

RL78/G23

SMS I2C 通信による HS300x 湿度センサ制御

要旨

本アプリケーションノートでは、湿度センサ（HS300x）を RL78/G23 で制御し、屋内空気質を測定する例を説明します。SNOOZE モード・シーケンサ（SMS）、データ・トランスファ・コントローラ（DTC）およびシリアル・インタフェース IICA（IICA）を使って SNOOZE モード期間中に I²C 通信で HS300x を制御します。SMS で I²C 通信を行うことで、CPU で処理したときより低消費電力で通信が可能です。

動作確認デバイス

RL78/G23

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、そのマイコンの使用に合わせて変更し、十分評価してください。

目次

1. 仕様	3
2. 動作確認条件	6
3. 関連アプリケーションノート	7
4. ハードウェア説明	8
4.1 ハードウェア構成例	8
4.2 使用端子一覧	9
5. ソフトウェア説明	10
5.1 動作概要	10
5.2 フォルダ構成	11
5.3 オプション・バイトの設定一覧	12
5.4 定数一覧	12
5.5 変数一覧	13
5.6 関数一覧	13
5.7 関数仕様	13
5.8 フローチャート	16
5.8.1 メイン処理	16
5.8.2 SMS 初期化処理	17
5.8.3 SMS 動作開始処理	17
5.8.4 SMS ステータス確認処理	18
5.8.5 SMS 割り込み処理	18
5.8.6 IICA0 ステータス確認処理	18
5.8.7 データのコピー処理	19
5.9 SNOOZE モード・シーケンサの設定	20
6. 応用例	27
6.1 r01an6065jj0110_sms_humidity_sensor.scfg	27
6.1.1 クロック	29
6.1.2 システム	29
6.1.3 r_bsp	29
6.1.4 Config_LVD0	29
6.1.5 Config_DTC	29
6.1.6 Config_IT000_ITL001	29
6.1.7 Config_PORT	29
6.1.8 Config_IICA0	29
7. サンプルコード	30
8. 参考ドキュメント	30
改訂記録	31

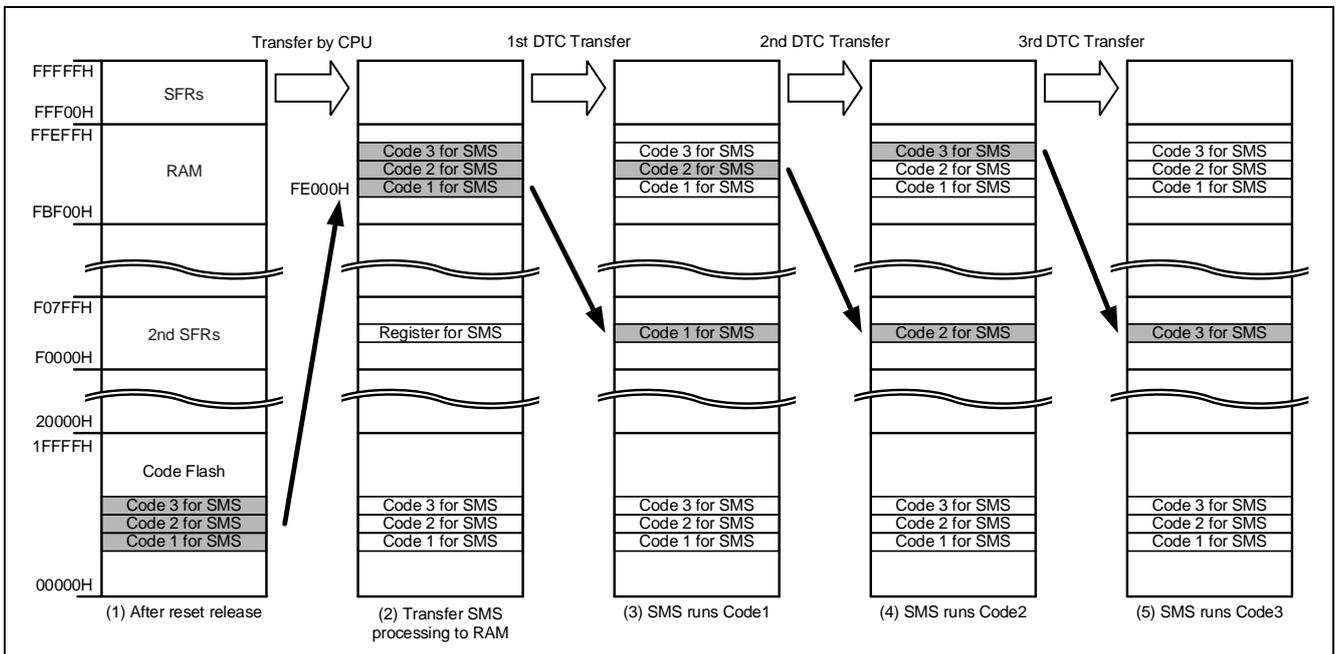
1. 仕様

SMS はあらかじめシーケンサ・インストラクション・レジスタ SMSI0~SMSI31 に設定した最大 32 の処理を順次実行します。しかし、I²C の一般的な通信規格は 32 の処理では収まりません。このため本アプリケーションノートでは、SMS で実行する処理を 3 つに分け、その処理を SMS 動作中に SMSI レジスタに順次更新することで I²C 通信を行う方法を示します。SMSI レジスタの更新には、DTC を使用します。

図 1-1 にメモリマップを示します。SMSI0~SMSI31 は、拡張特殊機能レジスタ（2nd SFR）領域に割り当てられています。

RAM に SMS で使用する処理をあらかじめ格納しておきます（図 1-1 の(1), (2)）。I²C 通信を行う間隔はインターバル・タイマ（TML32）を用いて設定し、SMS は TML32 の割り込み（INTITL）をトリガに起動します。SMS の起動後、DMATRGR 命令により DTC を起動し、RAM から SMS のレジスタに転送します。SMS は転送された処理を順次実行し、I²C 通信を行います（図 1-1 の(3) - (5)）。

図 1-1 メモリマップ



本アプリケーションノートで使用する湿度センサ（HS300x）は、受信したデータ（Humidity[13:0]）から以下の式を使って湿度を求めることができます。

$$\text{Humidity}[\%RH] = \left(\frac{\text{Humidity}[13:0]}{2^{14} - 1} \right) * 100$$

サンプルコードは受信したデータの Humidity[13:8]の値を閾値と比較します。サンプルコードの閾値は湿度 50%に相当する 20H が設定されています。

HS300x の詳細は、HS300x のマニュアルを参照してください。

図 1-2 にシステム構成を、図 1-3 にシステム全体のフローチャートを示します。

図 1-2 システム構成

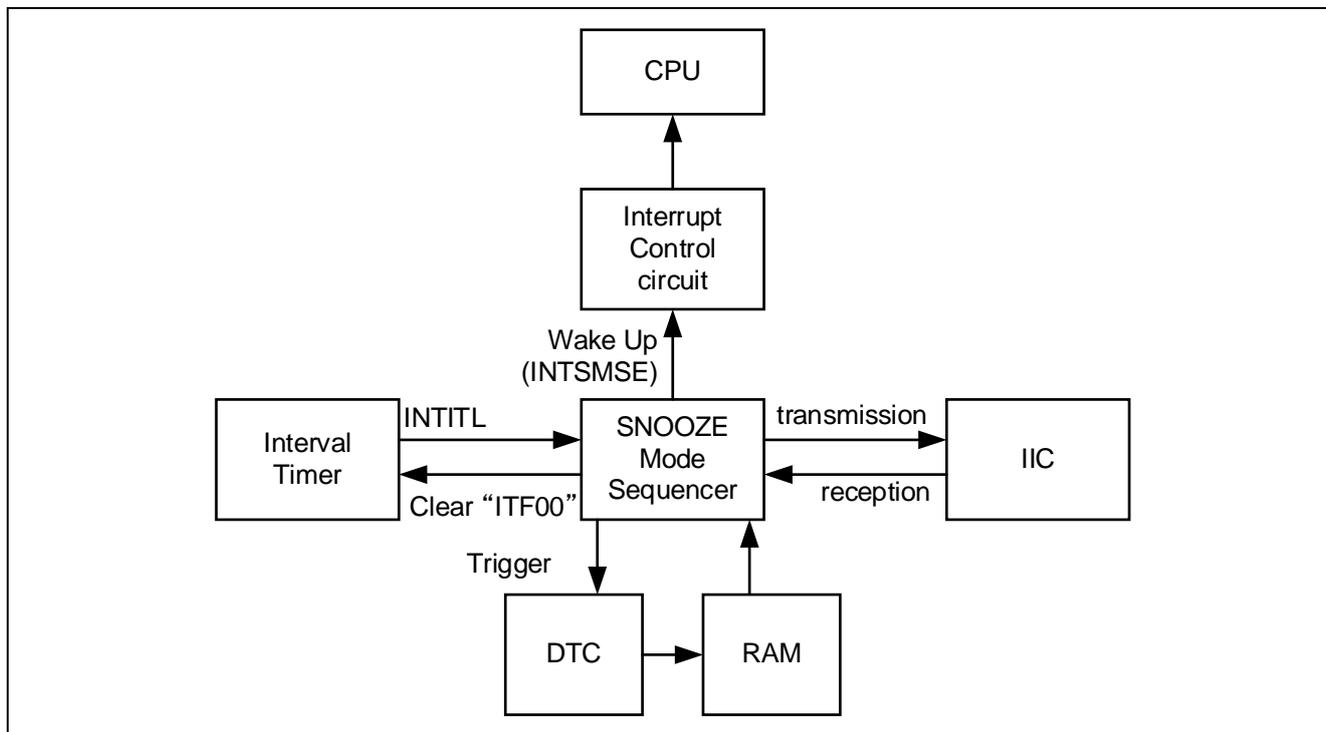
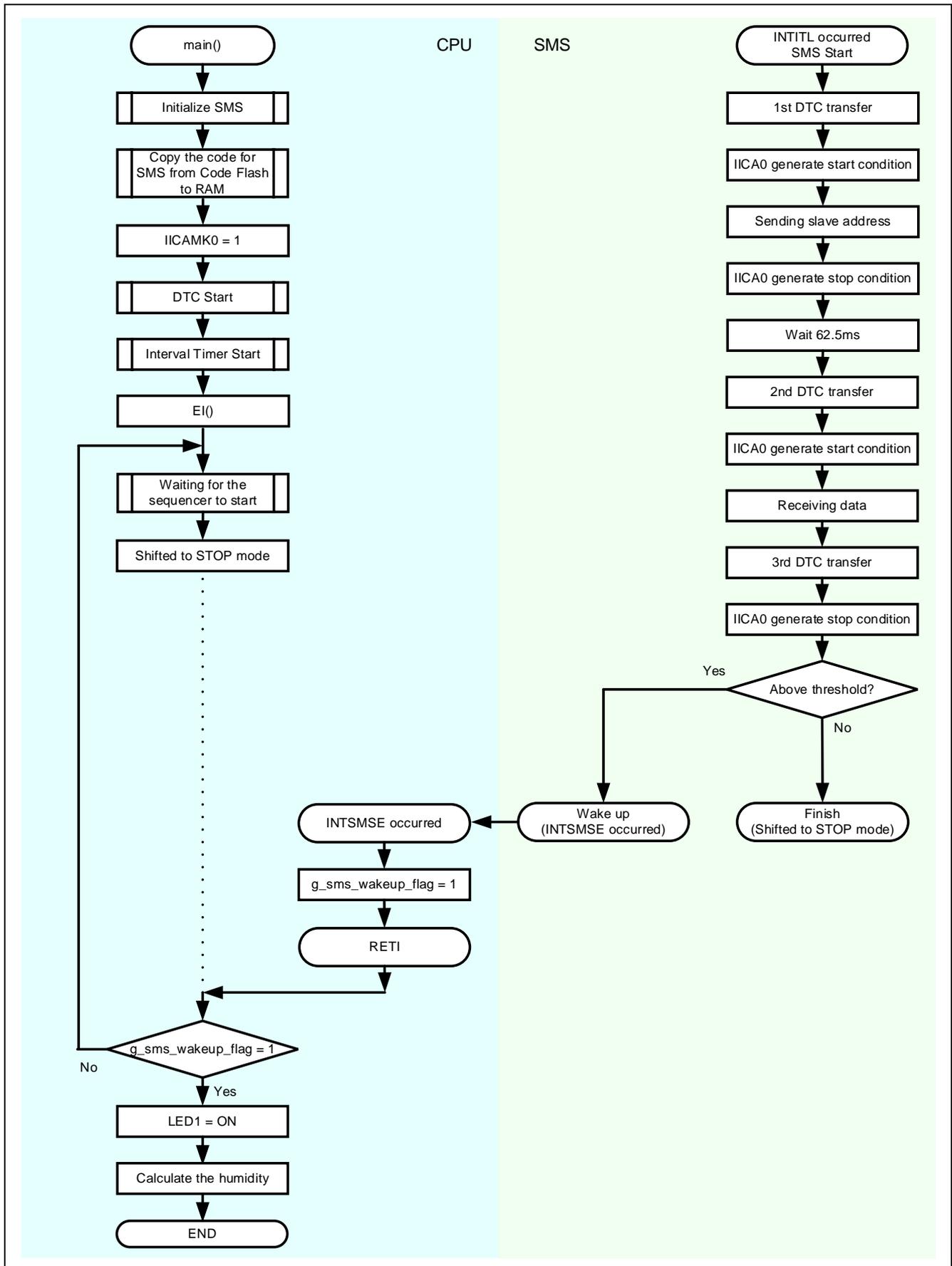


図 1-3 全体フローチャート



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	RL78/G23 (R7F100GLG)
動作周波数	<ul style="list-style-type: none"> 高速オンチップ・オシレータ・クロック : 32MHz CPU/周辺ハードウェア・クロック : 32MHz
動作電圧	<ul style="list-style-type: none"> 3.3V LVD0 動作 (V_{LVD0}) : リセット・モード 立ち上がり時 TYP. 1.90V (1.84V~1.95V) 立ち下がり時 TYP. 1.86V (1.80V~1.91V)
統合開発環境 (CS+)	ルネサスエレクトロニクス製 CS+ for CC V8.07.00
C コンパイラ (CS+)	ルネサスエレクトロニクス製 CC-RL V1.11
統合開発環境 (e ² studio)	ルネサスエレクトロニクス製 e ² studio 2022-01 (22.1.0)
C コンパイラ (e ² studio)	ルネサスエレクトロニクス製 CC-RL V1.11
統合開発環境 (IAR)	IAR システム製
C コンパイラ (IAR)	IAR Embedded Workbench for Renesas RL78 V4.21.1
スマート・コンフィグレータ	V.1.2.0
ボードサポートパッケージ (r_bsp)	V.1.13
エミュレータ	CS+、e ² studio : COM ポート IAR : E2 エミュレータ Lite
使用ボード	RL78/G23 Fast Prototyping Board (RTK7RLG230CLG000BJ)

3. 関連アプリケーションノート

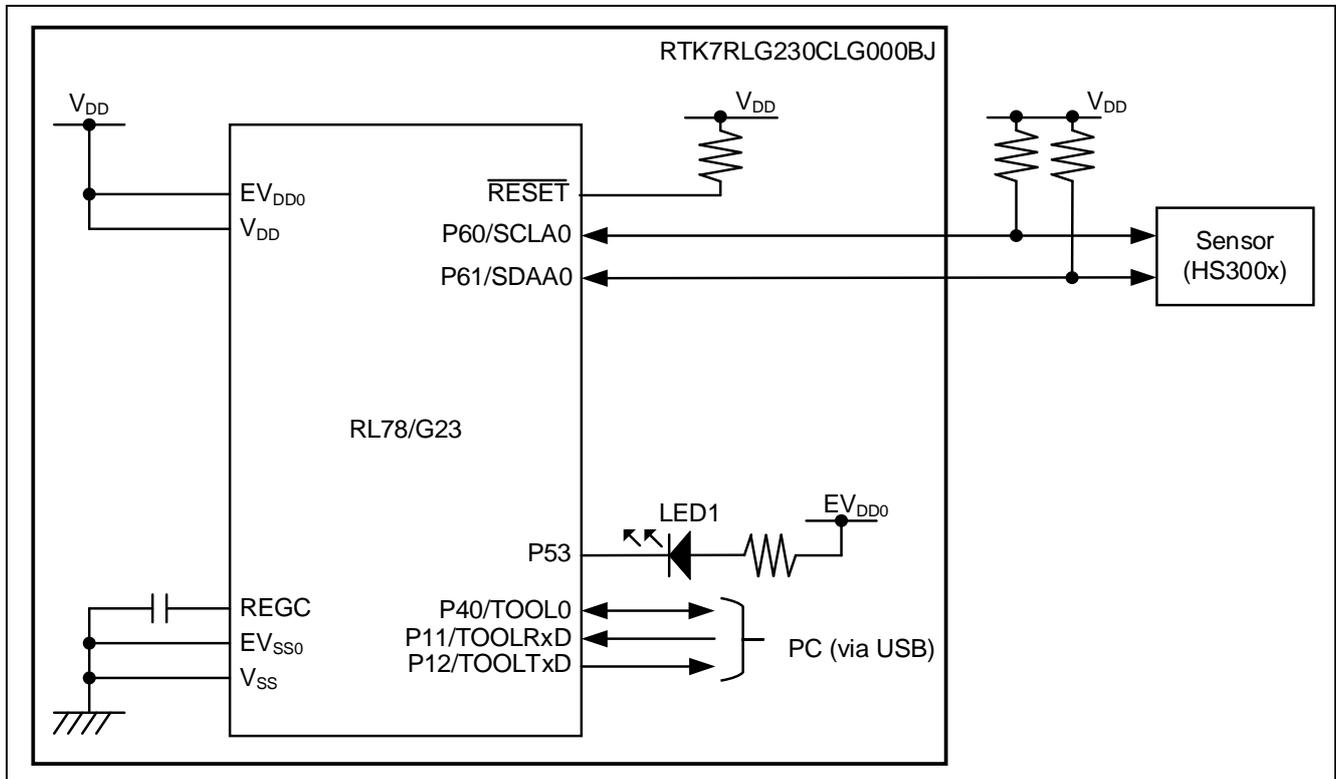
本アプリケーションノートに関連するアプリケーションノートを以下に示します。
併せて参照してください。

4. ハードウェア説明

4.1 ハードウェア構成例

図 4-1 に本アプリケーションで使用するハードウェア構成例を示します。

図 4-1 ハードウェア構成例



- 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 V_{DD} 又は V_{SS} に接続して下さい）。
- 注意 2. EV_{SS} で始まる名前の端子がある場合には V_{SS} に、 EV_{DD} で始まる名前の端子がある場合には V_{DD} にそれぞれ接続してください。
- 注意 3. V_{DD} は $LVD0$ にて設定したリセット解除電圧 (V_{LVD0}) 以上にしてください。

本アプリケーションノートでは湿度センサ (HS300x) を使用します。HS300x の詳細は、HS300x のマニュアルを参照してください。

4.2 使用端子一覧

表 4-1 に使用端子と機能を示します。

表 4-1 使用端子と機能

端子名	入出力	内容
P53	出力	LED1 点灯 (Low Active)
P60	入出力	シリアル・クロック
P61	入出力	シリアル・データ・バス

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。実際に回路を作成される場合は、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。

5. ソフトウェア説明

5.1 動作概要

本サンプルコードでは、32 ビット・インターバル・タイマ (TML32) の割り込み要求 (INTITL) によって STOP モードから SNOOZE モードに移行し、SNOOZE モード・シーケンサ (SMS) から DTC を起動します。DTC は前もって RAM に格納していたコードをシーケンサ・インストラクション・レジスタ p (SMSIp (p = 3-31)) に転送します。転送されたコードは SMS が実行し、I²C 通信を行ってセンサからデータを取得し閾値と比較します。閾値を超えた場合は割り込み要求信号 (INTSMSE) を出力して CPU を起動させます。

表 5-1 に本サンプルコードの動作概要を示します。

表 5-1 動作概要

No.	各機能の動作		
	CPU	SMS	DTC
(1)	SMS 用のコードを Code Flash から RAM に格納	—	—
(2)	TML32 のカウント開始	起動トリガ待ち状態	起動トリガ待ち状態
(3)	STOP モードへ移行	↑	↑
(4)	STOP 中	TML32 のコンペアー致で SMS 起動	↑
(5)	↑	DTC 起動処理を実行	↑
(6)	↑	DTC 転送完了待ち	RAM から SMSIp レジスタにデータを転送
(7)	↑	センサに MR コマンドを送信し、62.5ms 待つ	起動トリガ待ち状態
(8)	↑	DTC 起動処理を実行	↑
(9)	↑	DTC 転送完了待ち	RAM から SMSIp レジスタにデータを転送
(10)	↑	センサからデータを取得し、RAM に格納	起動トリガ待ち状態
(11)	↑	DTC 起動処理を実行	↑
(12)	↑	DTC 転送完了待ち	RAM から SMSIp レジスタにデータを転送
(13)	↑	取得した結果が閾値を超えていない場合は (14) へ、超えた場合は (15) へ	起動トリガ待ち状態
(14)	↑	SMS は待機状態へ移行し(4)へ	↑
(15)	↑	CPU を起動	↑
(16)	LED1 点灯	—	—
(17)	RAM に格納されているデータから湿度を計算	—	—

5.2 フォルダ構成

表 5-2 にサンプルコードの使用しているソースファイル／ヘッダファイルの構成を示します。なお、統合開発環境で自動生成されるファイル、bsp 環境のファイルは除きます。

表 5-2 フォルダ構成

フォルダ、ファイル名	説明	スマート・コンフィグレータを使用
¥r01an6065jj0110_sms_humidity_sensor<DIR>	サンプルコードのフォルダ	
¥src<DIR>	プログラム格納用フォルダ	
main.c	サンプルコードソースファイル	
sms_p.c	SMS 用コード格納ファイル	
¥SMS<DIR>	SMS 用プログラム格納フォルダ	
r_sms.c	SMS 用ソースプログラム	
r_sms.h	SMS 用ヘッダファイル	
r_sms_ASM.smsasm	SMS 用 ASM ソースファイル	
¥smc_gen<DIR> ^{注2}	スマート・コンフィグレータ生成フォルダ	√
¥Config_DTC<DIR>	DTC 用プログラム格納フォルダ	√
Config_DTC.c	DTC 用ソースプログラム	√
Config_DTC.h	DTC 用ヘッダファイル	√
Config_DTC_user.c	DTC 用割り込みソースファイル	√ ^{注1}
¥Config_IICA0<DIR>	IICA0 用プログラム格納フォルダ	√
Config_IICA0.c	IICA0 用ソースファイル	√
Config_IICA0.h	IICA0 用ヘッダファイル	√
Config_IICA0_user.c	IICA0 用割り込みソースファイル	√ ^{注1}
¥Config_ITL000_ITL001<DIR>	TML32 用プログラム格納フォルダ	√
Config_ITL000_ITL001.c	TML32 用ソースファイル	√
Config_ITL000_ITL001.h	TML32 用ヘッダファイル	√
Config_ITL000_ITL001_user.c	TML32 用割り込みソースファイル	√ ^{注1}
¥Config_PORT<DIR>	PORT 用プログラム格納フォルダ	√
Config_PORT.c	PORT 用ソースファイル	√
Config_PORT.h	PORT 用ヘッダファイル	√
Config_PORT_user.c	PORT 用割り込みソースファイル	√ ^{注1}
¥general<DIR>	初期化、共通プログラム格納フォルダ	√
¥r_bsp<DIR>	BSP プログラム格納フォルダ	√
¥r_config<DIR>	BSP_CFG プログラム格納フォルダ	√

補足 ” <DIR> ” は、ディレクトリを意味します。

注 1. 本サンプルコードでは使用しません。

注 2. IAR 版のサンプルコードは構成が異なります。詳細は IAR 版のサンプルコードを確認してください。また、r01an6065jj0110_sms_humidity_sensor.ipcf を格納しています。詳細は、「RL78 スマート・コンフィグレータ ユーザーガイド：IAR 編 (R20AN0581)」を確認してください。

5.3 オプション・バイトの設定一覧

表 5-3 にオプション・バイト設定を示します。

表 5-3 オプション・バイト設定

アドレス	設定値	内容
000C0H/040C0H	1110 1111B (EFH)	ウォッチドッグ・タイマ動作停止 (リセット解除後、カウント停止)
000C1H/040C1H	1111 1110B (FEH)	LVD 動作：リセット・モード 立ち上がり時 TYP.1.90V (1.84V~1.95V) 立ち下がり時 TYP.1.86V (1.80V~1.91V)
000C2H/040C2H	1110 1000B (E8H)	フラッシュ動作モード：高速メインモード 高速オンチップ・オシレータの周波数：32MHz
000C3H/040C3H	1000 0101B (85H)	オンチップ・デバッグ動作許可

5.4 定数一覧

表 5-4 にサンプルコードで使用する定数を示します。

表 5-4 サンプルコードで使用する定数

定数名	設定値	内容	ファイル
LED1	P5_bit.no3	P53	main.c
LED_ON	0	LED を ON するための設定値	main.c
THRESHOLD	20H	湿度閾値 (上位 2 バイト)	main.c
DATA_BYTE	27*2*3	SMS に転送するコードのバイト数 (27*2byte)*3times = 162	main.c
NUMBER_OF_DATA	4	センサのデータ (バイト数)	main.c
ERROR	FFFFH	通信エラー時の設定値	main.c
NO_DATA	-1	通信エラー時の設定値	main.c
sensor_add[]	88H	センサのアドレス (送信用)	main.c
	89H	センサのアドレス (受信用)	
g_dtc_data[]	注	SMS に転送するコード	sms_p.c

注 詳細は表 5-7~表 5-10 を参照してください。

5.5 変数一覧

表 5-5 にサンプルコードで使用するグローバル変数を示します。

表 5-5 サンプルコードで使用するグローバル変数

型	変数名	内容	使用関数
float	g_result	湿度計算結果	main
uint8_t	g_sms_wakeup_flag	SMS ウェイクアップフラグ	r_sms_getstatus r_sms_start r_sms_interrupt
uint16_t	g_communication_status	IICA0 ステータスフラグ	r_iica0_getstatus

5.6 関数一覧

表 5-6 にサンプルコードで使用する関数を示します。ただし、スマート・コンフィグレータで生成された関数の内、変更を行っていないものは除きます。

表 5-6 関数一覧

関数名	概要	ソースファイル
main	メイン処理	main.c
r_sms_create	SMS 初期化処理	r_SMS.c
r_sms_start	SMS 動作開始処理	r_SMS.c
r_sms_getstatus	SMS ステータス確認処理	r_SMS.c
r_sms_interrupt	SMS 割り込み処理	r_SMS.c
r_iica0_getstatus	IICA0 ステータス確認処理	Config_IICA0_user.c
memcpy_ff ^注	データのコピー処理	main.c

注 IAR 版のみ。CC-RL 版は標準関数「_COM_memcpy_ff」を使用します。

5.7 関数仕様

サンプルコードの関数仕様を示します。

[関数名] main

概要	メイン処理
ヘッダ	r_smc_entry.h, string.h, r_sms.h
宣言	void main (void);
説明	SMS の初期設定を行い、SMS 起動時に DTC が転送するコード (g_dtc_data[]) をコードフラッシュ領域から RAM に転送します。 TML32 の動作を開始し、DTC をトリガ待ち状態にして STOP モードへ移行します。スタンバイ・モードから復帰した際、CPU 起動要求フラグの確認へ移行します。CPU 起動要求フラグ (g_sms_wakeup_flag) がセットされていない場合は再度 STOP モードへ移行、CPU 起動要求フラグがセットされている場合は LED1 を点灯します。
引数	なし
リターン値	なし
備考	なし

[関数名] r_sms_create

概要	SMS の初期化処理
ヘッダ	r_SMS.h, r_cg_macrodriver.h, r_cg_userdefine.h, r_SMS_ASM.h
宣言	void r_sms_create (void)
説明	SMS の構成、SMS 命令のコピー、SMS データのコピーなど、SMS モジュールを制御する前に必要な初期化処理を実行します。
引数	なし
リターン値	なし
備考	なし

[関数名] r_sms_start

概要	SMS の動作を開始
ヘッダ	r_SMS.h, r_cg_macrodriver.h, r_cg_userdefine.h, r_SMS_ASM.h
宣言	void r_sms_start (uint16_t addr_iic_sl, uint16_t addr_iic_rx_s, uint16_t addr_iic_rx_e, uint16_t addr_dtc_init, uint16_t val_threshold)
説明	引数から SMS データを設定し、SMS モジュールの動作を開始します。
引数	addr_iic_sl, addr_iic_rx_s, addr_iic_rx_e, val_threshold
リターン値	なし
備考	なし

[関数名] r_sms_getstatus

概要	SMS ウェイクアップ・ステータスの確認
ヘッダ	r_SMS.h, r_cg_macrodriver.h, r_cg_userdefine.h, r_SMS_ASM.h
宣言	uint8_t r_sms_getstatus (void)
説明	SMS ウェイクアップ・ステータスを確認します。
引数	なし
リターン値	なし
備考	なし

[関数名] r_sms_interrupt

概要	INTSMSE 割り込みに伴う処理
ヘッダ	r_SMS.h, r_cg_macrodriver.h, r_cg_userdefine.h, r_SMS_ASM.h
宣言	#pragma interrupt r_sms_interrupt(vect=INTSMSE)
説明	g_sms_wakeup_flag = 1 に設定します。
引数	なし
リターン値	なし
備考	なし

[関数名] r_iica0_getstatus

概要	IICA0 のステータス確認処理
ヘッダ	r_cg_macrodriver.h, r_cg_userdefine.h, Config_IICA0.h
宣言	uint16_t r_iica0_getstatus (void) (void)
説明	IICA0 のステータスを確認します。
引数	なし
リターン値	なし
備考	なし

[関数名] memcpy_ff

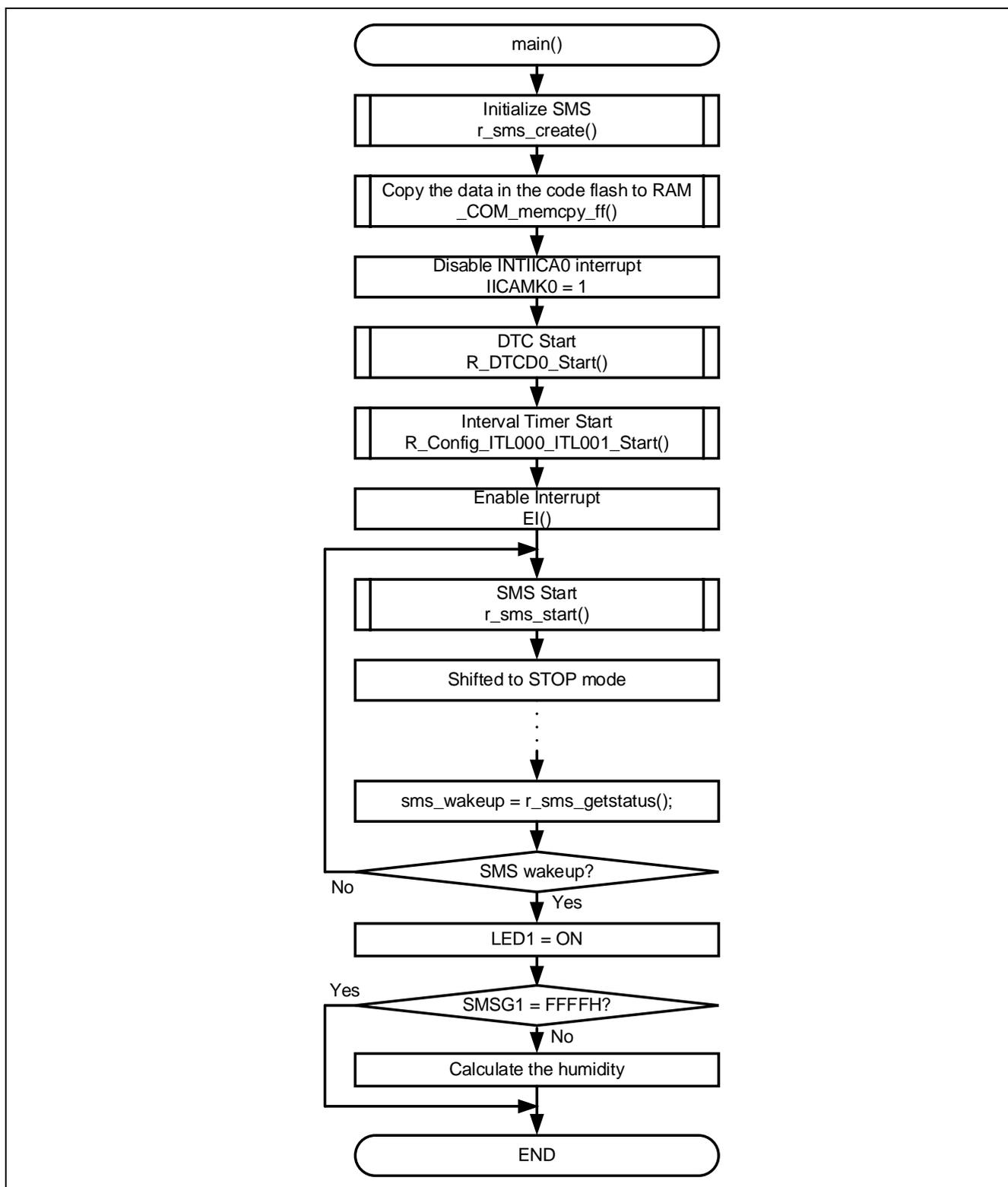
概要	データのコピー
ヘッダ	r_smc_entry.h, r_SMS.h
宣言	__far_func void __far * memcpy_ff(void __near *_Restrict s1, const void __far *_Restrict s2, uint32_t n)
説明	s2 のデータを s1 に n バイト分コピーします。
引数	s1, s2, n
リターン値	なし
備考	なし

5.8 フローチャート

5.8.1 メイン処理

図 5-1 にメイン処理のフローチャートを示します。

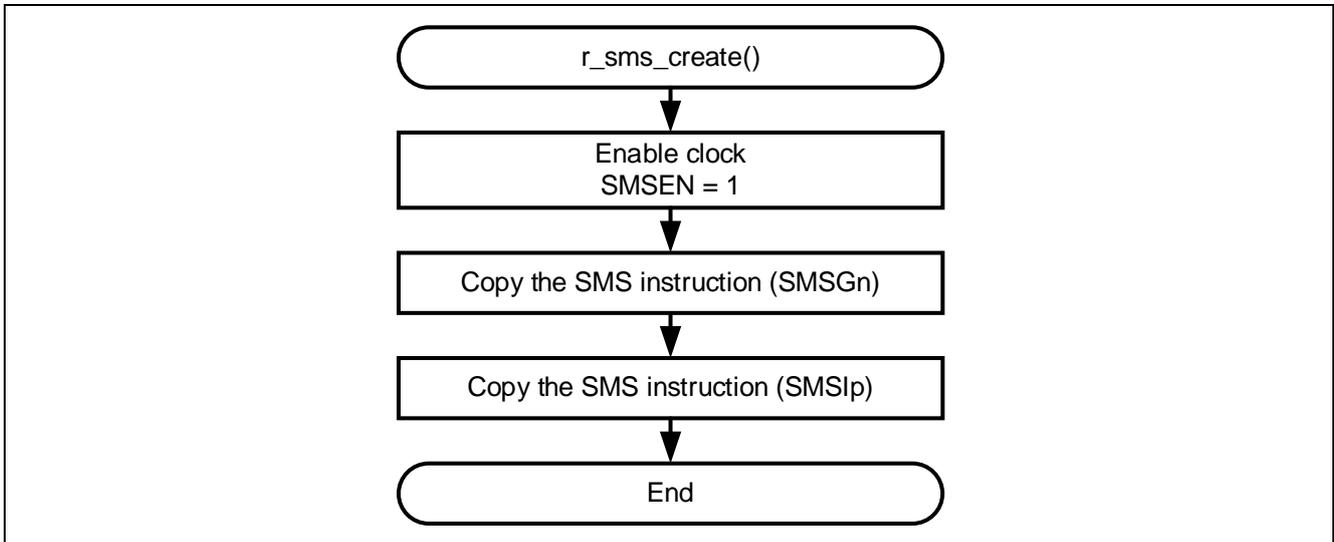
図 5-1 メイン処理



5.8.2 SMS 初期化処理

図 5-2 に SMS 初期化処理のフローチャートを示します。

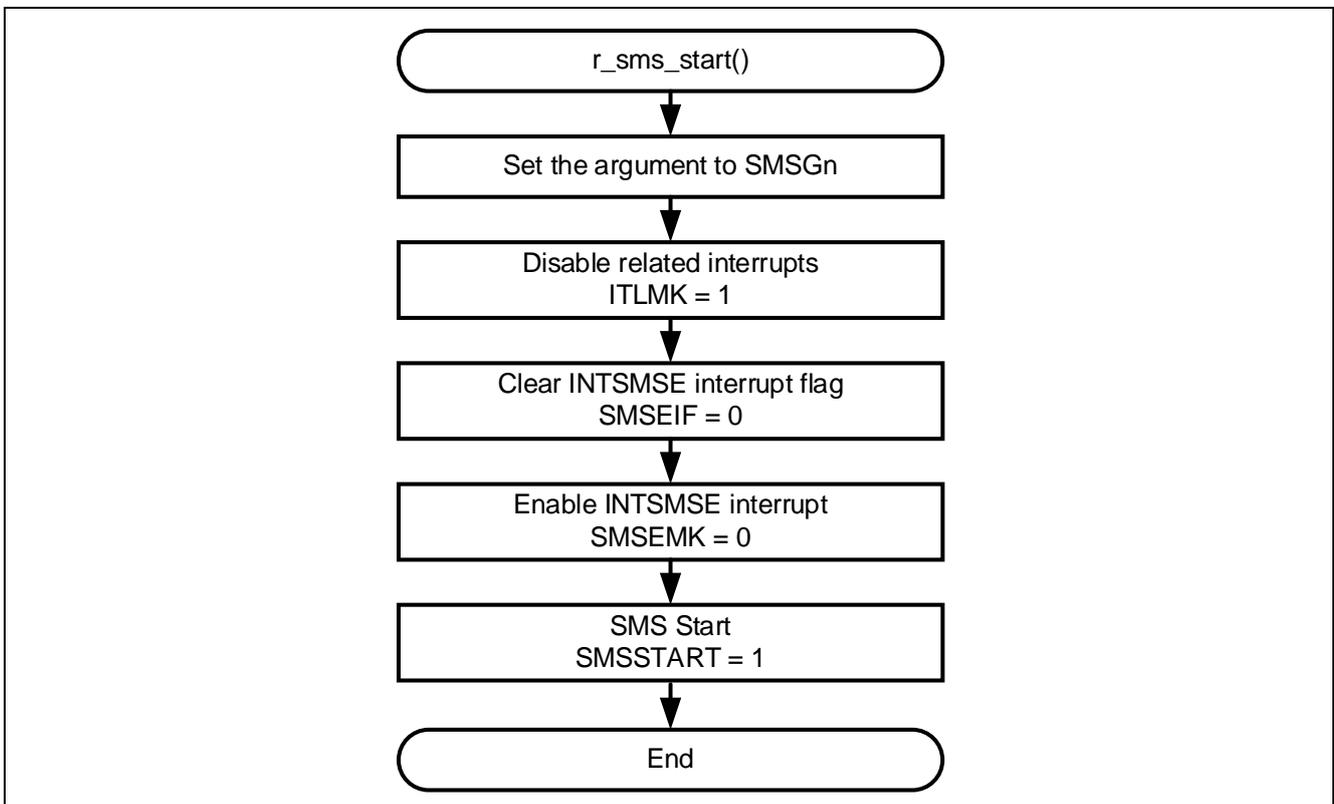
図 5-2 SMS 初期化処理



5.8.3 SMS 動作開始処理

図 5-3 に SMS 動作開始処理のフローチャートを示します。

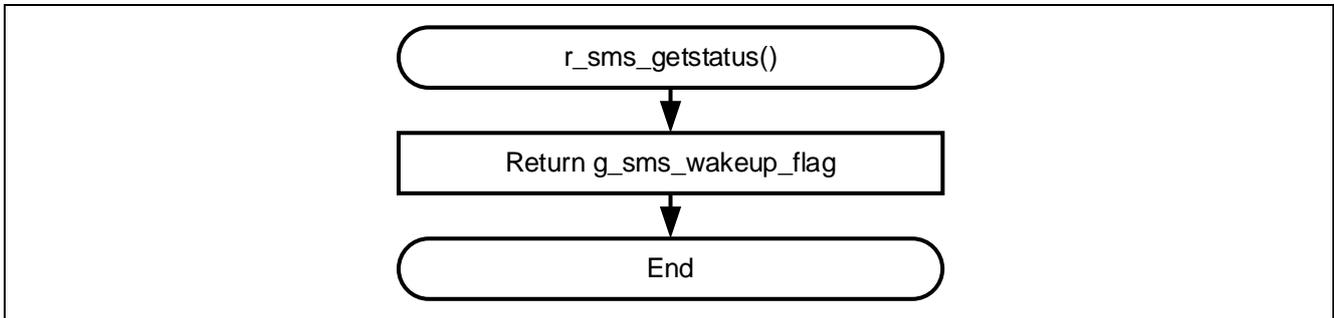
図 5-3 SMS 動作開始処理



5.8.4 SMS ステータス確認処理

図 5-4 に SMS ステータス確認処理のフローチャートを示します。

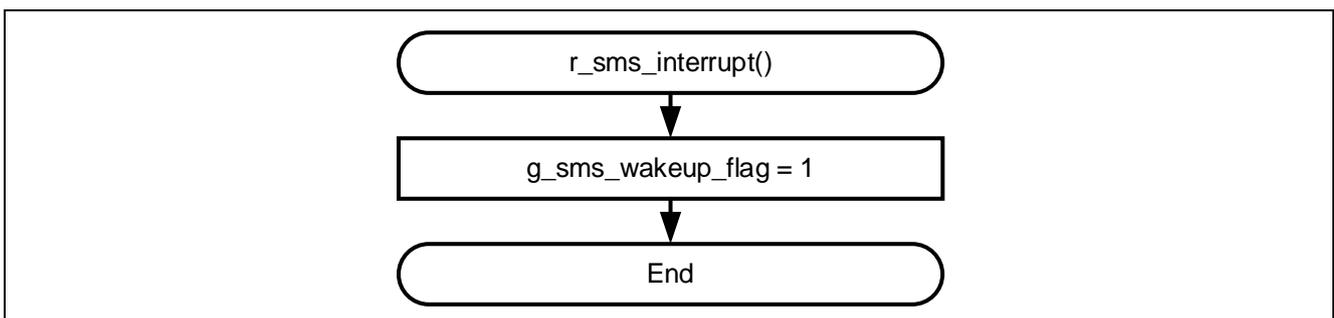
図 5-4 SMS ステータス確認処理



5.8.5 SMS 割り込み処理

図 5-5 に SMS 割り込み処理のフローチャートを示します。

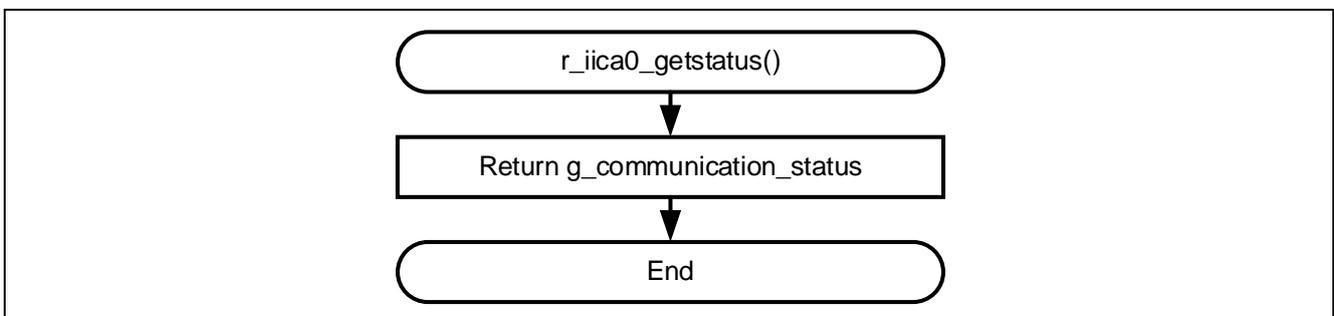
図 5-5 SMS 割り込み処理



5.8.6 IICA0 ステータス確認処理

図 5-6 に IICA0 ステータス確認処理のフローチャートを示します。

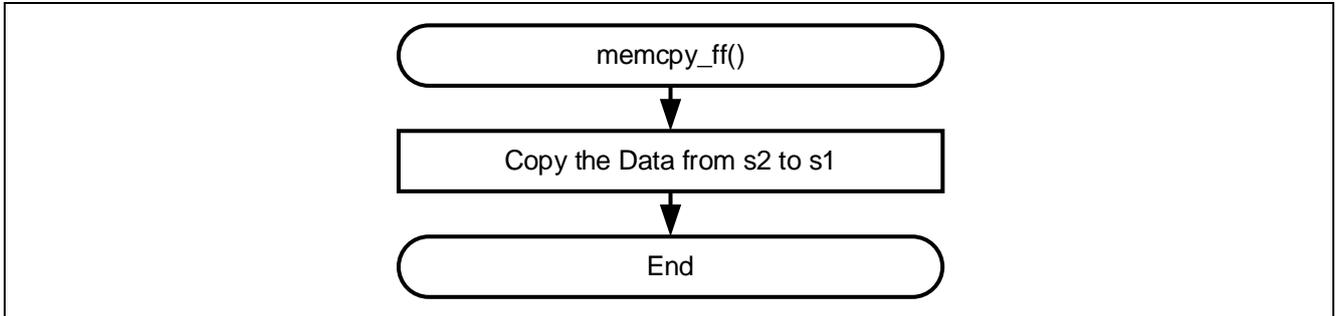
図 5-6 IICA0 ステータス確認処理



5.8.7 データのコピー処理

図 5-7 にデータのコピー処理のフローチャートを示します。

図 5-7 データのコピー処理



5.9 SNOOZE モード・シーケンサの設定

起動トリガに設定したイベントが発生すると SMS はシーケンサ・インストラクション・レジスタ (SMSIO-31) に格納された処理コマンドを順次実行します。処理コマンドの実行では、シーケンサ汎用レジスタ (SMSG0-15) をソース・アドレスやディスティネーション・アドレス、演算データなどの格納に使用します。

SMSIO-31 と SMSG0-15 は、SMS 用プログラム (.SMSASM ファイル) をアセンブリ言語で記述することで設定します。作成した SMS 用プログラムは SMS 用アセンブラで C 言語ファイルへ変換されプログラムに組み込まれます。

サンプルコードで実行する SMS 処理の仕様を以下に示します。

概要	SMS 処理
説明	TML32 割り込みにより SMS が起動し、センサに MR コマンドを送信し、センサの動作安定待ち時間 62.5ms を待ちます。待ち時間経過後、センサの測定結果を入手し、RAM に格納します。 RAM に格納した湿度データが閾値以上であれば CPU を起動し、閾値より小さければ SMS 処理を終了し、再度 STOP モードへ移行します。 I ² C 通信を処理する命令を前もって RAM に格納しておき、DTC で 3 回に分けて転送し実行します。
引数 ^{注1}	addr_iic_sl : センサのスレーブアドレスを格納したバッファの開始アドレス addr_iic_rx_s : データ格納領域の開始アドレス addr_iic_rx_e : データ格納領域の終了アドレス val_threshold : 湿度データの閾値
リターン値	なし
備考	なし

注1 r_sms_start 関数設定で指定する引数です。

図 5-8、図 5-9 に SMS 処理のフローチャートを示します。

表 5-7～表 5-10 に SNOOZE モード・シーケンサを制御するレジスタの設定値を示します。

SMS は I²C 通信を使って、HS300x プロトコルに準じて制御しています。HS300x プロトコルの詳細は、HS300x のマニュアルを参照してください。

図 5-8 SMS 処理

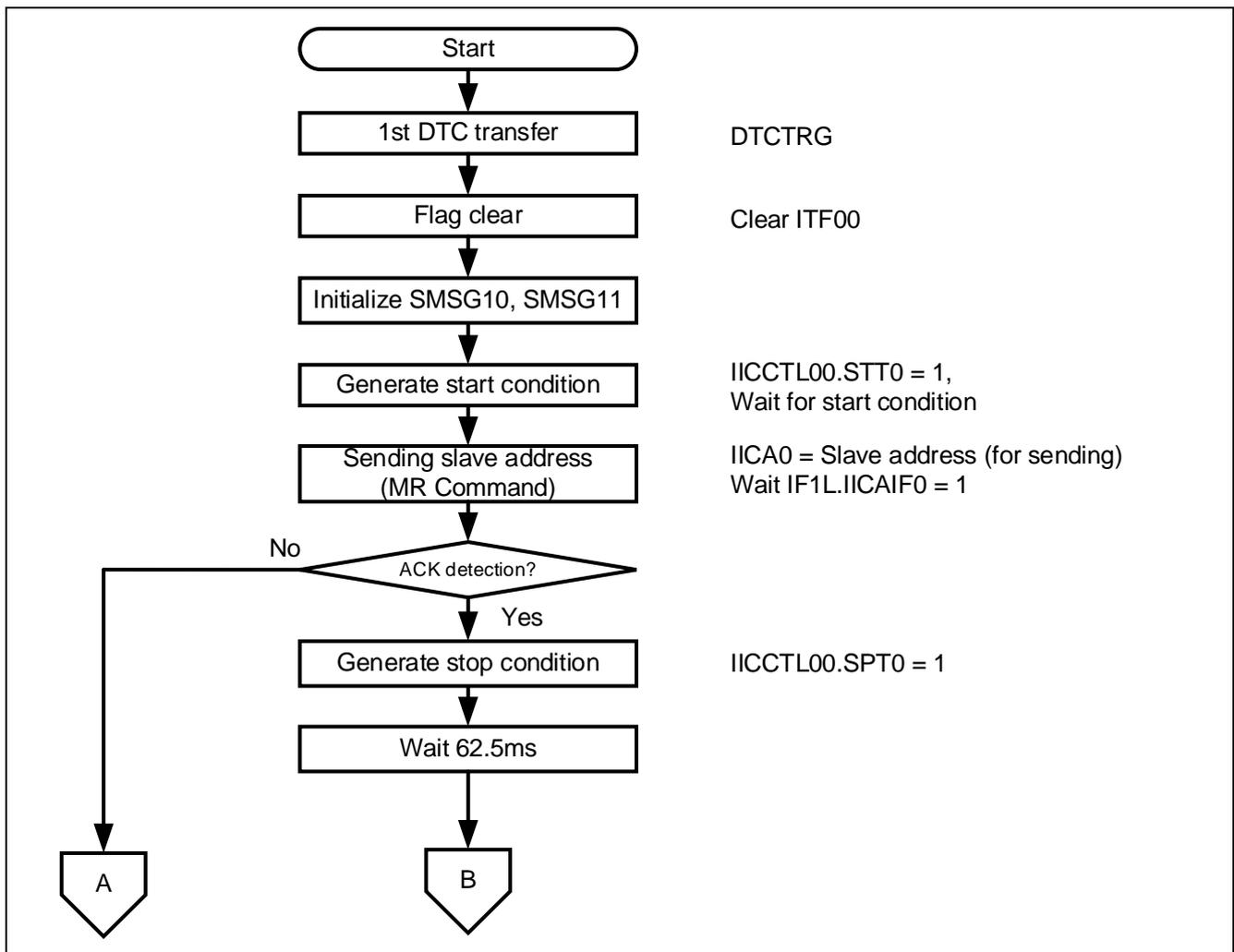


図 5-9 SMS 処理

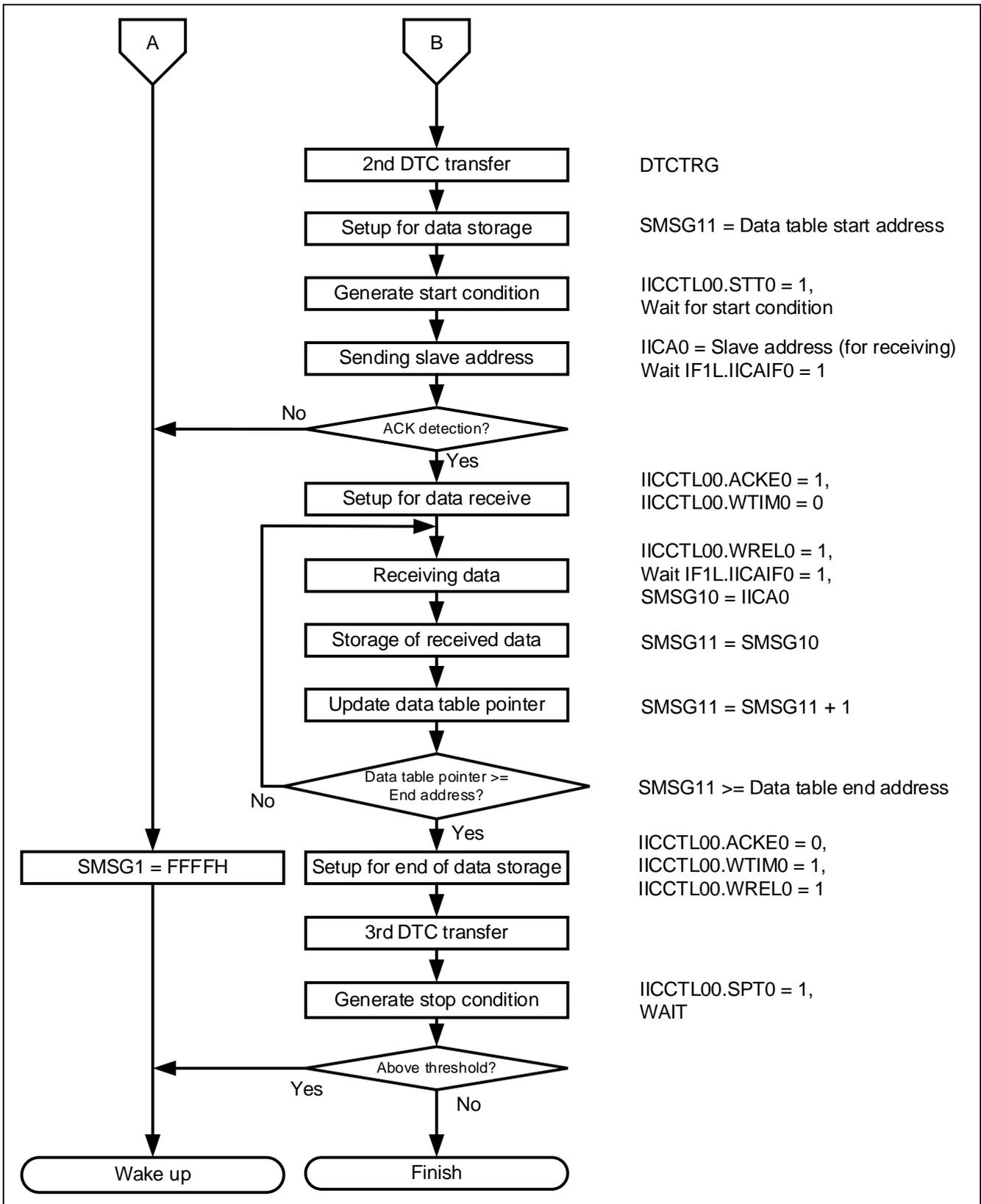


表 5-7 シーケンサ汎用レジスタ 0-15

レジスタ呼称	設定値	備考
SMSG0	0000H	固定値 : 0000H
SMSG1	&ITLS0	ITLS0 アドレス
SMSG2	&IICCTL00	IICCTL00 アドレス
SMSG3	&IF1L	IF1L アドレス
SMSG4	&IICA0	IICA0 アドレス
SMSG5	0000H	センサのスレーブアドレス
SMSG6	0000H	データテーブル開始アドレス
SMSG7	0000H	データテーブル終了アドレス
SMSG8	1	固定値 : 1
SMSG9	0000H	湿度の閾値
SMSG10	0000H	演算用
SMSG11	0000H	演算用
SMSG12	&SMSG1	SMSG1 アドレス
SMSG13	&SMSI28	SMSI28 アドレス
SMSG14	&SMSG11	SMSG11 アドレス
SMSG15	FFFFH	固定値 : FFFFH

表 5-8 シーケンサ・インストラクション・レジスタ 0-31 (1 回目の DTC 転送後)

レジスタ呼称	設定値	内容	備考
SMSI0	2DF2H	MOVW [MSG13+2], MSG15	SMSI29 ← FFFFH
SMSI1	F002H	DTCTRG	DTC 起動
SMSI2	AD02H	WHILE1 [MSG13+2].0	DTC 転送完了待ち
SMSI3	0100H	MOV [MSG1+0], MSG0	ITLS0 ← 00H
SMSI4	30A0H	MOVW MSG10, [MSG0+0]	MSG10 の初期化
SMSI5	30B0H	MOVW MSG11, [MSG0+0]	MSG11 の初期化
SMSI6	4210H	SET1 [MSG2+0].1	IICCTL00.STT0 ← 1
SMSI7	9640H	WAIT 100, 0	スタートコンディション待ち
SMSI8	15A0H	MOV MSG10, [MSG5+0]	MSG10 ← スレーブアドレス送信用
SMSI9	04A0H	MOV [MSG4+0], MSG10	IICAn ← MSG10
SMSI10	B330H	WHILE0 [MSG3+0].3	IFxx.IICAIFn = 1 まで待つ
SMSI11	5330H	CLR1 [MSG3+0].3	IF1L.IICAIF0 ← 0
SMSI12	6421H	MOV1 SCY, [MSG4+1].2	SCY ← IICS0.ACKD0
SMSI13	8051H	BNC 5	ACK=0 の場合 SMSI18 に分岐
SMSI14	4200H	SET1 [MSG2+0].0	IICCTLn0.SPT0 ← 1
SMSI15	9804H	WAIT 128, 4	62.5ms 待つ
SMSI16	7FF2H	CMPW MSG15, MSG15	SMSI0 に分岐するための比較
SMSI17	8EF2H	BZ -17	SMSI0 に分岐
SMSI18	2CF0H	MOVW [MSG12+0], MSG15	MSG1 ← FFFFH
SMSI19	F001H	WAKEUP	WAKEUP
SMSI20	0000H	未使用	—
SMSI21	0000H	未使用	—
SMSI22	0000H	未使用	—
SMSI23	0000H	未使用	—
SMSI24	0000H	未使用	—
SMSI25	0000H	未使用	—
SMSI26	0000H	未使用	—
SMSI27	0000H	未使用	—
SMSI28	0000H	未使用	—
SMSI29	0000H	未使用	—
SMSI30	0000H	未使用	—
SMSI31	0000H	未使用	—

補足 SMSI0~SMSI2 の設定値は、r_SMS_ASM.smsasm で設定します。

DTC で転送される SMSI3~SMSI29 のデータは、g_dtc_data に格納します。

表 5-9 シーケンサ・インストラクション・レジスタ 0-31 (2 回目の DTC 転送後)

レジスタ呼称	設定値	内容	備考
SMSI0	2DF2H	MOVW [MSG13+2], MSG15	SMSI29 ← FFFFH
SMSI1	F002H	DTCTRG	DTC 起動
SMSI2	AD02H	WHILE1 [MSG13+2].0	DTC 転送完了待ち
SMSI3	2E60H	MOVW [MSG14+0], MSG6	MSG11 ← MSG6 の値
SMSI4	4210H	SET1 [MSG2+0].1	IICCTL00.STT0 ← 1
SMSI5	9640H	WAIT 100, 0	スタートコンディション待ち
SMSI6	15A1H	MOV MSG10, [MSG5+1]	MSG10 ← スレーブアドレス受信用
SMSI7	04A0H	MOV [MSG4+0], MSG10	IICAn ← MSG10
SMSI8	B330H	WHILE0 [MSG3+0].3	IFxx.IICAIFn = 1 まで待つ
SMSI9	5330H	CLR1 [MSG3+0].3	IF1L.IICAIF0 ← 0
SMSI10	6421H	MOV1 SCY, [MSG4+1].2	SCY ← IICS0.ACKD0
SMSI11	8101H	BNC 16	ACK=0 の場合 SMSI27 に分岐
SMSI12	4220H	SET1 [MSG2+0].2	IICCTL00.ACKE0 ← 1
SMSI13	5230H	CLR1 [MSG2+0].3	IICCTL00.WTIM0 ← 0
SMSI14	4250H	SET1 [MSG2+0].5	IICCTL00.WRELO ← 1
SMSI15	B330H	WHILE0 [MSG3+0].3	IFxx.IICAIFn = 1 まで待つ
SMSI16	5330H	CLR1 [MSG3+0].3	IF1L.IICAIF0 ← 0
SMSI17	14A0H	MOV MSG10, [MSG4+0]	MSG10 ← IICAn
SMSI18	0BA0H	MOV [MSG11+0], MSG10	格納されているアドレス ← MSG10
SMSI19	7B80H	ADDW MSG11, MSG8	MSG11 ← MSG11 + 1
SMSI20	7B72H	CMPW MSG11, MSG7	MSG11 < MSG7 の時
SMSI21	8F90H	BC -7	SMSI14 に分岐
SMSI22	5220H	CLR1 [MSG2+0].2	IICCTL00.ACKE0 ← 0
SMSI23	4230H	SET1 [MSG2+0].3	IICCTL00.WTIM0 ← 1
SMSI24	4250H	SET1 [MSG2+0].5	IICCTL00.WRELO ← 1
SMSI25	7FF2H	CMPW MSG15, MSG15	SMSI0 に分岐するための比較
SMSI26	8E62H	BZ -26	SMSI0 に分岐
SMSI27	2CF0H	MOVW [MSG12+0], MSG15	MSG1 ← FFFFH
SMSI28	F001H	WAKEUP	WAKEUP
SMSI29	0000H	未使用	—
SMSI30	0000H	未使用	—
SMSI31	0000H	未使用	—

補足 SMSI0~SMSI2 の設定値は、r_SMS_ASM.smsasm で設定します。

DTC で転送される SMSI3~SMSI29 のデータは、g_dtc_data に格納します。

表 5-10 シーケンサ・インストラクション・レジスタ 0-31 (3 回目の DTC 転送後)

レジスタ呼称	設定値	内容	備考
SMSI0	2DF2H	MOVW [MSG13+2], MSG15	SMSI29 ← FFFFH
SMSI1	F002H	DTCTRG	DTC 起動
SMSI2	AD02H	WHILE1 [MSG13+2].0	DTC 転送完了待ち
SMSI3	B330H	WHILE0 [MSG3+0].3	IFxx.IICAIFn == 1 まで待つ
SMSI4	5330H	CLR1 [MSG3+0].3	IF1L.IICAIF0 ← 0
SMSI5	4200H	SET1 [MSG2+0].0	IICCTLn0.SPT0 ← 1
SMSI6	9A10H	WAIT 161, 0	ストップコンディション待ち
SMSI7	16A0H	MOV MSG10, [MSG6+0]	MSG10 ← 変数 humi_h
SMSI8	79A2H	CMPW MSG9, MSG10	閾値 < humi_h の時
SMSI9	8020H	BC 2	SMSI11 に分岐
SMSI10	F000H	FINISH	FINISH
SMSI11	F001H	WAKEUP	WAKEUP
SMSI12	0000H	未使用	—
SMSI13	0000H	未使用	—
SMSI14	0000H	未使用	—
SMSI15	0000H	未使用	—
SMSI16	0000H	未使用	—
SMSI17	0000H	未使用	—
SMSI18	0000H	未使用	—
SMSI19	0000H	未使用	—
SMSI20	0000H	未使用	—
SMSI21	0000H	未使用	—
SMSI22	0000H	未使用	—
SMSI23	0000H	未使用	—
SMSI24	0000H	未使用	—
SMSI25	0000H	未使用	—
SMSI26	0000H	未使用	—
SMSI27	0000H	未使用	—
SMSI28	0000H	未使用	—
SMSI29	0000H	未使用	—
SMSI30	0000H	未使用	—
SMSI31	0000H	未使用	—

補足 SMSI0~SMSI2 の設定値は、r_SMS_ASM.smsasm で設定します。

DTC で転送される SMSI3~SMSI29 のデータは、g_dtc_data に格納します。

6. 応用例

本アプリケーションノートは、サンプルコードの他に以下のスマート・コンフィグレータの設定ファイルを格納しています。

r01an6065jj0110_sms_humidity_sensor.scfg

ファイルの説明と使用する上での設定例および注意事項を以下に示します。

6.1 r01an6065jj0110_sms_humidity_sensor.scfg

サンプルコードで使用しているスマート・コンフィグレータの設定ファイルです。スマート・コンフィグレータで設定されている全ての機能が含まれています。サンプルコードの設定は以下の通りです。

表 6-1 スマート・コンフィグレータの設定

タグ名	コンポーネント	内容
クロック	-	動作モード：高速メインモード 2.4 (V) ~5.5 (V) EV _{DD} 設定：1.8V ≤ EV _{DD0} < 5.5V 高速オンチップ・オシレータ：32MHz f _{IHP} ：32MHz f _{CLK} ：32MHz（高速オンチップ・オシレータ） f _{SXP} ：32.768kHz（低速オンチップ・オシレータ）
システム	-	オンチップ・デバッグ動作設定：COMポート ^{注1} 疑似 RRM/DMM 機能設定：使用する Start/Stop 関数機能設定：使用しない トレース機能設定：使用する セキュリティ ID 設定：セキュリティ ID を設定する セキュリティ ID：0x00000000000000000000 セキュリティ ID 認証失敗時の設定：フラッシュ・メモリのデータを消去しない
コンポーネント	r_bsp	Start up select：Enable (use BSP startup) Control of invalid memory access detection：Disable RAM guard space (GRAM0-1)：Disabled Guard of control registers of port function (GPORT)：Disabled Guard of registers of interrupt function (GINT)：Disabled Guard of control registers of clock control function, voltage detector, and RAM parity error detection function (GCSC)：Disabled Data flash access control (DFLEN)：Disables Initialization of peripheral functions by Code Generator/Smart Configurator：Enable API functions disable：Enable Parameter check enable：Enable Setting for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode：High-speed Enable user warm start callback (PRE)：Unused Enable user warm start callback (POST)：Unused Watchdog Timer refresh enable：Unused
	Config_LVDD0	動作モード設定：リセット・モード 電圧検出設定：リセット発生電圧 (V _{LVDD0})：1.86 (V)

表 6-2 スマート・コンフィグレータの設定値

タグ名	コンポーネント	内容
コンポーネント	Config_DTC	DTC ベース・アドレス : 0xFFD00 コントロール・データ 0 (DTCD0) 起動要因 (DTCD0) : SNOOZE モード・シーケンサからのイベント出力 [DTCD0] 転送モード設定 : リピート・モード 転送データ・サイズ設定 : 16 ビット リピート・モード割り込み設定 : 禁止 リピート・エリア設定 : 転送元 転送元アドレス : 0xE000 転送先アドレス : 0x0386 アドレス固定 転送回数 : 3 転送ブロック・サイズ : 54
	Config_ITL000 _ITL001	コンポーネント : インターバル・タイマ 動作モード : 16 ビット・カウンタ・モード リソース : ITL000_ITL001 動作クロック : f _{SXP} クロックソース : f _{ITL0} /128 インターバル時間 : 5000ms 割り込み設定 : 使用しない
	Config_PORT	コンポーネント : ポート ポート選択 : PORT5 P53 : 出力 (1 を出力)
	Config_IICA0	クロック・モード設定 : f _{CLK} /2 アドレス : 16 動作モード設定 : 標準 転送クロック (f _{SCL}) : 100000 (bps) 優先順位 : レベル 3 コールバック機能設定 : マスタ送信完了、マスタ受信完了

注 1. IAR 使用時は以下の設定にしてください。

オンチップ・デバッグ動作設定 : エミュレータを使う

エミュレータ設定 : E2 エミュレータ Lite

6.1.1 クロック

サンプルコードで使用するクロックの設定を行います。

6.1.2 システム

サンプルコードのオンチップ・デバッグ設定を行います。

「オンチップ・デバッグ動作設定」、「セキュリティ ID 認証失敗時の設定」は、「表 5-3 オプション・バイト設定」の「オンチップ・デバッグ動作許可」に影響を与えます。設定を変更する際は注意してください。

6.1.3 r_bsp

サンプルコードのスタートアップの設定を行います。

6.1.4 Config_LVD0

サンプルコードの電源管理の設定を行います。

「表 5-3 オプション・バイト設定」の「LVD0 の設定」に影響を与えます。設定を変更する際は注意してください。

6.1.5 Config_DTC

サンプルコードで使用する DTC の設定を行います。

サンプルコードでは、コントロール・データ 0 を使用し、あらかじめ RAM（転送元アドレス：FE000H（設定値：0xE000））に格納していた SMS の処理を SMS のレジスタ（転送先アドレス：F0386H（設定値：0x0386））に転送します。RAM への格納は、main.c で行います。

6.1.6 Config_IT000_ITL001

サンプルコードのインターバル・タイマの初期設定を行います。

サンプルコードの SMS の起動にインターバル・タイマ割り込み（INTITL）を使用します。そのため、「割り込み設定」を「使用しない」に設定しています。「割り込み設定」を「使用する」に変更することも可能です。R_Config_SMS_Start 関数で INTITL はマスクされますので、STOP モード、SNOOZE モード中に INTITL が発生しても CPU が起動することはありません。STOP モード、SNOOZE モードから復帰後は、INTITL はマスクされた状態ですので、必要なときは INTITL のマスクを解除してください。

6.1.7 Config_PORT

サンプルコードのポートの設定を行います。サンプルコードでは LED1 の制御に P53 を使用します。

6.1.8 Config_IICA0

サンプルコードの IICA0 の設定を行います。

7. 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。

8. 参考ドキュメント

RL78/G23 ユーザーズマニュアル ハードウェア編 (R01UH0896J)

RL78 ファミリ ユーザーズマニュアル ソフトウェア編 (R01US0015J)

SMS アセンブラ ユーザーズマニュアル (R20UT4792J)

RL78 スマート・コンフィグレータ ユーザーガイド : CS+編 (R20AN0580J)

RL78 スマート・コンフィグレータ ユーザーガイド : e² studio 編 (R20AN0579J)

RL78 スマート・コンフィグレータ ユーザーガイド : IAR 編 (R20AN0581J)

HS300x Datasheet

(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

(最新版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。)

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	Sep.30.21	-	初版発行
1.10	Feb.15.22	3	湿度 70%に相当する 2CH が設定されています。⇒ 湿度 50%に相当する 20H が設定されています。
		5	図 1-3 全体フローチャート Wait 4s ⇒ Wait 62.5ms
		6	ツールバージョンを更新 表 2-1 動作確認条件 統合開発環境 (CS+) : CS+ for CC V8.06.00 ⇒ V8.07.00 C コンパイラ (CS+) : CC-RL V1.10 ⇒ V1.11 統合開発環境 (e ² studio) : e ² studio 2021-07 (21.7.0) ⇒ 2022-01 (22.1.0) C コンパイラ (e ² studio) : CC-RL V1.10 ⇒ V1.11 スマート・コンフィグレータ : V.1.0.1 ⇒ V.1.2.0 ボードサポートパッケージ (r_bsp) : V.1.11 ⇒ V.1.13
		10	表 5-1 動作概要 4 秒待つ ⇒ 62.5ms 待つ
		11	フォルダ名を更新
		12	表 5-4 サンプルコードで使用する定数 THRESHOLD : 2CH ⇒ 20H dtc_init[] を削除
		13	[関数名] main float.h ⇒ r_sms.h
		14	[関数名] r_sms_start addr_dtc_init 削除
		20	4 秒を待ちます ⇒ 62.5ms を待ちます
		21	図 5-8 SMS 処理 Initialize SMSG11, SMSG12 ⇒ Initialize SMSG10, SMSG11 Wait 4s ⇒ 62.5ms
		22	図 5-9 SMS 処理 SMSG12 ⇒ SMSG11 SMSG11 ⇒ SMSG10
		23	表 5-1 シーケンサ汎用レジスタ 0-15 サンプルプログラム更新に伴う更新
		24-26	表 5-2 シーケンサ・インストラクション・レジスタ 0-31 表 5-3 シーケンサ・インストラクション・レジスタ 0-31 表 5 10 シーケンサ・インストラクション・レジスタ 0-31 サンプルプログラム更新に伴う更新
		27	ファイル名を更新

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。