

S1C17W22/S1C17W23
温湿度計/塩分濃度計(RFC)
アプリケーションノート

評価ボード・キット、開発ツールご使用上の注意事項

1. 本評価ボード・キット、開発ツールは、お客様での技術的評価、動作の確認および開発のみに用いられることを想定し設計されています。それらの技術評価・開発等の目的以外には使用しないで下さい。本品は、完成品に対する設計品質に適合していません。
2. 本評価ボード・キット、開発ツールは、電子エンジニア向けであり、消費者向け製品ではありません。お客様において、適切な使用と安全に配慮願います。弊社は、本品を用いることで発生する損害や火災に対し、いかなる責も負いかねます。通常の使用においても、異常がある場合は使用を中止して下さい。
3. 本評価ボード・キット、開発ツールに用いられる部品は、予告無く変更されることがあります。

本資料のご使用につきましては、次の点にご留意願います。

本資料の内容については、予告無く変更することがあります。

1. 本資料の一部、または全部を弊社に無断で転載、または、複製など他の目的に使用することは堅くお断りいたします。
2. 本資料に掲載される応用回路、プログラム、使用方法等はあくまでも参考情報であり、これらに起因する第三者の知的財産権およびその他の権利侵害あるいは損害の発生に対し、弊社はいかなる保証を行うものではありません。また、本資料によって第三者または弊社の知的財産権およびその他の権利の実施権の許諾を行うものではありません。
3. 特性値の数値の大小は、数直線上の大小関係で表しています。
4. 製品および弊社が提供する技術を輸出等するにあたっては「外国為替および外国貿易法」を遵守し、当該法令の定める手続きが必要です。大量破壊兵器の開発等およびその他の軍事用途に使用する目的をもって製品および弊社が提供する技術を費消、再販売または輸出等しないでください。
5. 本資料に掲載されている製品は、生命維持装置その他、きわめて高い信頼性が要求される用途を前提としていません。よって、弊社は本（当該）製品をこれらの用途に用いた場合のいかなる責任についても負いかねます。
6. 本資料に掲載されている会社名、商品名は、各社の商標または登録商標です。

要旨

本資料は、S1C17W22 または S1C17W23 にて、R/F 変換器 (RFC) を用いて、以下のどちらかの測定を行うための参考資料です。

1. 温度センサと湿度センサの抵抗値の変化を計測し、温度と相対湿度を測定。
2. 電極間の抵抗値と温度センサの抵抗値を計測し、水溶液の塩分濃度と温度を測定。

動作環境

- S5U1C17W23T (以下 SVT17W23: Software Evaluation Tool for S1C17W23)
ICDmini との接続には専用ケーブルが必要です
- PC
 - ・ GNU17 (S5U1C17001C) 開発ツールインストール済み ※
 - ・ ICDmini USB ドライバインストール済み
- ICDmini (S5U1C17001H)
PC との接続には USB ケーブルが必要です
- S1C17W22/S1C17W23 用の以下のプログラミングパッケージ (本パッケージ)
 - ・ 温湿度センサのカウント値の評価用 プログラミングパッケージ
電極間抵抗の塩分濃度応答カウント値の評価にも用いています
(slc17w22_w23_rfc_temp_hum_analy_gnu17vx)
 - ・ 温湿度換算値のテスト用 プログラミングパッケージ
(slc17w22_w23_rfc_therm_hygro_meter_gnu17vx)
 - ・ 塩分濃度換算値のテスト用 プログラミングパッケージ
(slc17w22_w23_rfc_saltConc_meter_gnu17vx)

※ 本パッケージは、GNU17 v2.4.0/v3.0.5 で動作確認しています。

目 次

要旨	i
1. 仕様	3
2. 使用機能説明	4
3. 動作原理	6
3.1. R/F 変換器による温湿度計測	6
3.2. 湿度計算の原理	9
3.3. 湿度センサの抵抗計測方法について	11
3.4. R/F 変換器による塩分濃度計測	13
3.5. 塩分濃度計測の原理	14
4. ソフトウェア説明	15
4.1 s1c17w22_w23_rfc_temp_hum_analy_gnu17vx について	15
4.1.1 ファイル構成	15
4.1.2 モジュール説明	15
4.1.3 構造体	16
4.1.4 グローバル変数	16
4.1.5 操作手順	17
4.1.6 対話的操作で温湿度センサの特性を把握	18
4.1.7 サンプルプログラム動作概要	19
4.2 s1c17w22_w23_rfc_therm_hygro_meter_gnu17vx について	22
4.2.1 ファイル構成	22
4.2.2 モジュール説明	23
4.2.3 構造体	25
4.2.4 グローバル変数	26
4.2.5 操作手順	27
4.2.6 サンプルプログラム動作概要	28
4.3 s1c17w22_w23_rfc_saltConc_meter_gnu17vx について	32
4.3.1 ファイル構成	32
4.3.2 モジュール説明	33
4.3.3 構造体	35
4.3.4 グローバル変数	36
4.3.5 操作手順	37
4.3.6 サンプルプログラム動作概要	38
APPENDIX A. 温湿度テーブルの作成例	42
(1) センサの抵抗範囲の確認	42
(2) 外付け抵抗によるダイナミックレンジの調整	42
(3) 温度センサのカウント値の温度依存性の測定	43
(4) 湿度センサのカウント値の温湿度依存性の測定	44
(5) 温度測定結果の誤差の確認	44
(6) 湿度測定結果の誤差の確認	45
(7) 処理時間	46
(8) 計測カウント値の範囲	46
(9) 湿度の OSC1 の発振への影響	46
APPENDIX B. 塩分濃度換算用テーブルの作成例	47
(1) 塩分濃度センサ用電極の構造	47
(2) 塩分濃度センサの抵抗範囲の見積もり	47
(3) 外付け抵抗による抵抗値の調整	48

(4) 温度センサのカウント値の温度依存性の測定	48
(5) 塩分濃度センサのカウント値の 塩分濃度・温度 依存性の測定	49
(6) 塩分濃度測定結果の誤差の確認	49
(7) 処理時間	50
改訂履歴表	51

1. 仕様

本アプリケーションノートでは、S1C17W22/S1C17W23 に内蔵されている R/F 変換器 (RFC) を用いて、以下のどちらかの測定を行います。

1. 温度センサと湿度センサの抵抗値の変化を計測し、温度と相対湿度を測定。
2. 電極間の抵抗値と温度センサの抵抗値を計測し、水溶液の塩分濃度と温度を測定。

サンプルプログラムのうち、`slc17w22_w23_rfc_temp_hum_analy_gnu17vx` は、タイムベースカウンタの値を変えて、温度センサや湿度センサの抵抗値変化による計測カウント値を評価するためのもので、温度や湿度を変えてセンサの抵抗値変化を評価することに主眼を置いて作られたものです。このサンプルプログラムは設定パラメータの値を変えるだけで、電極間抵抗の塩分濃度応答カウント値の評価にも用いることができます。

またサンプルプログラムのうち、`slc17w22_w23_rfc_therm_hygro_meter_gnu17vx` は、16 ビットタイマ (T16) で一定時間ごと繰り返し温湿度センサの抵抗値を計測し、温湿度に換算した値を LCD に表示します。

同様にサンプルプログラムのうち、`slc17w22_w23_rfc_saltConc_meter_gnu17vx` は、16 ビットタイマ (T16) で一定時間ごと繰り返し電極間の抵抗値と温度センサの抵抗値を計測し、計測温度を用いて補正した塩分濃度の換算値と、温度に換算した値、SVD で計測した電源電圧の値を LCD に表示します。

2. 使用機能説明

2. 使用機能説明

RFC Ch.0	AC バイアス抵抗性センサである湿度センサを取り扱うため、AC 発振モードを使用して抵抗値を計測します。電極間の抵抗値を測定する場合も、電気二重層による容量の影響を最小化するため、AC 発振モードを使用して抵抗値を計測します。
RFC Ch.1	DC バイアス抵抗性センサである温度センサを取り扱うため、DC 発振モードを使用して抵抗値を計測します。
T16 Ch.0	一定時間毎に RFC による計測動作を開始して、温湿度センサの抵抗値、または、電極間抵抗値と温度センサの抵抗値を得るためのインターバルタイマ。

以下は、slc17w22_w23_rfc_therm_hygro_meter_gnu17vx の場合です。

LCD24	温湿度センサの抵抗値を温湿度に換算した値を LCD に表示します。
COPRO2	乗除算の演算を高速で処理するために使用しています。

動作モード	RFC Ch.0： センサ A 発振完了割り込み機能を用いて、湿度センサの抵抗値に依存した計測カウンタ値を得ます。 RFC Ch.1： センサ A 発振完了割り込み機能を用いて、温度センサの抵抗値に依存した計測カウンタ値を得ます。
-------	--

システムクロック	システムクロックは OSC1 (32.768kHz) を使用します。
----------	------------------------------------

割り込み	RFC Ch.0 のベクタ番号とベクタアドレスは以下の通りです。 RFC Ch.0 ベクタ番号： 23 (0x17) ベクタアドレス： 0x805c 本サンプルでは、次の 4 種類の割り込みを使用します。 基準発振完了割り込み センサ A 発振完了割り込み 計測カウンタオーバーフローエラー割り込み タイムベースカウンタオーバーフローエラー割り込み
------	--

	RFC Ch.1 のベクタ番号とベクタアドレスは以下の通りです。 RFC Ch.1 ベクタ番号： 24 (0x18) ベクタアドレス： 0x8060 本サンプルでは、次の 4 種類の割り込みを使用します。 基準発振完了割り込み センサ A 発振完了割り込み 計測カウンタオーバーフローエラー割り込み タイムベースカウンタオーバーフローエラー割り込み
--	--

	T16 Ch.0 のベクタ番号とベクタアドレスは以下の通りです。 T16 Ch.0 ベクタ番号： 9 (0x09) ベクタアドレス： 0x8024 本サンプルでは、次の割り込みを使用します。 アンダーフロー割り込み
--	--

以下は、slc17w22_w23_rfc_saltConc_meter_gnu17vx の場合です。

SVD	電極間抵抗値に影響を与える電源電圧値を確認するために、電源電圧を計測します。
LCD24	塩分濃度の計測値、温度センサの抵抗値を温度に換算した値、SVD で計測した電源電圧値を LCD に表示します。

COPRO2	乗除算の演算を高速で処理するために使用しています。
動作モード	RFC Ch.0： センサ A 発振完了割り込み機能を用いて、電極間の抵抗値に依存した計測カウンタ値を得ます。 RFC Ch.1： センサ A 発振完了割り込み機能を用いて、温度センサの抵抗値に依存した計測カウンタ値を得ます。
システムクロック	システムクロックは OSC1 (32.768kHz) を使用します。
割り込み	RFC Ch.0 のベクタ番号とベクタアドレスは以下の通りです。 RFC Ch.0 ベクタ番号： 23 (0x17) ベクタアドレス： 0x805c 本サンプルでは、次の 4 種類の割り込みを使用します。 基準発振完了割り込み センサ A 発振完了割り込み 計測カウンタオーバーフローエラー割り込み タイムベースカウンタオーバーフローエラー割り込み RFC Ch.1 のベクタ番号とベクタアドレスは以下の通りです。 RFC Ch.1 ベクタ番号： 24 (0x18) ベクタアドレス： 0x8060 本サンプルでは、次の 4 種類の割り込みを使用します。 基準発振完了割り込み センサ A 発振完了割り込み 計測カウンタオーバーフローエラー割り込み タイムベースカウンタオーバーフローエラー割り込み T16 Ch.0 のベクタ番号とベクタアドレスは以下の通りです。 T16 Ch.0 ベクタ番号： 9 (0x09) ベクタアドレス： 0x8024 本サンプルでは、次の割り込みを使用します。 アンダーフロー割り込み

図 1 に RFC の構成を示します。

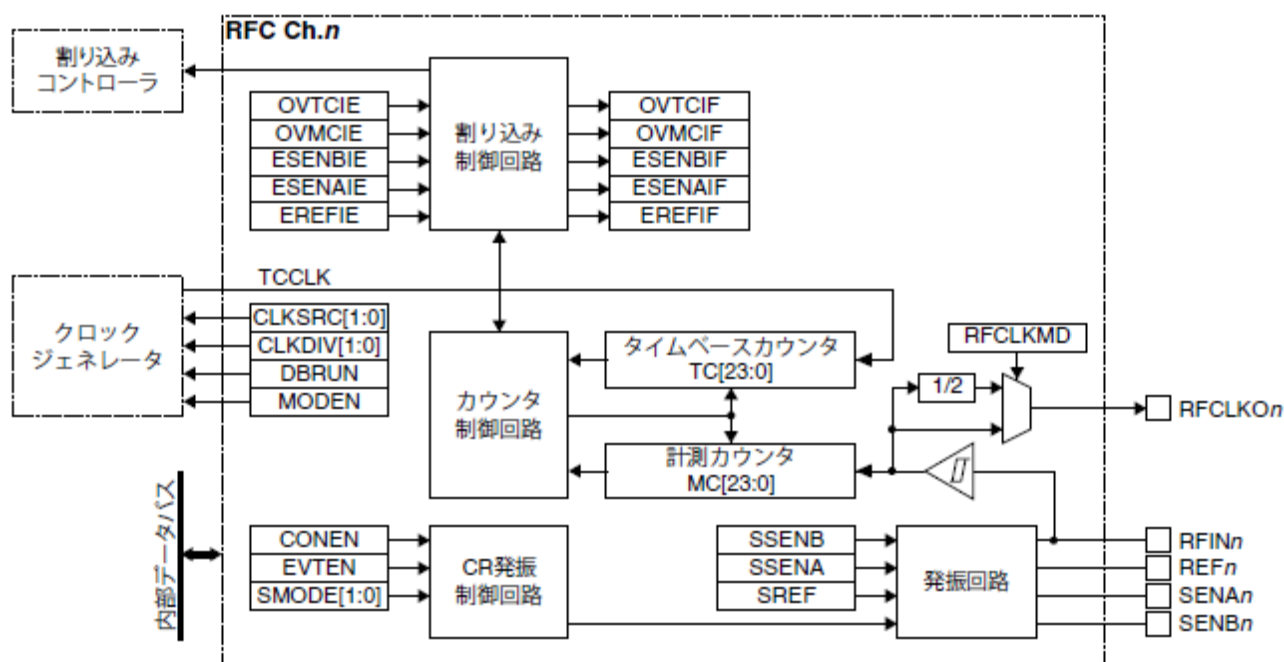


図 1 RFC の構成

3. 動作原理

3. 動作原理

この章では R/F 変換器と温湿度計測、塩分濃度計測の原理について説明します。

3.1. R/F 変換器による温湿度計測

S1C17 ファミリーには、CR 発振方式の A/D 変換器が内蔵されており、これを R/F 変換器と呼んでいます。R/F 変換器に接続されたセンサは、CR 発振回路によって発振クロックに変換され、その発振クロックを計測カウンタで計測します。このカウント値がデジタル化された値となります。図 3-1 に示すように、計測カウンタに初期値 (0x000000-n) を設定し、基準抵抗による発振 (基準発振) により計測カウンタがオーバーフローするまでの時間を求めます。次にこの時間を基準として、タイムベースカウンタがアンダーフローするまで、抵抗性温湿度センサによる発振 (センサ発振) によるクロック数をカウントします。

温度センサの場合は、このカウント値、事前に準備していたテーブルを用い、温度の値を算出します。

湿度センサの場合は、測定された温度の値、湿度センサのカウント値、事前に準備していたテーブルを用い、相対湿度の値を算出します。すなわち、相対湿度は温度依存性があるため、温度による補正が必要です。

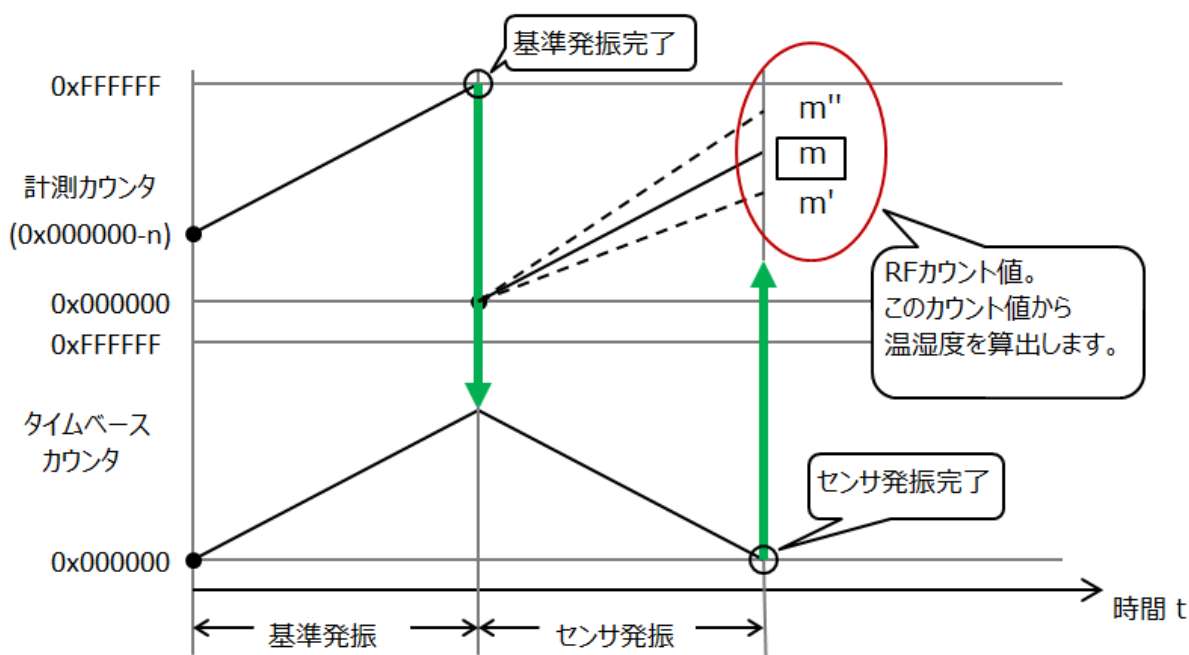


図 3-1 R/F 変換器動作 (基準発振とセンサ発振)

サンプルソフトウェアでは、サーミスタ (103AT-2B : SEMITEC 株式会社製) を接続した R/F 変換器の DC 発振モードにより温度計測を行います。また、抵抗性湿度センサ (CL-M52R : 神栄テクノロジー株式会社製) を接続した R/F 変換器の AC 発振モードにより、カウント値を得ます。サンプルソフトウェア中に用意してある湿度変換テーブルと測定された温度の値、湿度センサのカウント値より、一次補間して相対湿度の値を決定します。

サンプルソフトウェアは、図 3-2、図 3-3、表 3-1 に示した温度・湿度計測回路を前提としています。なお、この回路の部品の記号は、SVT17W23 (S5U1C17W23T) に使用されている記号をもとに割り振られています。

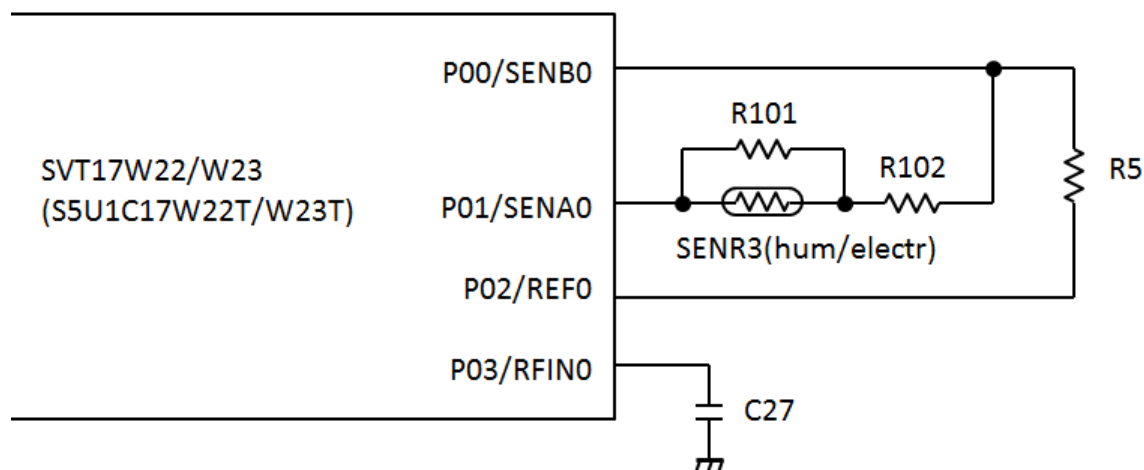


図 3-2 R/F 変換器 Ch0 湿度センサ計測用/塩分濃度センサ計測用 AC 発振モード接続

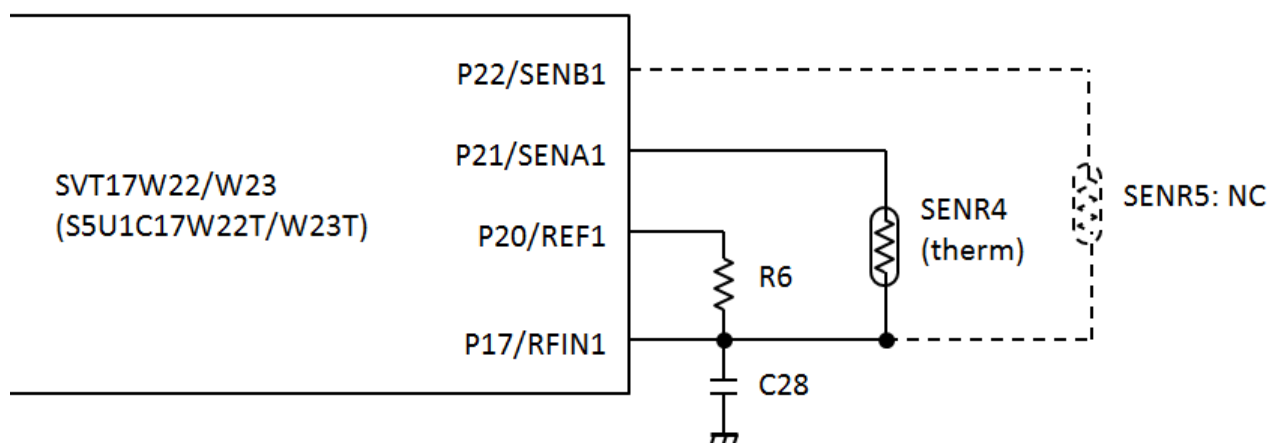


図 3-3 R/F 変換器 Ch1 温度センサ計測用 DC 発振モード接続

ここで用いている抵抗性湿度センサは、数 $k\Omega$ （高湿高温）から数十 $M\Omega$ （低湿低温）とダイナミックレンジが広いので、抵抗値が小さい領域では、IC 内部の抵抗による影響を強く受け、抵抗値が大きい領域では、カウント値が小さくなることにより誤差が増大します。これらの影響を軽減する為、計測に適する範囲になるように並列抵抗 R101 と直列抵抗 R102 を追加することが必要です。ソフトウェア設計時には、それらを考慮して湿度変換テーブルを作成します。

また SVT17W23（S5U1C17W23T）を用いて機能評価する場合、配線長を最小化するため、基板表面にシルク印刷で部品記号が示されている RF converter parts area（Ch0/Ch1）に部品を実装するようにし、拡張インターフェースコネクタ J1 は使用しないようにしてください。また部品のリード線が必要以上に長い場合、機械的な振動の影響を受けて計測値が不安定になる可能性があるため、考慮するようにしてください。

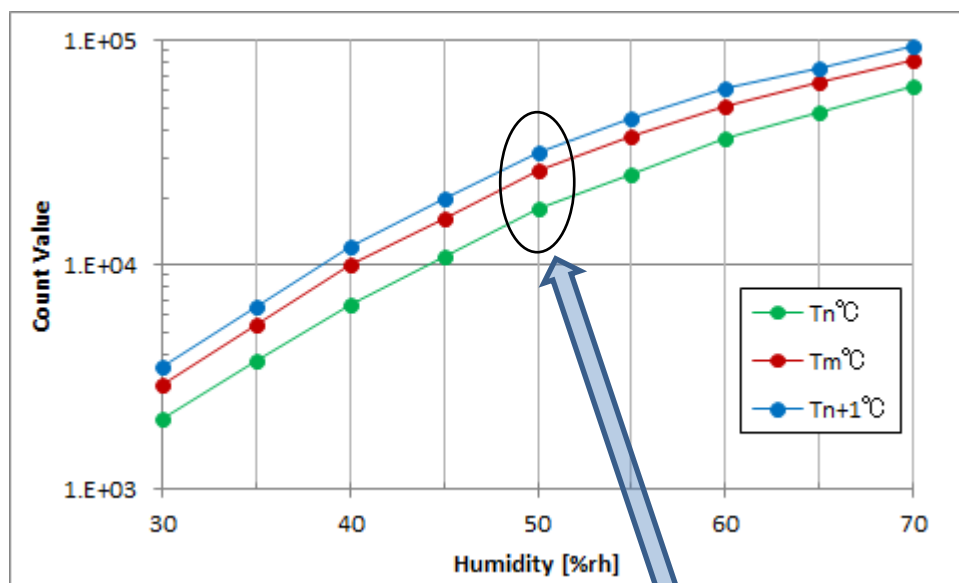
3. 動作原理

表 3-1 図 3-2 (湿度計測)、図 3-3 の R/F 変換器の BOM

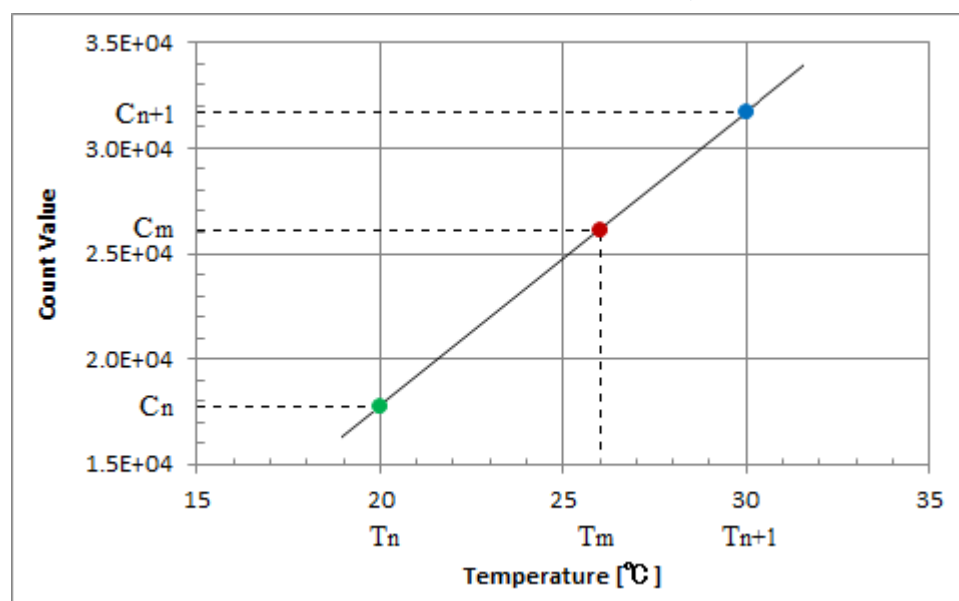
品名	記号	メーカー	メーカー品番	スペック
湿度センサ	SENR3	神栄テクノロジー	CL-M52R	31.0k Ω (typ.) at 25°C
温度センサ	SENR4	SEMITEC	103AT-2B	10.18k Ω $\pm$ 1% at 25°C
コンデンサ	C27,C28	muRata	GRM1552C1H102JA01	1000pF/50V 1005 CH
抵抗	R5	ROHM	MCR01 Series / MCR006YRTx Series	47k Ω 0603 F
	R6			10k Ω 0603 F
	R101			10M Ω 0603 F
	R102			4.3k Ω 0603 F

表 3-1 の BOM 中にあるセラミックコンデンサは、温度係数の小さな、スペックコード C0G、CH を使う方が、温湿度測定用途の場合はトラブルになりにくいと考えます。

3.2. 湿度計算の原理



(a)



(b)

図 3-4 湿度応答カウント値の温度補正

3. 動作原理

サーミスタで測定した抵抗値を温度の値に変換する際には、単純に一定ステップで温度を振った時の抵抗値のテーブルの値をもとに比例計算するだけで可能です。しかし湿度センサの抵抗値は、湿度のみならず温度にも依存して変化するため、それを考慮して変換する必要があります。

そのため、サーミスタで測定した温度の値を用いて、湿度換算に使用するテーブルデータを測定温度値に近接する2つの温度のテーブルから一次補間して作成します。例えば、 $T_m=26^{\circ}\text{C}$ の場合、テーブルが 10°C ごとに用意されているとすると、 $T_n=20^{\circ}\text{C}$ と $T_{n+1}=30^{\circ}\text{C}$ の2つのカウンタ値のテーブル、 C_n 、 C_{n+1} を使用し、これから温度値 $T_m=26^{\circ}\text{C}$ に相当するカウント値 C_m を計算により求めます。(図3-4の(a)のグラフの縦軸は対数目盛ですが、補間計算時は図3-4の(b)のグラフのようにリニア目盛として扱います。これは、対数で扱うと計算負荷が大きいので、それを回避するのがその理由です。)

$$C_m = C_n + (T_m - T_n) \cdot \frac{C_{n+1} - C_n}{T_{n+1} - T_n}$$

C_m : 一次補間で求められるカウント値

C_n : 温度 T_n におけるカウント値 = あらかじめ用意されたテーブル値

C_{n+1} : 温度 T_{n+1} におけるカウント値 = あらかじめ用意されたテーブル値

T_m : サーミスタで測定した温度値

T_n : 温度レンジ境界温度 (0°C 、 10°C 、 $\dots$ 50°C の 10°C ステップ)

T_{n+1} : $T_n+10^{\circ}\text{C}$ の温度レンジ境界温度

3.3. 湿度センサの抵抗計測方法について

本アプリケーションノートで使用している湿度センサは、高分子膜を湿度検出に用いるタイプで、抵抗測定時には交流電圧を印加する必要があります。

RFC を抵抗値の計測手段として用いる方法は、厳密な意味では交流電圧を印加する計測方法ではないため、RFC 方式で問題が生じないか、実際に湿度センサに RFC からの交流電圧を印加しながら、カウント値の挙動を調査しました。同時に、湿度センサに印加される交流電圧の周波数・波形がどのように変化するかについても確認するため、P40 端子を RFCLK00 に割り当て、その出力をオシロスコープで観察しました。

図 3-4 は、雰囲気温度 50℃に於いて、計測カウンタの初期値 (0x000000-n) の n の値 (n for MC と表記) を振った時に、湿度センサの計測カウンタの値 (Measured Count) の湿度依存性を調査した結果です。また、図 3-5 は、同計測時の RFCLK00 の周波数の湿度依存性の結果です。

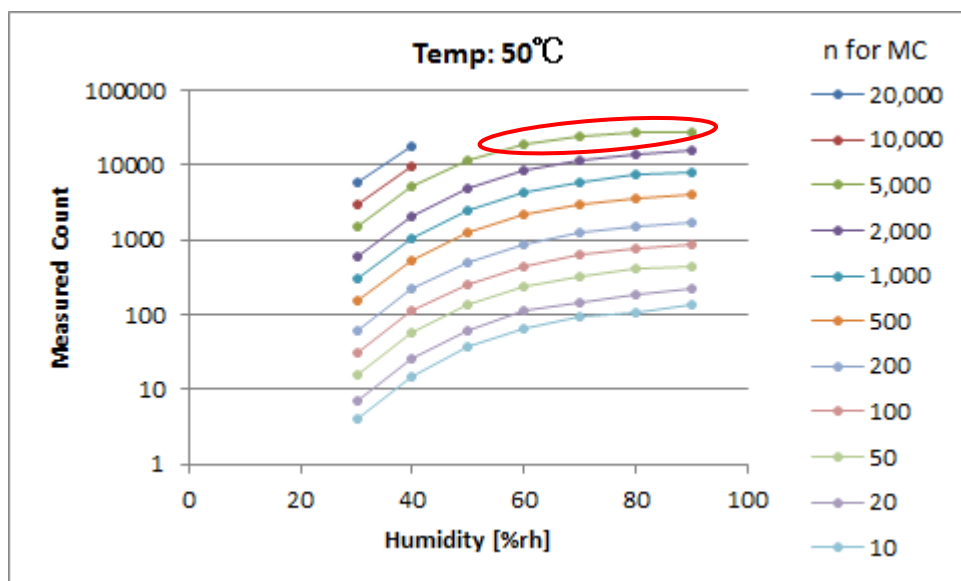


図 3-4 湿度センサの計測カウンタの値の湿度依存性

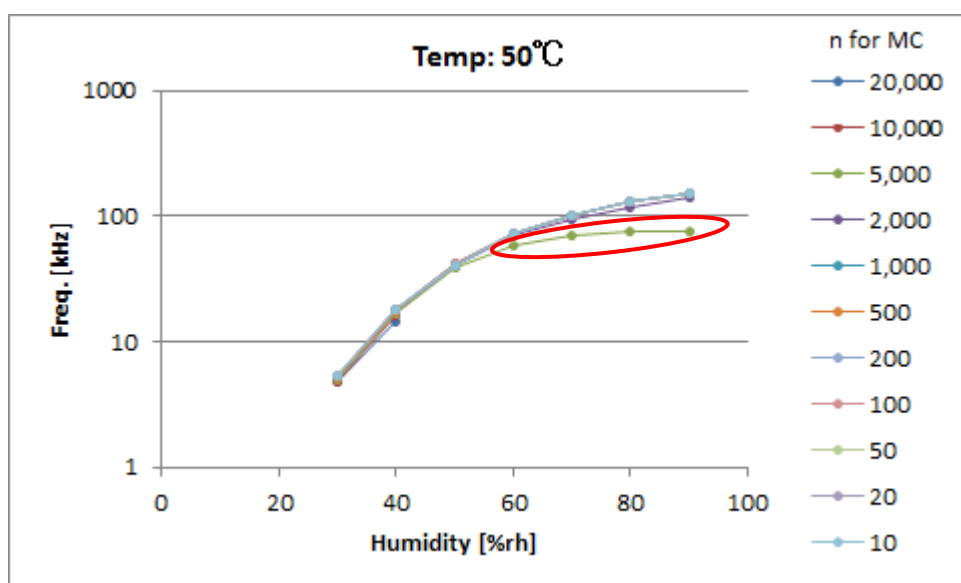


図 3-5 湿度センサの RFCLK00 の周波数の湿度依存性

3. 動作原理

両グラフの赤丸で囲んだエリアのデータは、他のデータと傾向が異なることを示しています。

図 3-4 の囲んだエリアのデータは、 $n=5,000$ の時の結果ですが、湿度応答が低下し、カーブが寝てきているのが見てとれると思います。よくよく見ると、その下の $n=2,000$ の時の結果も、若干カーブが寝てきているように見えます。このような挙動は、他の雰囲気温度に於いても、計測カウンタの値（縦軸の Measured Count のこと。基準発振用の n for MC ではありません）が 10,000 を超えたあたりから生じているように見受けられます。

また、図 3-5 の対応する結果では、RFCLK00 の周波数の湿度依存性の変化が他の条件の時から外れ、湿度が上がっても周波数が上がらないという応答になっていることがわかります。（注：25℃、30%rh の雰囲気下、周波数 100Hz～100kHz、交流バイアス 0.1V～1.0V の範囲で変えても、湿度センサの抵抗値は 340 k Ω ～500k Ω と、大きく変化しないことは LCR メーターで確認済みです。すなわち、湿度センサに印加される交流の周波数が計測途中に変わっても、抵抗値への影響は極端ではありません。）

このような現象が生じた時、湿度センサに印加される交流電圧の波形を RFCLK00 でモニターしたのが図 3-6 です。RFC0CTL.RFCLKMD レジスタの値=1 にして、1/2 分周クロックを出力しています。モニターした波形のタイミングは、湿度センサ印加交流の発振が停止する直前です。

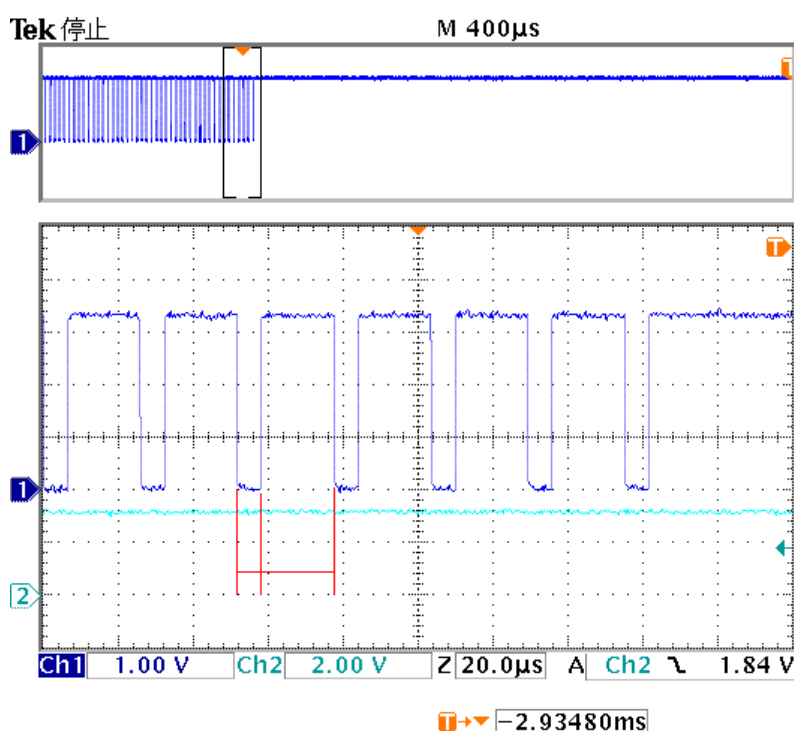


図 3-6 湿度センサに印加される交流波形の RFCLK00 によるモニター結果

発振開始からしばらくの間は当然 Duty 比=50%なのですが、発振停止する直前は Duty 比が 50%から大きく外れているのが観察されます。（更に n の値が大きくなると、Duty 比の外れはよりひどくなる。）

このように Duty 比が 50%から大きく外れた使い方は、高分子膜タイプの湿度センサの使い方のスペック（交流波形印加）を満たしているものではなく、 n の値をより大きくして計測カウンタの値（Measured Count）がだんだんと大きくなるに従い、計測値はますます不安定になる傾向が観察されます。具体的には、測定中に計測カウンタの値（Measured Count）がどんどん変わってしまい、一定の値に収束しないという状況に陥ります。すなわち、そうならないような使用条件にすることが必須で、その条件が先に示した、「計測カウンタの値（Measured Count）が 10,000 を超えないようにする。」です。

逆に計測カウンタの値（Measured Count）が小さすぎる場合には、量子化誤差の観点から精度が低下することになります。従い、計測カウンタの値（Measured Count）が一定の値の範囲になるように管理しながら RFC による湿度センサの抵抗値測定を行う必要があり、「Appendix A. 温湿度テーブルの作成例」ではその具体的な値について触れています。

3.4. R/F 変換器による塩分濃度計測

R/F 発振器の使い方は、温湿度計測の章での説明と基本的には同じです。特に水溶液の温度の測定については、使用しているテーブルの温度範囲とステップが異なる以外は全く同じです。

本アプリケーションノートの塩分濃度計は、水溶液の温度が計測値に与える影響を調査し、補正計算することで計測誤差をどこまで小さくできるかに主眼を置いて作製しました。そのため、測定された温度の値、塩分濃度センサのカウント値、事前に準備していたテーブルを用い、塩分濃度の値を算出します。同時に塩分濃度の計算値に影響を与える電源電圧の値も SVD で計測して表示しています。

サンプルソフトウェアでは、塩分濃度センサとしての電極を接続した AC 発振モードの R/F 変換器により、カウント値を得ます。サンプルソフトウェア中に用意してある塩分濃度変換テーブルと測定された温度の値、塩分濃度センサのカウント値より、一次補間して塩分濃度の値を決定します。

サンプルソフトウェアは、図 3-2、図 3-3、表 3-2 に示した塩分濃度・温度計測回路を前提としています。なお、この回路の部品の記号は、SVT17W23 (S5U1C17W23T) に使用されている記号をもとに割り振られています。

ここで用いている塩分濃度センサとしての電極は、高温・高塩分濃度の場合に抵抗が小さくなりすぎて、R/F 変換器の電気的特性のスペック、 $R_{REF}(\min)=1k\Omega$ を満たさなくなる可能性があります。また逆に水溶液に触れておらず、電極が空気中に置かれている場合は抵抗が ∞ となり、RFC0INTFOVMCIF ビット= 1 (計測カウンタオーバーフローエラー) でその状態は検出可能なものの、断線などの故障状態と区別するための方法も検討しておく必要があります。そこで湿度センサ同様、並列抵抗 R101 と直列抵抗 R102 を追加しました。ソフトウェア設計時には、これらを考慮して塩分濃度変換テーブルを作成します。

また SVT17W23 (S5U1C17W23T) を用いて機能評価する場合、温湿度計同様、配線長を最小化するため、基板表面にシルク印刷で部品記号が示されている RF converter parts area (Ch0/Ch1) に部品を実装するようにし、拡張インターフェースコネクタ J1 は使用しないようにしてください。また部品のリード線が必要以上に長い場合、機械的な振動の影響を受けて計測値が不安定になる可能性があるので、考慮するようにしてください。

表 3-2 図 3-2 (塩分濃度計測)、図 3-3 の R/F 変換器の BOM

品名	記号	メーカー	メーカー品番	スペック
塩分濃度センサ	SENR3	-	-	APPENDIX B. 参照
温度センサ	SENR4	SEMITEC	103AT-2B	10.18k Ω $\pm$ 1% at 25 $^{\circ}$ C
コンデンサ	C27,C28	muRata	GRM1552C1H102JA01	1000pF/50V 1005 CH
抵抗	R5	ROHM	MCR01 Series	2.2k Ω 0603 F
	R6			10k Ω 0603 F
	R101			1M Ω 0603 F
	R102			1k Ω 0603 F

表 3-2 の BOM 中にあるセラミックコンデンサは、温度係数の小さな、スペックコード C0G、CH を使う方が、塩分濃度測定用途の場合もトラブルになりにくいと考えます。

3. 動作原理

3.5. 塩分濃度計測の原理

本アプリケーションノートでは、身近な食品の塩分濃度を測ることを想定しています。電気伝導度と塩分濃度は大まかには比例関係にあり、精度を必要としない場合には簡便な塩分濃度計として使用することが可能です。塩分濃度センサには一対の電極を用いており、それらの間の電気的抵抗に相当する値を R/F 変換器で計測することで塩分濃度計を実現しています。

ただし、水溶液と電極の界面には電気二重層が存在するため、図 3-7 のような等価回路が電極間に形成されることになります。この等価回路に、周波数を変化させた微小交流電圧を印加し、電流の応答特性からインピーダンス (Z) を複素平面上にプロットしたものが図 3-8 のナイキストプロット (コールコールプロット) です。

十分に周波数が高い交流波形を電極間に印加することで、電気二重層の影響を取り去り、塩分濃度に応じて変化する溶液の電気的抵抗 R_s のみを抽出することができることがわかんと思います。

このように交流を用いて抵抗値を計測することで、電気分解が起こるほどの直流高電圧を印加して被測定物の組成を乱すことなく、溶液抵抗値を測定することができます。

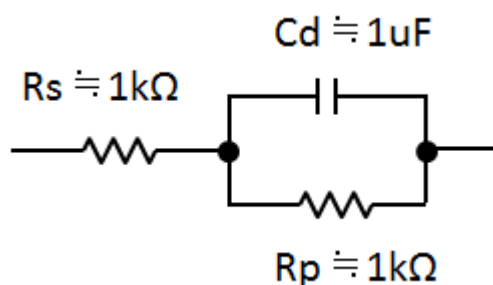


図 3-7 水溶液中の電極の等価回路

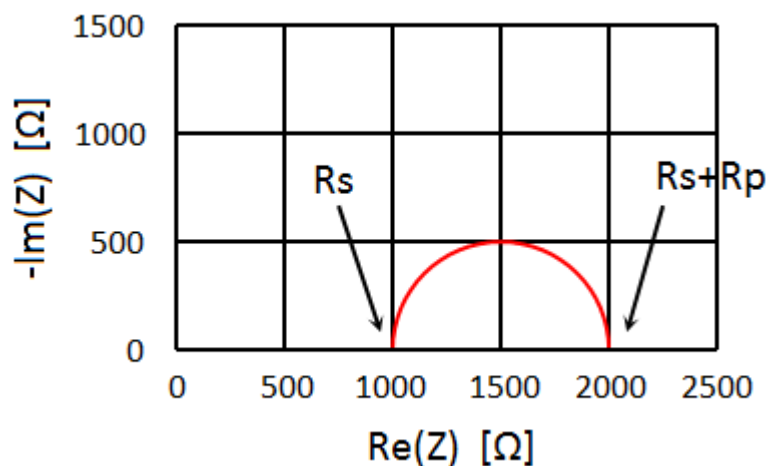


図 3-8 水溶液中の電極の等価回路のナイキストプロット (コールコールプロット)

湿度センサ同様、水溶液に浸した電極間の抵抗値は、塩分濃度のみならず温度にも依存して変化するため、それを考慮して変換する必要があります。

そのため、サーミスタで測定した溶液の温度の値を用いて、塩分濃度換算に使用するテーブルデータを測定温度値に近接する 2 つの温度のテーブルから一次補間して作成しますが、その計算は湿度計算の場合と同じですので、詳細については、「3.2. 温湿度計算の原理」の章を参照してください。

4. ソフトウェア説明

4.1 s1c17w22_w23_rfc_temp_hum_analy_gnu17vx について

温湿度を変えてセンサの抵抗値変化を評価する際に用いることに主眼を置いて作られた、タイムベースカウンタの値を変えて、温度センサや湿度センサの抵抗値変化による計測カウント値を評価するためのソフトウェア、s1c17w22_w23_rfc_temp_hum_analy_gnu17vx について説明します。

4.1.1 ファイル構成

ファイル名	機能
src/boot.c	スタートアップモジュールファイル
src/clg.c	CLG ドライバファイル
src/init.c	初期化関数ファイル
src/main.c	メイン関数ファイル
src/rfc.c	RFC ドライバファイル
src/t16_ch0.c	T16 Ch.0 ドライバファイル
src/crt0.h	ブート処理定義ファイル (GNU17 Version3 用)
src/clg.h	CLG ドライバヘッダ定義ファイル
src/init.h	初期化関数ヘッダ定義ファイル
src/rfc.h	RFC ドライバヘッダ定義ファイル
src/t16_ch0.h	T16 Ch.0 ドライバヘッダ定義ファイル
inc/reg	周辺機器レジスタ定義ファイル格納フォルダ
inc/c17w22_reg.h	周辺機器ヘッダ定義ファイル
inc/c17w23_reg.h	

4.1.2 モジュール説明

ファイル中のモジュールのうち、一般的でないものを中心に、関数名とその機能について説明します。

ファイル名： main.c

関数名	機能
measRfc	タイムベースカウンタの値を更新したのち、RFC の変換動作を開始します。
intT16Ch0	T16 Ch.0 割り込み関数。上記関数 measRfc を呼びます。

4. ソフトウェア説明

4.1.3 構造体

サンプルプログラム内で定義した構造体について記します。

構造体名： TEMP_HUM_ANALY

変数名	型	機能
stat0	unsigned short	RFC Ch.0 のステータス。
stat1	unsigned short	RFC Ch.1 のステータス。
measCountIndex0	unsigned short	RFC Ch.0 に設定する計測カウンタの初期値 (0x000000-n) のインデックス。
measCountIndex1	unsigned short	RFC Ch.1 に設定する計測カウンタの初期値 (0x000000-n) のインデックス。
dataPtr0	unsigned short	同じ構造体の sensorCount0*[#]の、どの#に RFC Ch.0 の計測値を格納するかを示すためのポインタ。
dataPtr1	unsigned short	同じ構造体の sensorCount1*[#]の、どの#に RFC Ch.1 の計測値を格納するかを示すためのポインタ。
sensorCount0L	unsigned short	同じ構造体の sensorCount0 の、下位 16bit。
sensorCount0H	unsigned short	同じ構造体の sensorCount0 の、上位 8bit。
sensorCount1L	unsigned short	同じ構造体の sensorCount1 の、下位 16bit。
sensorCount1H	unsigned short	同じ構造体の sensorCount1 の、上位 8bit。
measCount0	unsigned long	RFC Ch.0 に設定する計測カウンタの初期値 (0x000000-n)。
measCount1	unsigned long	RFC Ch.1 に設定する計測カウンタの初期値 (0x000000-n)。
sensorCount0	unsigned long	RFC Ch.0 で計測されたセンサのカウント値。
sensorCount1	unsigned long	RFC Ch.1 で計測されたセンサのカウント値。

4.1.4 グローバル変数

サンプルプログラム内で使用しているグローバル変数について記します。

ファイル名： src/rfc.c

変数名	型	機能
tha	TEMP_HUM_ANALY	RFC Ch.0、Ch.1 に関する設定値、計測値を保持するための構造体変数。

4.1.5 操作手順

このサンプルソフトウェアは、GNU17 Ver.2(以下、GNU17v2)および GNU17 Ver.3(以下、GNU17v3)の2種類のプロジェクト、および S1C17W22 と S1C17W23 の2種類のヘッダファイルを含みます。

サンプルソフトウェア使用前に以下の手順に従って、GNU17v2 または GNU17v3 のサンプルソフトウェアをターゲット機種に対応させてください。

- 1) 使用するターゲット機種のヘッダファイルを、サンプルソフトウェアの[inc]フォルダへコピーする。

例：ターゲット機種が S1C17W23 の場合、[inc_w23]フォルダの下の

"c17w23_reg.h"ファイルと[reg]フォルダを、サンプルソフトウェアの

[s1c17w22_w23_rfc_temp_hum_analy_gnu17vx]フォルダの下の[inc]フォルダへコピーしてください。

- 2) インポートウィザードから、「既存プロジェクトをワークスペースに」を開く。

[ファイル]>[インポート]>[一般]>[既存プロジェクトをワークスペースへ]

または

[File]>[Import]>[General]>[Existing Projects into Workspace]

- 3) サンプルプログラムが含まれるプロジェクトフォルダを選択。

GNU17v2 の場合 "s1c17w22_w23_rfc_temp_hum_analy_gnu17v2"プロジェクトを選択。

GNU17v3 の場合 "s1c17w22_w23_rfc_temp_hum_analy_gnu17v3"プロジェクトを選択。

- 4) "プロジェクトをワークスペースにコピー" または、"Copy projects into workspace"にチェックをつけ、[完了/Finish]ボタンを押す。

- 5) インポートしたプロジェクトを選択して機種設定を変更します。GNU17 のメニューから **GNU17v2 の場合**

[プロジェクト]>[プロパティ]を選択。

表示されたプロパティリストの中から[GNU17 一般設定]ダイアログボックスを選択し、[ターゲット CPU 機種]プルダウンメニューで対象機種を選択。[適用]ボタンを押す。

GNU17v3 の場合

[project]>[propaties]を選択。

表示されたプロパティリストの中から[GNU17 Setting]ダイアログボックスを選択し、[Target CPU]プルダウンメニューで対象機種を選択。[Apply]ボタンを押す。

4. ソフトウェア説明

6) (GNU17v2 のみ)

さらに、プロパティリストの中から[GNU17 GDB コマンド]ダイアログボックスを選択し、[雛形からコマンドを生成する]ボタンを押し、[デバッガ]プルダウンメニューから ICD Mini を選択、Flash ROM 書き込み実行チェック BOX にチェックを入れ、[上書き]ボタンを押し。

7) プロジェクトの"src"フォルダにあるヘッダファイル"init.h"の機種設定を変更します。

例： ターゲットが S1C17W23 の場合

```
// #define MCUSEL_C17W22          /// S1C17W22
#define MCUSEL_C17W23          /// S1C17W23
```

8) ビルド

IDE を使用して、「s1c17w22_w23_rfc_temp_hum_analy_gnu17vx」プロジェクトをビルドしてください。

9) 接続、電源投入

- (1) SVT17W23、ICDmini、USB ケーブル、PC を接続してください。
- (2) SVT17W23、ICDmini をリセットしてください。

10) 実行

IDE を使用して、「s1c17w22_w23_rfc_temp_hum_analy_gnu17vx」プロジェクトを実行してください。

4.1.6 対話的操作で温湿度センサの特性を把握

本サンプルプログラムは、計測カウンタの初期値 (0x000000-n) の n の値を動的に変更して、センサのカウント値を計測することが可能のように構成されています。

その計測カウンタの初期値 (0x000000-n) の n の値は、10、20、50、100、... 5M、10M の何れかになるようにするためには、tha.measCountIndex0、tha.measCountIndex1 の値を 0 (n=10) から 18 (n=10M) に変えます。すなわち、都度 (0x000000-n) の値を n から計算して設定しなくても良いように構成されています。

デフォルトの T16_CH0_INTERVAL_TIME の値の設定では、5 秒毎に温湿度センサの抵抗値を計測し、メモリ内にその結果を格納するよう動作します。その際、一度に RFC_DATA_SET_NUM=16 個の計測値をメモリ内に順に格納しますので、5 秒×(16-1)=75 秒間 実行すると、格納可能な領域すべてが、最新の計測データで更新されることになります。すなわち、プログラムの実行後 75 秒間強たつところで実行停止することで、一度に 16 セットの計測値の収集が可能です。

実際に実行するときには、計測値の収集の都度 tha.measCountIndex0、tha.measCountIndex1 の値を所望の値に変えてから実行することで、計測カウンタの初期値 (0x000000-n) の値を変えた場合のセンサのカウント値のデータを収集することになります。

なお、構造体 tha の値を変更するためには、Watch Expressions Window で、tha を入力後、[Add Watch] ボタンを押し、対応する変数の値を変えるだけです。

4.1.7 サンプルプログラム動作概要

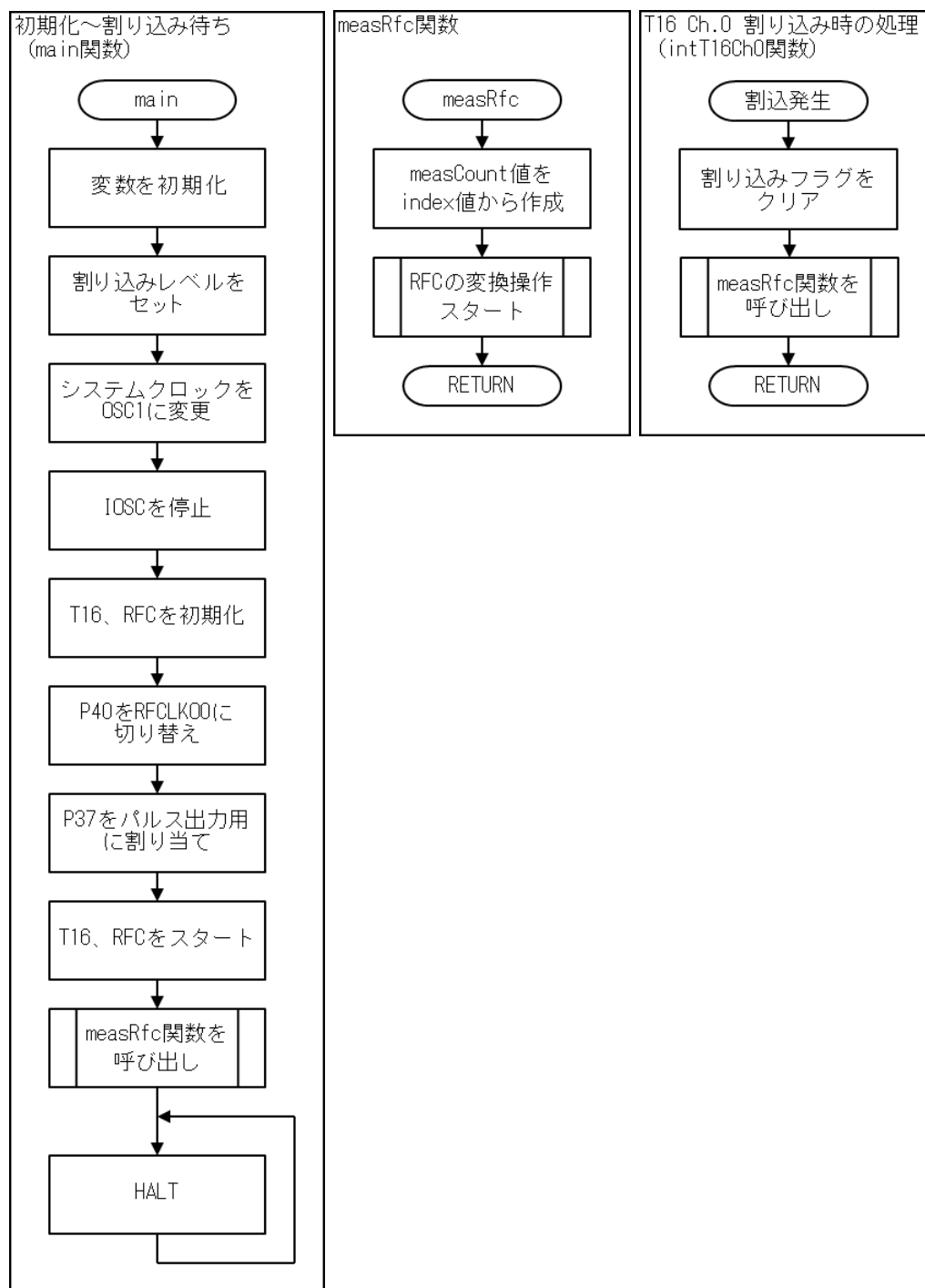
- ① パラメータや割り込みレベルの初期化を行います。
- ② システムクロックを IOSC から OSC1 (32.768kHz) に切り替え、IOSC を停止します。
- ③ T16 Ch.0、RFC Ch.0、RFC Ch.1 を初期化し、P40 を RFCLKO0 信号出力用に切り替え、P37 を RFC Ch.0 の変換期間モニターパルス出力用に設定します。
- ④ T16 Ch.0 のカウント動作を開始、RFC Ch.0、RFC Ch.1 を起動して計測開始関数を呼びます。
- ⑤ T16 Ch.0、RFC Ch.0、RFC Ch.1 の割り込みを待つために、HALT ループを繰り返します。
- ⑥ T16 Ch.0 割り込みがあるたび、計測開始関数を呼びます。

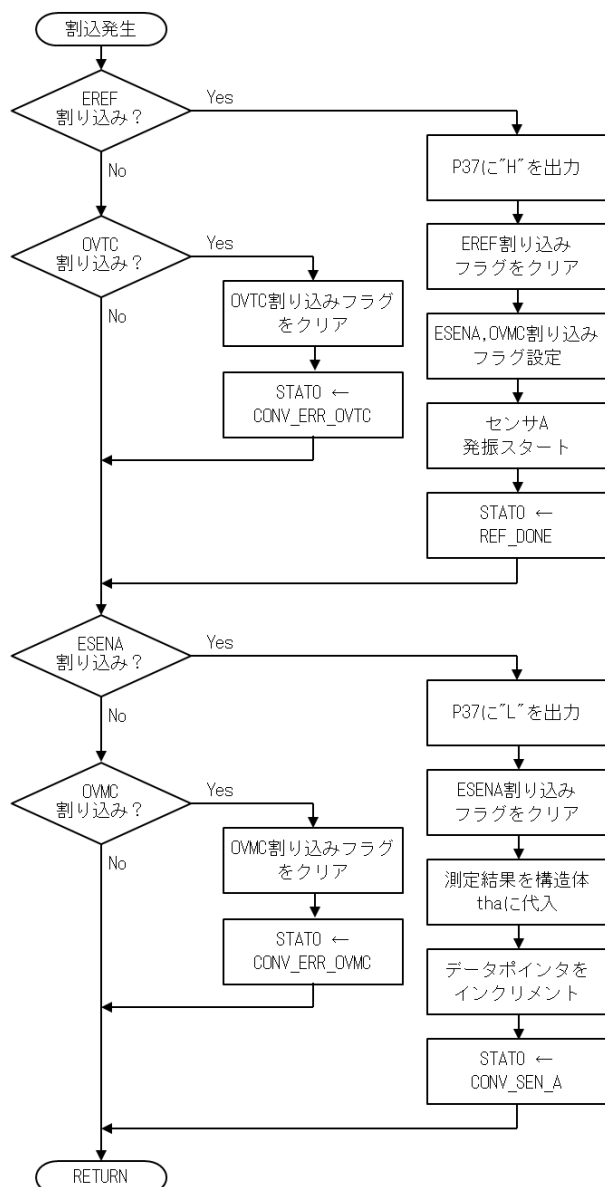
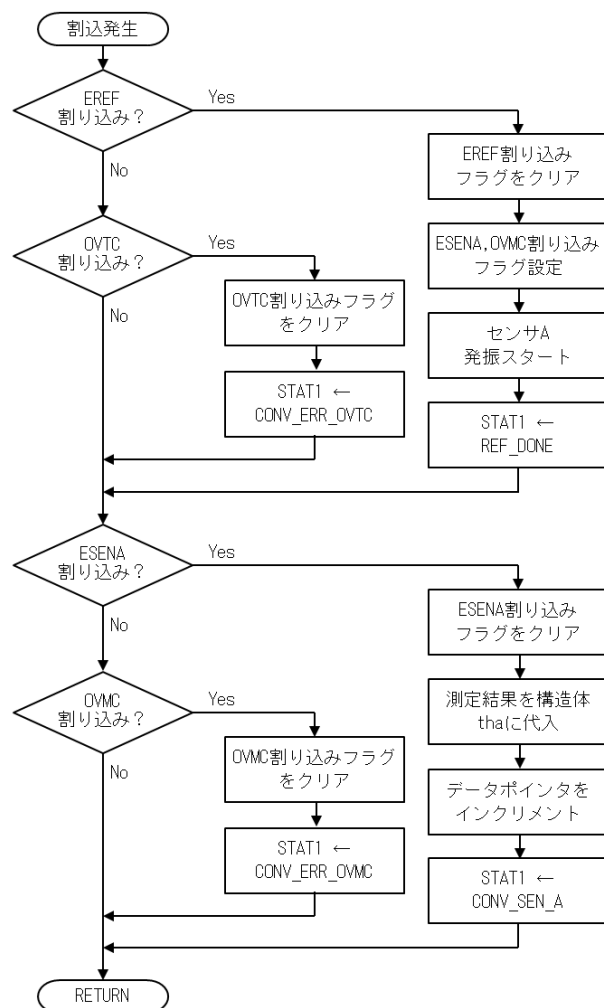
上記④の、RFC Ch.0、RFC Ch.1 の計測開始関数の動作について説明します。

- ① `tha.measCountIndex0`、`tha.measCountIndex1` の値に応じた計測カウンタの初期値 (0x000000-n) が、`tha.measCount0`、`tha.measCount1` に設定されます。
- ② RFC Ch.0、RFC Ch.1 による計測を開始します。
- ③ RFC Ch.0、RFC Ch.1 の割り込みがあるたび、割り込み要因を分析してステータスを更新し、センサのカウント値のデータを収集します。RFC Ch.0 の割り込みの時には、P37 に変換期間モニター用のパルスを出力します。

4. ソフトウェア説明

フローチャートを以下に記します。



RFC Ch.0 割り込み時の処理
(intRfc0関数)RFC Ch.1 割り込み時の処理
(intRfc1関数)

4. ソフトウェア説明

4.2 s1c17w22_w23_rfc_therm_hygro_meter_gnu17vx について

温湿度センサの抵抗値変化に応じて変わる RFC の計測カウント値を、温度と湿度の値に換算し、LCD にその値を表示するためのソフトウェア、s1c17w22_w23_rfc_therm_hygro_meter_gnu17vx について説明します。

4.2.1 ファイル構成

ファイル名	機能
src/boot.c	スタートアップモジュールファイル
src/clg.c	CLG ドライバファイル
src/display.c	LCD 表示関数ファイル
src/init.c	初期化関数ファイル
src/main.c	メイン関数ファイル
src/rfc_tbl.c	RFC 温湿度変換用テーブル、関数ファイル
src/rfc.c	RFC ドライバファイル
src/t16_ch0.c	T16 Ch.0 ドライバファイル
src/crt0.h	ブート処理定義ファイル (GNU17 Version3 用)
src/clg.h	CLG ドライバヘッダ定義ファイル
src/display.h	LCD 表示関数ヘッダ定義ファイル
src/init.h	初期化関数ヘッダ定義ファイル
src/lcd_font.h	LCD 表示フォント定義ファイル
src/rfc_tbl.h	RFC 温湿度変換用ヘッダ定義ファイル
src/rfc.h	RFC ドライバヘッダ定義ファイル
src/t16_ch0.h	T16 Ch.0 ドライバヘッダ定義ファイル
inc/reg	周辺機器レジスタ定義ファイル格納フォルダ
inc/c17w22_reg.h	周辺機器ヘッダ定義ファイル
inc/c17w23_reg.h	

4.2.2 モジュール説明

ファイル中のモジュールのうち、一般的でないものを中心に、関数名とその機能、配列変数名とその内容について説明します。

ファイル名： display.c

関数名	機能
selectLcdFont	LCD の指定フォントの配列のポインタを返します。
displayChar	LCD の指定行、指定文字位置に文字を表示します。
displayStr	LCD の指定行に文字列を表示します。
clearLcdLine	LCD の指定行の表示をクリアします。

ファイル名： main.c

関数名	機能
initLcd	SVT17W23 搭載の LCD を初期化します。
measRfc	湿度計測用の RFC Ch.0 のタイムベースカウンタの値を更新したのち、RFC Ch.0、RFC Ch.1 の変換動作を開始します。
intT16Ch0	T16 Ch.0 割り込み関数。上記関数 measRfc を呼びます。
val2Str3	unsigned short 型の値を、3 桁までの指定桁数の数値に変換して、文字列の指定位置に代入します。
dispVal	温度と湿度の値を LCD に表示します。測定範囲外の場合、その旨表示します。

ファイル名： rfc_tbl.c

配列変数名	内容
thermSensCount	-20℃から 50℃まで、1℃ステップの温度センサのカウント値。RFC Ch.1 に設定する計測カウンタの初期値（0x000000-n）の n=THERM_MEAS_COUNT=20,000 の時の RF カウントの実測値。
hygroCoef	0℃から 50℃まで、10℃ステップ、10%rh から 90%rh まで、5%rh ステップの湿度センサのカウント値。RFC Ch.0 に設定する計測カウンタの初期値（0x000000-n）の n=HYGRO_COEF_MULT=16,384 の時の RF カウントの実測値。

4. ソフトウェア説明

ファイル名 : rfc_tbl.c

関数名	機能
initRfcTbl	グローバル変数の構造体 tht の配列要素の値を初期化します。
getTemp	RFC Ch.1 で計測されたセンサのカウント値から温度を計算し、thm.temp100 に温度を 100 倍した値を、thm.tempReason に MEAS_DONE を設定します。範囲外の場合、thm.temp100 に OUT_OF_RANGE を、thm.tempReason に OUT_OF_LOWER_LIM または OUT_OF_UPPER_LIM を設定します。
getHum	thm.temp100 をもとに当該温度での湿度テーブルを作成し、RFC Ch.0 で計測されたセンサのカウント値から湿度を計算し、thm.hum10 に湿度を 10 倍した値を、thm.humReason に MEAS_DONE を設定します。範囲外の場合、thm.hum10 に OUT_OF_RANGE を設定しますが、thm.humReason には、温度が範囲外の場合は OUT_OF_TEMP_RANGE を、温度は範囲内でも湿度が範囲外の場合は OUT_OF_LOWER_LIM または OUT_OF_UPPER_LIM を設定します。
setHumMeasCount	thm.measCountIndex0 で示された計測カウント値を、thm.measCount0 に格納します。
changeHumMeasCount	thm.sensorCount0 の値が指定範囲内に収まるよう、thm.measCountIndex0 の値を調整します。

4.2.3 構造体

サンプルプログラム内で定義した構造体について記します。

構造体名： THERM_HYGRO_METER (ファイル：src/rfc.h)

変数名	型	機能
stat0	unsigned short	RFC Ch.0 のステータス。
stat1	unsigned short	RFC Ch.1 のステータス。
temp100	signed short	RFC Ch.1 で計測されたセンサのカウント値から計算した温度を 100 倍した値。
hum10	unsigned short	同じ構造体の temp100 をもとに当該温度での湿度テーブルを作成し、RFC Ch.0 で計測されたセンサのカウント値から計算した湿度を 10 倍した値。
tempReason	unsigned short	同じ構造体の temp100 が OUT_OF_RANGE の時、OUT_OF_LOWER_LIM (下限より低い温度) または、OUT_OF_UPPER_LIM (上限より高い温度)。
humReason	unsigned short	同じ構造体の hum10 が OUT_OF_RANGE の時、OUT_OF_TEMP_RANGE (温度が範囲外) または、OUT_OF_LOWER_LIM (下限より低い湿度) または、OUT_OF_UPPER_LIM (上限より高い湿度)。
measCountIndex0	unsigned short	RFC Ch.0 に設定する計測カウンタの初期値 (0x000000-n) のインデックス。
measCount0	unsigned long	RFC Ch.0 に設定する計測カウンタの初期値 (0x000000-n)。
sensorCount0	unsigned long	RFC Ch.0 で計測されたセンサのカウント値。
sensorCount1	unsigned long	RFC Ch.1 で計測されたセンサのカウント値。

4. ソフトウェア説明

構造体名： THERM_HYGRO_TABLE (ファイル：src/rfc_tbl.h)

変数名	型	機能
dThermTemp100	unsigned short	THERM_TEMP_STEP (配列 thermSensCount の温度ステップ) の値を 100 倍した値。
dHygroTemp10	unsigned short	HYGRO_TEMP_STEP (配列 hygroCoef の温度ステップ) の値を 10 倍した値。
dHygroHum10	unsigned short	HYGRO_HUM_STEP (配列 hygroCoef の湿度ステップ) の値を 10 倍した値。
thermTemp100	signed short	配列 thermSensCount の各温度の値を 100 倍した値。
dThermSensCount	unsigned short	配列 thermSensCount の各カウンタ値間の差 (勾配に比例する値)。
hygroTemp10	signed short	配列 hygroCoef の各温度の値を 10 倍した値。
hygroHum10	unsigned short	配列 hygroCoef の各湿度の値を 10 倍した値。
newHygroCoef	unsigned long	計測された温度 thm.temp100 をもとに補正した、配列 hygroCoef から作成した湿度テーブル。

4.2.4 グローバル変数

サンプルプログラム内で使用しているグローバル変数について記します。

ファイル名： src/rfc.c

変数名	型	機能
thm	THERM_HYGRO_METER	RFC Ch.0、Ch.1 から得られた温湿度の計算結果、測定条件を保持するための構造体変数。

ファイル名： src/rfc_tbl.c

変数名	型	機能
tht	THERM_HYGRO_TABLE	RFC Ch.0、Ch.1 から得られた計測値を温湿度に換算する際に、計算を高速化するために用いる値を保持するための構造体変数。

4.2.5 操作手順

このサンプルソフトウェアは、GNU17 Ver.2(以下、GNU17v2)および GNU17 Ver.3(以下、GNU17v3)の2種類のプロジェクト、および S1C17W22 と S1C17W23 の2種類のヘッダファイルを含みます。

サンプルソフトウェア使用前に以下の手順に従って、GNU17v2 または GNU17v3 のサンプルソフトウェアをターゲット機種に対応させてください。

- 1) 使用するターゲット機種のヘッダファイルを、サンプルソフトウェアの[inc]フォルダへコピーする。

例：ターゲット機種が S1C17W23 の場合、[inc_w23]フォルダの下の

"c17w23_reg.h"ファイルと[reg]フォルダを、サンプルソフトウェアの

[s1c17w22_w23_rfc_therm_hygro_meter_gnu17vx]フォルダの下の[inc]フォルダへコピーしてください。

- 2) インポートウィザードから、「既存プロジェクトをワークスペースに」を開く。

[ファイル]>[インポート]>[一般]>[既存プロジェクトをワークスペースへ]

または

[File]>[Import]>[General]>[Existing Projects into Workspace]

- 3) サンプルプログラムが含まれるプロジェクトフォルダを選択。

GNU17v2 の場合 "s1c17w22_w23_rfc_therm_hygro_meter_gnu17v2"プロジェクトを選択。

GNU17v3 の場合 "s1c17w22_w23_rfc_therm_hygro_meter_gnu17v3"プロジェクトを選択。

- 4) "プロジェクトをワークスペースにコピー" または、"Copy projects into workspace"にチェックをつけ、[完了/Finish]ボタンを押す。

- 5) インポートしたプロジェクトを選択して機種設定を変更します。GNU17 のメニューから **GNU17v2 の場合**

[プロジェクト]>[プロパティ]を選択。

表示されたプロパティリストの中から[GNU17 一般設定]ダイアログボックスを選択し、[ターゲット CPU 機種]プルダウンメニューで対象機種を選択。[適用]ボタンを押す。

GNU17v3 の場合

[project]>[propaties]を選択。

表示されたプロパティリストの中から[GNU17 Setting]ダイアログボックスを選択し、[Target CPU]プルダウンメニューで対象機種を選択。[Apply]ボタンを押す。

4. ソフトウェア説明

6) (GNU17v2 のみ)

さらに、プロパティリストの中から[GNU17 GDB コマンド]ダイアログボックスを選択し、[雛形からコマンドを生成する]ボタンを押し、[デバッガ]プルダウンメニューから ICD Mini を選択、Flash ROM 書き込み実行チェック BOX にチェックを入れ、[上書き]ボタンを押し。

7) プロジェクトの"src"フォルダにあるヘッダファイル"init.h"の機種設定を変更します。

例： ターゲットが S1C17W23 の場合

```
// #define MCUSEL_C17W22          /// S1C17W22  
#define MCUSEL_C17W23          /// S1C17W23
```

8) ビルド

IDE を使用して、「s1c17w22_w23_rfc_therm_hygro_meter_gnu17vx」プロジェクトをビルドしてください。

9) 接続、電源投入

- (1) SVT17W23、ICDmini、USB ケーブル、PC を接続してください。
- (2) SVT17W23、ICDmini をリセットしてください。

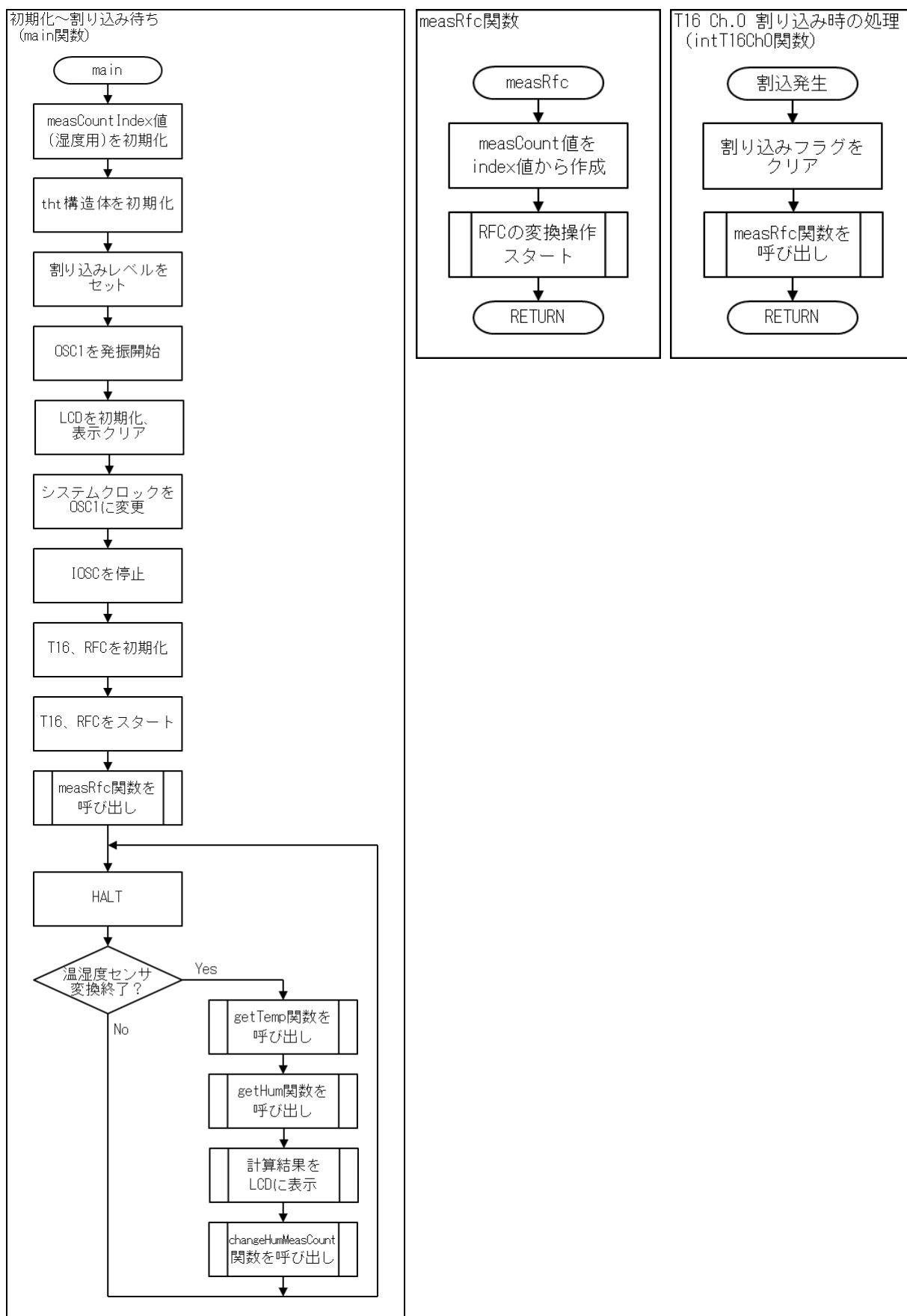
10) 実行

IDE を使用して、「s1c17w22_w23_rfc_therm_hygro_meter_gnu17vx」プロジェクトを実行してください。

4.2.6 サンプルプログラム動作概要

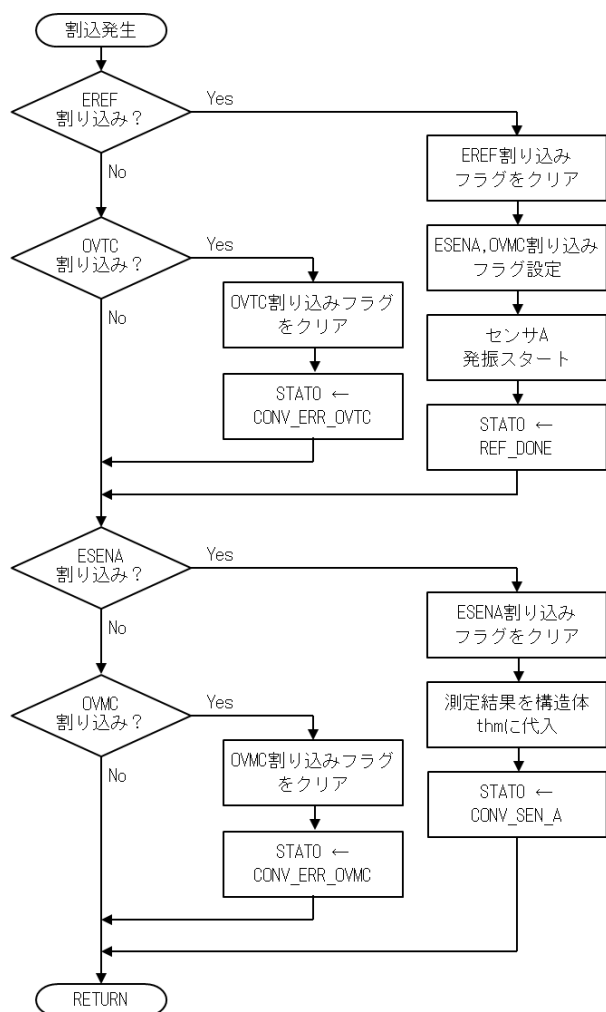
- ① 構造体変数 tht の値や割り込みレベルの初期化を行います。
- ② OSC1 (32.768kHz) を発振開始します。
- ③ LCD を初期化し、表示をクリアします。
- ④ システムクロックを IOSC から OSC1 に切り替え、IOSC を停止します。
- ⑤ T16 Ch.0、RFC Ch.0、RFC Ch.1 を初期化します。
- ⑥ T16 Ch.0 のカウント動作を開始、RFC Ch.0、RFC Ch.1 を起動して計測開始関数を呼びます。
- ⑦ T16 Ch.0、RFC Ch.0、RFC Ch.1 の割り込みを待つために、HALT します。
- ⑧ RFC Ch.0、RFC Ch.1 の両方とも計測が終了したなら、温度と湿度の値を計算し、結果を LCD に表示します。その後、関数 changeHumMeasCount を呼び、thm.sensorCount0 の値が指定範囲内に収まるよう、thm.measCountIndex0 の値を調整した後、⑦に処理を移して HALT します。
- ⑨ T16 Ch.0 割り込みがあるたび、計測開始関数を呼びます。

フローチャートを以下に記します。

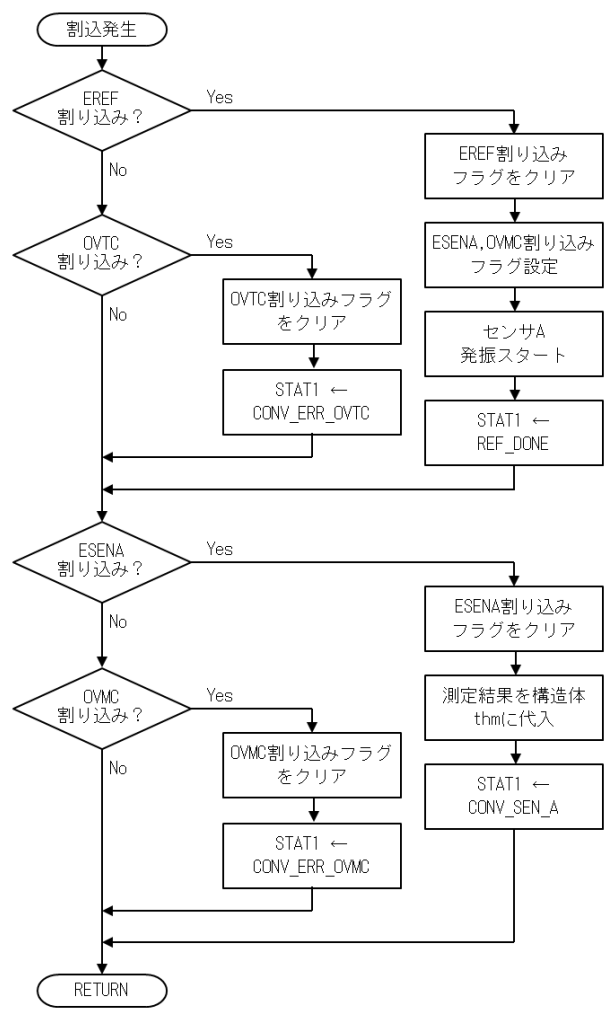


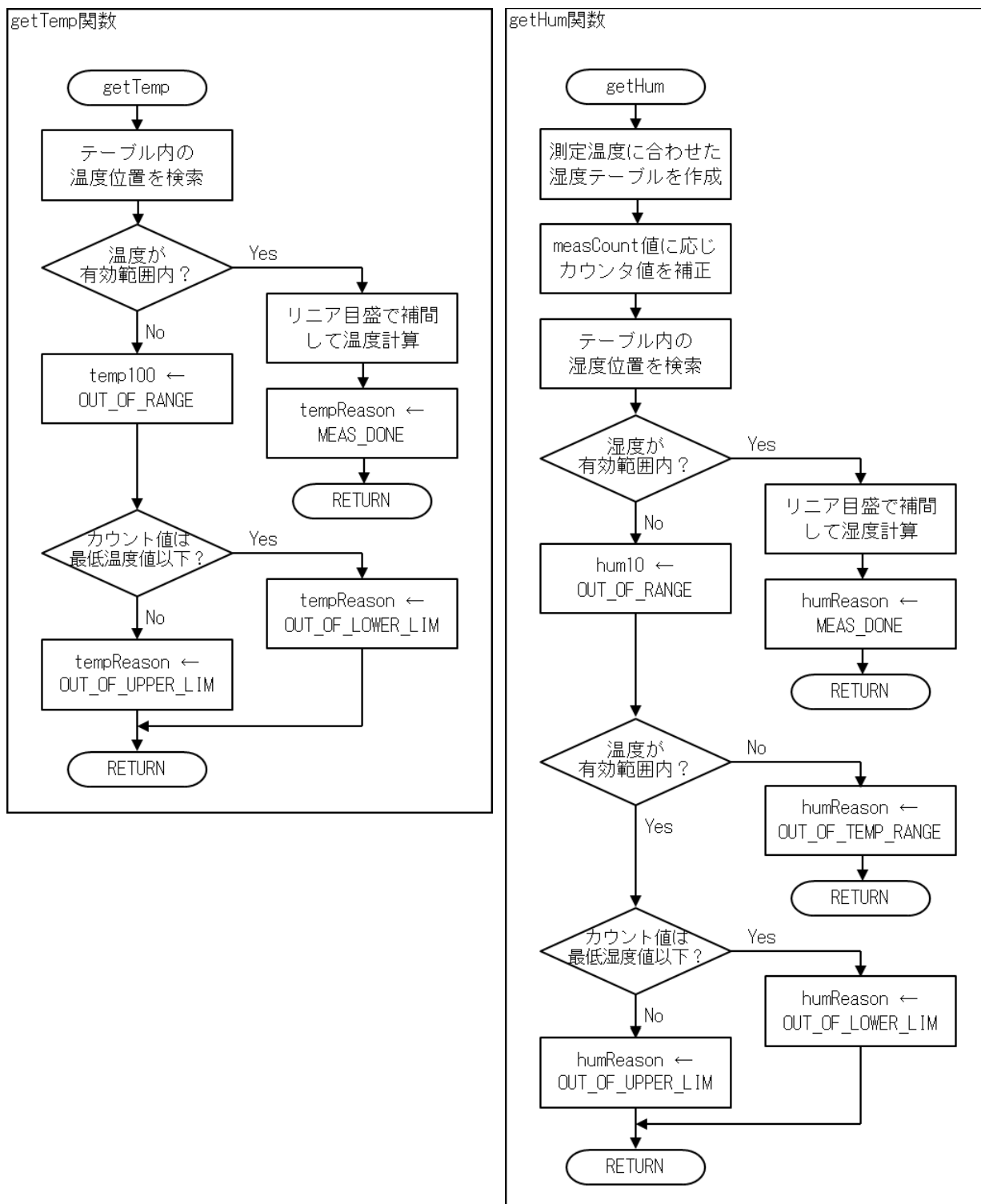
4. ソフトウェア説明

RFC Ch.0 割り込み時の処理
(intRfc0関数)



RFC Ch.1 割り込み時の処理
(intRfc1関数)





4. ソフトウェア説明

4.3 s1c17w22_w23_rfc_saltConc_meter_gnu17vx について

電極間の抵抗値と温度センサの抵抗値に応じて変わる RFC の計測カウント値を、塩分濃度の値と温度に換算し、LCD にその値を表示するためのソフトウェア、s1c17w22_w23_rfc_saltConc_meter_gnu17vx について説明します。

4.3.1 ファイル構成

ファイル名	機能
src/boot.c	スタートアップモジュールファイル
src/clg.c	CLG ドライバファイル
src/display.c	LCD 表示関数ファイル
src/init.c	初期化関数ファイル
src/main.c	メイン関数ファイル
src/rfc_tbl.c	RFC 塩分濃度、温度変換用のテーブル、関数ファイル
src/rfc.c	RFC ドライバファイル
src/svd.c	SVD ドライバファイル
src/t16_ch0.c	T16 Ch.0 ドライバファイル
src/crt0.h	ブート処理定義ファイル (GNU17 Version3 用)
src/clg.h	CLG ドライバヘッダ定義ファイル
src/display.h	LCD 表示関数ヘッダ定義ファイル
src/init.h	初期化関数ヘッダ定義ファイル
src/lcd_font.h	LCD 表示フォント定義ファイル
src/rfc_tbl.h	RFC 塩分濃度、温度変換用ヘッダ定義ファイル
src/rfc.h	RFC ドライバヘッダ定義ファイル
src/svd.h	SVD ドライバヘッダ定義ファイル
src/t16_ch0.h	T16 Ch.0 ドライバヘッダ定義ファイル
inc/reg	周辺機器レジスタ定義ファイル格納フォルダ
inc/c17w22_reg.h inc/c17w23_reg.h	周辺機器ヘッダ定義ファイル

4.3.2 モジュール説明

ファイル中のモジュールのうち、一般的でないものを中心に、関数名とその機能、配列変数名とその内容について説明します。（ただし、プログラミングパッケージ slc17w22_w23_rfc_therm_hygro_meter_gnu17vx と同じ内容のファイルの説明は割愛します。）

ファイル名： main.c

関数名	機能
initLcd	SVT17W23 搭載の LCD を初期化します。
measRfc	RFC Ch.0、RFC Ch.1 の変換動作を開始します。
intT16Ch0	T16 Ch.0 割り込み関数。上記関数 measRfc を呼びます。
val2Str3	unsigned short 型の値を、3 桁までの指定桁数の数値に変換して、文字列の指定位置に代入します。
dispVal	塩分濃度の値、温度、電源電圧の値を LCD に表示します。測定範囲外の場合、その旨表示します。

ファイル名： rfc_tbl.c

配列変数名	内容
thermSensCount	0℃から 100℃まで、2℃ステップの温度センサのカウンタ値。RFC Ch.1 に設定する計測カウンタの初期値（0x000000-n）の n=THERM_MEAS_COUNT=5,000 の時の RF カウンタの実測値。
salconCoef	Vdd=3.3V の時の、5℃から 85℃まで、20℃ステップ、0.40%から 1.40%まで、0.25%ステップの計測用電極（塩分濃度センサ）のカウンタ値。RFC Ch.0 に設定する計測カウンタの初期値（0x000000-n）の n=SALCON_MEAS_COUNT=1,000 の時の RF カウンタの実測値。

4. ソフトウェア説明

ファイル名 : rfc_tbl.c

関数名	機能
initRfcTbl	グローバル変数の構造体 tst の配列要素の値を初期化します。
getTemp	RFC Ch.1 で計測されたセンサのカウント値から温度を計算し、tsm.temp100 に温度を 100 倍した値を、tsm.tempReason に MEAS_DONE を設定します。範囲外の場合、tsm.temp100 に OUT_OF_RANGE を、tsm.tempReason に OUT_OF_LOWER_LIM または OUT_OF_UPPER_LIM を設定します。
getConc	SVD で計測した電源電圧に応じた値を tsm.svdVolt にセットし、tsm.temp100 をもとに当該温度での塩分濃度テーブルを作成し、RFC Ch.0 で計測されたセンサのカウント値から塩分濃度を計算し、tsm.conc1000 に塩分濃度%値を 1000 倍した値を、tsm.concReason に MEAS_DONE を設定します。範囲外の場合、tsm.conc1000 に OUT_OF_RANGE を設定しますが、tsm.concReason には、温度が範囲外の場合は OUT_OF_TEMP_RANGE を、温度は範囲内でも塩分濃度が範囲外の場合は OUT_OF_LOWER_LIM または OUT_OF_UPPER_LIM を設定します。

4.3.3 構造体

サンプルプログラム内で定義した構造体について記します。

構造体名： THERM_SALCON_METER (ファイル：src/rfc.h)

変数名	型	機能
stat0	unsigned short	RFC Ch.0 のステータス。
stat1	unsigned short	RFC Ch.1 のステータス。
temp100	signed short	RFC Ch.1 で計測されたセンサのカウント値から計算した温度を 100 倍した値。
conc1000	unsigned short	同じ構造体の temp100 をもとに当該温度での塩分濃度テーブルを作成し、RFC Ch.0 で計測されたセンサのカウント値から計算した塩分濃度%値を 1000 倍した値。
svdVolt	unsigned short	SVD で計測された電源電圧を、inc/svd.h に定義された SVDCTL_SVDC_????MV で示した値。
tempReason	unsigned short	同じ構造体の temp100 が OUT_OF_RANGE の時、OUT_OF_LOWER_LIM (下限より低い温度) または、OUT_OF_UPPER_LIM (上限より高い温度)。
concReason	unsigned short	同じ構造体の conc1000 が OUT_OF_RANGE の時、OUT_OF_TEMP_RANGE (温度が範囲外) または、OUT_OF_LOWER_LIM (下限より低い塩分濃度) または、OUT_OF_UPPER_LIM (上限より高い塩分濃度)。
sensorCount0	unsigned long	RFC Ch.0 で計測されたセンサのカウント値。
sensorCount1	unsigned long	RFC Ch.1 で計測されたセンサのカウント値。

4. ソフトウェア説明

構造体名： THERM_SALCON_TABLE (ファイル：src/rfc_tbl.h)

変数名	型	機能
dThermTemp100	unsigned short	THERM_TEMP_STEP (配列 thermSensCount の温度ステップ) の値を 100 倍した値。
dSalconTemp10	unsigned short	SALCON_TEMP_STEP (配列 salconCoef の温度ステップ) の値を 10 倍した値。
dSalconConc1000	unsigned short	SALCON_CONC_STEP100 (配列 salconCoef の塩分濃度ステップ、%単位×100) の値を 10 倍した値。
thermTemp100	signed short	配列 thermSensCount の各温度の値を 100 倍した値。
dThermSensCount	unsigned short	配列 thermSensCount の各カウンタ値間の差 (勾配に比例する値)。
salconTemp10	signed short	配列 salconCoef の各温度の値を 10 倍した値。
salconConc1000	unsigned short	配列 salconCoef の各塩分濃度の%値を 1000 倍した値。
newSalconCoef	unsigned long	計測された温度 tsm.temp100 をもとに補正した、配列 salconCoef から作成した塩分濃度テーブル。

4.3.4 グローバル変数

サンプルプログラム内で使用しているグローバル変数について記します。

ファイル名： src/rfc.c

変数名	型	機能
tsm	THERM_SALCON_METER	RFC Ch.0、Ch.1 から得られた塩分濃度、温度の計算結果、測定条件を保持するための構造体変数。

ファイル名： src/rfc_tbl.c

変数名	型	機能
tst	THERM_SALON_TABLE	RFC Ch.0、Ch.1 から得られた計測値を塩分濃度、温度に換算する際に、計算を高速化するために用いる値を保持するための構造体変数。

4.3.5 操作手順

このサンプルソフトウェアは、GNU17 Ver.2(以下、GNU17v2)および GNU17 Ver.3(以下、GNU17v3)の2種類のプロジェクト、および S1C17W22 と S1C17W23 の2種類のヘッダファイルを含みます。

サンプルソフトウェア使用前に以下の手順に従って、GNU17v2 または GNU17v3 のサンプルソフトウェアをターゲット機種に対応させてください。

- 1) 使用するターゲット機種のヘッダファイルを、サンプルソフトウェアの[inc]フォルダへコピーする。

例：ターゲット機種が S1C17W23 の場合、[inc_w23]フォルダの下の

"c17w23_reg.h"ファイルと[reg]フォルダを、サンプルソフトウェアの

[s1c17w22_w23_rfc_saltConc_meter_gnu17vx]フォルダの下の[inc]フォルダへコピーしてください。

- 2) インポートウィザードから、「既存プロジェクトをワークスペースに」を開く。

[ファイル]>[インポート]>[一般]>[既存プロジェクトをワークスペースへ]

または

[File]>[Import]>[General]>[Existing Projects into Workspace]

- 3) サンプルプログラムが含まれるプロジェクトフォルダを選択。

GNU17v2 の場合 "s1c17w22_w23_rfc_saltConc_meter_gnu17v2"プロジェクトを選択。

GNU17v3 の場合 "s1c17w22_w23_rfc_saltConc_meter_gnu17v3"プロジェクトを選択。

- 4) "プロジェクトをワークスペースにコピー" または、"Copy projects into workspace"にチェックをつけ、[完了/Finish]ボタンを押す。

- 5) インポートしたプロジェクトを選択して機種設定を変更します。GNU17 のメニューから **GNU17v2 の場合**

[プロジェクト]>[プロパティ]を選択。

表示されたプロパティリストの中から[GNU17 一般設定]ダイアログボックスを選択し、[ターゲット CPU 機種]プルダウンメニューで対象機種を選択。[適用]ボタンを押す。

GNU17v3 の場合

[project]>[propaties]を選択。

表示されたプロパティリストの中から[GNU17 Setting]ダイアログボックスを選択し、[Target CPU]プルダウンメニューで対象機種を選択。[Apply]ボタンを押す。

4. ソフトウェア説明

6) (GNU17v2 のみ)

さらに、プロパティリストの中から[GNU17 GDB コマンド]ダイアログボックスを選択し、[雛形からコマンドを生成する]ボタンを押し、[デバッガ]プルダウンメニューから ICD Mini を選択、Flash ROM 書き込み実行チェック BOX にチェックを入れ、[上書き]ボタンを押し。

7) プロジェクトの"src"フォルダにあるヘッダファイル"init.h"の機種設定を変更します。

例： ターゲットが S1C17W23 の場合

```
// #define MCUSEL_C17W22          /// S1C17W22  
#define MCUSEL_C17W23          /// S1C17W23
```

8) ビルド

IDE を使用して、「s1c17w22_w23_rfc_saltConc_meter_gnu17vx」プロジェクトをビルドしてください。

9) 接続、電源投入

(1) SVT17W23、ICDmini、USB ケーブル、PC を接続してください。

(2) SVT17W23、ICDmini をリセットしてください。

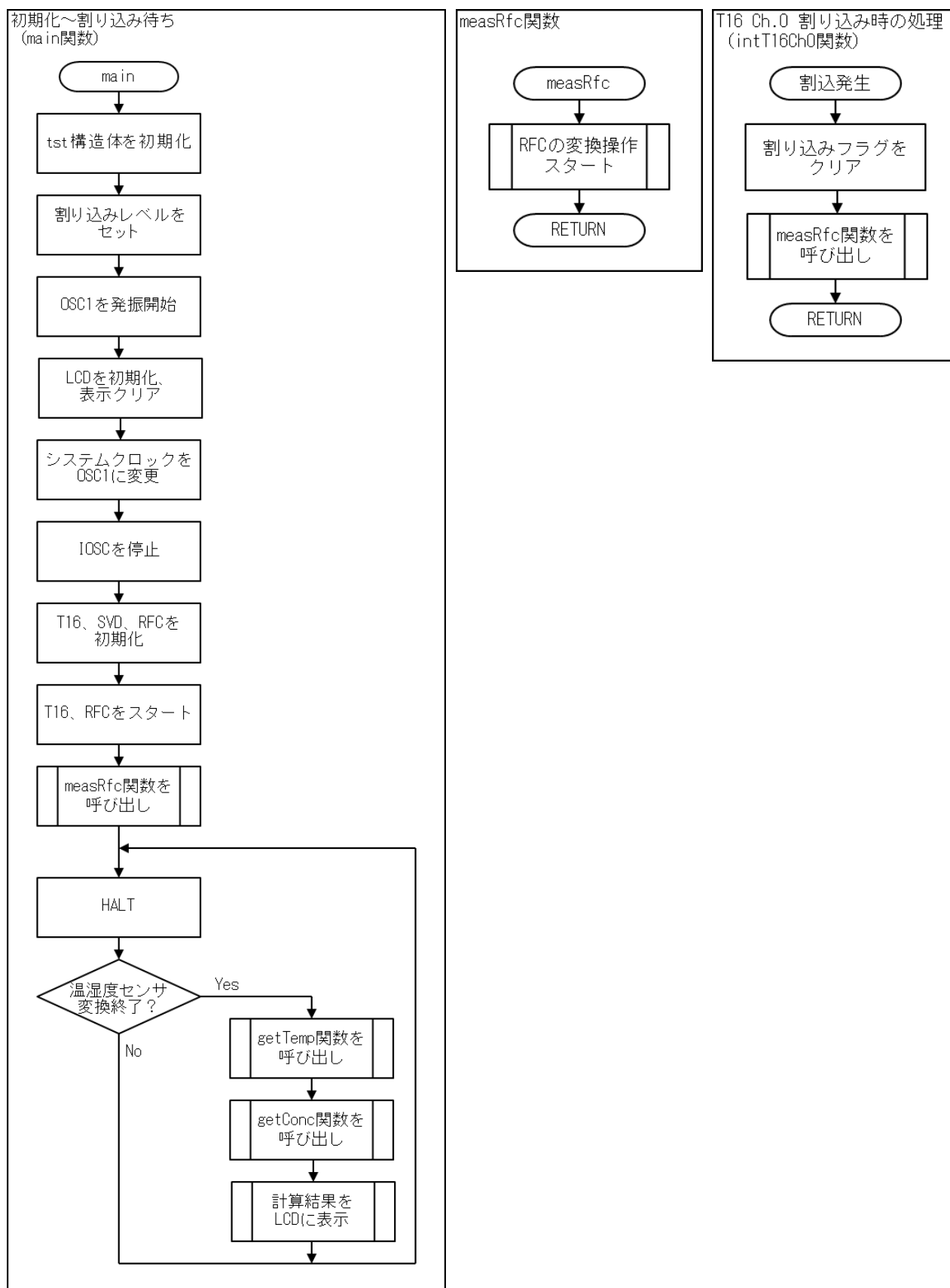
10) 実行

IDE を使用して、「s1c17w22_w23_rfc_saltConc_meter_gnu17vx」プロジェクトを実行してください。

4.3.6 サンプルプログラム動作概要

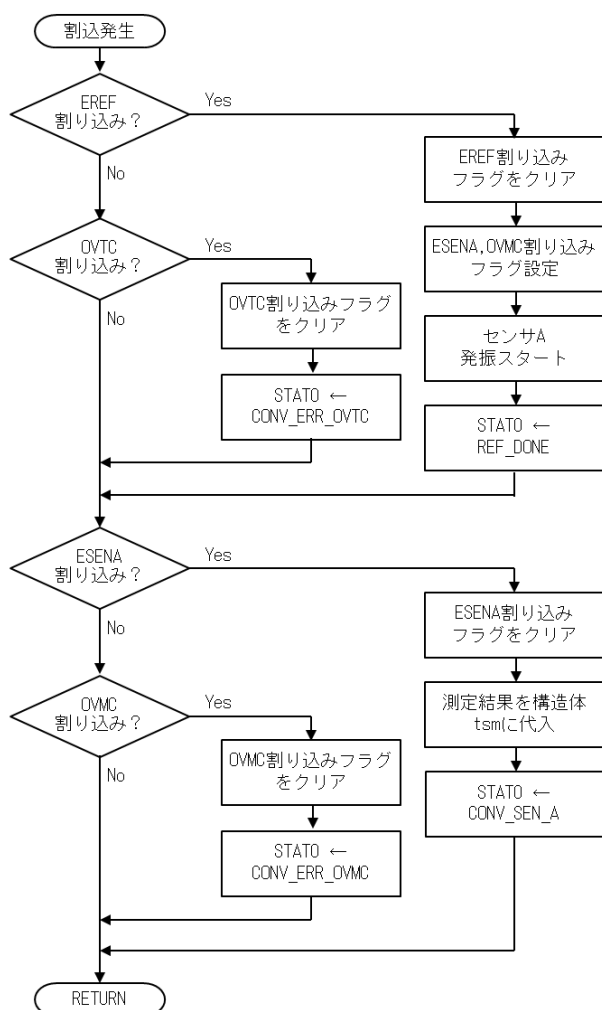
- ① 構造体変数 tst の値や割り込みレベルの初期化を行います。
- ② OSC1 (32.768kHz) を発振開始します。
- ③ LCD を初期化し、表示をクリアします。
- ④ システムクロックを IOSC から OSC1 に切り替え、IOSC を停止します。
- ⑤ T16 Ch.0、SVD、RFC Ch.0、RFC Ch.1 を初期化します。
- ⑥ T16 Ch.0 のカウント動作を開始、RFC Ch.0、RFC Ch.1 を起動して計測開始関数を呼びます。
- ⑦ T16 Ch.0、RFC Ch.0、RFC Ch.1 の割り込みを待つために、HALT します。
- ⑧ RFC Ch.0、RFC Ch.1 の両方とも計測が終了したなら、塩分濃度、温度の値を計算し、SVD による電源電圧の検出値と共に結果を LCD に表示します。その後、⑦に処理を移して HALT します。
- ⑨ T16 Ch.0 割り込みがあるたび、計測開始関数を呼びます。

フローチャートを以下に記します。

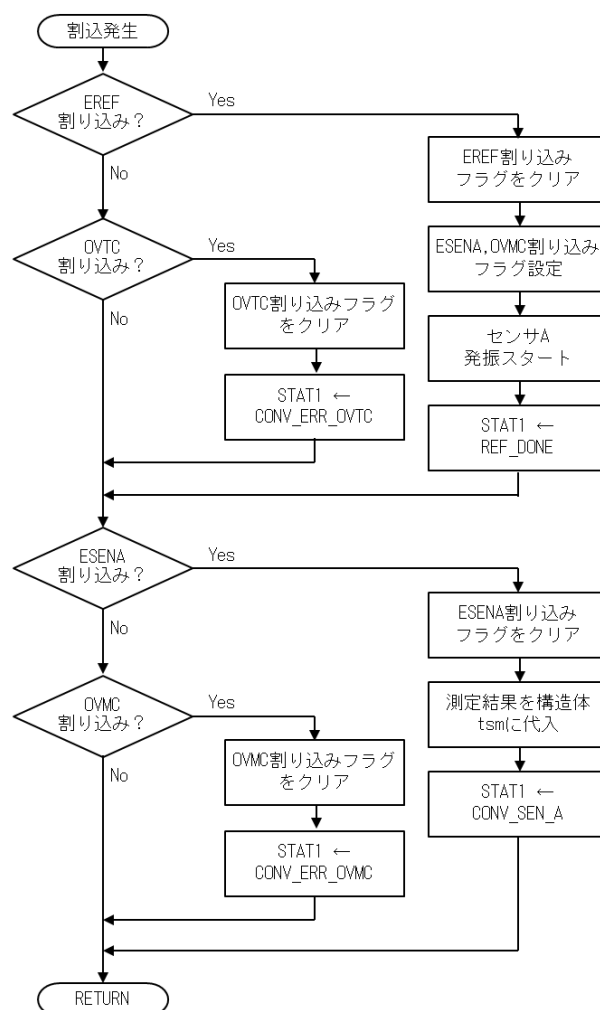


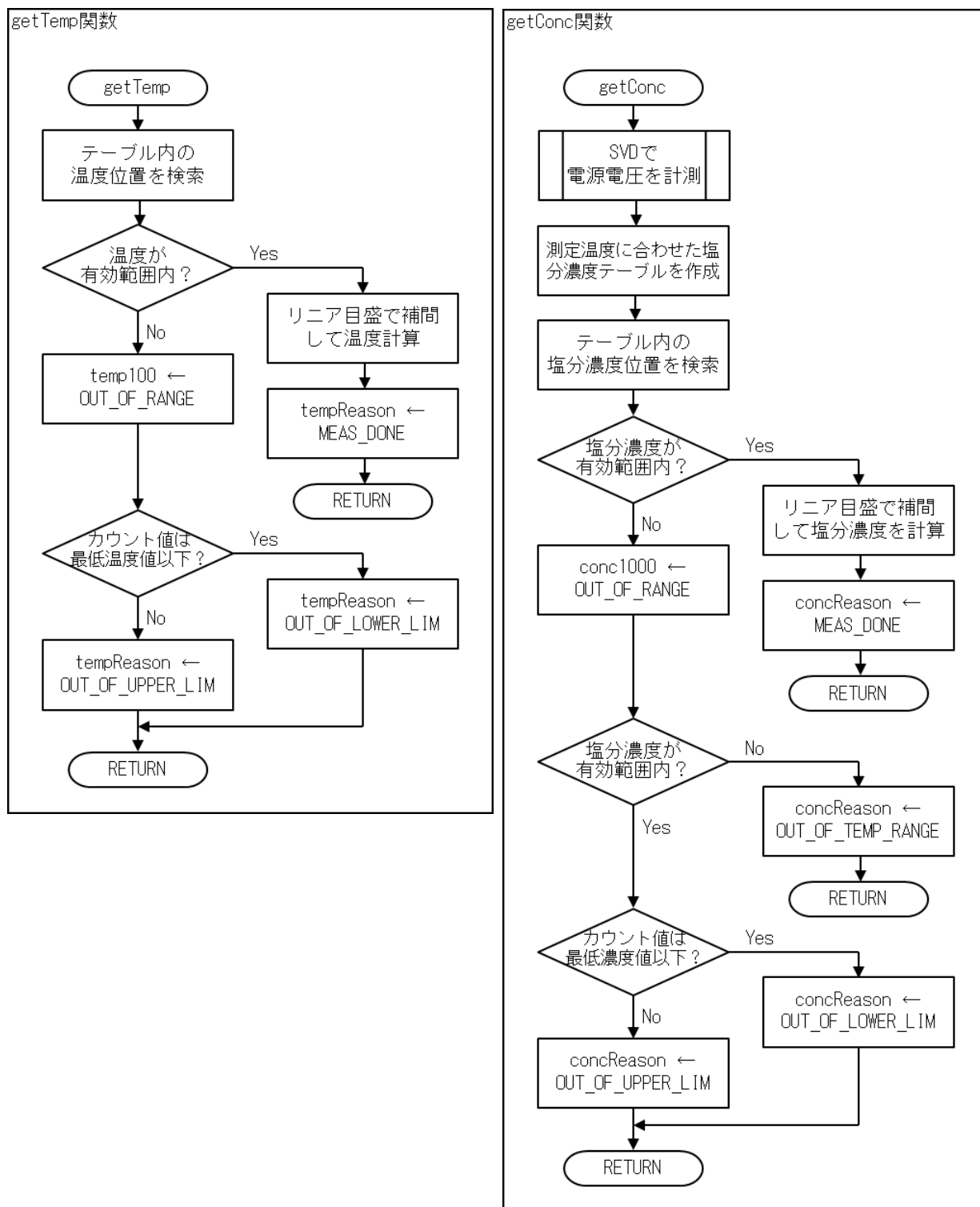
4. ソフトウェア説明

RFC Ch.0 割り込み時の処理
(intrRfc0関数)



RFC Ch.1 割り込み時の処理
(intrRfc1関数)





APPENDIX A. 温湿度テーブルの作成例

ここでは、「R/F 変換カウント値－温度」変換テーブル、「R/F 変換カウント値－湿度」変換テーブルの作成方法についての一例を紹介します。変換テーブルの作成は、何点かの条件の時のセンサの抵抗値を測定してセンサのデータシートをもとに換算して抵抗値のテーブルを作成し、その値の抵抗器を RFC のセンサ部に接続してカウンタ値を計測するやり方や、実際にセンサと恒温槽を用いて何点かの条件のカウンタ値を測定する方法などがありますが、ここでは後者の方法で行いました。

測定の際には、図 3-2、図 3-3 の回路と、表 3-1 の BOM で示した部品を用いました。なお、実験には以下の恒温槽を使用しましたが、本資料の実験結果は参考にとどめるようにしてください。

● 恒温槽： ESPEC Temp. & Humid. Chamber SH-240

今回作成したサンプルプログラムで計測できる温湿度の範囲は、以下を目標としました。

温度： -20.0 °C ～ 50.0 °C 分解能 0.1 °C
湿度： 10 %rh ～ 90 %rh 分解能 1 %rh

(1) センサの抵抗範囲の確認

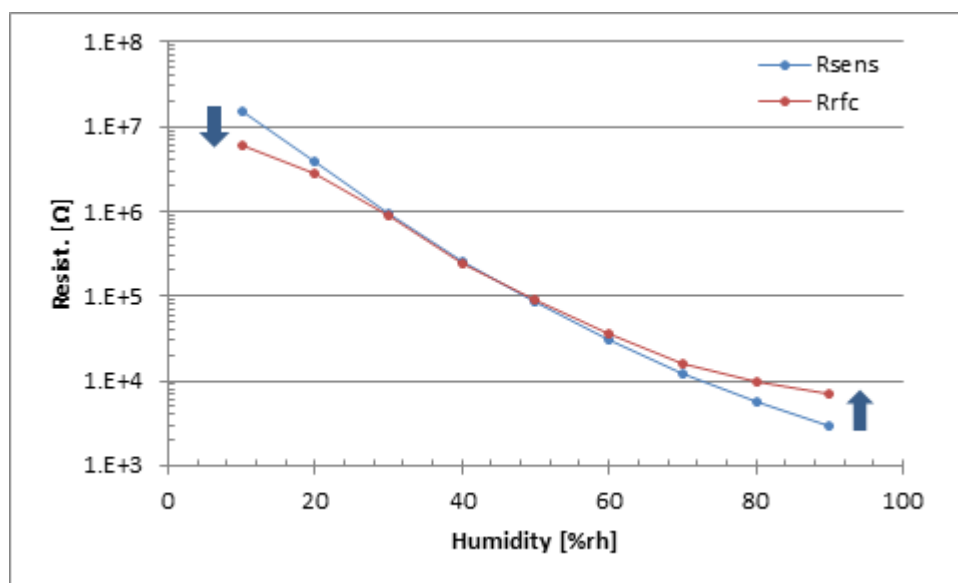
使用するセンサの特性から、測定したい範囲の抵抗値の最小値 (Rmin) と最大値 (Rmax) を確認します。

温度センサ： SEMITEC 株式会社製 103AT-2B
Rmin(typ.)=10.18k Ω (at 50°C) Rmax(typ.)=69.01k Ω (at -20°C)

湿度センサ： 神栄テクノロジー株式会社製 CL-M52R
Rmin $\approx$ 3k Ω (at 90%rh, 25°C) Rmax $\approx$ 15M Ω (at 10%rh, 25°C)

(2) 外付け抵抗によるダイナミックレンジの調整

湿度センサの抵抗値変化のダイナミックレンジが広いと、並列抵抗 R101 と直列抵抗 R102 を追加します。図 3-2 の回路で、表 3-1 の BOM に示した R101=10M Ω 、R102=4.3k Ω を採用した場合、応答特性は以下のように Rsens から Rrfc のようにダイナミックレンジが狭まる方向に変化します。



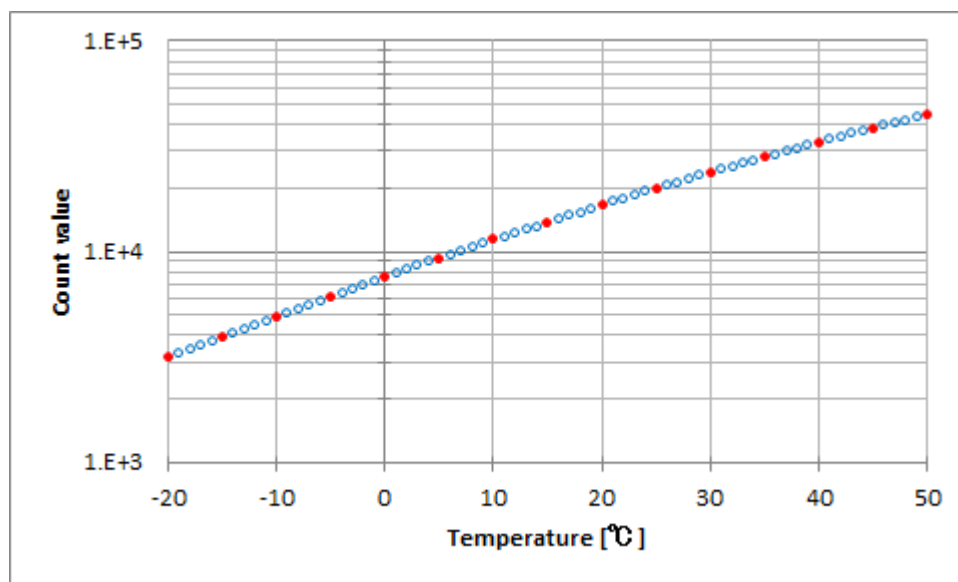
テクニカルマニュアルの電気的特性の R/F 変換器 (RFC) 特性に記載されているように、基準抵抗や抵抗性センサの抵抗値は、 $1\text{k}\Omega$ 以上でなければなりません。このスペックはもう一つ、 $1\text{k}\Omega$ に近い値まで抵抗値が小さくなると、IC の製造ばらつきなどセンサ以外の要因で、誤差が大きくなるという意味もあります。上記応答特性の高湿度領域で、カーブが寝てきているのは、このような意図、すなわち抵抗を $1\text{k}\Omega$ に近づけたくないという意思を反映させたからです。

同様に低湿度時に、極端に大きな抵抗にならないようにしているのは、発振周波数が低くなりすぎて規定時間内の計測カウント値が小さくなり、計測値の誤差が大きくなるのを回避するためです。

(3) 温度センサのカウント値の温度依存性の測定

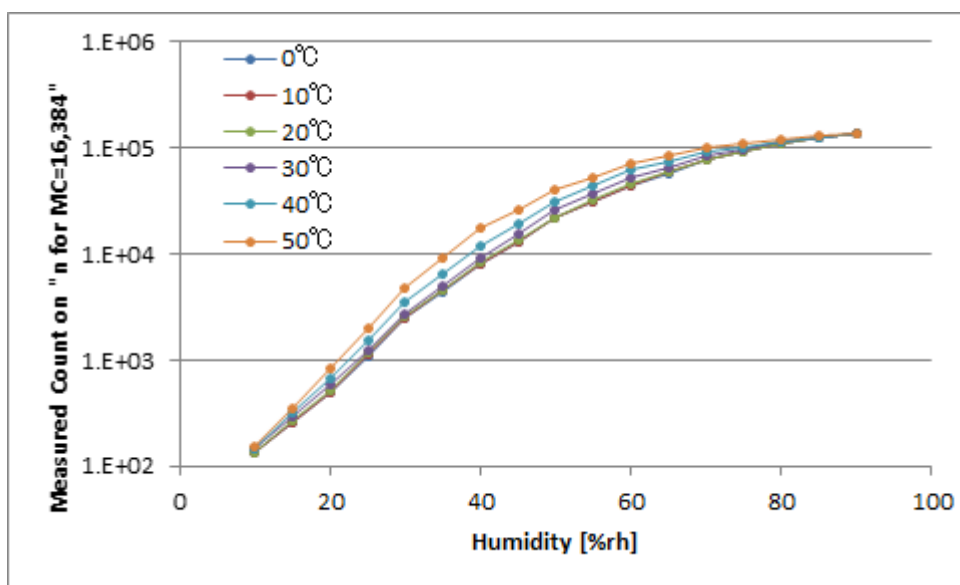
恒温槽を用いて、温度センサのカウント値の温度依存性を確認します。今回は -20°C から 50°C の範囲、 5°C ステップで測定しました。得られた結果をもとに 1°C ステップになるよう、対数目盛で補間してカウント値を求めました。以下のグラフの赤い点が実測値、青い丸が補間して求めた値です。

このように補間して、ステップが小さくなるようにテーブルデータを増やすのは、MCU で温度を計算する際、図 3-4 の (b) のグラフのようにリニア目盛として計算するので、その誤差をできるだけ小さくするためです。



(4) 湿度センサのカウント値の温湿度依存性の測定

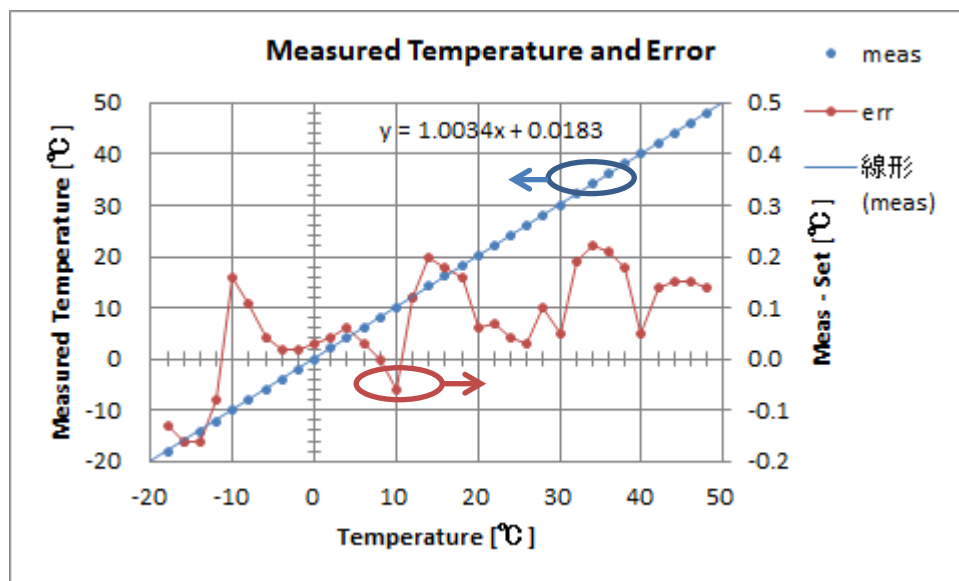
恒温槽を用いて、湿度センサのカウント値の温湿度依存性を確認します。今回は 10℃ から 50℃ の範囲、10℃ ステップ、30%rh から 90%rh の範囲、10%rh ステップで測定しました。得られた結果をもとに 5%rh ステップになるよう、対数目盛で補間してカウント値を求めましたが、それは温度センサの時と同じ理由からです。補間できない低温低湿度領域は、プロットのカーブが滑らかに変化するような値を選びました。すなわち、以上の処理の結果、0℃ から 50℃ の範囲、10℃ ステップ、10%rh から 90%rh の範囲、5%rh ステップの組み合わせで、湿度センサのカウント値の温湿度依存性のデータを得ることができました。



(5) 温度測定結果の誤差の確認

以上のようにして得られたパラメータを用いて、温度を -20℃ から 50℃ の範囲、2℃ ステップで振って、計測された温度の値と恒温槽の温度設定値を比較しました。なお計測温度は、構造体 thm の要素 thm.temp100 の値を 100 分の 1 した値を用いました。

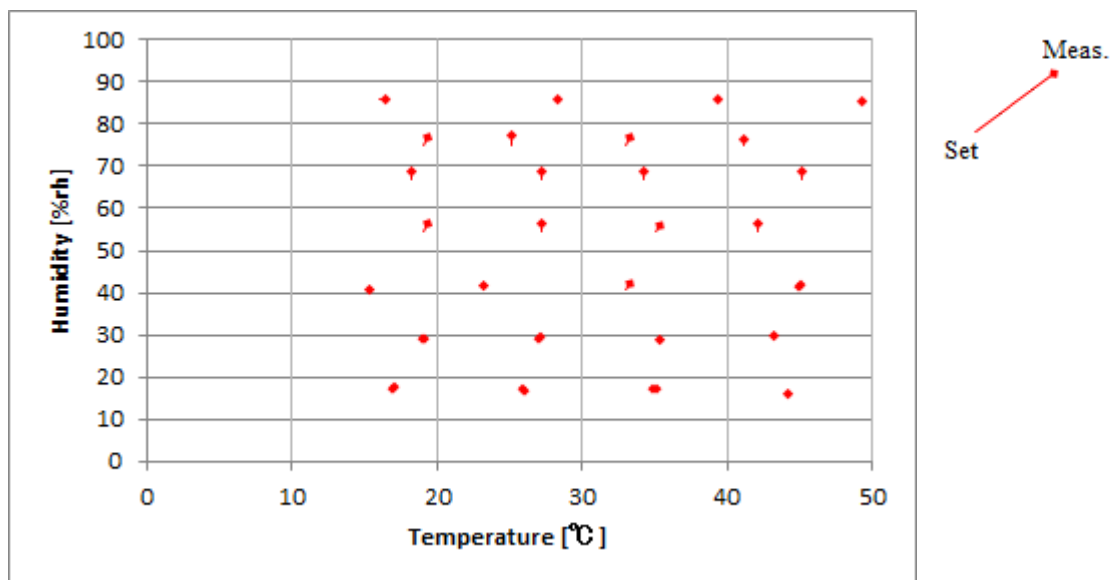
得られた結果をプロットしたものが以下のグラフです。横軸は恒温槽の設定温度、青い点は温度を変えた時の温度の測定結果で、それを線形近似したのが青色の線、ともにグラフの縦軸は左側の目盛です。赤色のプロットは、測定温度の値から設定温度の値を引いた誤差で、グラフの縦軸は右側の目盛です。



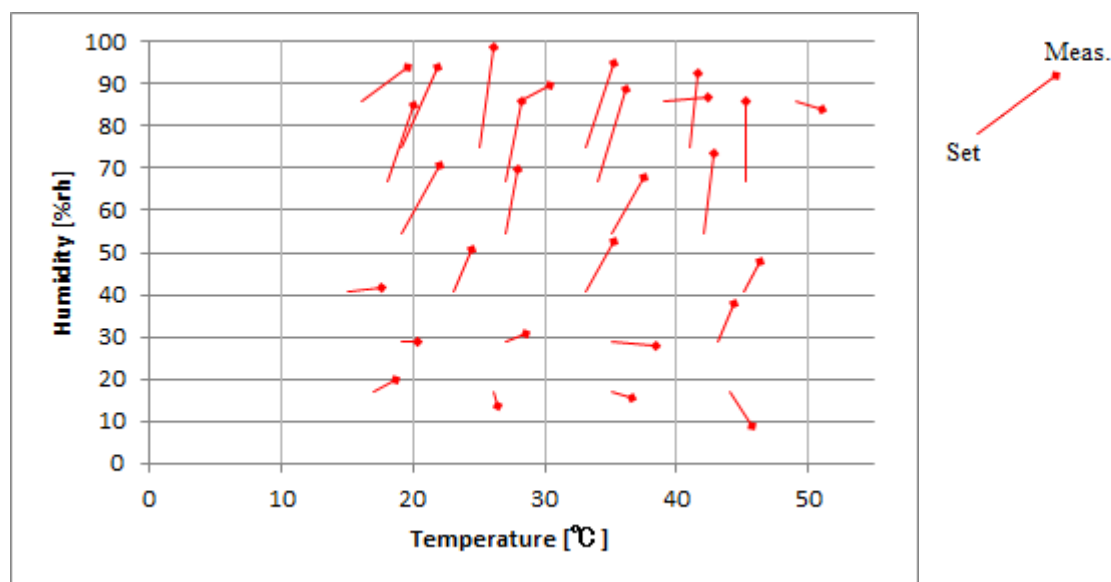
(6) 湿度測定結果の誤差の確認

得られたパラメータを用いて、恒温槽の温度を 15℃ から 49℃ の範囲、湿度 17%rh から 86%rh の範囲のいくつかのポイントで設定し、計測された温度と湿度の値と、恒温槽の設定値を比較しました。なお計測温度は先と同様 `thm.temp100` の値を 100 分の 1 した値を、計測湿度は構造体 `thm` の要素 `thm.hum10` の値を 10 分の 1 した値を用いました。

得られた結果をプロットしたものが以下のグラフです。横軸は温度、縦軸は相対湿度です。マッチ棒に例えて説明すると、マッチ棒の軸端が設定値、頭の火薬の位置が測定値です。



上のグラフでは設定値と計測値が近接しているため、誤差を 10 倍（マッチ棒の軸端の位置を保ったまま、火薬の位置に至るベクトルの長さを 10 倍）にしてプロットし直したのが以下のグラフです。



APPENDIX A. 温湿度テーブルの作成例

(7) 処理時間

[c17 clock]コマンドで計測した、処理時間の実測値です。これは参考にするための情報で、処理時間を保証するものではありません。

計測結果： 気温 24.4℃、湿度 15%rh

温度計測 (runRfcConvergingOperation～intRfc1)	134 ms
湿度計測 (runRfcConvergingOperation～intRfc0)	510 ms
温度計算 (getTemp 関数実行時間)	135 ms
湿度計算 (getHum 関数実行時間)	1070 ms
結果表示 (dispVal 実行時間)	147 ms

並行処理されるので、
長い方の時間の経過で
計測終了します。

(8) 計測カウンタ値の範囲

ところで、「3.3. 湿度センサの抵抗計測方法について」の章で、計測カウンタの値 (Measured Count) が一定の値の範囲になるように管理しながら RFC による湿度センサの抵抗値測定を行う必要があると記しましたが、この Appendix A ではその範囲を以下のように決めてコーディングしました。(src/rfc_tbl.c の中で値を定義しています。)

HUM_COUNTER_LOWER_LIM	500
HUM_COUNTER_UPPER_LIM	5,000

HUM_COUNTER_UPPER_LIM=5,000 なのは、「計測カウンタの値 (Measured Count) が 10,000 を超えないようにする。」という制約条件に基づくものですが、HUM_COUNTER_LOWER_LIM=500 なのは、湿度の分解能の 1%rh の表示値に影響を与えないのに十分な値を選んだだけです。

(9) 湿度の OSC1 の発振への影響

S1C17 シリーズの OSC1 の発振回路は、非常にわずかな電力で発振するよう設計されています。そのため湿度が高くなると、リーク電流の影響で発振停止してしまうことがあります。

通常の生活条件での使用の場合は、CLGOSC1.INVIN レジスタの値を変更して、発振インバータのゲインを上げてやることでその影響を低減させることが可能です。

しかし著しい高湿度環境下では、それだけでは影響を完全に防ぐことができません。その場合は防湿加工を少なくとも OSC1 の系統、望ましくはシステム全体に対して施す必要があります。

APPENDIX B. 塩分濃度換算用テーブルの作成例

ここでは、「R/F 変換カウント値－塩分濃度」変換テーブルの作成方法についての一例を紹介します。変換テーブルの作成は、湿度センサの場合と同様、実際に塩分濃度センサ用電極と恒温槽を用いて何点かの条件のカウント値を測定する方法で行いました。

測定の際には、図 3-2、図 3-3 の回路と、表 3-2 の BOM で示した部品を用いました。なお、実験には以下の恒温槽を使用しましたが、本資料の実験結果は参考にとどめるようにしてください。なお、恒温槽の代わりにウォーターバスを使う方が、定温に到達する時間が短くなるのが一般的です。

● 恒温槽： ESPEC Temp. & Humid. Chamber SH-240

今回作成したサンプルプログラムで計測できる塩分濃度の範囲は、以下を目標としました。

塩分濃度： 0.40 % ～ 1.40 % 分解能 0.01%

その際、温度範囲、電源電圧は以下の範囲、値であることを想定しています。

温度範囲： 5 °C ～ 85 °C（テーブルは、0 °C ～ 100 °C の範囲を用意）

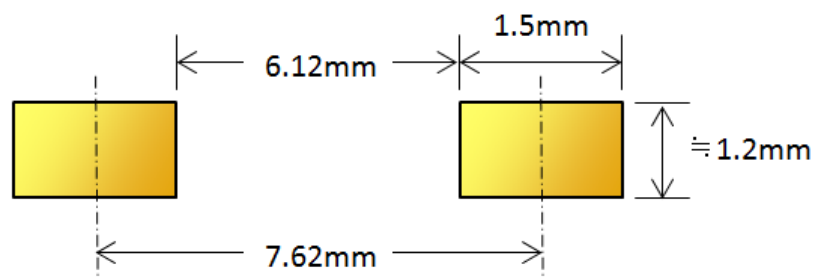
電源電圧： 3.3V

上記の想定で、電源電圧を一定の 3.3V としたのは、電極間抵抗の温度依存性が比較的大きいことが予備実験でわかっていたため、更にパラメータが増えることを嫌ったためです。すなわち、実際の塩分濃度計では、4.5V 程度の電池電圧を、電源レギュレータで 3.3V に安定化して MCU に供給するやり方を採用することになると仮定しています。

なお、水溶液の温度の測定については、温湿度計測の温度計測と比べ、使用しているテーブルの温度範囲とステップが異なる以外は全く同じなので、手順等は「Appendix A. 温湿度テーブルの作成例」を参考にしてください。

(1) 塩分濃度センサ用電極の構造

塩分濃度センサとして、ここでは以下のディメンジョンの電極を用いました。



電界集中のことを考慮したら、円形、または球の一部を形成する形状が望ましいのですが、本アプリケーションノートでは上記の長方形の形状の銅箔表面に金メッキされたものを電極として用いました。実際に食品の塩分濃度を計測するための製品の場合、食品衛生法に適合するため、電極の基材としては鉛など人体に有害な金属を含まない材料を用いることはもちろん、金メッキの厚さについても慎重に検討する必要があります。

(2) 塩分濃度センサの抵抗範囲の見積もり

今回想定している塩分濃度計の濃度範囲は、0.40%～1.40%ですが、その場合の 25°C に於ける導電率 σ は、 $\sigma = 7.4\text{mS/cm} \sim 23\text{mS/cm}$ 程度の範囲で変化します。先のディメンジョンの電極を、 $A = 1.5\text{mm} \times 1.2\text{mm}$ の面積を持つ長方形電極が、 $d = 7.62\text{mm}$ の距離で対抗しているものとして扱い、電極間の抵抗範囲を求めると、以下の式より、 $5.7\text{k}\Omega \sim 1.8\text{k}\Omega$ という値が得られます。

$$R = \frac{1}{\sigma} \cdot \frac{d}{A}$$

(3) 外付け抵抗による抵抗値の調整

「3.4 R/F 変換器による塩分濃度計測」の章で説明したように、塩分濃度センサとしての電極は、高塩分濃度の場合に抵抗が小さくなりすぎて R/F 変換器の電気的特性のスペック、 $R_{REF}(\min)=1k\Omega$ を満たさなくなることを避けるため、また水溶液に触れておらず、電極が空気中に置かれている場合の ∞ の抵抗を検出するため、並列抵抗 R101 と直列抵抗 R102 を追加しました。具体的な抵抗値は、表 3-2 の BOM に示したように、R101=1M Ω 、R102=1k Ω です。

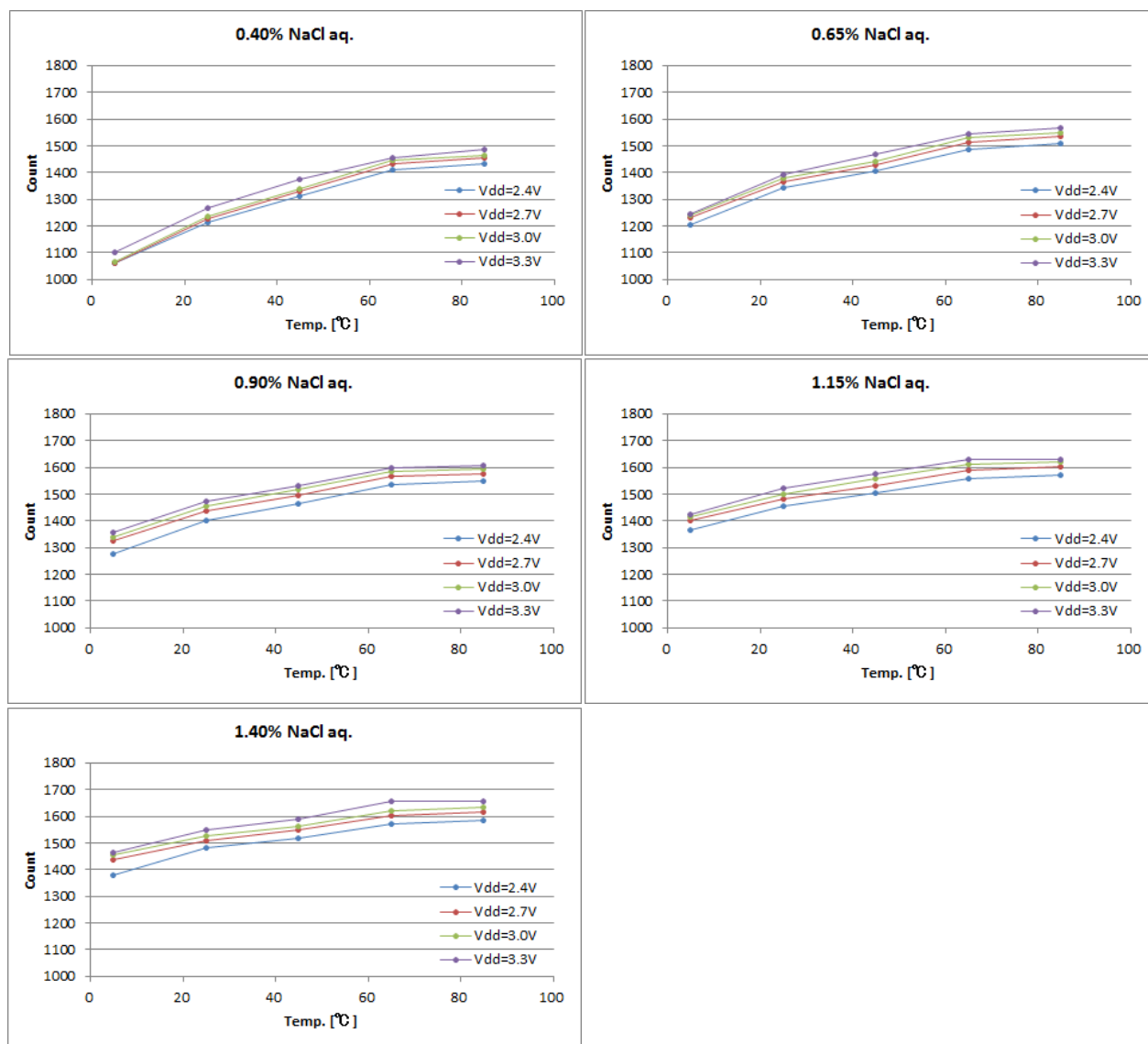
(4) 温度センサのカウント値の温度依存性の測定

恒温槽を用いて、温度センサのカウント値の温度依存性を確認します。今回は 0℃から 100℃の範囲、5℃ステップで測定しました。得られた結果をもとに 2℃ステップになるよう、対数目盛で補間してカウント値を求めました。

このように補間して、ステップが小さなテーブルデータを増やすのは、「Appendix A. 温湿度テーブルの作成例」に示した理由と同じですが、塩分濃度計での温度は塩分濃度の補正用に用いるだけなので、温湿度計の場合よりもステップを大きくしました。

(5) 塩分濃度センサのカウント値の 塩分濃度・温度 依存性の測定

恒温槽を用いて、塩分濃度センサのカウント値の塩分濃度・温度依存性を確認します。今回は 0.40% から 1.40% の範囲、0.25% ステップ、5℃ から 85℃ の範囲、20℃ ステップで測定しました。最終的に、電源電圧 Vdd=3.3V の時の結果のみをテーブルに用いますが、ここでは電源電圧の影響が無視できないことを確認するために、2.4V から 3.3V の範囲、0.3V ステップで測定しました。



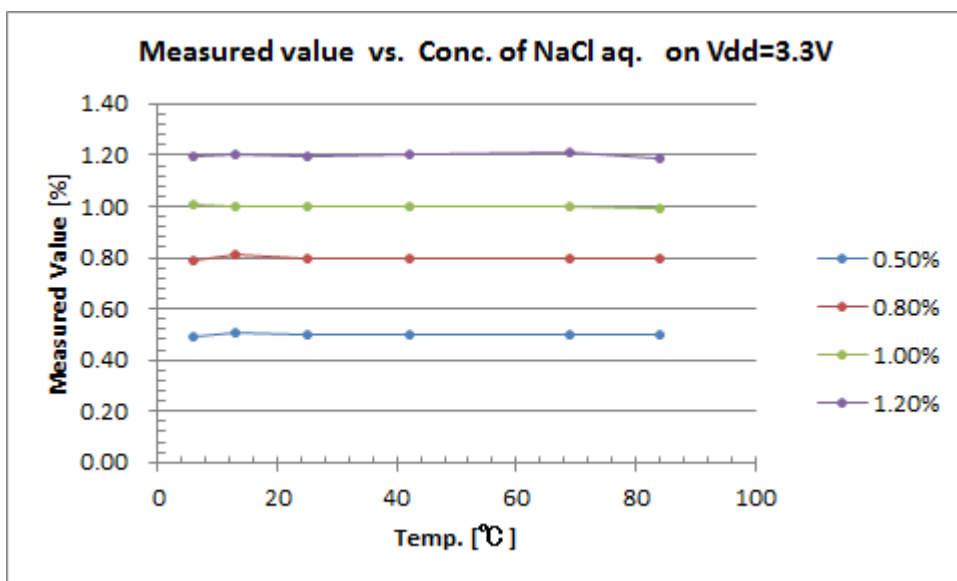
なお、上記の結果は、サンプルプログラム `s1c17w22_w23_rfc_temp_hum_analy_gnu17vx` の、`main.c` で定義している定数値を、以下のように変更して測定したものです。

```
HUM_TIME_BASE_COUNT_INDEX    6           // COUNT_1K
```

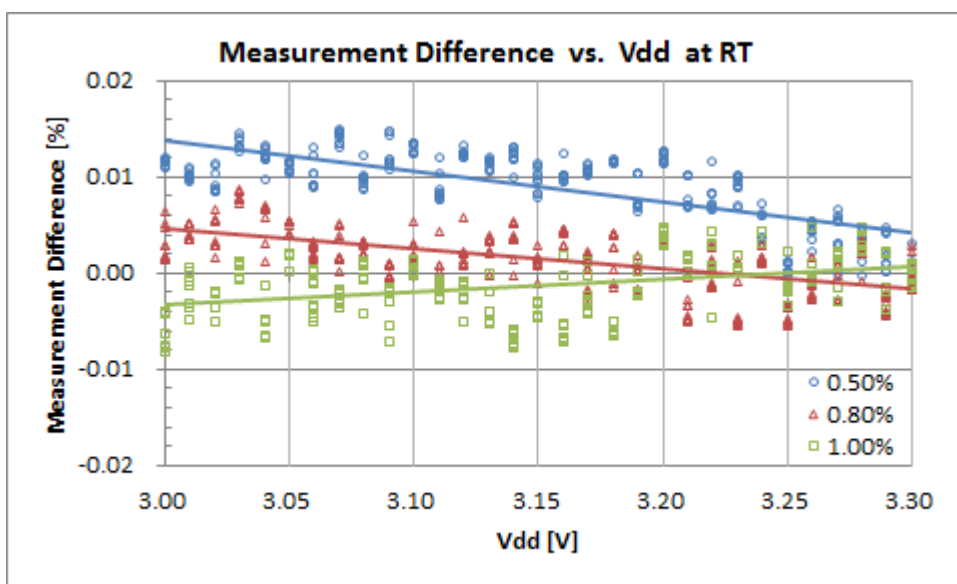
(6) 塩分濃度測定結果の誤差の確認

得られたパラメータを用いて、恒温槽の温度を 6℃ から 84℃ の範囲、塩分濃度 0.50% から 1.20% の範囲のいくつかのポイントで設定し、計測された塩分濃度の値をプロットしました。なお計測された塩分濃度%値は `tsm.conc1000` の値を 1000 分の 1 した値を用いました。

APPENDIX B. 塩分濃度換算用テーブルの作成例



また、繰り返し測定を行った時に塩分濃度の計測値がどの程度ばらつくか評価するため、3種類の塩分濃度の水溶液について、常温（約25℃）でVddを3.00Vから3.30Vまで0.01Vステップで変化させ、各電圧で10回測定を繰り返したときの塩分濃度の計測値のばらつき（調製濃度との差分の%値、すなわち、調製濃度0.80%の計測値が0.81%なら+0.01%）を以下に示します。



(7) 処理時間

[c17 clock]コマンドで計測した、処理時間の実測値です。これは参考にするための情報で、処理時間を保証するものではありません。

計測結果： 塩分濃度 1.00%、液温 24.4℃

塩分濃度計測 (runRfcConvergingOperation～intRfc0)
 温度計測 (runRfcConvergingOperation～intRfc1)
 塩分濃度計算 (getConc 関数実行時間)
 温度計算 (getTemp 関数実行時間)
 結果表示 (dispVal 実行時間)

7 ms
 37 ms
 354 ms
 77 ms
 152 ms

並行処理されるので、
 長い方の時間の経過で
 計測終了します。

[illegible]

セイコーエプソン株式会社

マイクロデバイス事業部 デバイス営業部

東京 〒191-8501 東京都日野市日野 421-8
TEL (042) 587-5313 (直通) FAX (042) 587-5116

大阪 〒541-0059 大阪市中央区博労町 3-5-1 御堂筋グランタワー15F
TEL (06) 6120-6000 (代表) FAX (06) 6120-6100

ドキュメントコード : 412808002

2014 年 8 月 作成

2016 年 4 月 作成