お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ（http://www.renesas.com）

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社（http://www.renesas.com）

【問い合わせ先】http://japan.renesas.com/inquiry

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。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。
3. 当社製品を改造、改変、複製等しないでください。
4. 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。
5. 輸出に際しては、「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。本資料に記載されている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用途の目的で使用しないでください。また、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に使用することができません。
6. 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ものですが、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。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されてい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生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

   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

   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、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

   特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社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。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、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、社会的損害など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、機器またはシステムとしての出荷保証をお願いいたします。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、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、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て、当社は、一切その責任を負いません。
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固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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。

RENESAS

アプリケーション・ノート

78K0/Lx3

サンプル・プログラム（温度測定）

ポートおよびタイマ機能による温度測定プログラム編

この資料は，サンプル・プログラムの動作概要や使用方法について説明したものです。本サンプル・プログラムは，A/Dコンバータを使用せず，ポート機能（ロウ・レベル入力認識）およびタイマを利用して温度測定を行うものです。具体的には，コンデンサに充電した電荷を，サーミスタ（抵抗）を介して放電させます。コンデンサの電位をロウ・レベルと認識するまでの放電時間をタイマで測定し，放電時間からサーミスタの抵抗値を算出します。あらかじめ存在するサーミスタ抵抗値－温度のテーブル・データより，測定時の温度を確定します。

対象デバイス

78K0/LC3マイクロコントローラ
78K0/LD3マイクロコントローラ
78K0/LE3マイクロコントローラ
78K0/LF3マイクロコントローラ

目　次

資料番号　U19542JJ1V0AN00（第1版）
発行年月　November 2008 NS

- **本資料に記載されている内容は2008年11月現在のもので，今後，予告なく変更することがあります。**
  量産設計の際には最新の個別データ・シート等をご参照ください。
- 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。当社は，本資料の誤りに関し，一切その責を負いません。
- 当社は，本資料に記載された当社製品の使用に関連し発生した第三者の特許権，著作権その他の知的財産権の侵害等に関し，一切その責を負いません。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。
- 本資料に記載された回路，ソフトウエ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。お客様の機器の設計において，回路，ソフトウエ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，一切その責を負いません。
- 当社は，当社製品の品質，信頼性の向上に努めておりますが，当社製品の不具合が完全に発生しないことを保証するものではありません。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。当社製品をお客様の機器にご使用の際には，当社製品の不具合の結果として，生命，身体および財産に対する損害や社会的損害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を行ってください。
- 当社は，当社製品の品質水準を「標準水準」，「特別水準」およびお客様に品質保証プログラムを指定していただく「特定水準」に分類しております。また，各品質水準は，以下に示す用途に製品が使われることを意図しておりますので，当社製品の品質水準をご確認ください。

  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

  特別水準：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，生命維持を目的として設計されていない医療機器

  特定水準：航空機器，航空宇宙機器，海底中継機器，原子力制御システム，生命維持のための医療機器，生命維持のための装置またはシステム等

  当社製品のデータ・シート，データ・ブック等の資料で特に品質水準の表示がない場合は，標準水準製品であることを表します。意図されていない用途で当社製品の使用をお客様が希望する場合には，事前に当社販売窓口までお問い合わせください。

（注）

（1）本事項において使用されている「当社」とは，NECエレクトロニクス株式会社およびNEC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う。

（2）本事項において使用されている「当社製品」とは，（1）において定義された当社の開発，製造製品をいう。

M8E0710J

第1章　概　　要

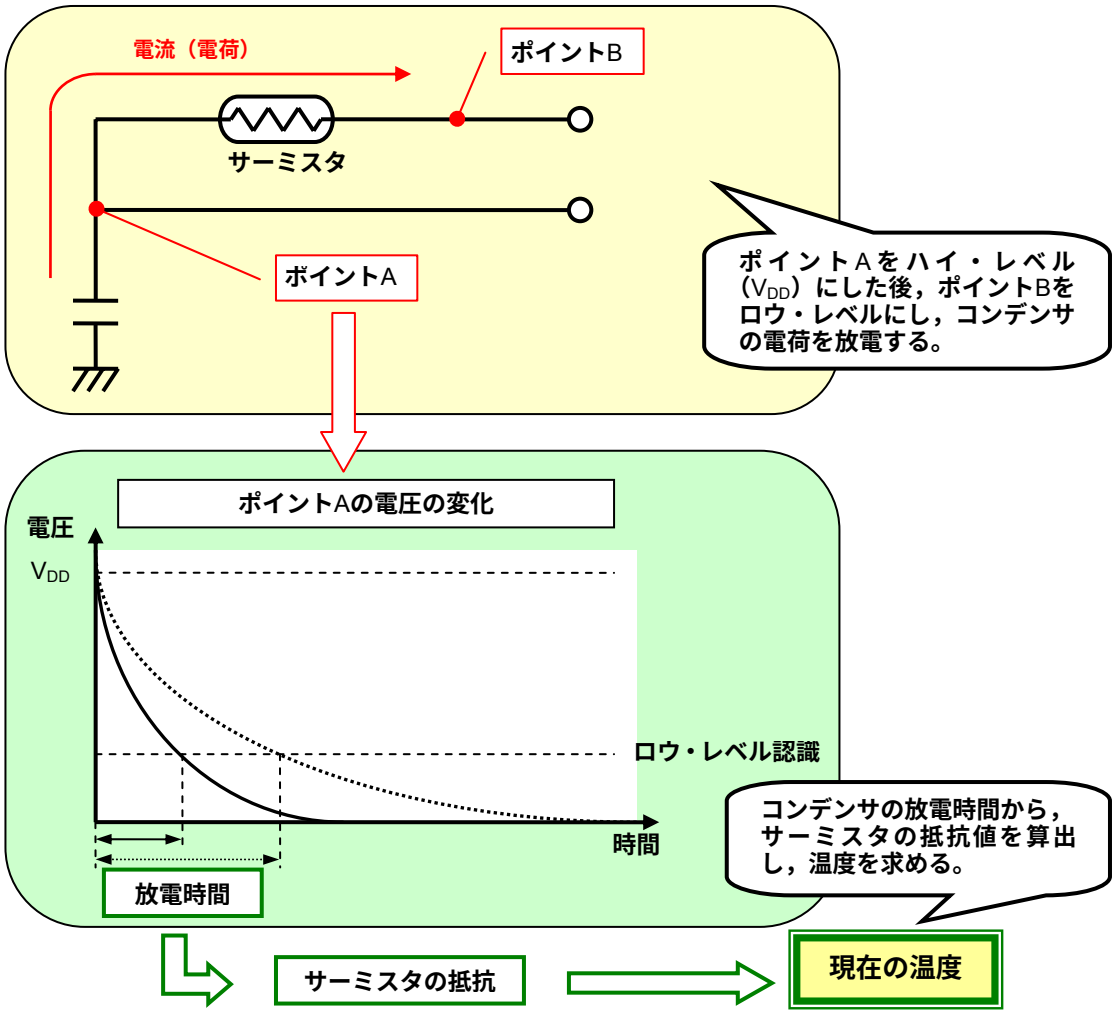
本サンプル・プログラムは，A/Dコンバータを使用せず，ポート機能（ロウ・レベル入力認識）およびタイマを利用して温度測定を行うものです。具体的には，コンデンサに充電した電荷を，サーミスタ（抵抗）を介して放電させます。コンデンサの電位をロウ・レベルと認識するまでの放電時間をタイマで測定し，放電時間からサーミスタの抵抗値を算出します。あらかじめ存在するサーミスタ抵抗値－温度のテーブル・データより，測定時の温度を確定します。メイン処理では，パルス幅測定処理，温度取得処理，UART送信処理を呼び出します。

パルス幅測定処理では，キャリブレーション用固定抵抗，およびサーミスタを使用してコンデンサの放電を行い，放電時間を測定します。放電時間は，16ビット・タイマ／イベント・カウンタ00のパルス幅測定の機能により測定します。

温度取得処理では，測定したコンデンサの放電時間からサーミスタの抵抗値を算出し，サーミスタR－T特性仕様を使用して抵抗値を温度に変換します。サーミスタの抵抗値は，コンデンサの放電時間より，抵抗値とコンデンサの放電時間が比例することを利用して算出します。サーミスタR－T特性仕様に対応した温度変換テーブル[注]より，サーミスタの抵抗値に対する温度を取得します。

UART送信処理では，温度測定結果をASCIIコードに変換し，シリアル・インタフェースUART6により送信します。

注　サーミスタR－T特性仕様に対応した温度変換テーブルの詳細は，2. 2 **抵抗値の温度変換**を参照してください。


電流（電荷）
ポイントB
サーミスタ
ポイントA
ポイントAをハイ・レベル（VDD）にした後，ポイントBをロウ・レベルにし，コンデンサの電荷を放電する。
ポイントAの電圧の変化
電圧
VDD
ロウ・レベル認識
時間
放電時間
コンデンサの放電時間から，サーミスタの抵抗値を算出し，温度を求める。
サーミスタの抵抗
現在の温度

(1) 使用する周辺の初期設定の主な内容

使用する周辺の初期設定の主な内容は次のとおりです。

- 割り込みの禁止
- レジスタ・バンクの設定
- スタック・ポインタの設定
- ROM/RAMサイズの設定
- ポートの設定
- CPUクロックを高速内蔵発振動作（8 MHz）に設定
- 周辺ハードウエア・クロックを高速内蔵発振動作（8 MHz）に設定
- 16ビット・タイマ／イベント・カウンタ00をパルス幅測定としての動作に設定
- 8ビット・タイマH2を温度測定タイミングのベース・タイマ（100 msのインターバル・タイマ）に設定
- シリアル・インタフェースUART6をデータ送信用に設定
- 割り込みマスクの設定
- 割り込みの許可

(2) メイン処理の内容

放電時間測定処理，温度取得処理，UART送信処理を呼び出します。
処理を呼び出す周期は約1秒で，8ビット・タイマH2をベース・タイマとしてカウントします。

(3) パルス幅測定処理の内容

16ビット・タイマ／イベント・カウンタ00のパルス幅測定の機能を使用し，コンデンサの放電時間を測定します。

コンデンサの電圧レベルはTI000端子で検出します。16ビット・タイマ／イベント・カウンタ00はTI000の立ち下がりで16ビット・タイマ・カウンタ00をキャプチャし，割り込み信号（INTTM010）を発生させるように設定します。

コンデンサを十分に充電したあと，16ビット・タイマ／イベント・カウンタ00を動作許可し，キャリブレーション用抵抗またはサーミスタを使用してコンデンサの放電を行います。コンデンサの放電が終わりTI000がロウ・レベルになると，INTTME010信号が発生し，16ビット・タイマ・キャプチャ／コンペア・レジスタ010より放電時間が取得できます。

(4) 温度取得処理の内容

測定したコンデンサの放電時間からサーミスタの抵抗値を算出し，サーミスタR－T特性仕様を使用して抵抗値を温度に変換する処理です。

サーミスタの抵抗値は，抵抗値とコンデンサの放電時間が比例することを利用し，コンデンサの放電時間より算出します。詳細は，2. 1 **サーミスタの抵抗値算出**を参照してください。

サーミスタR－T特性仕様に対応した温度変換テーブルを使用し，サーミスタの抵抗値を温度に変換します。詳細は，2. 2 **抵抗値の温度変換**を参照してください。

(5) UART送信処理の内容

温度測定結果をASCIIコードに変換し，シリアル・インタフェースUART6にて送信する処理です。

UART通信の設定内容詳細，および送信データの詳細については，4. 4 UART**送信データのフォーマット**を参照してください。

第2章　温度測定アルゴリズム

この章では，このサンプル・プログラムで使用する温度測定のアルゴリズムを説明します。温度測定は，サーミスタの抵抗値を算出し，抵抗値を温度に変換するという手順で行います。

2. 1　サーミスタの抵抗値算出

図2－1　回路例

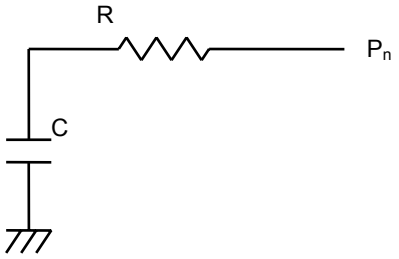


図2－1のような回路を使用し，コンデンサを充電した状態でポートP_nをロウ・レベルにすると，コンデンサから放電が始まります。そのとき，コンデンサが充電した電圧の37 %まで放電するためにかかる時間は，下の式で求められます。

$$t = R \times C$$

t　：抵抗Rを使用したコンデンサの放電時間［秒］
R　：抵抗Rの抵抗値［Ω］
C　：コンデンサの容量［F］

上の式より，放電時間と抵抗値は比例することを前提とし、サーミスタの抵抗値を算出します。
上の式では，固定抵抗を使用した場合のコンデンサの放電時間は，一定となります。キャリブレーション用固定抵抗，およびサーミスタを使用してコンデンサの放電時間を測定し，次の関係式よりサーミスタの抵抗値を算出します。

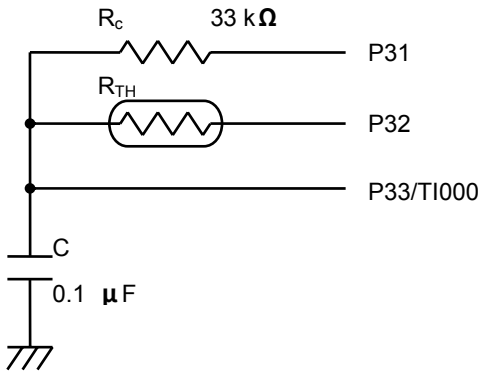
$$t_C : R_C = t_{TH} : R_{TH} \Rightarrow R_{TH} = \frac{R_C \times t_{TH}}{t_C}$$

t_C　：キャリブレーション用固定抵抗を使用したコンデンサの放電時間［秒］
R_C　：キャリブレーション用固定抵抗［Ω］
t_{TH}　：サーミスタを使用したコンデンサの放電時間［秒］
R_{TH}　：サーミスタの抵抗値［Ω］

(1) コンデンサの放電時間を測定

コンデンサの放電時間を測定するために，図2－2の回路を使用します。コンデンサの放電時間は16ビット・タイマ／イベント・カウンタ00のパルス幅測定の機能を使用し，TI000がハイ・レベルの時間を測定します。16ビット・タイマ／イベント・カウンタの設定の詳細は，5. 1 **使用する周辺の初期設定**を参照してください。

図2－2　サーミスタの抵抗値算出用回路



R_{TH}　：サーミスタ

R_C　：キャリブレーション用固定抵抗

P31　：キャリブレーション用固定抵抗を使用する場合の放電用ポートです。

P32　：サーミスタを使用する場合の放電用ポートです。

P33/TI000　：コンデンサの充電（P33）と，コンデンサの放電完了の検出（TI000）に使用します。

備考　コンデンサの容量と，キャリブレーション用固定抵抗は，放電時間を測定する際に16ビット・タイマ・カウンタ00のオーバフローが発生しないように選定しています。

放電時間を測定する手順の概要は以下のとおりです。詳細は，5. 3 **パルス幅測定処理**を参照してください。

① P33をハイ・レベル出力にし，コンデンサを充電します。[注1]
② P33をTI000端子として使用するため，入力ポートに設定します。
③ 放電に使用するポート（P31またはP32）[注2]をロウ・レベル出力に設定し，コンデンサの放電を開始します。
④ ③のコンデンサの放電の開始と同時に，16ビット・タイマ／イベント・カウンタ00を動作許可し，放電時間の測定を開始します。
⑤ コンデンサの放電が終了するまでウエイトします。ウエイト中に16ビット・タイマ・カウンタ00のオーバフローがあれば，その回数をカウントします。
⑥ 16ビット・タイマ・キャプチャ／コンペア・レジスタ010から，測定したコンデンサの放電時間を取得します。
⑦ 16ビット・タイマ／イベント・カウンタ00を動作禁止とし，放電に使用したポート（P31またはP32）[注2]を入力ポートに設定します。

注1. ハイ・レベル出力のマイコンのCMOS出力抵抗値を2 kΩと想定すると，時定数τ（電源電圧（V_{DD}）の63 %まで充電するために必要な時間）は次の式で求められます。

$$\tau = 0.1\ [\mu F] \times 2\ [k\Omega] = 0.2\ [ms]$$

本サンプル・プログラムでは，コンデンサを十分に充電するため，充電時間を5τ = 2 msとしています。

2. コンデンサの放電にキャリブレーション用固定抵抗を使用する場合はP31，サーミスタを使用する場合はP32となります。

(2) サーミスタの抵抗値を算出

測定したコンデンサの放電時間より，下の式にてサーミスタの抵抗値を算出します。計算で使用するコンデンサの放電時間は16ビット・タイマ・キャプチャ／コンペア・レジスタ010から読み出した16ビットの値です。サーミスタを使用したコンデンサの放電時間には，オーバフロー回数を含めて最大17ビット[注1]の値を使用して計算します。

$$R_{TH} = \frac{R_C \times (CNT_{TH} + \text{オーバフロー回数} \times 10000H)}{CNT_C}$$

R_{TH}　：サーミスタの抵抗値［100 Ω］
R_C　：キャリブレーション用固定抵抗［100 Ω］
CNT_{TH}　：サーミスタを使用したコンデンサの放電時間[注2]
CNT_C　：キャリブレーション用固定抵抗を使用したコンデンサの放電時間[注2]
オーバフロー回数　：サーミスタを使用したコンデンサの放電時間を測定したときの16ビット・タイマ・カウンタ00のオーバフロー回数[注2]

注1. 本サンプル・プログラムの温度の測定範囲（42.0～32.0 ℃）に対する抵抗値の測定範囲は，24.5～37.0 kΩとなります。下の式で計算すると，オーバフロー回数が2回以上になると，抵抗値の測定範囲外となります。オーバフロー回数が2回以上になった場合は，サーミスタの抵抗値を算出せず，測定エラーとして処理します。

2. コンデンサの放電パルス幅は，周辺ハードウエア・クロック（f_{PRS}）をカウント・クロックとして測定しています。よって，サーミスタの抵抗値算出で使用するコンデンサの放電時間（CNT_CおよびCNT_{TH}）とコンデンサの放電時間（t_Cおよびt_{TH}［秒］）は，次の式で求めることができます。

$$t_C \text{［秒］} = \frac{CNT_C}{f_{PRS}}$$

$$t_{TH} \text{［秒］} = \frac{CNT_{TH} + \text{オーバフロー回数} \times 10000H}{f_{PRS}}$$

t_C　：キャリブレーション用固定抵抗を使用したコンデンサの放電時間［秒］
t_{TH}　：サーミスタを使用したコンデンサの放電時間［秒］
f_{PRS}　：周辺ハードウエア・クロック周波数［Hz］

2. 2　抵抗値の温度変換

サーミスタの抵抗値に対する温度を温度変換テーブルから取得し，サーミスタの抵抗値を温度変換します。温度変換テーブルは，サーミスタR－T特性仕様に対応しています。

本サンプル・プログラムでは，石塚電子株式会社製のサーミスタ 503ETを使用しています。503ETのサーミスタR－T特性仕様を表2－1に示します。

表2－1　サーミスタR－T特性仕様（32.0～42.0 ℃）

温度［℃］	最大抵抗［kΩ］	標準抵抗［kΩ］	最小抵抗［kΩ］	温度許容誤差［℃］
31	39.82	38.56	37.30	－0.8～+0.8
32	38.18	36.95	35.74	－0.8～+0.8
33	36.62	35.43	34.25	－0.8～+0.9
34	35.13	33.98	33.98	－0.8～+0.9
35	33.71	32.59	32.83	－0.9～+0.9
36	32.36	31.27	31.27	－0.9～+0.9
37	31.07	30.01	30.01	－0.9～+0.9
38	29.84	28.81	28.81	－0.9～+0.9
39	28.67	27.67	27.67	－0.9～+1.0
40	27.55	26.58	26.58	－0.9～+1.0
41	26.46	25.52	25.52	－0.9～+1.0
42	25.43	24.51	24.51	－1.0～+1.0

備考　温度測定範囲（32.0～42.0 ℃）のデータと，温度変換テーブル作成の際に計算に使用するデータの抜粋です。

表2－1の温度と標準抵抗を基に，表2－2の温度変換テーブルを作成します。表2－1を標準抵抗に対して100 Ω単位で直線になるように補正し，抵抗に対する温度を0.1 ℃単位で算出します。抵抗値に対する温度の算出は，下の式を使用します。下の式は，サーミスタの温度が1 ℃変化するときのサーミスタの抵抗値と温度の変化が，比例することを前提としています。

$$T_{TH} = T_0 - \frac{R_{TH} - R_0}{R_0 - R_1}$$

③　①　②

T_{TH}　：サーミスタの温度［℃］

T_0　：基準温度［℃］ 注

R_{TH}　：サーミスタの抵抗値［100 Ω］

R_0　：基準温度での標準抵抗値［100 Ω］ 注

R_1　：基準温度－1 ℃での標準抵抗値［100 Ω］ 注

① サーミスタの抵抗値が含まれる範囲の1 ℃あたりの抵抗値の変化量を求めます。

② サーミスタの抵抗値の基準温度での抵抗値からの変化量を求めます。

③ ①と②からサーミスタの抵抗値に対する温度の基準温度からの変化量を求め，基準温度から変化量を引くことでサーミスタの温度を求めます。

注　$R_1 < R_{TH} < R_0$ となるように設定します。例えばR_{TH} = 25 kΩの場合，T_0 = 42 ℃，R_0 = 24.51 kΩ，R_1 = 25.52 kΩとなります。

表2－2　温度変換テーブル

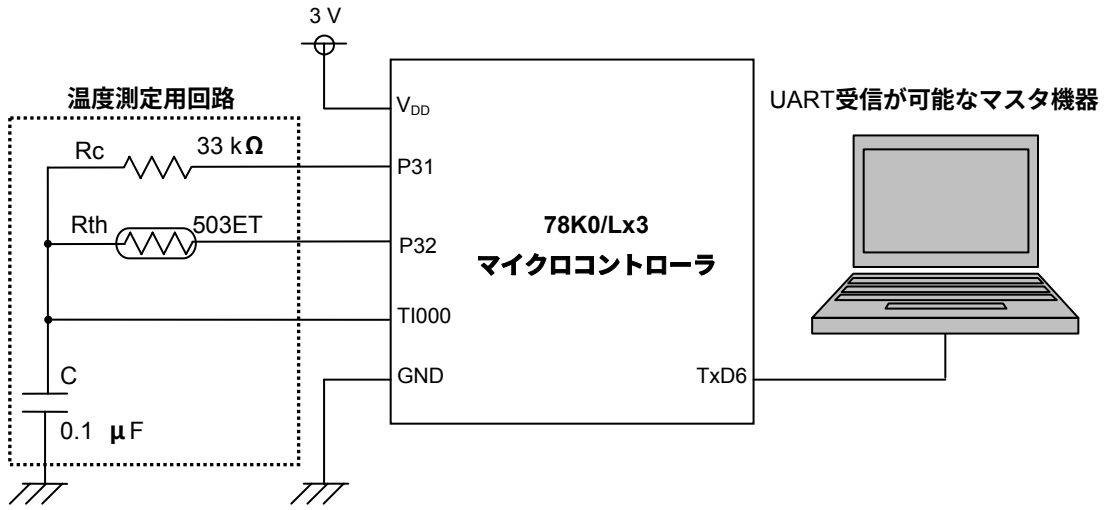
抵抗値［100Ω］	温度［℃］	抵抗値［100Ω］	温度［℃］	抵抗値［100Ω］	温度［℃］
245	42.0	287	38.1	329	34.8
246	41.9	288	38.0	330	34.7
247	41.8	289	37.9	331	34.6
248	41.7	290	37.8	332	34.6
249	41.6	291	37.8	333	34.5
250	41.5	292	37.7	334	34.4
251	41.4	293	37.6	335	34.3
252	41.3	294	37.5	336	34.3
253	41.2	295	37.4	337	34.2
254	41.1	296	37.3	338	34.1
255	41.0	297	37.3	339	34.1
256	40.9	298	37.2	340	34.0
257	40.8	299	37.1	341	33.9
258	40.7	300	37.0	342	33.8
259	40.6	301	36.9	343	33.8
260	40.5	302	36.8	344	33.7
261	40.5	303	36.8	345	33.6
262	40.4	304	36.7	346	33.6
263	40.3	305	36.6	347	33.5
264	40.2	306	36.5	348	33.4
265	40.1	307	36.5	349	33.4
266	40.0	308	36.4	350	33.3
267	39.9	309	36.3	351	33.2
268	39.8	310	36.2	352	33.2
269	39.7	311	36.1	353	33.1
270	39.6	312	36.1	354	33.0
271	39.5	313	36.0	355	33.0
272	39.4	314	35.9	356	32.9
273	39.3	315	35.8	357	32.8
274	39.2	316	35.8	358	32.8
275	39.2	317	35.7	359	32.7
276	39.1	318	35.6	360	32.6
277	39.0	319	35.5	361	32.6
278	38.9	320	35.4	362	32.5
279	38.8	321	35.4	363	32.4
280	38.7	322	35.3	364	32.4
281	38.6	323	35.2	365	32.3
282	38.5	324	35.1	366	32.2
283	38.4	325	35.1	367	32.2
284	38.4	326	35.0	368	32.1
285	38.3	327	34.9	369	32.0
286	38.2	328	34.8	370	32.0

第3章　回路図

この章では，このサンプル・プログラムで使用する回路図および周辺ハードウエアを説明します。

3. 1　回路図

回路図を次に示します。



注意1. AV$_{REF}$**端子は**V$_{DD}$**（3 V供給）に直接接続してください。**

2. AV$_{SS}$**端子は**GND**に直接接続してください。**
3. **回路図中の端子および**AV$_{REF}$, AV$_{SS}$**端子以外の未使用端子はすべて出力ポートのため，オープン（未接続）にしてください。**
4. TxD6**端子は，**UART**受信が可能な機器に接続します。**

3. 2　周辺ハードウエア

使用する周辺ハードウエアを次に示します。

（1）UART通信用機器（TxD6）

TxD6端子にUART受信用の機器を接続します。

（2）温度測定用回路（TI000, P31, P32）

TI000端子に0.1 μFのコンデンサを，P31端子に33 kΩのキャリブレーション用固定抵抗を，P32端子にサーミスタを接続します。

本サンプル・プログラムでは，サーミスタに石塚電子製高感度サーミスタ 503ETを使用しています。

注意1. AV_{REF}端子はV_{DD}（3 V供給）に直接接続してください。

2. AV_{SS}端子はGNDに直接接続してください。

第4章　ソフトウエアについて

この章では，ダウンロードする圧縮ファイルのファイル構成，使用するマイコンの内蔵周辺機能，サンプル・プログラムの初期設定と動作概要，UART送信データのフォーマット，およびフロー・チャートを説明します。

4. 1　ファイル構成

ダウンロードする圧縮ファイルのファイル構成は，次のようになっています。

ファイル名	説 明	同封圧縮（*.zip）ファイル	
			
main.asm （アセンブリ言語版）	マイコンのハードウエア初期化処理，メイン処理，パルス幅測定処理，温度取得処理とUART送信処理のソース・ファイル	●注	●注
main.c （C言語版）			
op.asm	オプション・バイト設定用アセンブラ・ソース・ファイル （ウォッチドッグ・タイマの動作設定，低速内蔵発振器の設定などを行います）	●	●
78K0Lx3_Thermistor.prw	統合開発環境 PM plus用ワーク・スペース・ファイル		●
78K0Lx3_Thermistor.prj	統合開発環境 PM plus用プロジェクト・ファイル		●

注　アセンブリ言語版には「main.asm」，C言語版には「main.c」が同封されています。

備考　  　：ソース・ファイルのみ同封

 　：統合開発環境 PM plusで使用するファイルを同封

4. 2　使用する内蔵周辺機能

このサンプル・プログラムでは，マイコンに内蔵する次の周辺機能を使用します。

- 高速内蔵発振回路

  CPUクロック，周辺ハードウエア・クロック用。
- 8ビット・タイマH2

  温度測定タイミング作成用に，100 msのインターバル・タイマとして使用します。
- 16ビット・タイマ／イベント・カウンタ00

  コンデンサの放電パルス幅測定用。パルス幅測定機能を使用します。
- シリアル・インタフェースUART6

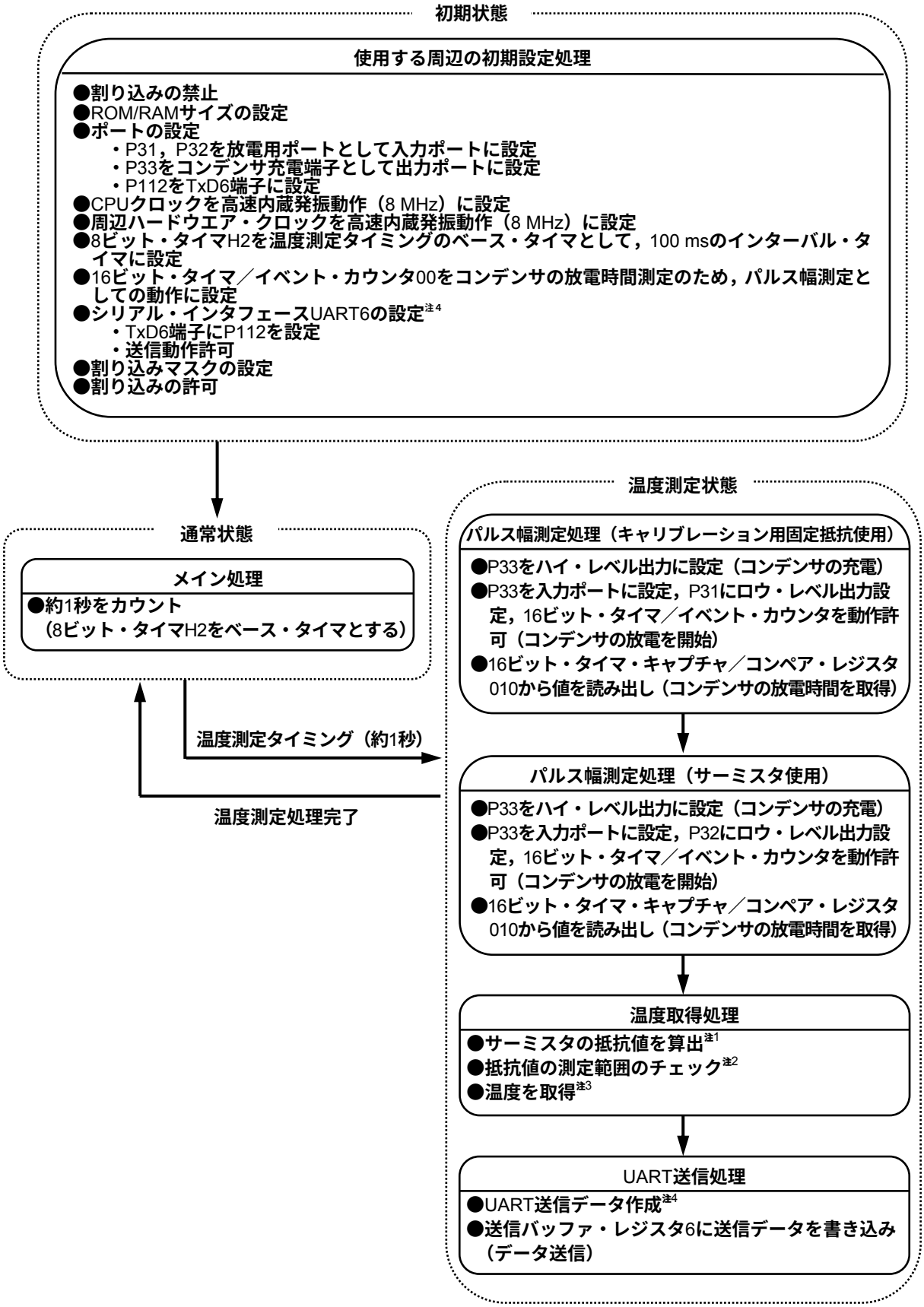
  温度測定結果の送信に使用します。

4. 3　初期設定と動作概要

このサンプル・プログラムでは，使用する周辺の初期設定にて，ポートの設定や，クロック周波数の選択，8ビット・タイマH2の設定，16ビット・タイマ／イベント・カウンタ00の設定，シリアル・インタフェースUART6の設定などを行います。使用する周辺の初期設定完了後，約1秒ごとにキャリブレーション用固定抵抗を使用したコンデンサの放電時間の測定，サーミスタを使用したコンデンサの放電時間の測定，コンデンサの放電時間から温度の取得，および温度測定結果のUART送信を行います。約1秒のタイミングは，8ビット・タイマH2をベース・タイマとしてカウントします。

詳細については，次の状態遷移図（ステート・チャート）に示します。

注　UART通信の設定内容詳細，および送信データの詳細については，4. 4　UART**送信データのフォーマット**を参照してください。



注1. 処理の詳細については，2. 1　**サーミスタの抵抗値算出**を参照してください。

2. 本サンプル・プログラムの温度の測定範囲（42.0～32.0 ℃）に対する抵抗値の測定範囲は，24.5～37.0 kΩです。

3. 処理の詳細については，2. 2　**抵抗値の温度変換**を参照してください。

4. UART通信の設定内容詳細，および送信データの詳細については，4. 4　UART**送信データのフォーマット**を参照してください。

4. 4　UART送信データのフォーマット

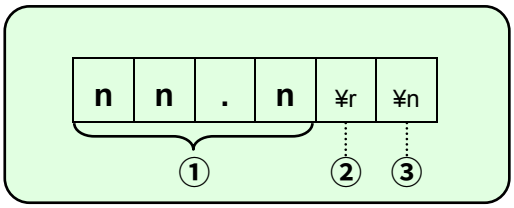
シリアル・インタフェースUART6により送信するデータの内容を説明します。

シリアル・インタフェースUART6の設定は下記のとおりです。

設定項目	設定内容
ボー・レート	115200 bps
送信データのキャラクタ長	8ビット
パリティ・ビット	出力しない
ストップ・ビット数	1
先頭ビット	LSB

データの送信は，約1秒に1回行います。1回の送信データの長さは6バイトです。温度測定結果をASCIIコードに変換して送信します。送信データのフォーマットを図4－1に示します。

図4－1　UART送信データのフォーマット



①測定した温度を小数点付きで表します。温度の測定範囲は32.0～42.0 ℃です。温度測定結果が測定エラーの場合，送信データは「＊＊．＊」となります。（n = 0-9，＊）

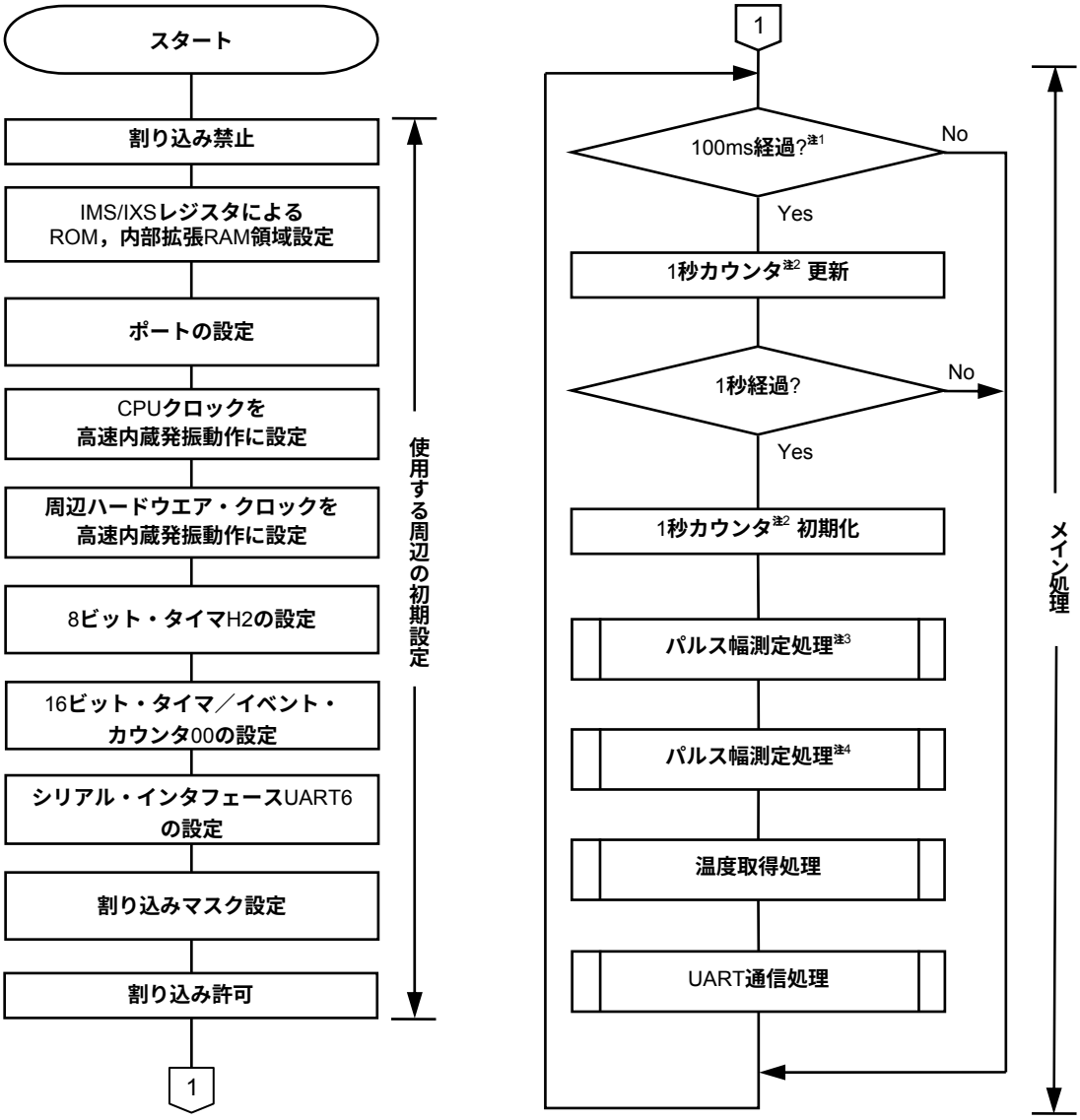
②復帰コードです。

③改行コードです。

4.5　フロー・チャート

このサンプル・プログラムのフロー・チャートを次に示します。

＜使用する周辺の初期設定処理＞



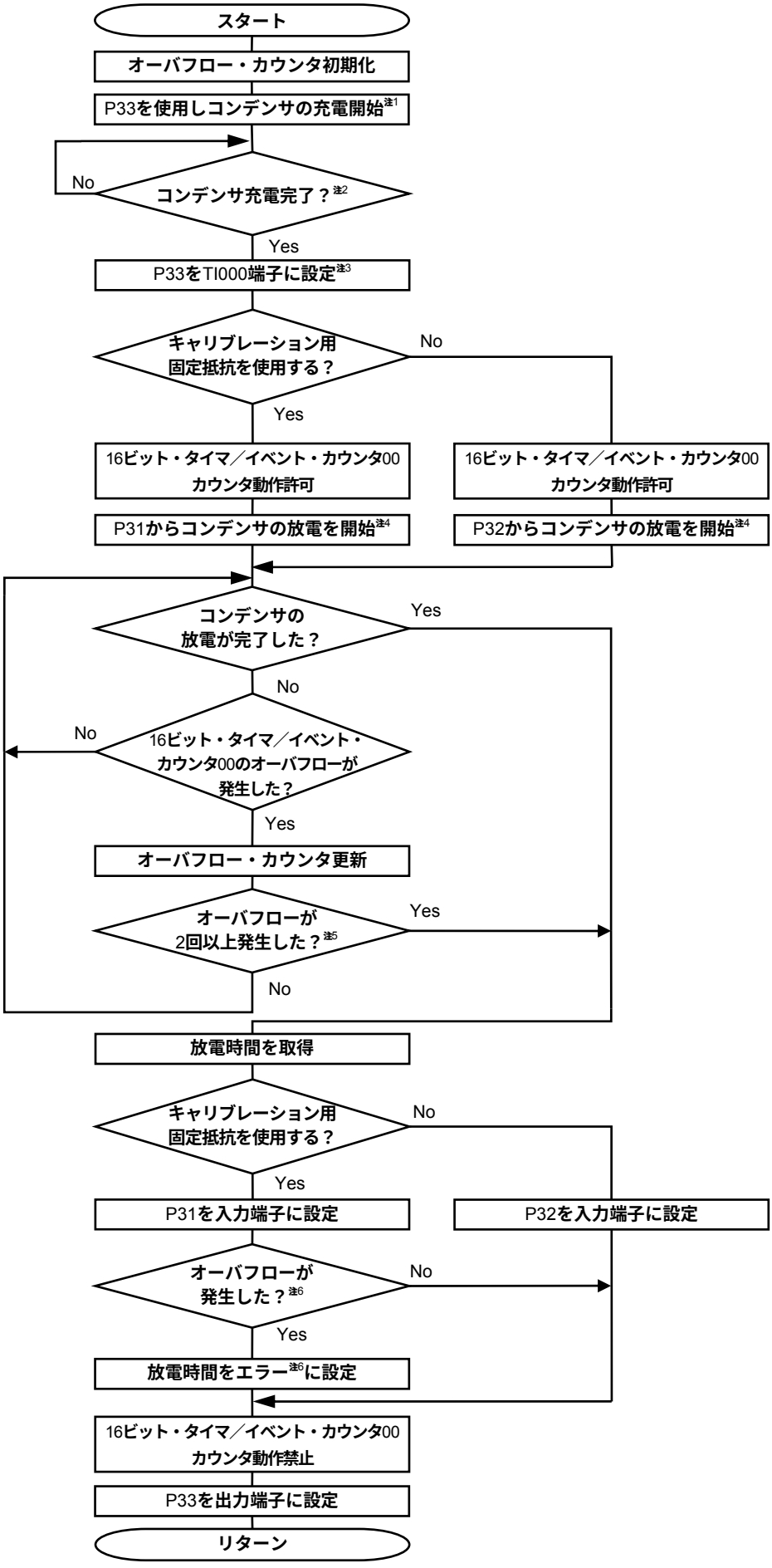
注1. 8ビット・タイマH2をベース・タイマとして使用しています。

2. 8ビット・タイマH2をベース・タイマとして，約1秒のタイミングをカウントする変数です。

3. キャリブレーション用固定抵抗を使用してコンデンサの放電パルス幅を測定します。

4. サーミスタを使用してコンデンサの放電パルス幅を測定します。

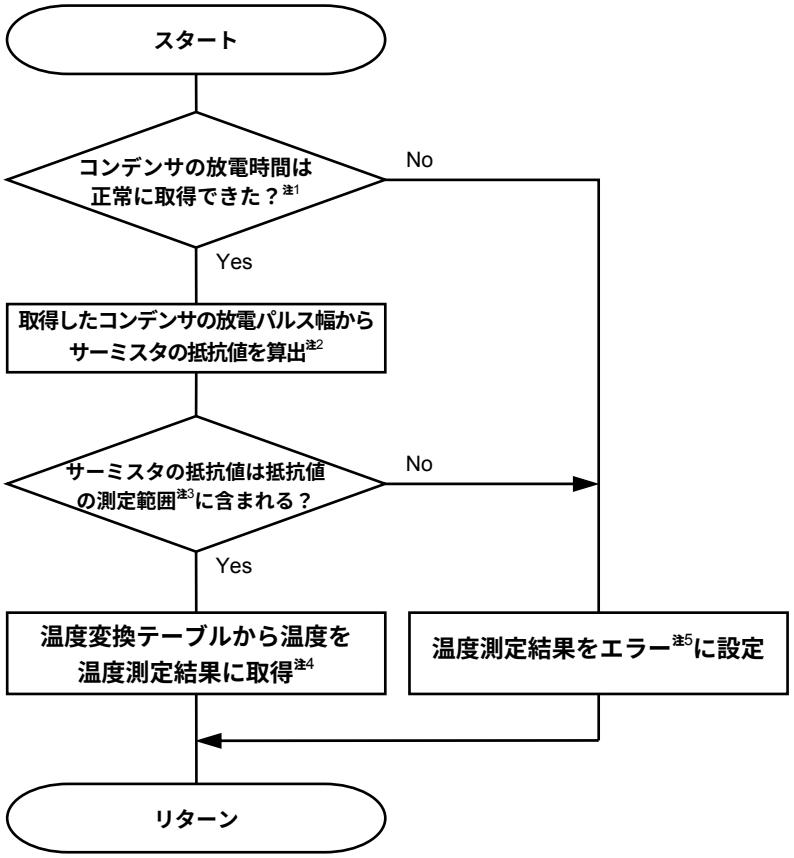
＜パルス幅測定処理＞


スタート
オーバフロー・カウンタ初期化
P33を使用しコンデンサの充電開始注1
No
コンデンサ充電完了？注2
Yes
P33をTI000端子に設定注3
キャリブレーション用
固定抵抗を使用する？
No
Yes
16ビット・タイマ／イベント・カウンタ00
カウンタ動作許可
P31からコンデンサの放電を開始注4
16ビット・タイマ／イベント・カウンタ00
カウンタ動作許可
P32からコンデンサの放電を開始注4
コンデンサの
放電が完了した？
Yes
No
No
16ビット・タイマ／イベント・
カウンタ00のオーバフローが
発生した？
Yes
オーバフロー・カウンタ更新
オーバフローが
2回以上発生した？注5
Yes
No
放電時間を取得
キャリブレーション用
固定抵抗を使用する？
No
Yes
P31を入力端子に設定
P32を入力端子に設定
オーバフローが
発生した？注6
No
Yes
放電時間をエラー注6に設定
16ビット・タイマ／イベント・カウンタ00
カウンタ動作禁止
P33を出力端子に設定
リターン

注1. P33をハイ・レベル出力に設定します。

2. コンデンサの充電完了待ちのため，約2 msウエイトします。
3. P33を入力ポートに設定します。
4. P31またはP32をロウ・レベル出力に設定します。
5. キャリブレーション用固定抵抗を使用して放電時間を測定する場合,キャリブレーション用固定抵抗とコンデンサは，16ビット・タイマ・カウンタ00がオーバフローしないように選択しています。また，サーミスタを使用して放電時間を測定する場合にオーバフローが2回以上発生すると，サーミスタの抵抗値の計算結果が測定範囲外になります。そのため，オーバフローが2回以上発生した場合は，測定エラーと判断し，コンデンサの放電時間の測定を中断します。
6. キャリブレーション用固定抵抗とコンデンサは,キャリブレーション用固定抵抗を使用して放電時間を測定する場合，16ビット・タイマ・カウンタ00がオーバフローしないように選択しています。そのため，キャリブレーション用放電パルス幅の測定でオーバフローが発生すると，測定エラーとして放電時間に0を設定します。

＜温度取得処理＞



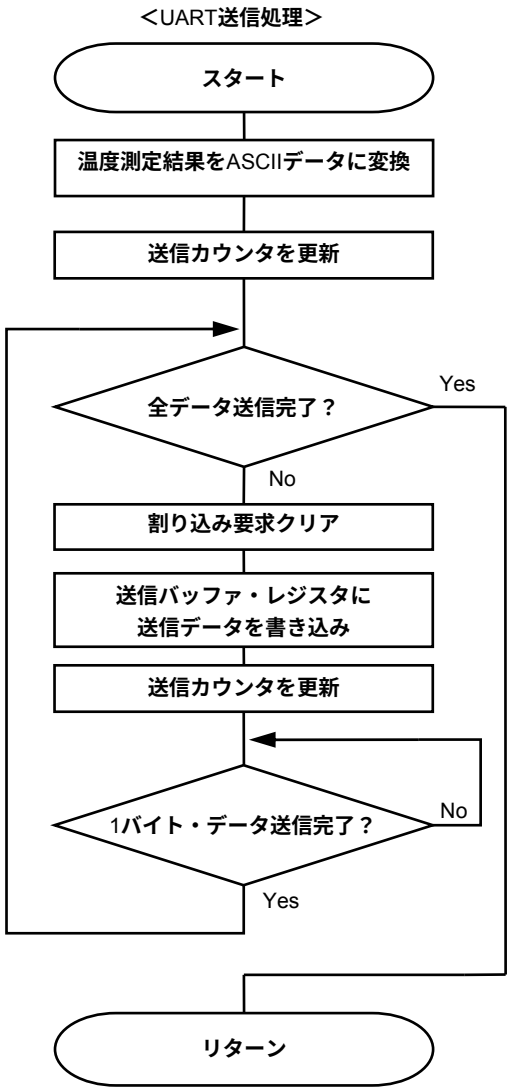
注1. キャリブレーション用固定抵抗を使用したコンデンサの放電時間が正常に取得できているか，またサーミスタを使用したコンデンサの放電時間の測定でオーバフローが2回以上発生していないかを判定します。

2. 放電時間と抵抗値の比より算出します。詳細については，2. 1　**サーミスタの抵抗値算出**を参照してください。

3. 温度の測定範囲（32.0～42.0 ℃）に対する抵抗値の測定範囲は，37.0～24.5 kΩです。

4. 温度変換テーブルから取得します。詳細については，2. 2　**抵抗値の温度変換**を参照してください。

5. FFFFHを設定します。


＜UART送信処理＞
スタート
温度測定結果をASCIIデータに変換
送信カウンタを更新
全データ送信完了？
Yes
No
割り込み要求クリア
送信バッファ・レジスタに
送信データを書き込み
送信カウンタを更新
1バイト・データ送信完了？
No
Yes
リターン

第5章　設定方法について

この章では，使用する周辺の初期設定，およびサンプル・プログラムの処理内容について説明します。説明するサンプル・プログラムは，78K0/LF3のものです。

オプション・バイト，ベクタ・テーブル，メモリ空間の設定，スタック・ポインタ，クロック周波数，レジスタ設定方法の詳細については，各製品のユーザーズ・マニュアル（78K0/Lx3）とサンプル・プログラムを参照してください。

アセンブラ命令については，78K0**マイクロコントローラ　命令編　ユーザーズ・マニュアル**を参照してください。

5.1　使用する周辺の初期設定

(1) 変数の定義

アセンブリ言語で定義する変数は次のとおりです。

① R1SECCNT　：1秒カウンタです。8ビット・タイマH2をベース・タイマとしてカウントします。
② ROVFCNT　：コンデンサの放電時間を測定する際，オーバフロー回数をカウントします。
③ RTXBUF　：UART通信で送信するデータを格納する配列です。温度測定結果をASCIIコードに変換して格納します。
④ RCALBCNT　：キャリブレーション用固定抵抗を使用して測定したコンデンサの放電時間を格納します。
⑤ RTHERMCNT：サーミスタを使用して測定したコンデンサの放電時間を格納します。
⑥ RHEAT　：温度測定結果を格納します。
⑦ RTEMP16A　：16ビットの演算用領域です。
⑧ RTEMP16B　：16ビットの演算用領域です。
⑨ RTEMP32　：32ビットの演算用領域です。

```
;=============================================================================
;
;       RAMの定義
;
;=============================================================================
DTHERMO         DSEG    SADDR
①-----R1SECCNT:       DS      1               ;100ms(TMH2)をベース・タイマとして時間をカウントする
 C1SEC          EQU     (1000/100)      ;1秒カウント用

②-----ROVFCNT:        DS      1               ;TM00オーバフロー回数カウンタ

③-----RTXBUF:         DS      6               ;送信データ･バッファ

DTHERMOP        DSEG    SADDRP  ;温度測定関連RAM
④-----RCALBCNT:       DS      2               ;キャリブレーション用TI000パルス幅測定値取得用
⑤-----RTHERMCNT:      DS      2               ;サーミスタ用TI000パルス幅測定値取得用
⑥-----RHEAT:          DS      2               ;算出した温度を保存 ※測定エラーの場合は"FFFFH"となる

⑦-----RTEMP16A:       DS      2               ;抵抗値計算用変数
⑧-----RTEMP16B:       DS      2               ;抵抗値計算用変数
⑨-----RTEMP32:        DS      4               ;抵抗値計算用変数
```

C言語で定義する変数は次のとおりです。

① uc1secCnt　：1秒カウンタです。8ビット・タイマH2をベース・タイマとしてカウントします。

② ushCalibrationCnt　：キャリブレーション用固定抵抗を使用して測定したコンデンサの放電時間を格納します。

③ ushThermistorCnt　：サーミスタを使用して測定したコンデンサの放電時間を格納します。

④ ucOVFcnt　：コンデンサの放電時間を測定する際，オーバフロー回数をカウントします。

⑤ ushHeatData　：温度測定結果を格納します。

⑥ ucTxBuffer[6]　：UART通信で送信するデータを格納する配列です。温度測定結果をASCIIコードに変換して格納します。

⑦ ucTxBufferCounter　：UART送信処理で，送信したデータ数をカウントします。

```
①-----unsigned char uc1secCnt;            /* 100ms(TMH2)をベース・タイマとして1秒をカウントする */
       #define         TMH2_1SEC       (1000/100)      /* 1秒カウント用 */

②-----unsigned short ushCalibrationCnt;   /* キャリブレーション用TI000パルス幅測定値取得用 */
③-----unsigned short ushThermistorCnt;    /* サーミスタ用TI000パルス幅測定値取得用 */
④-----unsigned char ucOVFcnt;             /* TM00オーバフロー回数カウンタ */
⑤-----unsigned short ushHeatData;         /* 算出した温度を保存　※測定エラーの場合は"FFFFH"となる */

⑥-----unsigned char ucTxBuffer[6];        /* 送信データ･バッファ */
⑦-----unsigned char ucTxBufferCounter;    /* 送信カウンタ */
```

(2) テーブルの定義

アセンブリ言語で，抵抗値を温度に変換するために使用する温度変換テーブルを，以下のように定義します。サーミスタの抵抗値が測定範囲内であった場合，サーミスタの抵抗値について，測定範囲の最小抵抗値（24.5 kΩ）からのオフセットを求め，温度変換テーブルの先頭アドレスからのオフセットに変換して温度（BCD）を取得します。温度変換テーブルのデータの詳細については，2. 2　**抵抗値の温度変換**を参照してください。

```
;============================================================================
;
;       ROMの定義
;
;============================================================================
CREGACC CSEG    UNITP
;----------------------------------------------------------------------------
;       抵抗を温度に変換するテーブル
;----------------------------------------------------------------------------
;       24.5 kΩを基準としたオフセット[100Ω]より，温度を参照します。
;       参照される温度はBCD[0.1℃]です。
;----------------------------------------------------------------------------
TR2HEAT:
        DW      0420H           ;24.5 kΩ → 42.0
        DW      0419H           ;24.6 kΩ → 41.9
        DW      0418H           ;24.7 kΩ → 41.8
        DW      0417H           ;24.8 kΩ → 41.7
        DW      0416H           ;24.9 kΩ → 41.6
        DW      0415H           ;25.0 kΩ → 41.5
        DW      0414H           ;25.1 kΩ → 41.4
        DW      0413H           ;25.2 kΩ → 41.3
        DW      0412H           ;25.3 kΩ → 41.2
        DW      0411H           ;25.4 kΩ → 41.1
        DW      0410H           ;25.5 kΩ → 41.0
        DW      0409H           ;25.6 kΩ → 40.9
        DW      0408H           ;25.7 kΩ → 40.8
        DW      0407H           ;25.8 kΩ → 40.7
        DW      0406H           ;25.9 kΩ → 40.6
        DW      0405H           ;26.0 kΩ → 40.5
        DW      0405H           ;26.1 kΩ → 40.5
        DW      0404H           ;26.2 kΩ → 40.4
        DW      0403H           ;26.3 kΩ → 40.3
        DW      0402H           ;26.4 kΩ → 40.2
        DW      0401H           ;26.5 kΩ → 40.1
        DW      0400H           ;26.6 kΩ → 40.0
        DW      0399H           ;26.7 kΩ → 39.9
        DW      0398H           ;26.8 kΩ → 39.8
        DW      0397H           ;26.9 kΩ → 39.7
        DW      0396H           ;27.0 kΩ → 39.6
        DW      0395H           ;27.1 kΩ → 39.5
        DW      0394H           ;27.2 kΩ → 39.4
        DW      0393H           ;27.3 kΩ → 39.3
        DW      0392H           ;27.4 kΩ → 39.2
        DW      0392H           ;27.5 kΩ → 39.2
        DW      0391H           ;27.6 kΩ → 39.1
        DW      0390H           ;27.7 kΩ → 39.0
        DW      0389H           ;27.8 kΩ → 38.9
        DW      0388H           ;27.9 kΩ → 38.8
        DW      0387H           ;28.0 kΩ → 38.7
        DW      0386H           ;28.1 kΩ → 38.6
        DW      0385H           ;28.2 kΩ → 38.5
        DW      0384H           ;28.3 kΩ → 38.4
        DW      0384H           ;28.4 kΩ → 38.4
        DW      0383H           ;28.5 kΩ → 38.3
        DW      0382H           ;28.6 kΩ → 38.2
        DW      0381H           ;28.7 kΩ → 38.1
        DW      0380H           ;28.8 kΩ → 38.0
        DW      0379H           ;28.9 kΩ → 37.9
        DW      0378H           ;29.0 kΩ → 37.8
        DW      0378H           ;29.1 kΩ → 37.8
        DW      0377H           ;29.2 kΩ → 37.7
        DW      0376H           ;29.3 kΩ → 37.6
        DW      0375H           ;29.4 kΩ → 37.5
        DW      0374H           ;29.5 kΩ → 37.4
        DW      0373H           ;29.6 kΩ → 37.3
        DW      0373H           ;29.7 kΩ → 37.3
        DW      0372H           ;29.8 kΩ → 37.2
        DW      0371H           ;29.9 kΩ → 37.1
        DW      0370H           ;30.0 kΩ → 37.0
```

```
        DW      0369H           ;30.1 kΩ → 36.9
        DW      0368H           ;30.2 kΩ → 36.8
        DW      0368H           ;30.3 kΩ → 36.8
        DW      0367H           ;30.4 kΩ → 36.7
        DW      0366H           ;30.5 kΩ → 36.6
        DW      0365H           ;30.6 kΩ → 36.5
        DW      0365H           ;30.7 kΩ → 36.5
        DW      0364H           ;30.8 kΩ → 36.4
        DW      0363H           ;30.9 kΩ → 36.3
        DW      0362H           ;31.0 kΩ → 36.2
        DW      0361H           ;31.1 kΩ → 36.1
        DW      0361H           ;31.2 kΩ → 36.1
        DW      0360H           ;31.3 kΩ → 36.0
        DW      0359H           ;31.4 kΩ → 35.9
        DW      0358H           ;31.5 kΩ → 35.8
        DW      0358H           ;31.6 kΩ → 35.8
        DW      0357H           ;31.7 kΩ → 35.7
        DW      0356H           ;31.8 kΩ → 35.6
        DW      0355H           ;31.9 kΩ → 35.5
        DW      0354H           ;32.0 kΩ → 35.4
        DW      0354H           ;32.1 kΩ → 35.4
        DW      0353H           ;32.2 kΩ → 35.3
        DW      0352H           ;32.3 kΩ → 35.2
        DW      0351H           ;32.4 kΩ → 35.1
        DW      0351H           ;32.5 kΩ → 35.1
        DW      0350H           ;32.6 kΩ → 35.0
        DW      0349H           ;32.7 kΩ → 34.9
        DW      0348H           ;32.8 kΩ → 34.8
        DW      0348H           ;32.9 kΩ → 34.8
        DW      0347H           ;33.0 kΩ → 34.7
        DW      0346H           ;33.1 kΩ → 34.6
        DW      0346H           ;33.2 kΩ → 34.6
        DW      0345H           ;33.3 kΩ → 34.5
        DW      0344H           ;33.4 kΩ → 34.4
        DW      0343H           ;33.5 kΩ → 34.3
        DW      0343H           ;33.6 kΩ → 34.3
        DW      0342H           ;33.7 kΩ → 34.2
        DW      0341H           ;33.8 kΩ → 34.1
        DW      0341H           ;33.9 kΩ → 34.1
        DW      0340H           ;34.0 kΩ → 34.0
        DW      0339H           ;34.1 kΩ → 33.9
        DW      0338H           ;34.2 kΩ → 33.8
        DW      0338H           ;34.3 kΩ → 33.8
        DW      0337H           ;34.4 kΩ → 33.7
        DW      0336H           ;34.5 kΩ → 33.6
        DW      0336H           ;34.6 kΩ → 33.6
        DW      0335H           ;34.7 kΩ → 33.5
        DW      0334H           ;34.8 kΩ → 33.4
        DW      0334H           ;34.9 kΩ → 33.4
        DW      0333H           ;35.0 kΩ → 33.3
        DW      0332H           ;35.1 kΩ → 33.2
        DW      0332H           ;35.2 kΩ → 33.2
        DW      0331H           ;35.3 kΩ → 33.1
        DW      0330H           ;35.4 kΩ → 33.0
        DW      0330H           ;35.5 kΩ → 33.0
        DW      0329H           ;35.6 kΩ → 32.9
        DW      0328H           ;35.7 kΩ → 32.8
        DW      0328H           ;35.8 kΩ → 32.8
        DW      0327H           ;35.9 kΩ → 32.7
        DW      0326H           ;36.0 kΩ → 32.6
        DW      0326H           ;36.1 kΩ → 32.6
        DW      0325H           ;36.2 kΩ → 32.5
        DW      0324H           ;36.3 kΩ → 32.4
        DW      0324H           ;36.4 kΩ → 32.4
        DW      0323H           ;36.5 kΩ → 32.3
        DW      0322H           ;36.6 kΩ → 32.2
        DW      0322H           ;36.7 kΩ → 32.2
        DW      0321H           ;36.8 kΩ → 32.1
        DW      0320H           ;36.9 kΩ → 32.0
        DW      0320H           ;37.0 kΩ → 32.0
TR2HEATE:
```

C言語でも，アセンブリ言語と同様に温度変換テーブルの定義を行います。

```
/*==========================================================================

        ＲＯＭの定義

==========================================================================*/
/*--------------------------------------------------------------------------
        抵抗を温度に変換するテーブル
--------------------------------------------------------------------------
        24.5 kΩを基準としたオフセット[100Ω]より，温度を参照します。
        参照される温度はBCD[0.1℃]です。
--------------------------------------------------------------------------*/
const unsigned short tR2Heat[] =
{
         0x0420         /* 24.5 kΩ → 42.0 */
        ,0x0419         /* 24.6 kΩ → 41.9 */
        ,0x0418         /* 24.7 kΩ → 41.8 */
        ,0x0417         /* 24.8 kΩ → 41.7 */
        ,0x0416         /* 24.9 kΩ → 41.6 */
        ,0x0415         /* 25.0 kΩ → 41.5 */
        ,0x0414         /* 25.1 kΩ → 41.4 */
        ,0x0413         /* 25.2 kΩ → 41.3 */
        ,0x0412         /* 25.3 kΩ → 41.2 */
        ,0x0411         /* 25.4 kΩ → 41.1 */
        ,0x0410         /* 25.5 kΩ → 41.0 */
        ,0x0409         /* 25.6 kΩ → 40.9 */
        ,0x0408         /* 25.7 kΩ → 40.8 */
        ,0x0407         /* 25.8 kΩ → 40.7 */
        ,0x0406         /* 25.9 kΩ → 40.6 */
        ,0x0405         /* 26.0 kΩ → 40.5 */
        ,0x0405         /* 26.1 kΩ → 40.5 */
        ,0x0404         /* 26.2 kΩ → 40.4 */
        ,0x0403         /* 26.3 kΩ → 40.3 */
        ,0x0402         /* 26.4 kΩ → 40.2 */
        ,0x0401         /* 26.5 kΩ → 40.1 */
        ,0x0400         /* 26.6 kΩ → 40.0 */
        ,0x0399         /* 26.7 kΩ → 39.9 */
        ,0x0398         /* 26.8 kΩ → 39.8 */
        ,0x0397         /* 26.9 kΩ → 39.7 */
        ,0x0396         /* 27.0 kΩ → 39.6 */
        ,0x0395         /* 27.1 kΩ → 39.5 */
        ,0x0394         /* 27.2 kΩ → 39.4 */
        ,0x0393         /* 27.3 kΩ → 39.3 */
        ,0x0392         /* 27.4 kΩ → 39.2 */
        ,0x0392         /* 27.5 kΩ → 39.2 */
        ,0x0391         /* 27.6 kΩ → 39.1 */
        ,0x0390         /* 27.7 kΩ → 39.0 */
        ,0x0389         /* 27.8 kΩ → 38.9 */
        ,0x0388         /* 27.9 kΩ → 38.8 */
        ,0x0387         /* 28.0 kΩ → 38.7 */
        ,0x0386         /* 28.1 kΩ → 38.6 */
        ,0x0385         /* 28.2 kΩ → 38.5 */
        ,0x0384         /* 28.3 kΩ → 38.4 */
        ,0x0384         /* 28.4 kΩ → 38.4 */
        ,0x0383         /* 28.5 kΩ → 38.3 */
        ,0x0382         /* 28.6 kΩ → 38.2 */
        ,0x0381         /* 28.7 kΩ → 38.1 */
        ,0x0380         /* 28.8 kΩ → 38.0 */
        ,0x0379         /* 28.9 kΩ → 37.9 */
        ,0x0378         /* 29.0 kΩ → 37.8 */
        ,0x0378         /* 29.1 kΩ → 37.8 */
        ,0x0377         /* 29.2 kΩ → 37.7 */
        ,0x0376         /* 29.3 kΩ → 37.6 */
        ,0x0375         /* 29.4 kΩ → 37.5 */
        ,0x0374         /* 29.5 kΩ → 37.4 */
        ,0x0373         /* 29.6 kΩ → 37.3 */
        ,0x0373         /* 29.7 kΩ → 37.3 */
        ,0x0372         /* 29.8 kΩ → 37.2 */
        ,0x0371         /* 29.9 kΩ → 37.1 */
        ,0x0370         /* 30.0 kΩ → 37.0 */
        ,0x0369         /* 30.1 kΩ → 36.9 */
        ,0x0368         /* 30.2 kΩ → 36.8 */
        ,0x0368         /* 30.3 kΩ → 36.8 */
        ,0x0367         /* 30.4 kΩ → 36.7 */
        ,0x0366         /* 30.5 kΩ → 36.6 */
```

```
        ,0x0365         /* 30.6 kΩ → 36.5 */
        ,0x0365         /* 30.7 kΩ → 36.5 */
        ,0x0364         /* 30.8 kΩ → 36.4 */
        ,0x0363         /* 30.9 kΩ → 36.3 */
        ,0x0362         /* 31.0 kΩ → 36.2 */
        ,0x0361         /* 31.1 kΩ → 36.1 */
        ,0x0361         /* 31.2 kΩ → 36.1 */
        ,0x0360         /* 31.3 kΩ → 36.0 */
        ,0x0359         /* 31.4 kΩ → 35.9 */
        ,0x0358         /* 31.5 kΩ → 35.8 */
        ,0x0358         /* 31.6 kΩ → 35.8 */
        ,0x0357         /* 31.7 kΩ → 35.7 */
        ,0x0356         /* 31.8 kΩ → 35.6 */
        ,0x0355         /* 31.9 kΩ → 35.5 */
        ,0x0354         /* 32.0 kΩ → 35.4 */
        ,0x0354         /* 32.1 kΩ → 35.4 */
        ,0x0353         /* 32.2 kΩ → 35.3 */
        ,0x0352         /* 32.3 kΩ → 35.2 */
        ,0x0351         /* 32.4 kΩ → 35.1 */
        ,0x0351         /* 32.5 kΩ → 35.1 */
        ,0x0350         /* 32.6 kΩ → 35.0 */
        ,0x0349         /* 32.7 kΩ → 34.9 */
        ,0x0348         /* 32.8 kΩ → 34.8 */
        ,0x0348         /* 32.9 kΩ → 34.8 */
        ,0x0347         /* 33.0 kΩ → 34.7 */
        ,0x0346         /* 33.1 kΩ → 34.6 */
        ,0x0346         /* 33.2 kΩ → 34.6 */
        ,0x0345         /* 33.3 kΩ → 34.5 */
        ,0x0344         /* 33.4 kΩ → 34.4 */
        ,0x0343         /* 33.5 kΩ → 34.3 */
        ,0x0343         /* 33.6 kΩ → 34.3 */
        ,0x0342         /* 33.7 kΩ → 34.2 */
        ,0x0341         /* 33.8 kΩ → 34.1 */
        ,0x0341         /* 33.9 kΩ → 34.1 */
        ,0x0340         /* 34.0 kΩ → 34.0 */
        ,0x0339         /* 34.1 kΩ → 33.9 */
        ,0x0338         /* 34.2 kΩ → 33.8 */
        ,0x0338         /* 34.3 kΩ → 33.8 */
        ,0x0337         /* 34.4 kΩ → 33.7 */
        ,0x0336         /* 34.5 kΩ → 33.6 */
        ,0x0336         /* 34.6 kΩ → 33.6 */
        ,0x0335         /* 34.7 kΩ → 33.5 */
        ,0x0334         /* 34.8 kΩ → 33.4 */
        ,0x0334         /* 34.9 kΩ → 33.4 */
        ,0x0333         /* 35.0 kΩ → 33.3 */
        ,0x0332         /* 35.1 kΩ → 33.2 */
        ,0x0332         /* 35.2 kΩ → 33.2 */
        ,0x0331         /* 35.3 kΩ → 33.1 */
        ,0x0330         /* 35.4 kΩ → 33.0 */
        ,0x0330         /* 35.5 kΩ → 33.0 */
        ,0x0329         /* 35.6 kΩ → 32.9 */
        ,0x0328         /* 35.7 kΩ → 32.8 */
        ,0x0328         /* 35.8 kΩ → 32.8 */
        ,0x0327         /* 35.9 kΩ → 32.7 */
        ,0x0326         /* 36.0 kΩ → 32.6 */
        ,0x0326         /* 36.1 kΩ → 32.6 */
        ,0x0325         /* 36.2 kΩ → 32.5 */
        ,0x0324         /* 36.3 kΩ → 32.4 */
        ,0x0324         /* 36.4 kΩ → 32.4 */
        ,0x0323         /* 36.5 kΩ → 32.3 */
        ,0x0322         /* 36.6 kΩ → 32.2 */
        ,0x0322         /* 36.7 kΩ → 32.2 */
        ,0x0321         /* 36.8 kΩ → 32.1 */
        ,0x0320         /* 36.9 kΩ → 32.0 */
        ,0x0320         /* 37.0 kΩ → 32.0 */
};
```

(3) 使用する周辺の初期設定処理

アセンブリ言語での使用する周辺の初期設定処理では，次の動作を行います。

① 割り込みを禁止します。

② レジスタ・バンクを設定します。

③ スタック・ポインタの設定を行います。

④ メモリ・サイズと内部拡張RAMサイズの設定を行います。

使用するマイコンに適したIMS, IXS[注]を設定してください。

⑤ ポートの設定を行います。

P112をTxD6端子として使用するため，ハイ・レベル出力に設定します。P33をコンデンサ充電用ポートとして使用するため，ロウ・レベル出力に設定します。P31，およびP32はコンデンサの放電時間を測定する際に放電用ポートとして使用するため，初期設定として入力ポートに設定します。その他のポートはロウ・レベル出力に設定します。

⑥ クロック周波数の設定を行います。

CPUクロック，および周辺ハードウエア・クロックを高速内蔵発振クロック動作に設定します。

⑦ 8ビット・タイマH2の設定を行います。

温度測定処理のタイミング作成のベース・タイマとするため，100 ms周期のインターバル・タイマに設定します。

⑧ 16ビット・タイマ／イベント・カウンタ00を設定します。

コンデンサの放電パルス幅を測定するため，パルス幅測定（TI000端子の有効エッジ入力によるクリア＆スタート・モード）としての動作に設定します。16ビット・タイマ・キャプチャ／コンペア・レジスタ010の動作モードはキャプチャ・レジスタとして動作を選択し，16ビット・タイマ・キャプチャ／コンペア・レジスタ000のキャプチャ・トリガはTI000端子の有効エッジの逆相でキャプチャするを選択します。TI000端子の有効エッジは立ち下がりエッジを選択します。16ビット・タイマ／イベント・カウンタ00の動作許可は，パルス幅測定処理にて行います。

⑨ シリアル・インタフェースUART6を下記のとおり設定します。

・ボー・レート　　：115200 bps
・データのキャラクタ長　：8ビット
・パリティ・ビット　　：出力しない
・ストップ・ビット数　：1
・先頭ビット　　：LSB

また，TxD6端子の入力をP112に設定し，TxD6端子の入力許可を設定します。

⑩ 割り込みマスクを設定します。

すべての割り込みをマスクします。

⑪ 割り込みを許可します。

注　78K0/LF3, 78K0/LE3のみ

```
;*******************************************************************************
;
;
;       使用する周辺の初期設定
;
;
;*******************************************************************************
XMAIN   CSEG    UNIT
RESET_START:

;-------------------------------------------------------------------------------
;       割り込み禁止
;-------------------------------------------------------------------------------
①      DI
;-------------------------------------------------------------------------------
;       レジスタ・バンク設定
;-------------------------------------------------------------------------------
②      SEL     RB0
;-------------------------------------------------------------------------------
;       スタック・ポインタの設定
;-------------------------------------------------------------------------------
③      MOVW    SP,     #STACKTOP
;-------------------------------------------------------------------------------
;       ROM/RAMサイズの設定

;-------------------------------------------------------------------------------
;       モデルにより設定値が異なるので注意してください。
;       使用モデルの設定を有効にしてください。（デフォルトではuPD78F0485）
;-------------------------------------------------------------------------------
        ;uPD78F0471,uPD78F0481,uPD78F0491使用時の設定
④       ;MOV    IMS,    #04H                    ;ROMサイズの設定
        ;MOV    IXS,    #0CH                    ;内部拡張RAMサイズの設定

        ;uPD78F0472,uPD78F0482,uPD78F0492使用時の設定
        ;MOV    IMS,    #0C6H                   ;ROMサイズの設定
        ;MOV    IXS,    #0CH                    ;内部拡張RAMサイズの設定

        ;uPD78F0473,uPD78F0483,uPD78F0493使用時の設定
        ;MOV    IMS,    #0C8H                   ;ROMサイズの設定
        ;MOV    IXS,    #0CH                    ;内部拡張RAMサイズの設定

        ;uPD78F0474,uPD78F0484,uPD78F0494使用時の設定
        ;MOV    IMS,    #0CCH                   ;ROMサイズの設定
        ;MOV    IXS,    #0AH                    ;内部拡張RAMサイズの設定

        ;uPD78F0475,uPD78F0485,uPD78F0495使用時の設定
        MOV     IMS,    #0CFH                   ;ROMサイズの設定
        MOV     IXS,    #0AH                    ;内部拡張RAMサイズの設定

;-------------------------------------------------------------------------------
;       ポート1の設定
;-------------------------------------------------------------------------------
        MOV     P1,     #00000000B              ;P1初期値設定
                        ;++++++++-------------- P17/P16/P15/P14/P13/P12/P11/P10:未使用(0)
        MOV     PM1,    #00000000B              ;P1入出力設定
                        ;++++++++-------------- PM17/PM16/PM15/PM14/PM13/PM12/PM11/PM10:未使用(0)
;-------------------------------------------------------------------------------
;       ポート2の設定
⑤;-------------------------------------------------------------------------------
        MOV     P2,     #00000000B              ;P2初期値設定
                        ;++++++++-------------- P27/P26/P25/P24/P23/P22/P21/P20:未使用(0)
        MOV     PM2,    #00000000B              ;P1入出力設定
                        ;++++++++-------------- PM27/PM26/PM25/PM24/PM23/PM22/PM21/PM20:未使用(0)
;-------------------------------------------------------------------------------
;       ポート3の設定
;-------------------------------------------------------------------------------
        MOV     P3,     #00000000B              ;P3初期値設定
                        ;|||+++++-------------- P33/P32/P31/P34/P30:Lo(0)
                        ;+++-------------------- <000固定>
        MOV     PM3,    #11100110B              ;P3入出力設定
                        ;|||+|||+-------------- PM34/PM30:未使用(0)
                        ;|||  |++-------------- PM32/PM31:入力(1) 放電用ポートとして使用
                        ;|||  +---------------- PM33:出力(0) コンデンサ充電ポートとして使用
                        ;|||                       (パルス幅測定時はTI000として使用)
                        ;+++-------------------- <111固定>
```

⑤

```
;-------------------------------------------------------------------------------
;       ポート4の設定
;-------------------------------------------------------------------------------
        MOV     P4,     #00000000B              ;P4初期値設定
                        ;++++++++--------------- P47/P46/P45/P44/P43/P42/P41/P40:未使用(0)
        MOV     PM4,    #00000000B              ;P4入出力設定
                        ;++++++++--------------- PM47/PM46/PM45/PM44/PM43/PM42/PM41/PM40:未使用(0)
;-------------------------------------------------------------------------------
;       ポート8の設定
;-------------------------------------------------------------------------------
        MOV     P8,     #00000000B              ;P8初期値設定
                        ;||||++++--------------- P83/P82/P81/P80:未使用(0)
                        ;++++------------------- <0000固定>
        MOV     PM8,    #11110000B              ;P8入出力設定
                        ;||||++++--------------- PM83/PM82/PM81/PM80:未使用(0)
                        ;++++------------------- <1111固定>
;-------------------------------------------------------------------------------
;       ポート9の設定
;-------------------------------------------------------------------------------
        MOV     P9,     #00000000B              ;P9初期値設定
                        ;||||++++--------------- P93/P92/P91/P90:未使用(0)
                        ;++++------------------- <0000固定>
        MOV     PM9,    #11110000B              ;P9入出力設定
                        ;||||++++--------------- PM93/PM92/PM91/PM90:未使用(0)
                        ;++++------------------- <1111固定>
;-------------------------------------------------------------------------------
;       ポート10の設定
;-------------------------------------------------------------------------------
        MOV     P10,    #00000000B              ;P10初期値設定
                        ;||||++++--------------- P103/P102/P101/P100:未使用(0)
                        ;++++------------------- <0000固定>
        MOV     PM10,   #11110000B              ;P10入出力設定
                        ;||||++++--------------- PM103/PM102/PM101/PM100:未使用(0)
                        ;++++------------------- <1111固定>
;-------------------------------------------------------------------------------
;       ポート11の設定
;-------------------------------------------------------------------------------
        MOV     P11,    #00000100B              ;P11初期値設定
                        ;||||+|++--------------- P113/P111/P110:未使用(0)
                        ;||||  +---------------- P112:Hi(1)
                        ;++++------------------- <0000固定>
        MOV     PM11,   #11110000B      ;P11入出力設定
                        ;||||+|++--------------- PM113/PM111/PM110:未使用(0)
                        ;||||  +---------------- PM112:出力(0) TxD6として使用
                        ;++++------------------- <1111固定>
;-------------------------------------------------------------------------------
;       ポート12の設定
;-------------------------------------------------------------------------------
        MOV     P12,    #00000000B              ;P12初期値設定
                        ;|||||||+--------------- P120:未使用(0)
                        ;|||++++--------------- P124/P123/P122/P121:Read Only
                        ;+++-------------------- <000固定>
        MOV     PM12,   #11111110B              ;P12入出力設定
                        ;|||||||+--------------- PM120:未使用(0)
                        ;+++++++---------------- <1111111固定>
;-------------------------------------------------------------------------------
;       ポート13の設定
;-------------------------------------------------------------------------------
        MOV     P13,    #00000000B              ;P13初期値設定
                        ;||||++++--------------- P133/P132/P131/P130:未使用(0)
                        ;++++------------------- <0000固定>
        MOV     PM13,   #11110000B              ;P13入出力設定
                        ;||||++++--------------- PM133/PM132/PM131/PM130:未使用(0)
                        ;++++------------------- <1111固定>
;-------------------------------------------------------------------------------
;       ポート14の設定
;-------------------------------------------------------------------------------
        MOV     P14,    #00000000B              ;P14初期値設定
                        ;||||++++--------------- P143/P142/P141/P140:未使用(0)
                        ;++++------------------- <0000固定>
        MOV     PM14,   #11110000B              ;P14入出力設定
                        ;||||++++--------------- PM143/PM142/PM141/PM140:未使用(0)
                        ;++++------------------- <1111固定>
```

```
;-------------------------------------------------------------------------------
;       ポート15の設定
;-------------------------------------------------------------------------------
⑤       MOV     P15,    #00000000B              ;P15初期値設定
                        ;||||++++--------------- P153/P152/P151/P150:未使用(0)
                        ;++++------------------- <0000固定>
        MOV     PM15,   #11110000B              ;P15入出力設定
                        ;||||++++--------------- PM153/PM152/PM151/PM150:未使用(0)
                        ;++++------------------- <1111固定>

;-------------------------------------------------------------------------------
;       クロック周波数の設定
;-------------------------------------------------------------------------------
;       高速内蔵発振8MHz(TYP.)で動作を行うように設定します。
;-------------------------------------------------------------------------------
        MOV  OSCCTL, #00000000B  ;クロック動作モード
                    ;||||++++--- <0000固定>
                    ;|||+------- OSCSELS:入力ポート・モード
                    ;||+-------- <0固定>
                    ;++--------- EXCLK/OSCSEL:
                    ;              高速システム・クロック端子の動作モード：入力ポート・モード
                    ;              P121/X1,P122/X2/EXCLK：入力ポート

        MOV  MOC,    #10000000B  ;メインOSCコントロール
                    ;|+++++++--- <0000000固定>
                    ;+---------- X1発振回路停止,EXCLK端子からの外部クロック無効

        MOV  MCM,    #00000000B  ;供給クロック選択
⑥                   ;|||||+|+--- XSEL/MCM0:
                    ;|||||  |      メイン・システム・クロック(fXP) = 高速内蔵発振クロック(fRH)
                    ;|||||  |      周辺ハードウエア・クロック(fPRS) = 高速内蔵発振クロック(fRH)
                    ;|||||  +---- MCS: Read Only
                    ;+++++------ <00000固定>

        MOV  PCC,    #00000000B  ; CPUクロック(fCPU)の選択
                    ;|||+|+++--- CSS/PCC2/PCC1/PCC0:
                    ;|||  |          CPUクロック(fCPU)=fXP
                    ;|||  +------ <0固定>
                    ;||+--------- CLS: メイン・システム・クロック
                    ;++---------- <00固定>

        MOV  RCM,    #00000001B  ; CPUクロック(fCPU)の選択
                    ;|||||||+--- LSRSTOP:低速内蔵発信器の停止
                    ;|||||+---- RSTOP:高速内蔵発信器の発振
                    ;|+++++----- <00000固定>
                    ;+---------- RSTS: Read Only

;-------------------------------------------------------------------------------
;       8ビット・タイマH2
;-------------------------------------------------------------------------------
;       100msのインターバル・タイマに設定し，
;       温度測定及びUART送信(1s毎)のインターバル用に使用します。
;-------------------------------------------------------------------------------
        MOV     TMHMD2,#01100000B       ;タイマ･クロック選択レジスタ
⑦                   ;|||||||+------ TOEN2: タイマ出力禁止
                    ;||||||+------- TOLEV2: タイマ出力レベル 未使用
                    ;||||++--------- TMMD21/TMMD20: タイマ動作=インターバル
                    ;|+++----------- CKS22/CKS21/CKS20:
                    ;                 カウント･クロック fPRS/2^12 (fPRS = 8MHz より1953.125Hz)
                    ;+-------------- TMHE2: タイマ動作禁止(タイマ設定完了後許可)

        MOV     CMP02,  #(195-1)        ;100msインターバル：(fPRS/2^12)*0.1[sec]=195.3125

        SET1    TMHE2                   ;タイマ動作開始
        CLR1    TMHIF2                  ;割り込み要求クリア

        MOV     R1SECCNT,#C1SEC         ;TMH0ベース・タイマの1秒カウンタ初期化
```

⑧

```
;-------------------------------------------------------------------------
;       16ビット・タイマ／イベント・カウンタ00
;-------------------------------------------------------------------------
;       温度センサの抵抗値を測るため，コンデンサの放電時間(パルス幅)を測定します。
;-------------------------------------------------------------------------
        MOV     TMC00,  #00000000B  ;16ビット・タイマ・モード・コントロール・レジスタ00
                        ;|||||||+---- OVF00:TM00のオーバフロー・フラグ クリア
                        ;||||||+----- TMC001:タイマ出力（TO00）反転条件はTM00とCR000の一致，
                        ;                     TM00とCR010の一致
                        ;||||++------ TMC003/TMC002:16ビット・タイマ／イベント・カウンタ00動作禁止
                        ;++++-------- <0固定>
        MOV     CRC00,  #00000111B   ; キャプチャ／コンペア・コントロール・レジスタ00
                        ;|||||||+---- CRC000:CR000をキャプチャ・レジスタとして動作
                        ;||||||+----- CRC001:CR000のキャプチャ・トリガはTI000端子の有効エッジの逆相
                        ;|||||+------ CRC002:CR010をキャプチャ・レジスタとして動作
                        ;+++++------- <0固定>
        MOV     TOC00,  #0000000B    ;16ビット・タイマ出力コントロール・レジスタ00
                        ;|||||||+---- TOE00:TO00出力禁止
                        ;||||||+----- TOC001:CR000とTM00の一致によるTO00出力反転動作禁止
                        ;|||||++------ LVS00/LVR00:TO00端子出力の状態変化無し
                        ;||||+-------- TOC004:CR010とTM00の一致によるTO00出力反転動作禁止
                        ;|||+--------- OSPE00:ワンショット・パルス出力動作は連続パルス出力
                        ;||+---------- OSPT00:ソフトウエアによるワンショット・パルス出力トリガなし
                        ;++----------- <0固定>
        MOV     PRM00,  #00000000B   ;プリスケーラ・モード・レジスタ00
                        ;|||||+++---- PRM002/PRM001/PRM000:fPRS=fRHより設定禁止
                        ;|||||+------- <0固定>
                        ;||||++------- ES001/ES000:TI000端子の有効エッジ 立下りエッジ
                        ;++++---------- ES101/ES100:TI010端子の有効エッジ 立下りエッジ
```

⑨

```
;-------------------------------------------------------------------------
;       UART6の設定
;-------------------------------------------------------------------------
;       温度センサでの測定結果を送信します。
;-------------------------------------------------------------------------

        MOV     CKSR6,  #00000000B      ;UART6基本クロック選択
                        ;||||++++------- TPS63-60: 基本クロック(fXCLK6) = fPRS
                        ;++++----------- <0固定>

        ;ボー･レート用クロックの分周値設定
        MOV     BRGC6,  #35             ;ボーレート = 8*10^6[Hz]/(2 * 115200[bps]) = 34.72
                                        ;※誤差を小さくするため，小数点以下を切り上げ
                                        ;ボーレート：115200bps ← 114285bps(ERR:-0.79%)

        MOV     ASIM6,  #01000101B      ;UART6動作モード選択
                        ;|||||||+------- ISRM6: 受信エラー発生時にINTSR6を割り込み
                        ;||||||+-------- SL6: ストップ･ビット数=1
                        ;|||||+--------- CL6: データ長=8
                        ;|||++---------- PS61-60: パリティなし
                        ;||+------------ RXE6: 受信動作禁止
                        ;|+------------- TXE6: 送信動作許可
                        ;+-------------- POWER6: 内部動作クロックの動作禁止

        MOV     ASICL6, #00010110B      ;先頭ビット,TxD6出力反転選択
                        ;|||||||+------- TXDLV6: TxD6通常出力
                        ;||||||+-------- DIR6: 先頭ビットLSB
                        ;|||+++---------- SBL62-60: 未使用
                        ;||+------------ SBTT6: 未使用
                        ;|+------------- SBRT6: Read Only
                        ;+-------------- SBRF6: 未使用

        MOV     ISC,    #00001000B      ;入力切り替え制御
                        ;|||||||+------- ISC0: 未使用
                        ;||||||+-------- ISC1: TI000への入力ソースにP33/TI000端子からの入力信号を選択
                        ;|||||+--------- ISC2: 未使用
                        ;||||+---------- ISC3: RxD6/P113入力許可
                        ;|||++----------- ISC5-4: TxD6=P112,RxD6=P113
                        ;++-------------- <0固定>

        SET1    POWER6          ;内部動作クロックの動作許可
```

⑩
⑪

```
;-------------------------------------------------------------------------
;       割り込みマスクの設定
;-------------------------------------------------------------------------
        MOVW    MK0,#0FFFFH
        MOVW    MK1,#0FFFFH             ;全ての割り込みをマスク

;-------------------------------------------------------------------------
;       割り込み許可
;-------------------------------------------------------------------------
        EI
```

C言語の初期化処理も，アセンブリ言語と同様な動作を行います。

C言語では，hdwinit 関数を作成することにより，初期設定のタイミングを早くすることができます。

hdwinit 関数は，周辺装置（sfr）の初期設定をする関数としてユーザが必要に応じて作成する関数です。

```
/*****************************************************************************

        リセット解除後の初期化処理

*****************************************************************************/
void hdwinit(void)
{
        DI();                                   /* 割り込み禁止 */

/*---------------------------------------------------------------------------
        ROM/RAMサイズの設定
---------------------------------------------------------------------------/
        モデルにより設定値が異なるので注意してください。
        使用モデルの設定を有効にしてください。（デフォルトではuPD78F0485）
---------------------------------------------------------------------------*/
        /* uPD78F0471,uPD78F0481,uPD78F0491使用時の設定 */
        /*IMS = 0x04;                   /* ROMサイズの設定 */
        /*IXS = 0x0C;                   /* 内部拡張RAMサイズの設定 */

        /* uPD78F0472,uPD78F0482,uPD78F0492使用時の設定 */
        /*IMS = 0xC6;                   /* ROMサイズの設定 */
        /*IXS = 0x0C;                   /* 内部拡張RAMサイズの設定 */

        /* uPD78F0473,uPD78F0483,uPD78F0493使用時の設定 */
        /*IMS = 0xC8;                   /* ROMサイズの設定 */
        /*IXS = 0x0C;                   /* 内部拡張RAMサイズの設定 */

        /* uPD78F0474,uPD78F0484,uPD78F0494使用時の設定 */
        /*IMS = 0xCC;                   /* ROMサイズの設定 */
        /*IXS = 0x0A;                   /* 内部拡張RAMサイズの設定 */

        /* uPD78F0475,uPD78F0485,uPD78F0495使用時の設定 */
        IMS =   0xCF;                   /* ROMサイズの設定 */
        IXS =   0x0A;                   /* 内部拡張RAMサイズの設定 */

/*---------------------------------------------------------------------------
        ポートの設定（未使用ポートはLow出力に設定）
---------------------------------------------------------------------------*/
        /* ポート1 */
        P1 =    0b00000000;             /* P1初期値設定 */
                /*++++++++--------------- P17/P16/P15/P14/P13/P12/P11/P10:未使用(0) */
        PM1 =   0b00000000;             /* P1入出力設定 */
                /*++++++++--------------- PM17/PM16/PM15/PM14/PM13/PM12/PM11/PM10:未使用(0) */
        /* ポート2 */
        P2 =    0b00000000;             /* P2初期値設定 */
                /*++++++++--------------- P27/P26/P25/P24/P23/P22/P21/P20:未使用(0) */
        PM2 =   0b00000000;             /* P1入出力設定 */
                /*++++++++--------------- PM27/PM26/PM25/PM24/PM23/PM22/PM21/PM20:未使用(0) */
        /* ポート3 */
        P3 =    0b00000000;             /* P3初期値設定 */
                /*|||+++++--------------- P33/P32/P31/P34/P30:Lo(0) */
                /*+++-------------------- <000固定> */
        PM3 =   0b11100110;             /* P3入出力設定 */
                /*|||+|||+--------------- PM34/PM30:未使用(0) */
                /*|||  |++--------------- PM32/PM31:入力(1) 温度センサ接続ポートとして使用 */
                /*|||  +------------------ PM33:出力(0) コンデンサ充電ポートとして使用 */
                /*|||                                  （パルス幅測定時はTI000として使用) */
                /*+++-------------------- <111固定> */
        /* ポート4 */
        P4 =    0b00000000;             /* P4初期値設定 */
                /*++++++++--------------- P47/P46/P45/P44/P43/P42/P41/P40:未使用(0) */
        PM4 =   0b00000000;             /* P4入出力設定 */
                /*++++++++--------------- PM47/PM46/PM45/PM44/PM43/PM42/PM41/PM40:未使用(0) */
        /* ポート8 */
        P8 =    0b00000000;             /* P8初期値設定 */
                /*||||++++--------------- P83/P82/P81/P80:未使用(0) */
                /*++++-------------------- <0000固定> */
        PM8 =   0b11110000;             /* P8入出力設定 */
                /*||||++++--------------- PM83/PM82/PM81/PM80:未使用(0) */
                /*++++-------------------- <1111固定> */
```

```
        /* ポート9 */
        P9 =    0b00000000;             /* P9初期値設定 */
                /*||||++++--------------- P93/P92/P91/P90:未使用(0) */
                /*++++------------------- <0000固定> */
        PM9 =   0b11110000;             /* P9入出力設定 */
                /*||||++++--------------- PM93/PM92/PM91/PM90:未使用(0) */
                /*++++------------------- <1111固定> */
        /* ポート10 */
        P10 =   0b00000000;             /* P10初期値設定 */
                /*||||++++--------------- P103/P102/P101/P100:未使用(0) */
                /*++++------------------- <0000固定> */
        PM10 =  0b11110000;             /* P10入出力設定 */
                /*||||++++--------------- PM103/PM102/PM101/PM100:未使用(0) */
                /*++++------------------- <1111固定> */
        /* ポート11 */
        P11  =  0b00000100;             /* P11初期値設定 */
                /*||||+|++--------------- P113/P111/P110:未使用(0) */
                /*||||  +---------------- P112:Hi(1)*/
                /*++++------------------- <0000固定> */
        PM11 =  0b11110000;             /* P11入出力設定 */
                /*||||+|++--------------- PM113/PM111/PM110:未使用(0) */
                /*||||  +---------------- PM112:出力(0) TxD6として使用*/
                /*++++------------------- <1111固定> */
        /* ポート12 */
        P12 =   0b00000000;             /* P12初期値設定 */
                /*|||||||+--------------- P120:未使用(0) */
                /*|||++++----------------- P123/P122/P121:Read Only */
                /*+++--------------------- <000固定> */
        PM12 =  0b11111110;             /* P12入出力設定 */
                /*|||||||+--------------- PM120:未使用(0) */
                /*+++++++----------------- <1111111固定> */
        /* ポート13 */
        P13 =   0b00000000;             /* P13初期値設定 */
                /*||||++++--------------- P133/P132/P131/P130:未使用(0) */
                /*++++------------------- <0000固定> */
        PM13 =  0b11110000;             /* P13入出力設定 */
                /*||||++++--------------- PM133/PM132/PM131/PM130:未使用(0) */
                /*++++------------------- <1111固定> */
        /* ポート14 */
        P14 =   0b00000000;             /* P14初期値設定 */
                /*||||++++--------------- P143/P142/P141/P140:未使用(0) */
                /*++++------------------- <0000固定> */
        PM14 =  0b11110000;             /* P14入出力設定 */
                /*||||++++--------------- PM143/PM142/PM141/PM140:未使用(0) */
                /*++++------------------- <1111固定> */
        /* ポート15 */
        P15 =   0b00000000;             /* P15初期値設定 */
                /*||||++++--------------- P153/P152/P151/P150:未使用(0) */
                /*++++------------------- <0000固定> */
        PM15 =  0b11110000;             /* P15入出力設定 */
                /*||||++++--------------- PM153/PM152/PM151/PM150:未使用(0) */
                /*++++------------------- <1111固定> */
```

```
/*-------------------------------------------------------------------------
        クロック周波数の設定
-------------------------------------------------------------------------
        高速内蔵発振8MHz(TYP.)で動作を行うように設定します。
-------------------------------------------------------------------------*/
        OSCCTL =0b00000000;             /* クロック動作モード */
                /*|||+++---------------- <0000固定> */
                /*||+------------------- OSCSELS:入力ポート・モード */
                /*|+-------------------- <0固定> */
                /*++--------------------- EXCLK/OSCSEL: */
                /*                         高速システム・クロック端子の動作モード：入力ポート・モード */
                /*                         P121/X1,P122/X2/EXCLK：入力ポート */

        MOC = 0x80;                     /* X1発振回路停止,EXCLK端子からの外部クロック無効 */

        MCM =   0b00000000;             /* 供給クロック選択 */
                /*|||||+|+---------------- XSEL/MCM0: */
                /*|||||  |                  メイン・システム・クロック(fXP) = 高速内蔵発振クロック(fRH) */
                /*|||||  |                  周辺ハードウエア・クロック(fPRS) = 高速内蔵発振クロック(fRH) */
                /*|||||  +----------------- MCS: Read Only */
                /*+++++------------------- <00000固定> */

        PCC =   0b00000000;             /* CPUクロック(fCPU)の選択 */
                /*|||+|+++---------------- CSS/PCC2/PCC1/PCC0: */
                /*|||  |                          CPUクロック(fCPU)=fXP */
                /*|||  +------------------- <0固定> */
                /*||+---------------------- CLS: メイン・システム・クロック */
                /*++----------------------- <00固定> */

        RCM =   0b00000001;             /* CPUクロック(fCPU)の選択 */
                /*|||||||+---------------- LSRSTOP:低速内蔵発信器の停止 */
                /*||||||+----------------- RSTOP:高速内蔵発信器の発振 */
                /*|+++++------------------ <00000固定> */
                /*+-------------------------- RSTS: Read Only */

/*-------------------------------------------------------------------------
        8ビット・タイマH2
-------------------------------------------------------------------------/
        100msのインターバル・タイマに設定し，
        温度測定及びUART送信(1s毎)のインターバル用に使用します。
-------------------------------------------------------------------------*/
        TMHMD2 = 0b01100000;            /* タイマ･クロック選択レジスタ */
                /* |||||||+------------- TOEN2: タイマ出力禁止 */
                /* ||||||+-------------- TOLEV2: タイマ出力レベル 未使用 */
                /* |||+++---------------- TMMD21/TMMD20: タイマ動作=インターバル */
                /* |+++------------------ CKS22/CKS21/CKS20: カウント･クロック fPRS/2^12 */
                /* |                                          (fPRS = 8MHz より1953.125Hz) */
                /* +--------------------- TMHE2: タイマ動作禁止(タイマ設定完了後許可) */
        CMP02 = 195-1;                  /* 100msインターバル：(fPRS/2^12)*0.1[sec]=195.3125 */

        TMHE2 = 1;                      /* タイマ動作開始 */
        TMHIF2 = 0;                     /* 割り込み要求クリア */
        uc1secCnt = TMH2_1SEC;          /* TMH0ベース・タイマの1秒カウンタ初期化 */
```

```
/*-------------------------------------------------------------------------
        16ビット・タイマ／イベント・カウンタ00
-------------------------------------------------------------------------/
        温度センサの抵抗値を測るため，コンデンサの放電時間(パルス幅)を測定します。
-------------------------------------------------------------------------*/
        TMC00 = 0b00000000;     /* 16ビット・タイマ・モード・コントロール・レジスタ00 */
                /*||||||+--------------- OVF00:TM00のオーバフロー・フラグ クリア */
                /*|||||+---------------- TMC001:タイマ出力(T000)反転条件はTM00とCR000の一致,*/
                /*|||||                                                TM00とCR010の一致*/
                /*||||++----------------- TMC003/TMC002:16ビット･タイマ／イベント･カウンタ00動作禁止 */
                /*++++------------------- <0固定> */
        CRC00 = 0b00000111;     /* キャプチャ／コンペア・コントロール・レジスタ00 */
                /*||||||+--------------- CRC000:CR000をキャプチャ・レジスタとして動作 */
                /*|||||+---------------- CRC001:CR000のキャプチャ・トリガはTI000端子の有効エッジの逆相
*/
                /*||||+----------------- CRC002:CR010をキャプチャ・レジスタとして動作 */
                /*+++++------------------ <0固定> */
        TOC00 = 0b0000000;      /* 16ビット・タイマ出力コントロール・レジスタ00 */
                /*||||||+--------------- TOE00:T000出力禁止 */
                /*|||||+---------------- TOC001:CR000とTM00の一致によるT000出力反転動作禁止 */
                /*|||++----------------- LVS00/LVR00:T000端子出力の状態変化無し */
                /*||+------------------- TOC004:CR010とTM00の一致によるT000出力反転動作禁止 */
                /*|+-------------------- OSPE00:ワンショット・パルス出力動作は連続パルス出力 */
                /*+--------------------- OSPT00:ソフトウエアによるワンショット･パルス出力トリガなし */
        PRM00 = 0b00000000;     /* プリスケーラ・モード・レジスタ00 */
                /*||||+++---------------- PRM002/PRM001/PRM000:fPRS=fRHより設定禁止 */
                /*|||+------------------- <0固定> */
                /*||++-------------------- ES001/ES000:TI000端子の有効エッジ 立下りエッジ */
                /*++---------------------- ES101/ES100:TI010端子の有効エッジ 立下りエッジ */

/*-------------------------------------------------------------------------
        UART6の設定
-------------------------------------------------------------------------/
        温度センサでの測定結果を送信します。
-------------------------------------------------------------------------*/

        CKSR6 = 0b00000000;     /* UART6基本クロック選択 */
                /*||||++++-------- TPS63-60: 基本クロック(fXCLK6) = fPRS */
                /*++++------------ <0固定> */

        /* ボー･レート用クロックの分周値設定 */
        BRGC6 = 35;     /* ボーレート = 8*10^6[Hz]/(2 * 115200[bps]) = 34.72 */
                        /* ※誤差を小さくするため，小数点以下を切り上げ */
                        /* ボーレート：115200bps ← 114285bps(ERR:-0.79%) */

        ASIM6 = 0b01000101;             /* UART6動作モード選択 */
                /*||||||+--------------- ISRM6: 受信エラー発生時にINTSR6を割り込み */
                /*|||||+---------------- SL6: ストップ･ビット数=1 */
                /*||||+----------------- CL6: データ長=8 */
                /*|||++----------------- PS61-60: パリティなし */
                /*||+------------------- RXE6: 受信動作禁止 */
                /*|+-------------------- TXE6: 送信動作許可 */
                /*+--------------------- POWER6: 内部動作クロックの動作禁止 */

        ASICL6 =0b00010110;             /* 先頭ビット,TxD6出力反転選択 */
                /*||||||+--------------- TXDLV6: TxD6通常出力 */
                /*|||||+---------------- DIR6: 先頭ビットLSB */
                /*||+++----------------- SBL62-60: 未使用 */
                /*||+-------------------- SBTT6: 未使用 */
                /*|+--------------------- SBRT6: Read Only */
                /*+---------------------- SBRF6: 未使用 */

        ISC =   0b00001000;             /* 入力切り替え制御 */
                /*||||||+--------------- ISC0: 未使用 */
                /*|||||+---------------- ISC1:TI000への入力ソースにP33/TI000端子からの入力信号を選択 */
                /*||||+----------------- ISC2: 未使用 */
                /*|||+------------------ ISC3: RxD6/P113入力許可 */
                /*||++------------------- ISC5-4: TxD6=P112,RxD6=P113 */
                /*++--------------------- <0固定> */

        POWER6 =        1;              /* 内部動作クロックの動作許可 */

/*-------------------------------------------------------------------------
        割り込みマスクの設定
-------------------------------------------------------------------------*/
        MK0 = 0x0FFFF;
        MK1 = 0x0FFFF;                  /* 全ての割り込みをマスク */

        EI();                           /* 割り込み許可 */
}
```

5.2　メイン処理

アセンブリ言語のメイン処理では，次の動作を行います。

① 8ビット・タイマH2をベース・タイマとして，約1秒の温度測定のタイミングをカウントします。約1秒経過した場合は，②～⑤を行います。

② キャリブレーション用固定抵抗使用し，コンデンサの放電時間を測定します。パルス幅測定モードにキャリブレーション用固定抵抗使用を設定し，パルス幅測定処理を呼び出します。

③ サーミスタを使用し，コンデンサの放電時間を測定します。パルス幅測定モードにサーミスタ使用を設定し，パルス幅測定処理を呼び出します。

④ 温度取得処理を呼び出します。

⑤ UART送信処理を呼び出します。

⑥ ①へ分岐します。

```
;*****************************************************************************
;
;
;       メイン処理
;
;
;*****************************************************************************
MAIN_LOOP:
        ;*======================================*
        ;                                       ;
        ;           測定温度送信処理            ;
        ;                                       ;
        ;*======================================*
        ;----------------------------------------;
        ;           タイミング作成処理          ;
        ;----------------------------------------;
LMAIN100:
        BF      TMHIF2,$LMAIN500        ;100ms経過した? →NO
①       CLR1    TMHIF2                  ;割り込み要求クリア
        DEC     R1SECCNT                ;1秒カウンタ更新
        BNZ     $LMAIN500               ;1秒経過した? →NO
        MOV     R1SECCNT,#C1SEC         ;1秒カウンタ初期化

        ;----------------------------------------;
        ;             温度測定処理               ;
        ;----------------------------------------;
        MOV     B,#0             ;引数(パルス幅測定モード)設定
②       CALL    !SGETPULSE       ;キャリブレーション抵抗の放電パルス幅測定
        MOVW    RCALBCNT,AX      ;測定したパルス幅を取得

        MOV     B,#1             ;引数(パルス幅測定モード)設定
③       CALL    !SGETPULSE       ;サーミスタの放電パルス幅測定
        MOVW    RTHERMCNT,AX     ;測定したパルス幅を取得

LMAIN400:
④       CALL    !SGETHEAT        ;測定したパルス幅から抵抗を算出し，温度を取得

        ;----------------------------------------;
        ;     UART6送信データ作成＆データ送信    ;
        ;----------------------------------------;
⑤       CALL    !SUART6TX

LMAIN500:
        ;*======================================*
        ;                                       ;
        ;             各種メイン処理            ;
        ;                                       ;
        ;*======================================*

        ; 各種メイン処理があればここで行います。

⑥       BR      MAIN_LOOP
```

C言語のメイン処理も，アセンブリ言語と同様な動作を行います。

```
/************************************************************************************

        メイン・ループ

************************************************************************************/
void main(void)
{
        while(1)
        {
                /*****************************************/
                /*                                       */
                /*            測定温度送信処理           */
                /*                                       */
                /*****************************************/
                /*---------------------------------------*/
                /*           タイミング作成処理          */
                /*---------------------------------------*/
                if(TMHIF2)
                {/* 100ms経過  */
                        TMHIF2 = 0;             /* 割り込み要求クリア */
                        uc1secCnt--;            /* 1秒カウンタ更新 */
                }

                /*---------------------------------------*/
                /*             温度測定処理              */
                /*---------------------------------------*/
                if(uc1secCnt == 0)
                {/* 1秒経過 */
                        uc1secCnt = TMH2_1SEC;  /* 1秒カウンタクリア */

                        /* キャリブレーション抵抗の放電パルス幅測定 */
                        ushCalibrationCnt = fn_GetPulseTime(0);

                        /* サーミスタの放電パルス幅測定 */
                        ushThermistorCnt = fn_GetPulseTime(1);

                        /* 測定したパルス幅から抵抗を算出し，温度を取得 */
                        ushHeatData = fn_GetHeatData();

                        /* UART6送信データ作成＆データ送信 */
                        fn_UART6_Tx();
                }
                /*****************************************/
                /*                                       */
                /*             各種メイン処理            */
                /*                                       */
                /*****************************************/

                /* 各種メイン処理があればここで行います。 */
        }
}
```

5. 3　パルス幅測定処理

アセンブリ言語のパルス幅測定処理では，次の動作を行います。

① Bレジスタを③の処理で使用するため，パルス幅測定モードをAレジスタに保存します。
② オーバフロー回数カウンタを初期化します。
③ コンデンサを充電します。P33をハイ・レベル出力に設定し，2 msウエイトします。
④ P33をTI000端子として使用するため，入力ポートに設定します。
⑤ TI000端子の有効エッジ検出割り込みの割り込み要求をクリアします。
⑥ コンデンサの放電を開始します。パルス幅測定モードで指定された放電用ポート（P31またはP32）[注]をロウ・レベル出力に設定するとコンデンサの放電が開始します。
　(a) コンデンサの放電を開始させると同時に，16ビット・タイマ／イベント・カウンタ00の動作許可をTI000端子の有効エッジ入力でクリア&スタート・モードに設定し，放電時間の測定を開始させます。
⑦ コンデンサが放電し，TI000がロウ・レベルとなるのを待ちます。TI000がロウ・レベルになると，TI000端子の有効エッジ検出割り込み（INTTM010）が発生し，16ビット・タイマ・キャプチャ／コンペア・レジスタ010に16ビット・タイマ・カウンタ00の値がキャプチャされます。
　(a) 放電時間を測定中に16ビット・タイマ・カウンタ00がオーバフローした場合は，オーバフロー回数カウンタを更新します。キャリブレーション用固定抵抗を使用して放電時間を測定する場合，キャリブレーション用固定抵抗とコンデンサは，16ビット・タイマ・カウンタ00がオーバフローしないように選定しています。また，サーミスタを使用して放電時間を測定する場合にオーバフローが2回以上発生すると，サーミスタの抵抗値の計算結果が測定範囲外になります。そのため，オーバフローが2回以上発生した場合は，測定エラーと判断し，コンデンサの放電時間の測定を中断します。
⑧ 16ビット・タイマ・キャプチャ／コンペア・レジスタ010からコンデンサの放電時間を取得します。
⑨ 使用した放電用ポート（P31またはP32）[注]を入力ポートに設定します。
　(a) キャリブレーション用固定抵抗とコンデンサは，キャリブレーション用固定抵抗を使用して放電時間を測定する場合，16ビット・タイマ・カウンタ00がオーバフローしないように選定しています。そのため，キャリブレーション用放電パルス幅の測定でオーバフローが発生すると，測定エラーとして放電時間に0を設定します。
⑩ 16ビット・タイマ／イベント・カウンタ00の動作許可を16ビット・タイマ／イベント・カウンタ00動作禁止に設定します。
⑪ P33を入力ポートに設定します。

注 コンデンサの放電にキャリブレーション用固定抵抗を使用する場合はP31，サーミスタを使用する場合はP32となります。

```
;*******************************************************************************
;
;       コンデンサの放電時間測定（TI000パルス幅測定）
;
;-------------------------------------------------------------------------------
;       [  IN  ] B:パルス幅測定モード(0:キャリブレーション抵抗の放電パルス幅を測定
;                                     1:サーミスタの放電パルス幅を測定            )
;       [  OUT ] AX:測定した放電パルス幅
;                ROVFCNT:TM00オーバフロー回数
;
;       TI000のパルス幅測定機能を利用し，コンデンサの放電時間を測定します。
;       キャリブレーション抵抗かサーミスタのどちらの放電パルス幅を測定するか
;       引数で指定します。
;       戻り値として測定した放電パルス幅を返します。
;       パルス幅測定中にTM00がオーバフローした場合は，
;       その回数をオーバフロー回数カウンタに設定します。
;*******************************************************************************
SGETPULSE:
①       MOV     A,B                     ;パルス幅測定モードを取得
②       MOV     ROVFCNT,#0              ;オーバフロー回数カウンタクリア

        ;============    コンデンサ充電    ============
        SET1    P3.3
        CLR1    PM3.3                   ;コンデンサ充電開始
        ;充電完了するまで2msecウエイト
③       MOV     B,#93           ;[4clk]                 ↑
JGETP100:                       ;                       |
        MOV     C,#27           ;[4clk]↑                |4 + (166 + 6)*93 = 16000clk
JGETP101:                       ;       |4+6*27 = 166clk |16000 * 0.125[μs] = 2000[μs]
        DBNZ    C,$JGETP101     ;[6clk]↓                |
        DBNZ    B,$JGETP100     ;[6clk]                 ↓

④       SET1    PM3.3                   ;P33をTI000として使用
⑤       CLR1    TMIF010                 ;割り込み要求クリア

        ;============   コンデンサ放電開始   ============
        MOV     B,A                     ;パルス幅測定モードを保存
        CMP     A,#0                    ;キャリブレーション抵抗の放電パルス幅を測定する?
        BNZ     $JGETP200               ; →NO:サーミスタの放電パルス幅を測定する

⑥      CLR1    P3.1                    ;放電準備;P31をLo出力に設定すると放電が開始する
  (a    MOV     TMC00,#08H              ;16ビット・タイマ／イベント・カウンタ00でパルス幅測定開始
        CLR1    PM3.1                   ;放電開始
        BR      JGETP300
JGETP200:
        CLR1    P3.2                    ;放電準備;P32をLo出力に設定すると放電が開始する
  (a    MOV     TMC00,#08H              ;16ビット・タイマ／イベント・カウンタ00でパルス幅測定開始
        CLR1    PM3.2                   ;放電開始

JGETP300:
        ;==========  コンデンサ放電完了待ち  ==========
        BT      TMIF010,$JGETP500       ;コンデンサの放電完了? →YES

        BF      OVF00,$JGETP400         ;TM00のオーバーフロを検出した? →NO
        CLR1    OVF00                   ;TM00オーバフロー・フラグをクリア
⑦  (a   INC     ROVFCNT                 ;オーバフロー回数を更新
        CMP     ROVFCNT,#2              ;オーバフローが2回以上発生した?
        BZ      $JGETP500               ; →YES:温度測定エラー，パルス幅の測定を中断する
JGETP400:
        BR      JGETP300                ;引き続きコンデンサの放電完了を待つ

JGETP500:
        CLR1    TMIF010                 ;割り込み要求クリア
```

```
        ;==========    コンデンサ放電完了    ==========
⑧       MOVW    AX,CR010                ;測定したパルス幅を取得
        DEC     B
        BZ      $JGETP700               ;キャリブレーション抵抗の放電パルス幅を測定した?
                                        ; →NO:サーミスタの放電パルス幅を測定した
⑨       SET1    PM3.1                   ;キャリブレーション抵抗での放電用のポートを入力に戻す
(a      CMP     ROVFCNT,#0              ;パルス幅測定でオーバフローがあった?
        BZ      $JGETP800               ; →NO:測定したパルス幅を戻り値として返す
        MOVW    AX,#0                   ;戻り値を値をエラーにする
        BR      JGETP800
JGETP700:
        SET1    PM3.2                   ;サーミスタでの放電用のポートを入力に戻す

JGETP800:
⑩       MOV     TMC00,#0                ;16ビット・タイマ／イベント・カウンタ00動作停止
        CLR1    P3.3
⑪       CLR1    PM3.3                   ;TI000をL出力に戻す

        RET
```

C言語の処理も，アセンブリ言語と同様な動作を行います

```
/***************************************************************************
        コンデンサの放電時間測定（TI000パルス幅測定）

---------------------------------------------------------------------------
        [  IN  ] mode(0:キャリブレーション抵抗の放電パルス幅を測定
                      1:サーミスタの放電パルス幅を測定            )
        [  OUT ] 測定した放電パルス幅

        TI000のパルス幅測定機能を利用し，コンデンサの放電時間を測定します。
        キャリブレーション抵抗かサーミスタのどちらの放電パルス幅を測定するか
        引数で指定します。
        戻り値として測定した放電パルス幅を返します。
        パルス幅測定中にTM00がオーバフローした場合は，
        その回数をオーバフロー回数カウンタに設定します。
***************************************************************************/
static short fn_GetPulseTime(unsigned char mode)
{
        unsigned short ushRet;          /* 戻り値保存用 */
        unsigned short temp;            /* ワーク領域 */

        ucOVFcnt = 0;           /* オーバフロー回数カウンタクリア */

        /* コンデンサ充電 */
        P3.3 = 1;
        PM3.3 = 0;              /* コンデンサ充電開始 */
        for(temp = 224; temp > 0; temp--)
        {
                NOP();          /* 充電完了するまで2msec程度ウエイト */
        }
        PM3.3 = 1;              /* P33をTI000として使用 */
        TMIF010 = 0;            /* 割り込み要求クリア */

        /* コンデンサ放電開始 */
        if(mode == 0)
        {/* キャリブレーション抵抗の放電パルス幅を測定する */
                P3.1 = 0;               /* 放電準備 *//* P31をLo出力に設定すると放電が開始する */
                TMC00 = 0x08;           /* パルス幅測定開始 */
                PM3.1 = 0;              /* 放電開始 */
        }
        else
        {/* サーミスタの放電パルス幅を測定する */
                P3.2 = 0;               /* 放電準備 *//* P32をLo出力に設定すると放電が開始する */
                TMC00 = 0x08;           /* パルス幅測定開始 */
                PM3.2 = 0;              /* 放電開始 */
        }

        /* コンデンサ放電完了待ち */
        while(!TMIF010)
        {
                if(OVF00)
                {/* TM00のオーバーフロを検出ーした場合 */
                        OVF00 = 0;              /* TM00オーバフロー・フラグをクリア */
                        ucOVFcnt++;             /* オーバフロー回数を更新 */
                        if(ucOVFcnt >= 2)       /* オーバフロー回数が2回以上の場合は */
                                break;          /* 温度測定エラー，パルス幅の測定を中断する。 */
                }
        }
        TMIF010 = 0;            /* 割り込み要求クリア */
        ushRet = CR010;         /* 測定したパルス幅を取得 */
```

```
        /* コンデンサ放電完了 */
        if(mode == 0){          /* 放電用のポートを入力に戻す */
                PM3.1 = 1;      /* キャリブレーション抵抗を使用した場合 */
                if(ucOVFcnt > 0)
                        ushRet = 0;
        }
        else
        {
                PM3.2 = 1;      /* サーミスタを使用した場合 */
        }

        TMC00 = 0x00;           /* 16ビット・タイマ／イベント・カウンタ00動作停止 */
        P3.3 = 0;
        PM3.3 = 0;              /* TI000をL出力にもどす */

        return ushRet;          /* パルス幅を返す */
}
```

5. 4　温度取得処理

アセンブリ言語の温度取得処理では，次の動作を行います。

① コンデンサの放電時間の測定で，測定エラーになってないかを確認します。キャリブレーション用固定抵抗を使用した放電時間がエラーの場合，サーミスタを使用した放電パルス幅の測定時にオーバフローが2回以上発生した場合は，測定エラーとなります。測定エラーでない場合は②～④の処理を，測定エラーの場合は⑤を行います。

② サーミスタの抵抗値を算出します。［式1］を［式2］のように展開し，（a），（b），（c）の順で演算をします。

$$R_{TH} = \frac{R_C \times (CNT_{TH} + \text{オーバフロー回数} \times 10000H)}{CNT_C} \quad \cdots\cdots [式1]$$

$$R_{TH} = \{(R_C \times CNT_{TH}) + (R_C \times \text{オーバフロー回数} \times 10000H)\} \div CNT_C \quad \cdots\cdots [式2]$$

(a)　(b)　(c)

R_{TH}　：サーミスタの抵抗値［100 Ω］
R_C　：キャリブレーション用固定抵抗の抵抗値［100 Ω］
CNT_{TH}　：サーミスタを使用したコンデンサの放電時間
CNT_C　：キャリブレーション用固定抵抗を使用したコンデンサの放電時間

③ ②で算出したサーミスタの抵抗値が，温度の測定範囲（42.0～32.0 ℃）に対する抵抗値の測定範囲（24.5～37.0 kΩ）に含まれているかを判定します。測定範囲外であった場合，⑤を行います。

④ 温度変換テーブル[注]から，サーミスタの抵抗値に対する温度を取得します。サーミスタの抵抗値について測定範囲の最小抵抗値（24.5 kΩ）からのオフセット（100 Ω単位）を求め，温度変換テーブルの先頭アドレスからのオフセットに変換し，温度（BCD）を取得します。

⑤ 温度測定結果にエラー（FFFFH）を設定します。

⑥ ②で使用する掛け算の関数です。

⑦ ②で使用する割り算の関数です。

注　温度変換テーブルの詳細については，2. 2　**抵抗値の温度変換**を参照してください。

```
;*******************************************************************************
;
;       温度取得処理
;
;-------------------------------------------------------------------------------
;       [  IN  ] RCALBCNT:キャリブレーション抵抗の放電パルス幅
;                RTHERMCNT:サーミスタの放電パルス幅
;                ROVFCNT:TM00オーバフロー回数(サーミスタの放電パルス幅測定時)
;       [  OUT ] RHEAT:温度(BCD)
;
;       測定したパルス幅から抵抗を算出し，
;       抵抗を温度に変換するテーブルから温度を取得します。
;
;       ◎パルス幅から下記の式にて抵抗値を算出します。
;
;                      Rc × (CNTth + オーバフロー回数 × 10000H)
;               Rth = ------------------------------------------
;                                    CNTc
;
;               Rth  :サーミスタの抵抗値[100Ω]
;               Rc   :キャリブレーション抵抗の抵抗値 = 330 [100Ω]
;               CNTth:サーミスタの放電パルス幅
;               CNTc :キャリブレーション抵抗の放電パルス幅
;
;
;       ◎Rthの測定範囲に対する相対値を下記の式で求め，
;         その値をオフセットとして抵抗を温度に変換するテーブルから取得します。
;
;               Rrel = Rth - Rmin
;
;               Rrel:Rthの測定範囲に対する相対値[100Ω]
;               Rmin:測定範囲の最小抵抗値 = 245 [100Ω]
;*******************************************************************************
SGETHEAT:
        CMP     RCALBCNT,#0             ;キャリブレーション抵抗の放電パルス幅測定でエラーが起きた?
        BZ      $JGETH800               ; →YES:抵抗値算出不可。抵抗値算出を行わない

        CMP     ROVFCNT,#2              ;パルス幅の測定でオーバフローが2回以上起きた?
        BZ      $JGETH800               ; →YES:既に抵抗値が測定範囲外。抵抗値算出を行わない

        ;---------------------------------;
        ;   パルス幅から抵抗値を算出する   ;
        ;---------------------------------;
        MOVW    RTEMP32,#0              ;計算用変数下位16bitに0を保存
        MOVW    AX,RTHERMCNT
        MOVW    (RTEMP32+2),AX          ;計算用変数上位16bitにCNTthを保存
        MOVW    AX,#330
        MOVW    RTEMP16A,AX             ;計算用変数にRc(330)[100Ω]を保存
        CALL    !SMULT16                ;(Rc × CNTth) を計算

        ;(Rc × CNTth) の計算結果に (Rc × オーバフロー回数 × 10000H)を加える
        CMP     ROVFCNT,#0              ;オーバフローが発生した?
        BZ      $JGETH300               ; →NO:10000Hの加算をせず抵抗値算出を続行
        MOV     A,ROVFCNT
        MOV     B,A                     ;オーバフロー回数をカウンタに設定
        MOVW    AX,(RTEMP32+2)          ;(Rc × CNTth)の結果の上位16bitに対してRcを足しこむ
JGETH200:
        ADDW    AX,#330                 ;Rcを加算
        BC      $JGETH800               ;加算結果がオーバフロー? →YES:温度は測定範囲外
        DBNZ    B,$JGETH200             ;オーバフロー回数分Rcを足しこんだ? →NO
        MOVW    (RTEMP32+2),AX

JGETH300:
        MOVW    AX,RCALBCNT
        MOVW    RTEMP16A,AX           ;キャリブレーション抵抗の放電パルス幅を除数に設定
        CALL    !SDIV32               ;(Rc × (CNTth + オーバフロー回数 × 10000H))/CNTc の割り算を行う
```

① (a)

② (b) (c)

```
        ;-------------------------------------------------------------;
        ;   サーミスタの抵抗値が測定範囲(24.5kΩ～37.0kΩ)内であるか判定   ;
        ;-------------------------------------------------------------;
        MOVW    AX,(RTEMP32+2)      ;算出した抵抗値の上位16bitを取得
        CMPW    AX,#0000H           ;(370 = 172Hより)上位16bitを0と比較
        BNZ     $JGETH800           ;上位16bitが1以上の場合は，測定範囲外なので温度の値をエラーにする
JGETH400:
③      MOVW    AX,RTEMP32          ;算出した抵抗値の下位16bitを取得
        CMPW    AX,#371             ;算出した抵抗値は37.0kΩ以下?
        BNC     $JGETH800           ; →NO:温度の値をエラーにする
        CMPW    AX,#245             ;算出した抵抗値は24.5kΩ以上?
        BC      $JGETH800           ; →NO:温度の値をエラーにする

        ;-----------------------;
        ;   抵抗値を温度に変換   ;
        ;-----------------------;
JGETH500:       ;Rthの測定範囲に対する相対値を算出
        MOVW    AX,RTEMP32
        SUBW    AX,#245             ;Rrel = Rth - Rminを計算
        MOV     A,X                 ;下位8bit取得(測定範囲内であればRrelは8bitに収まる)
        ADD     A,A                 ;Rrelを2倍にし，
        MOV     B,A                 ;抵抗を温度に変換するテーブルのオフセットを取得
④      MOVW    HL,#TR2HEAT         ;HLに抵抗を温度に変換するテーブルのアドレスを設定
        MOV     A,[HL+B]            ;温度(下位8bit)を取得
        MOV     X,A
        INC     B
        MOV     A,[HL+B]            ;温度(上位8bit)を取得
        MOVW    RHEAT,AX            ;温度の値を変数に保存
        BR      JGETH900

        ;-----------------------------------------------;
        ;   温度測定でエラーが発生した場合の温度設定   ;
        ;-----------------------------------------------;
JGETH800:
⑤      MOVW    RHEAT,#0FFFFH       ;温度の値をエラーにする

JGETH900:
        RET
```

⑥

```
;*******************************************************************************
;
;       掛け算を行う関数 (16bit * 16bit)
;
;-------------------------------------------------------------------------------
;       [  IN  ] RTEMP16A:掛け算を行う値
;                RTEMP32:上位16bitに掛け算を行う数を,下位16bitは0を保存
;       [  OUT ] RTEMP32:演算結果
;*******************************************************************************
$MULT16:
        MOV     B,#16                   ;ビットカウンタを設定
JMLT120:
        CLR1    CY
        MOV     A,RTEMP32
        ROLC    A,1
        MOV     RTEMP32,A
        MOV     A,(RTEMP32+1)
        ROLC    A,1
        MOV     (RTEMP32+1),A
        MOV     A,(RTEMP32+2)
        ROLC    A,1
        MOV     (RTEMP32+2),A
        MOV     A,(RTEMP32+3)
        ROLC    A,1
        MOV     (RTEMP32+3),A           ;演算結果(被乗数を含む)を1bit左シフト
        BNC     $JMLT220                ;MSB = 1 ? →NO

        MOV     A,RTEMP16A
        ADD     A,RTEMP32
        MOV     RTEMP32,A
        MOV     A,(RTEMP16A+1)
        ADDC    A,(RTEMP32+1)
        MOV     (RTEMP32+1),A
        MOV     A,#0
        ADDC    A,RTEMP16A
        MOV     RTEMP16A,A              ;被乗数を加える
JMLT220:
        DBNZ    B,$JMLT120              ;16bit分処理が完了した? →NO,

        RET
```

⑦

```
;*******************************************************************************
;
;       割り算を行う関数 (32bit / 16bit)
;
;-------------------------------------------------------------------------------
;       [  IN  ] RTEMP16A:除数
;                RTEMP32:被除数
;       [  OUT ] RTEMP32:演算結果
;                RTEMP16B:余り
;*******************************************************************************
SDIV32:
        MOVW    RTEMP16B,#0             ;計算用変数を初期化

        MOV     B,#32                   ;ビットカウンタを設定
JDIV120:
        CLR1    CY
        MOV     A,RTEMP16B
        ROLC    A,1
        MOV     RTEMP16B,A
        MOV     A,(RTEMP16B+1)
        ROLC    A,1
        MOV     (RTEMP16B+1),A
        MOV     A,RTEMP32
        ROLC    A,1
        MOV     RTEMP32,A
        MOV     A,(RTEMP32+1)
        ROLC    A,1
        MOV     (RTEMP32+1),A
        MOV     A,(RTEMP32+2)
        ROLC    A,1
        MOV     (RTEMP32+2),A
        MOV     A,(RTEMP32+3)
        ROLC    A,1
        MOV     (RTEMP32+3),A           ;被除数を1bit左シフト
        MOV     A,#0
        ADDC    A,RTEMP16B
        MOV     RTEMP16B,A              ; MSB -> LSB

        SUB     A,RTEMP16A
        MOV     RTEMP16B,A
        MOV     A,(RTEMP16B+1)
        SUBC    A,(RTEMP16A+1)
        MOV     (RTEMP16B+1),A          ;RTEMP16B - RTEMP16A
        BT      RTEMP32.0,$JDIV220      ;桁借りできる? →YES
        BC      $JDIV180                ;RTEMP16B < RTEMP16A ? →YES

        SET1    RTEMP32.0               ;商をセット
        BR      JDIV220

JDIV180:
        MOV     A,RTEMP16B
        ADD     A,RTEMP16A
        MOV     RTEMP16B,A
        MOV     A,(RTEMP16B+1)
        ADDC    A,(RTEMP16A+1)
        MOV     (RTEMP16B+1),A
JDIV220:
        DBNZ    B,$JDIV120              ;32bit分処理が完了した? →NO

        RET
```

C言語の処理も，アセンブリ言語と同様な動作を行います。

```
/*************************************************************************

        温度取得処理

-------------------------------------------------------------------------
        [  IN  ] なし
        [  OUT ] 温度(BCD)

        測定したパルス幅から抵抗を算出し，
        抵抗を温度に変換するテーブルから温度を取得します。

        ◎パルス幅から下記の式にて抵抗値を算出します。
                (抵抗値とパルス幅は比例するより)
                Rc : CNTc = Rth : CNTth

                          Rc × (CNTth + オーバフロー回数 × 0x10000)
                → Rth = ------------------------------------------
                                        CNTc

                Rth  :サーミスタの抵抗値[100Ω]
                Rc   :キャリブレーション抵抗の抵抗値 = 330 [100Ω]
                CNTth:サーミスタの放電パルス幅
                CNTc :キャリブレーション抵抗の放電パルス幅


        ◎Rthの測定範囲に対する相対値を下記の式で求め，
          その値をオフセットとして抵抗を温度に変換するテーブルから取得します。

                Rrel = Rth - Rmin

                Rrel:Rthの測定範囲に対する相対値[100Ω]
                Rmin:測定範囲の最小抵抗値 = 245 [100Ω]
*************************************************************************/
static short fn_GetHeatData(void)
{
        unsigned short ushRet;                  /* 戻り値保存用 */
        unsigned long int ulTemp1;              /* 計算用RAM */
        unsigned char ucTemp2;                  /* 計算用RAM */

        if((ushCalibrationCnt != 0) && (ucOVFcnt < 2))
        {/* キャリブレーション抵抗の放電パルス幅が正常に測定でき，*/
         /*かつ，サーミスタ抵抗の放電パルス幅測定でオーバフローの回数が2回以上でなければ，*/
         /* パルス幅から抵抗値を算出する */
                /* オーバフロー分を合わせ，測定したサーミスタのパルス幅を32bitに拡張 */
                ulTemp1 = (unsigned long)(ucOVFcnt * 0x10000) + ushThermistorCnt;
                /* サーミスタの抵抗値を計算 */
                ushRet = (unsigned short)((ulTemp1 * 330) / ushCalibrationCnt);

                /* サーミスタの抵抗値が測定範囲(24.5kΩ～37.0kΩ)内であるか判定 */
                if((ushRet <= 370)&&(ushRet >= 245))
                {/* 抵抗値が測定範囲に含まれている → 抵抗値から温度を取得 */
                        ucTemp2 = (unsigned char)(ushRet - 245);
                        ushRet = tR2Heat[ucTemp2];
                }
                else
                {/* 抵抗値が測定範囲外 → 温度の値をエラーにする */
                        ushRet = 0xffff;
                }
        }
        else
        {/* サーミスタ放電パルス幅測定で，2回以上オーバフローが起こると */
        /* 既に抵抗値が測定範囲外となっている */
                ushRet = 0xffff;                /* 温度の値をエラーにする */
        }

        return ushRet;          /* 温度を返す */
}
```

5. 5　UART送信処理

アセンブリ言語のUART送信処理では，次の動作を行います。

① 温度測定結果をASCIIコードに変換します。[注]

② シリアル・インタフェースUART6にてASCIIコードに変換した温度測定結果を送信します。

注 UART通信の設定内容詳細，および送信データの詳細については，4. 4　UART送信データのフォーマットを参照してください。

```
;*******************************************************************************
;
;       UART6送信データ作成＆データ送信
;
;-------------------------------------------------------------------------------
;       [  IN  ] RHEAT:温度(BCD)
;       [  OUT ] なし
;
;       測定した温度を，ASCIIコードに変換して送信バッファに設定し，
;       送信します。
;
;       ＜ 送信データの例 ＞
;               ◎測定結果が38.5℃場合
;                 0   1   2   3   4   5
;               ┌───┬───┬───┬───┬───┬───┐
;               │ 3 │ 8 │ . │ 5 │\r │\n │
;               └───┴───┴───┴───┴───┴───┘
;
;               ◎測定でエラーが発生した場合
;                 0   1   2   3   4   5
;               ┌───┬───┬───┬───┬───┬───┐
;               │ * │ * │ . │ * │\r │\n │
;               └───┴───┴───┴───┴───┴───┘
;
;*******************************************************************************
SUART6TX:
        ;-----------------------------------------
        ;   送信バッファにUART6送信データ作成処理
        ;-----------------------------------------
        MOVW    AX,RHEAT
        CMPW    AX,#0FFFFH              ;温度測定が正常に終了した?
        BZ      $JU6TX100               ; →NO

        ;温度を送信バッファに設定
        AND     A,#0FH                  ;10の位の桁を取得
        ADD     A,#'0'                  ;ASCIIコードに変換
        MOV     RTXBUF,A        ;[0]温度10の位保存

        MOV     A,X                     ;1の位の桁，及び少数第一位を取得
①       ROR     A,1                     ;上位4bitを下位4bitに落とす
        ROR     A,1
        ROR     A,1
        ROR     A,1
        AND     A,#0FH                  ;下位4ビット1の位の桁を取得
        ADD     A,#'0'                  ;ASCIIコードに変換
        MOV     (RTXBUF+1),A    ;[1]温度1の位保存

        MOV     (RTXBUF+2),#'.' ;[2]小数点保存
```

```
        MOV     (RTXBUF+2),#'.' ;[2]小数点保存

        MOV     A,X                     ;1の位の桁，及び少数第一位を取得
        AND     A,#0FH                  ;少数第一位を取得
        ADD     A,#'0'                  ;ASCIIコードに変換
        MOV     (RTXBUF+3),A    ;[3]温度小数第一位保存

        BR      JU6TX200
①
JU6TX100:
        ;「＊＊.＊」を送信バッファに設定
        MOV     RTXBUF,#'*'     ;[0]アスタリスク保存
        MOV     (RTXBUF+1),#'*' ;[1]アスタリスク保存
        MOV     (RTXBUF+2),#'.' ;[2]小数点保存
        MOV     (RTXBUF+3),#'*' ;[3]アスタリスク保存

JU6TX200:
        MOV     (RTXBUF+4),#0DH ;[4]キャリッジリターン保存
        MOV     (RTXBUF+5),#0AH ;[5]ラインフィード保存

        ;----------------------------------------
        ;           UART6データ送信
        ;----------------------------------------
JU6TX500:
        ;=============    送信動作開始    =============
        MOV     B,#6                    ;送信カウンタ設定
        MOVW    HL,#RTXBUF

JU6TX600:
        CLR1    STIF6                   ;割り込み要求クリア
        MOV     A,[HL]                  ;送信バッファから送信データを取得
②       MOV     TXB6,A                  ;データ送信
JU6TX700:
        BF      STIF6,$JU6TX700         ;UART6 1Byte送信完了? →NO
        CLR1    STIF6                   ;割り込み要求クリア
        INCW    HL                      ;送信バッファの送信データ位置更新
        DBNZ    B,$JU6TX600             ;未送信データあり? →YES :次のデータを送信
JU6TX800:               ;送信終了
        RET
```

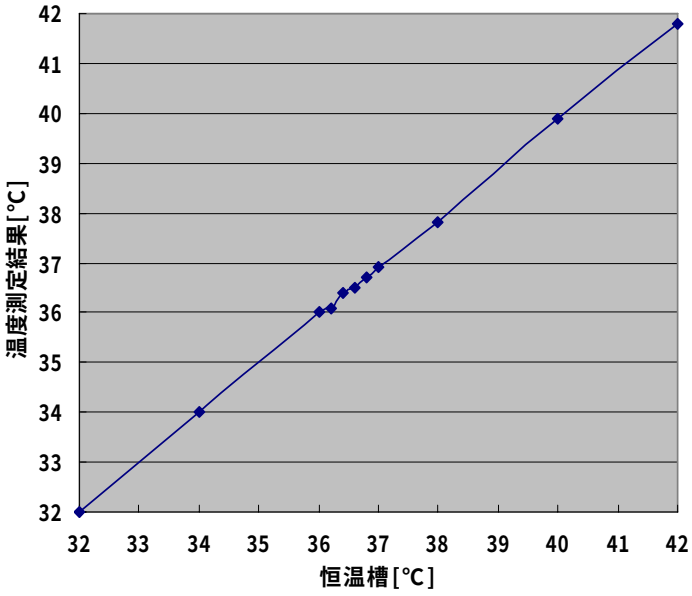
C言語の処理も，アセンブリ言語と同様な動作を行います。

```
/************************************************************************************

        送信データ作成＆UART6データ送信

------------------------------------------------------------------------------------
        [  IN  ] なし
        [  OUT ] なし

        測定した温度を，ASCIIコードに変換して送信バッファに設定し，
        送信します。

        ＜ 送信データの例 ＞
                ◎測定結果が38.5℃場合
                  0   1   2   3   4   5
                ┌───┬───┬───┬───┬───┬───┐
                │ 3 │ 8 │ . │ 5 │\r │\n │
                └───┴───┴───┴───┴───┴───┘

                ◎測定でエラーが発生した場合
                  0   1   2   3   4   5
                ┌───┬───┬───┬───┬───┬───┐
                │ * │ * │ . │ * │\r │\n │
                └───┴───┴───┴───┴───┴───┘

************************************************************************************/
static void fn_UART6_Tx(void)
{
        /*****************************************/
        /*                                       */
        /*          UART6送信データ作成          */
        /*                                       */
        /*****************************************/
        if(ushHeatData != 0xFFFF)
        {/* 温度測定が正常に終了 → 温度を送信バッファに設定 */
                 /* [0]温度10の位(ASCIIコードに変換) */
                ucTxBuffer[0] = (unsigned char)(((ushHeatData >> 8) & 0x000f) + '0');
                 /* [1]温度1の位(ASCIIコードに変換) */
                ucTxBuffer[1] = (unsigned char)(((ushHeatData >> 4) & 0x000f) + '0');
                 /* [2]小数点 */
                ucTxBuffer[2] = '.';
                 /* [3]温度小数第一位(ASCIIコードに変換) */
                ucTxBuffer[3] = (unsigned char)((ushHeatData & 0x000f) + '0');
        }
        else
        {/* 測定エラーが発生 → 「＊＊.＊」を送信バッファに設定 */
                ucTxBuffer[0] = '*';    /* [0]アスタリスク保存 */
                ucTxBuffer[1] = '*';    /* [1]アスタリスク保存 */
                ucTxBuffer[2] = '.';    /* [2]小数点保存 */
                ucTxBuffer[3] = '*';    /* [3]アスタリスク保存 */
        }
        ucTxBuffer[4] = '\r';           /* [4]キャリッジリターン */
        ucTxBuffer[5] = '\n';           /* [5]ラインフィード */

        /*****************************************/
        /*                                       */
        /*          UART6データ送信              */
        /*                                       */
        /*****************************************/
        for(ucTxBufferCounter = 0; ucTxBufferCounter < sizeof(ucTxBuffer); ucTxBufferCounter++)
        {/* 全データ送信完了するまで，送信動作継続 */
                STIF6 = 0;                              /* 割り込み要求クリア */
                TXB6 = ucTxBuffer[ucTxBufferCounter];   /* データ送信 */
                while(!STIF6)/* UART6で1Byte送信完了待ち */
                        NOP();
        }
}
```

第6章　デバイスでの動作確認例

この章では，温度測定結果の例を示します。デバイスを恒温槽で動作させたときの，恒温槽の温度と温度測定結果を以下にまとめます。

恒温槽（℃）	温度測定結果（℃）
32	32
34	34
36	36
36.2	36.1
36.4	36.4
36.6	36.5
36.8	36.7
37	36.9
38	37.8
40	39.9
42	41.8



第7章　関連資料

資　料　名		和文／英文
78K0/LC3　ユーザーズ・マニュアル		PDF
78K0/LD3　ユーザーズ・マニュアル		PDF
78K0/LE3　ユーザーズ・マニュアル		PDF
78K0/LF3　ユーザーズ・マニュアル		PDF
78K/0シリーズ　命令編　ユーザーズ・マニュアル		PDF
RA78K0　アセンブラ・パッケージ	言語編	PDF
ユーザーズ・マニュアル	操作編	PDF
CC78K0　Cコンパイラ	言語編	PDF
ユーザーズ・マニュアル	操作編	PDF
PM+　プロジェクト・マネージャ　ユーザーズ・マニュアル		PDF

付録A　プログラム･リスト

プログラム･リスト例として，78K0/LF3マイクロコントローラのソース･プログラムを次に示します。

●main.asm　（アセンブリ言語版）

```
;*******************************************************************************
;
;           NEC Electronics         78K0/Lx3シリーズ
;
;*******************************************************************************
;           78K0/LF3シリーズ              サンプル･プログラム
;*******************************************************************************
;           ポート及びタイマ機能による温度測定プログラム編
;*******************************************************************************
;【履歴】
;           2008.05.--          新規作成
;*******************************************************************************
;
;【概要】
;
; 本サンプルプログラムは，外部に接続したサーミスタを使用し温度を測定するプログラ
;ムです。
; 1秒毎に温度を測定し，シリアル・インタフェースUART6にて送信します。
; 温度の測定は，まずキャリブレーション抵抗とサーミスタ，それぞれを使用してコンデン
;サの放電時間を測定し，放電時間と抵抗値の比よりサーミスタの抵抗値を算出します。算
;出したサーミスタの抵抗ちを使用し，温度変換テーブルを使用して温度に変換します。
; 放電時間の測定は16ビット・タイマ／イベント・カウンタ00のパルス幅測定機能を使用し
;ます。
; 測定範囲は32.0℃から42.0℃です。測定範囲外の値を測定した場合は，UARTにてエラー
;を送信します。
; 本サンプルプログラムのUARTは，送信のみを行います。
;
;
;   ＜初期設定の主な設定内容＞
;
;   ・ベクタ・テーブルの設定
;   ・レジスタ・バンクの設定
;   ・スタック・ポインタの設定
;   ・ROM/RAMサイズの設定
```

```
;    ・ポートの設定
;    ・CPUクロック周波数を高速内蔵発振8MHz(TYP.)に設定
;    ・16ビット・タイマ／イベント・カウンタ00の設定
;    ・8ビット・タイマH2の設定
;    ・シリアル・インタフェースUART6の設定
;    ・割り込みマスクの設定
;
;
;    ＜メイン処理の主な内容＞
;
;    ・キャリブレーション抵抗の放電パルス幅取得処理
;    ・サーミスタの放電パルス幅取得処理
;    ・温度取得処理
;    ・UART6の送信データ作成＆データ送信処理
;
;
;    ＜放電パルス幅測定処理の主な内容＞
;
;    ・コンデンサの充電
;    ・コンデンサ放電開始
;    ・TM00カウント値(パルス幅)取得
;
;
;    ＜温度取得処理の主な内容＞
;
;    ・パルス幅からサーミスタの抵抗値を算出
;    ・サーミスタの抵抗値のエラー判定
;    ・サーミスタの抵抗値から温度を取得
;
;
;    ＜UART6の送信データ作成＆データ送信処理の主な内容＞
;
;    ・送信データ作成
;    ・通信動作の開始
;    ・送信データのカウント
;    ・送信データの設定
;
;*****************************************************************************

;=============================================================================
;
;        ベクタ・テーブルの設定
;
```

```
;================================================================================
TVCT1    CSEG    AT      000000H
         DW      RESET_START                 ;(00)   RESET入力,POC,LVI,WDT,TRAP
TVCT2    CSEG    AT      000004H
         DW      RESET_START                 ;(04)   INTLVI
         DW      RESET_START                 ;(06)   INTP0
         DW      RESET_START                 ;(08)   INTP1
         DW      RESET_START                 ;(0A)   INTP2
         DW      RESET_START                 ;(0C)   INTP3
         DW      RESET_START                 ;(0E)   INTP4
         DW      RESET_START                 ;(10)   INTP5
         DW      RESET_START                 ;(12)   INTSRE6
         DW      RESET_START                 ;(14)   INTSR6
         DW      RESET_START                 ;(16)   INTST6
         DW      RESET_START                 ;(18)   INTCSI10/INTST0
         DW      RESET_START                 ;(1A)   INTTMH1
         DW      RESET_START                 ;(1C)   INTTMH0
         DW      RESET_START                 ;(1E)   INTTM50
         DW      RESET_START                 ;(20)   INTTM000
         DW      RESET_START                 ;(22)   INTTM010
         DW      RESET_START                 ;(24)   INTAD
         DW      RESET_START                 ;(26)   INTSR0
         DW      RESET_START                 ;(28)   INTRTC
         DW      RESET_START                 ;(2A)   INTTM51
         DW      RESET_START                 ;(2C)   INTKR
         DW      RESET_START                 ;(2E)   INTRTCI
         DW      RESET_START                 ;(30)   INTDSAD
         DW      RESET_START                 ;(32)   INTTM52
         DW      RESET_START                 ;(34)   INTTMH2
         DW      RESET_START                 ;(36)   INTMCG
         DW      RESET_START                 ;(38)   INTRIN
         DW      RESET_START                 ;(3A)   INTRERR/INTGP/INTREND/INTDFULL
         DW      RESET_START                 ;(3C)   INTACSI
         DW      RESET_START                 ;(3E)   BRK

;================================================================================
;
;       スタック領域の確保
;
;================================================================================
DSTK            DSEG    AT      0FB00H          ;RAM先頭アドレス
STACKEND:
                DS      20H                     ;スタック領域を32バイト確保
```

```
STACKTOP:                                   ;スタック領域の先頭アドレス = FB20H

;=============================================================================
;
;       RAMの定義
;
;=============================================================================
DTHERMO         DSEG    SADDR
R1SECCNT:       DS      1               ;100ms(TMH2)をベース・タイマとして時間をカウントする
 C1SEC          EQU     (1000/100)      ;1秒カウント用

ROVFCNT:        DS      1               ;TM00オーバフロー回数カウンタ

RTXBUF:         DS      6               ;送信データ･バッファ

DTHERMOP        DSEG    SADDRP  ;温度測定関連RAM
RCALBCNT:       DS      2               ;キャリブレーション用TI000パルス幅測定値取得用
RTHERMCNT:      DS      2               ;サーミスタ用TI000パルス幅測定値取得用
RHEAT:          DS      2               ;算出した温度を保存　※測定エラーの場合は"FFFFH"となる

RTEMP16A:       DS      2               ;抵抗値計算用変数
RTEMP16B:       DS      2               ;抵抗値計算用変数
RTEMP32:        DS      4               ;抵抗値計算用変数

;=============================================================================
;
;       ROMの定義
;
;=============================================================================
CREGACC CSEG    UNITP
;-----------------------------------------------------------------------------
;       抵抗を温度に変換するテーブル
;-----------------------------------------------------------------------------
;       24.5 kΩを基準としたオフセット[100Ω]より，温度を参照します。
;       参照される温度はBCD[0.1℃]です。
;-----------------------------------------------------------------------------
TR2HEAT:
        DW      0420H           ;24.5 kΩ → 42.0
        DW      0419H           ;24.6 kΩ → 41.9
        DW      0418H           ;24.7 kΩ → 41.8
        DW      0417H           ;24.8 kΩ → 41.7
        DW      0416H           ;24.9 kΩ → 41.6
        DW      0415H           ;25.0 kΩ → 41.5
```

```
DW      0414H           ;25.1 kΩ → 41.4
DW      0413H           ;25.2 kΩ → 41.3
DW      0412H           ;25.3 kΩ → 41.2
DW      0411H           ;25.4 kΩ → 41.1
DW      0410H           ;25.5 kΩ → 41.0
DW      0409H           ;25.6 kΩ → 40.9
DW      0408H           ;25.7 kΩ → 40.8
DW      0407H           ;25.8 kΩ → 40.7
DW      0406H           ;25.9 kΩ → 40.6
DW      0405H           ;26.0 kΩ → 40.5
DW      0405H           ;26.1 kΩ → 40.5
DW      0404H           ;26.2 kΩ → 40.4
DW      0403H           ;26.3 kΩ → 40.3
DW      0402H           ;26.4 kΩ → 40.2
DW      0401H           ;26.5 kΩ → 40.1
DW      0400H           ;26.6 kΩ → 40.0
DW      0399H           ;26.7 kΩ → 39.9
DW      0398H           ;26.8 kΩ → 39.8
DW      0397H           ;26.9 kΩ → 39.7
DW      0396H           ;27.0 kΩ → 39.6
DW      0395H           ;27.1 kΩ → 39.5
DW      0394H           ;27.2 kΩ → 39.4
DW      0393H           ;27.3 kΩ → 39.3
DW      0392H           ;27.4 kΩ → 39.2
DW      0392H           ;27.5 kΩ → 39.2
DW      0391H           ;27.6 kΩ → 39.1
DW      0390H           ;27.7 kΩ → 39.0
DW      0389H           ;27.8 kΩ → 38.9
DW      0388H           ;27.9 kΩ → 38.8
DW      0387H           ;28.0 kΩ → 38.7
DW      0386H           ;28.1 kΩ → 38.6
DW      0385H           ;28.2 kΩ → 38.5
DW      0384H           ;28.3 kΩ → 38.4
DW      0384H           ;28.4 kΩ → 38.4
DW      0383H           ;28.5 kΩ → 38.3
DW      0382H           ;28.6 kΩ → 38.2
DW      0381H           ;28.7 kΩ → 38.1
DW      0380H           ;28.8 kΩ → 38.0
DW      0379H           ;28.9 kΩ → 37.9
DW      0378H           ;29.0 kΩ → 37.8
DW      0378H           ;29.1 kΩ → 37.8
DW      0377H           ;29.2 kΩ → 37.7
DW      0376H           ;29.3 kΩ → 37.6
```

```
DW      0375H           ;29.4 kΩ → 37.5
DW      0374H           ;29.5 kΩ → 37.4
DW      0373H           ;29.6 kΩ → 37.3
DW      0373H           ;29.7 kΩ → 37.3
DW      0372H           ;29.8 kΩ → 37.2
DW      0371H           ;29.9 kΩ → 37.1
DW      0370H           ;30.0 kΩ → 37.0
DW      0369H           ;30.1 kΩ → 36.9
DW      0368H           ;30.2 kΩ → 36.8
DW      0368H           ;30.3 kΩ → 36.8
DW      0367H           ;30.4 kΩ → 36.7
DW      0366H           ;30.5 kΩ → 36.6
DW      0365H           ;30.6 kΩ → 36.5
DW      0365H           ;30.7 kΩ → 36.5
DW      0364H           ;30.8 kΩ → 36.4
DW      0363H           ;30.9 kΩ → 36.3
DW      0362H           ;31.0 kΩ → 36.2
DW      0361H           ;31.1 kΩ → 36.1
DW      0361H           ;31.2 kΩ → 36.1
DW      0360H           ;31.3 kΩ → 36.0
DW      0359H           ;31.4 kΩ → 35.9
DW      0358H           ;31.5 kΩ → 35.8
DW      0358H           ;31.6 kΩ → 35.8
DW      0357H           ;31.7 kΩ → 35.7
DW      0356H           ;31.8 kΩ → 35.6
DW      0355H           ;31.9 kΩ → 35.5
DW      0354H           ;32.0 kΩ → 35.4
DW      0354H           ;32.1 kΩ → 35.4
DW      0353H           ;32.2 kΩ → 35.3
DW      0352H           ;32.3 kΩ → 35.2
DW      0351H           ;32.4 kΩ → 35.1
DW      0351H           ;32.5 kΩ → 35.1
DW      0350H           ;32.6 kΩ → 35.0
DW      0349H           ;32.7 kΩ → 34.9
DW      0348H           ;32.8 kΩ → 34.8
DW      0348H           ;32.9 kΩ → 34.8
DW      0347H           ;33.0 kΩ → 34.7
DW      0346H           ;33.1 kΩ → 34.6
DW      0346H           ;33.2 kΩ → 34.6
DW      0345H           ;33.3 kΩ → 34.5
DW      0344H           ;33.4 kΩ → 34.4
DW      0343H           ;33.5 kΩ → 34.3
DW      0343H           ;33.6 kΩ → 34.3
```

```
        DW      0342H           ;33.7 kΩ → 34.2
        DW      0341H           ;33.8 kΩ → 34.1
        DW      0341H           ;33.9 kΩ → 34.1
        DW      0340H           ;34.0 kΩ → 34.0
        DW      0339H           ;34.1 kΩ → 33.9
        DW      0338H           ;34.2 kΩ → 33.8
        DW      0338H           ;34.3 kΩ → 33.8
        DW      0337H           ;34.4 kΩ → 33.7
        DW      0336H           ;34.5 kΩ → 33.6
        DW      0336H           ;34.6 kΩ → 33.6
        DW      0335H           ;34.7 kΩ → 33.5
        DW      0334H           ;34.8 kΩ → 33.4
        DW      0334H           ;34.9 kΩ → 33.4
        DW      0333H           ;35.0 kΩ → 33.3
        DW      0332H           ;35.1 kΩ → 33.2
        DW      0332H           ;35.2 kΩ → 33.2
        DW      0331H           ;35.3 kΩ → 33.1
        DW      0330H           ;35.4 kΩ → 33.0
        DW      0330H           ;35.5 kΩ → 33.0
        DW      0329H           ;35.6 kΩ → 32.9
        DW      0328H           ;35.7 kΩ → 32.8
        DW      0328H           ;35.8 kΩ → 32.8
        DW      0327H           ;35.9 kΩ → 32.7
        DW      0326H           ;36.0 kΩ → 32.6
        DW      0326H           ;36.1 kΩ → 32.6
        DW      0325H           ;36.2 kΩ → 32.5
        DW      0324H           ;36.3 kΩ → 32.4
        DW      0324H           ;36.4 kΩ → 32.4
        DW      0323H           ;36.5 kΩ → 32.3
        DW      0322H           ;36.6 kΩ → 32.2
        DW      0322H           ;36.7 kΩ → 32.2
        DW      0321H           ;36.8 kΩ → 32.1
        DW      0320H           ;36.9 kΩ → 32.0
        DW      0320H           ;37.0 kΩ → 32.0
TR2HEATE:

;*******************************************************************************
;
;
;       使用する周辺の初期設定
;
;
;*******************************************************************************
```

```
XMAIN    CSEG     UNIT
RESET_START:

;---------------------------------------------------------------------------
;        割り込み禁止
;---------------------------------------------------------------------------
         DI
;---------------------------------------------------------------------------
;        レジスタ・バンク設定
;---------------------------------------------------------------------------
         SEL      RB0
;---------------------------------------------------------------------------
;        スタック・ポインタの設定
;---------------------------------------------------------------------------
         MOVW     SP,      #STACKTOP
;---------------------------------------------------------------------------
;        ROM/RAMサイズの設定

;---------------------------------------------------------------------------
;        モデルにより設定値が異なるので注意してください。
;        使用モデルの設定を有効にしてください。（デフォルトではuPD78F0485）
;---------------------------------------------------------------------------
         ;uPD78F0471,uPD78F0481,uPD78F0491使用時の設定
         ;MOV     IMS,     #04H                    ;ROMサイズの設定
         ;MOV     IXS,     #0CH                    ;内部拡張RAMサイズの設定

         ;uPD78F0472,uPD78F0482,uPD78F0492使用時の設定
         ;MOV     IMS,     #0C6H                    ;ROMサイズの設定
         ;MOV     IXS,     #0CH                    ;内部拡張RAMサイズの設定

         ;uPD78F0473,uPD78F0483,uPD78F0493使用時の設定
         ;MOV     IMS,     #0C8H                    ;ROMサイズの設定
         ;MOV     IXS,     #0CH                    ;内部拡張RAMサイズの設定

         ;uPD78F0474,uPD78F0484,uPD78F0494使用時の設定
         ;MOV     IMS,     #0CCH                    ;ROMサイズの設定
         ;MOV     IXS,     #0AH                    ;内部拡張RAMサイズの設定

         ;uPD78F0475,uPD78F0485,uPD78F0495使用時の設定
         MOV      IMS,     #0CFH                    ;ROMサイズの設定
         MOV      IXS,     #0AH                    ;内部拡張RAMサイズの設定

;---------------------------------------------------------------------------
```

```
;       ポート1の設定
;-------------------------------------------------------------------------------
        MOV     P1,     #00000000B              ;P1初期値設定
                        ;++++++++---------- P17/P16/P15/P14/P13/P12/P11/P10:未使用(0)
        MOV     PM1,    #00000000B              ;P1入出力設定
                        ;++++++++---------- PM17/PM16/PM15/PM14/PM13/PM12/PM11/PM10:未使用(0)
;-------------------------------------------------------------------------------
;       ポート2の設定
;-------------------------------------------------------------------------------
        MOV     P2,     #00000000B              ;P2初期値設定
                        ;++++++++---------- P27/P26/P25/P24/P23/P22/P21/P20:未使用(0)
        MOV     PM2,    #00000000B              ;P1入出力設定
                        ;++++++++---------- PM27/PM26/PM25/PM24/PM23/PM22/PM21/PM20:未使用(0)
;-------------------------------------------------------------------------------
;       ポート3の設定
;-------------------------------------------------------------------------------
        MOV     P3,     #00000000B              ;P3初期値設定
                        ;|||+++++---------- P33/P32/P31/P34/P30:Lo(0)
                        ;+++---------- <000固定>
        MOV     PM3,    #11100110B              ;P3入出力設定
                        ;|||+|||+---------- PM34/PM30:未使用(0)
                        ;||| |++---------- PM32/PM31:入力(1) 放電用ポートとして使用
                        ;|||  +----------  PM33:出力(0) コンデンサ充電ポートとして使用(パルス幅測定時は
TI000として使用)
                        ;+++---------- <111固定>
;-------------------------------------------------------------------------------
;       ポート4の設定
;-------------------------------------------------------------------------------
        MOV     P4,     #00000000B              ;P4初期値設定
                        ;++++++++---------- P47/P46/P45/P44/P43/P42/P41/P40:未使用(0)
        MOV     PM4,    #00000000B              ;P4入出力設定
                        ;++++++++---------- PM47/PM46/PM45/PM44/PM43/PM42/PM41/PM40:未使用(0)
;-------------------------------------------------------------------------------
;       ポート8の設定
;-------------------------------------------------------------------------------
        MOV     P8,     #00000000B              ;P8初期値設定
                        ;||||++++---------- P83/P82/P81/P80:未使用(0)
                        ;++++---------- <0000固定>
        MOV     PM8,    #11110000B              ;P8入出力設定
                        ;||||++++---------- PM83/PM82/PM81/PM80:未使用(0)
                        ;++++---------- <1111固定>
;-------------------------------------------------------------------------------
;       ポート9の設定
```

```
;-------------------------------------------------------------------------------
        MOV     P9,     #00000000B              ;P9初期値設定
                        ;||||++++---------------- P93/P92/P91/P90:未使用(0)
                        ;++++------------------- <0000固定>
        MOV     PM9,    #11110000B              ;P9入出力設定
                        ;||||++++---------------- PM93/PM92/PM91/PM90:未使用(0)
                        ;++++------------------- <1111固定>
;-------------------------------------------------------------------------------
;       ポート10の設定
;-------------------------------------------------------------------------------
        MOV     P10,    #00000000B              ;P10初期値設定
                        ;||||++++---------------- P103/P102/P101/P100:未使用(0)
                        ;++++------------------- <0000固定>
        MOV     PM10,   #11110000B              ;P10入出力設定
                        ;||||++++---------------- PM103/PM102/PM101/PM100:未使用(0)
                        ;++++------------------- <1111固定>
;-------------------------------------------------------------------------------
;       ポート11の設定
;-------------------------------------------------------------------------------
        MOV     P11,    #00000100B              ;P11初期値設定
                        ;||||+|++---------------- P113/P111/P110:未使用(0)
                        ;||||  +----------------- P112:Hi(1)
                        ;++++------------------- <0000固定>
        MOV     PM11,   #11110000B          ;P11入出力設定
                        ;||||+|++---------------- PM113/PM111/PM110:未使用(0)
                        ;||||  +----------------- PM112:出力(0) TxD6として使用
                        ;++++------------------- <1111固定>
;-------------------------------------------------------------------------------
;       ポート12の設定
;-------------------------------------------------------------------------------
        MOV     P12,    #00000000B              ;P12初期値設定
                        ;|||||||+---------------- P120:未使用(0)
                        ;|||++++----------------- P124/P123/P122/P121:Read Only
                        ;+++-------------------- <000固定>
        MOV     PM12,   #11111110B              ;P12入出力設定
                        ;|||||||+---------------- PM120:未使用(0)
                        ;+++++++----------------- <1111111固定>
;-------------------------------------------------------------------------------
;       ポート13の設定
;-------------------------------------------------------------------------------
        MOV     P13,    #00000000B              ;P13初期値設定
                        ;||||++++---------------- P133/P132/P131/P130:未使用(0)
                        ;++++------------------- <0000固定>
```

```
        MOV     PM13,   #11110000B              ;P13入出力設定
                        ;||||++++---------- PM133/PM132/PM131/PM130:未使用(0)
                        ;++++---------- <1111固定>
;-------------------------------------------------------------------------------
;       ポート14の設定
;-------------------------------------------------------------------------------
        MOV     P14,    #00000000B              ;P14初期値設定
                        ;||||++++---------- P143/P142/P141/P140:未使用(0)
                        ;++++---------- <0000固定>
        MOV     PM14,   #11110000B              ;P14入出力設定
                        ;||||++++---------- PM143/PM142/PM141/PM140:未使用(0)
                        ;++++---------- <1111固定>
;-------------------------------------------------------------------------------
;       ポート15の設定
;-------------------------------------------------------------------------------
        MOV     P15,    #00000000B              ;P15初期値設定
                        ;||||++++---------- P153/P152/P151/P150:未使用(0)
                        ;++++---------- <0000固定>
        MOV     PM15,   #11110000B              ;P15入出力設定
                        ;||||++++---------- PM153/PM152/PM151/PM150:未使用(0)
                        ;++++---------- <1111固定>

;-------------------------------------------------------------------------------
;       クロック周波数の設定
;-------------------------------------------------------------------------------
;       高速内蔵発振8MHz(TYP.)で動作を行うように設定します。
;-------------------------------------------------------------------------------
        MOV     OSCCTL, #00000000B              ;クロック動作モード
                        ;||||++++---------- <0000固定>
                        ;|||+---------- OSCSELS:入力ポート・モード
                        ;||+---------- <0固定>
                        ;++---------- EXCLK/OSCSEL:
                        ;                       高速システム・クロック端子の動作モード：入力ポート・
モード
                        ;                       P121/X1,P122/X2/EXCLK：入力ポート

        MOV     MOC,    #10000000B              ;メインOSCコントロール
                        ;|+++++++---------- <0000000固定>
                        ;+---------- X1発振回路停止,EXCLK端子からの外部クロック無効

        MOV     MCM,    #00000000B              ;供給クロック選択
                        ;|||||+|+---------- XSEL/MCM0:
                        ;||||| |                メイン・システム・クロック(fXP) = 高速内蔵発振クロック(fRH)
```

```
                        ;|||||  |                    周辺ハードウエア・クロック(fPRS) = 高速内蔵発振クロック
(fRH)
                        ;|||||  +---------------- MCS: Read Only
                        ;+++++------------------- <00000固定>

        MOV     PCC,    #00000000B                  ; CPUクロック(fCPU)の選択
                        ;|||+|+++---------------- CSS/PCC2/PCC1/PCC0:
                        ;|||  |                          CPUクロック(fCPU)=fXP
                        ;|||  +------------------ <0固定>
                        ;||+--------------------- CLS:  メイン・システム・クロック
                        ;++---------------------- <00固定>

        MOV     RCM,    #00000001B                  ; CPUクロック(fCPU)の選択
                        ;|||||||+--------------- LSRSTOP:低速内蔵発信器の停止
                        ;||||||+---------------- RSTOP:高速内蔵発信器の発振
                        ;|+++++------------------ <00000固定>
                        ;+---------------------- RSTS: Read Only

;--------------------------------------------------------------------------------
;       8ビット・タイマH2
;--------------------------------------------------------------------------------
;       100msのインターバル・タイマに設定し，
;       温度測定及びUART送信(1s毎)のインターバル用に使用します。
;--------------------------------------------------------------------------------
        MOV     TMHMD2,#01100000B               ;タイマ･クロック選択レジスタ
                        ;|||||||+--------------- TOEN2:  タイマ出力禁止
                        ;||||||+---------------- TOLEV2:  タイマ出力レベル 未使用
                        ;||||++----------------- TMMD21/TMMD20:  タイマ動作=インターバル
                        ;|+++------------------- CKS22/CKS21/CKS20:  カウント･クロック  fPRS/2^12  (fPRS =
8MHz より1953.125Hz)
                        ;+---------------------- TMHE2:  タイマ動作禁止(タイマ設定完了後許可)

        MOV     CMP02,  #(195-1)        ;100msインターバル：(fPRS/2^12)*0.1[sec]=195.3125

        SET1    TMHE2                   ;タイマ動作開始
        CLR1    TMHIF2                  ;割り込み要求クリア

        MOV     R1SECCNT,#C1SEC         ;TMH0ベース・タイマの1秒カウンタ初期化

;--------------------------------------------------------------------------------
;       16ビット・タイマ／イベント・カウンタ00
;--------------------------------------------------------------------------------
;       温度センサの抵抗値を測るため，コンデンサの放電時間(パルス幅)を測定します。
```

```
;-------------------------------------------------------------------------------
        MOV     TMC00,  #00000000B              ;16ビット・タイマ・モード・コントロール・レジスタ00
                ;|||||||+--------------- OVF00:TM00のオーバフロー・フラグ クリア
                ;||||||+---------------- TMC001:タイマ出力（TO00）反転条件はTM00とCR000の一致，TM00
とCR010の一致
                ;||||++----------------- TMC003/TMC002:16ビット・タイマ／イベント・カウンタ00動作禁止
                ;++++------------------- <0固定>
        MOV     CRC00,  #00000111B              ; キャプチャ／コンペア・コントロール・レジスタ00
                ;|||||||+--------------- CRC000:CR000をキャプチャ・レジスタとして動作
                ;||||||+---------------- CRC001:CR000のキャプチャ・トリガはTI000端子の有効エッジの逆相
                ;|||||+----------------- CRC002:CR010をキャプチャ・レジスタとして動作
                ;+++++------------------ <0固定>
        MOV     TOC00,  #0000000B               ;16ビット・タイマ出力コントロール・レジスタ00
                ;|||||||+--------------- TOE00:TO00出力禁止
                ;||||||+---------------- TOC001:CR000とTM00の一致によるTO00出力反転動作禁止
                ;||||++----------------- LVS00/LVR00:TO00端子出力の状態変化無し
                ;|||+-------------------- TOC004:CR010とTM00の一致によるTO00出力反転動作禁止
                ;||+--------------------- OSPE00:ワンショット・パルス出力動作は連続パルス出力
                ;|+---------------------- OSPT00:ソフトウエアによるワンショット・パルス出力トリガなし
                ;+----------------------- <0固定>
        MOV     PRM00,  #00000000B              ;プリスケーラ・モード・レジスタ00
                ;|||||+++----------------- PRM002/PRM001/PRM000:fPRS=fRHより設定禁止
                ;||||+------------------- <0固定>
                ;||++--------------------- ES001/ES000:TI000端子の有効エッジ 立下りエッジ
                ;++----------------------- ES101/ES100:TI010端子の有効エッジ 立下りエッジ


;-------------------------------------------------------------------------------
;       UART6の設定
;-------------------------------------------------------------------------------
;       温度センサでの測定結果を送信します。
;-------------------------------------------------------------------------------

        MOV     CKSR6,  #00000000B              ;UART6基本クロック選択
                ;||||++++----------------- TPS63-60:  基本クロック(fXCLK6) = fPRS
                ;++++-------------------- <0固定>

        ;ボー･レート用クロックの分周値設定
        MOV     BRGC6,  #35                     ;ボーレート  = 8*10^6[Hz]/(2 * 115200[bps]) = 34.72
                                                ;※誤差を小さくするため，小数点以下を切り上げ
                                                ;ボーレート：115200bps ← 114285bps(ERR:-0.79%)

        MOV     ASIM6,  #01000101B              ;UART6動作モード選択
                ;|||||||+--------------- ISRM6:  受信エラー発生時にINTSR6を割り込み
```

```
                ;||||||+---------------- SL6:  ストップ･ビット数=1
                ;|||||+----------------- CL6:  データ長=8
                ;|||++------------------ PS61-60:  パリティなし
                ;||+-------------------- RXE6:  受信動作禁止
                ;|+--------------------- TXE6:  送信動作許可
                ;+---------------------- POWER6:  内部動作クロックの動作禁止

        MOV     ASICL6, #00010110B              ;先頭ビット,TxD6出力反転選択
                ;|||||||+--------------- TXDLV6: TxD6通常出力
                ;||||||+---------------- DIR6:  先頭ビットLSB
                ;|||+++------------------ SBL62-60:  未使用
                ;||+-------------------- SBTT6:  未使用
                ;|+--------------------- SBRT6: Read Only
                ;+---------------------- SBRF6:  未使用

        MOV     ISC,    #00001000B              ;入力切り替え制御
                ;|||||||+--------------- ISC0:  未使用
                ;||||||+---------------- ISC1: TI000への入力ソースにP33/TI000端子からの入力信号を選択
                ;|||||+----------------- ISC2:  未使用
                ;||||+------------------ ISC3: RxD6/P113入力許可
                ;||++------------------- ISC5-4: TxD6=P112,RxD6=P113
                ;++---------------------- <0固定>

        SET1    POWER6          ;内部動作クロックの動作許可

;---------------------------------------------------------------------------
;       割り込みマスクの設定
;---------------------------------------------------------------------------
        MOVW    MK0,#0FFFFH
        MOVW    MK1,#0FFFFH             ;全ての割り込みをマスク

;---------------------------------------------------------------------------
;       割り込み許可
;---------------------------------------------------------------------------
        EI

;***************************************************************************
;
;
;       メイン処理
;
;
;***************************************************************************
```

```
MAIN_LOOP:
        ;*****************************************;
        ;                                         ;
        ;              測定温度送信処理           ;
        ;                                         ;
        ;*****************************************;
        ;-----------------------------------------;
        ;              タイミング作成処理         ;
        ;-----------------------------------------;
LMAIN100:
        BF      TMHIF2,$LMAIN500        ;100ms経過した? →NO
        CLR1    TMHIF2                  ;割り込み要求クリア
        DEC     R1SECCNT                ;1秒カウンタ更新
        BNZ     $LMAIN500               ;1秒経過した? →NO
        MOV     R1SECCNT,#C1SEC         ;1秒カウンタ初期化

        ;-----------------------------------------;
        ;              温度測定処理               ;
        ;-----------------------------------------;
        MOV     B,#0                    ;引数(パルス幅測定モード)設定
        CALL    !SGETPULSE              ;キャリブレーション抵抗の放電パルス幅測定
        MOVW    RCALBCNT,AX             ;測定したパルス幅を取得

        MOV     B,#1                    ;引数(パルス幅測定モード)設定
        CALL    !SGETPULSE              ;サーミスタの放電パルス幅測定
        MOVW    RTHERMCNT,AX            ;測定したパルス幅を取得

LMAIN400:
        CALL    !SGETHEAT               ;測定したパルス幅から抵抗を算出し，温度を取得

        ;-----------------------------------------;
        ;     UART6送信データ作成＆データ送信     ;
        ;-----------------------------------------;
        CALL    !SUART6TX

LMAIN500:
        ;*****************************************;
        ;                                         ;
        ;              各種メイン処理             ;
        ;                                         ;
        ;*****************************************;

        ; 各種メイン処理があればここで行います。
```

```
        BR      MAIN_LOOP

;*************************************************************************
;
;       コンデンサの放電時間測定 (TI000パルス幅測定)
;
;-------------------------------------------------------------------------
;       [  IN  ] B:パルス幅測定モード(0:キャリブレーション抵抗の放電パルス幅を測定
;                                      1:サーミスタの放電パルス幅を測定          )
;       [  OUT ] AX:測定した放電パルス幅
;                ROVFCNT:TM00オーバフロー回数
;
;       TI000のパルス幅測定機能を利用し，コンデンサの放電時間を測定します。
;       キャリブレーション抵抗かサーミスタのどちらの放電パルス幅を測定するか
;       引数で指定します。
;       戻り値として測定した放電パルス幅を返します。
;       パルス幅測定中にTM00がオーバフローした場合は，
;       その回数をオーバフロー回数カウンタに設定します。
;*************************************************************************
SGETPULSE:
        MOV     A,B                     ;パルス幅測定モードを取得
        MOV     ROVFCNT,#0              ;オーバフロー回数カウンタクリア

        ;============    コンデンサ充電    ============
        SET1    P3.3
        CLR1    PM3.3                   ;コンデンサ充電開始
        ;充電完了するまで2msecウエイト
        MOV     B,#93           ;[4clk]                 ↑
JGETP100:                       ;                       |
        MOV     C,#27           ;[4clk]↑                | 4 + (166 + 6)*93 = 16000clk
JGETP101:                       ;       | 4+6*27 = 166clk | 16000 * 0.125[μs] = 2000[μs]
        DBNZ    C,$JGETP101     ;[6clk]↓                |
        DBNZ    B,$JGETP100     ;[6clk]                 ↓

        SET1    PM3.3                   ;P33をTI000として使用
        CLR1    TMIF010                 ;割り込み要求クリア

        ;===========    コンデンサ放電開始    ===========
        MOV     B,A                     ;パルス幅測定モードを保存
        CMP     A,#0                    ;キャリブレーション抵抗の放電パルス幅を測定する?
        BNZ     $JGETP200               ; →NO:サーミスタの放電パルス幅を測定する
```

```
        CLR1    P3.1                ;放電準備;P31をLo出力に設定すると放電が開始する
        MOV     TMC00,#08H          ;16ビット・タイマ／イベント・カウンタ00でパルス幅測定開始
        CLR1    PM3.1               ;放電開始
        BR      JGETP300
JGETP200:
        CLR1    P3.2                ;放電準備;P32をLo出力に設定すると放電が開始する
        MOV     TMC00,#08H          ;16ビット・タイマ／イベント・カウンタ00でパルス幅測定開始
        CLR1    PM3.2               ;放電開始

JGETP300:
        ;==========  コンデンサ放電完了待ち  ==========
        BT      TMIF010,$JGETP500   ;コンデンサの放電完了? →YES

        BF      OVF00,$JGETP400     ;TM00のオーバーフロを検出した? →NO
        CLR1    OVF00               ;TM00オーバフロー・フラグをクリア
        INC     ROVFCNT             ;オーバフロー回数を更新
        CMP     ROVFCNT,#2          ;オーバフローが2回以上発生した?
        BZ      $JGETP500           ; →YES:温度測定エラー，パルス幅の測定を中断する
JGETP400:
        BR      JGETP300            ;引き続きコンデンサの放電完了を待つ

JGETP500:
        CLR1    TMIF010             ;割り込み要求クリア

        ;===========  コンデンサ放電完了  ===========
        MOVW    AX,CR010            ;測定したパルス幅を取得
        DEC     B
        BZ      $JGETP700           ;キャリブレーション抵抗の放電パルス幅を測定した?
                                    ; →NO:サーミスタの放電パルス幅を測定した
        SET1    PM3.1               ;キャリブレーション抵抗での放電用のポートを入力に戻す
        CMP     ROVFCNT,#0          ;パルス幅測定でオーバフローがあった?
        BZ      $JGETP800           ; →NO:測定したパルス幅を戻り値として返す
        MOVW    AX,#0               ;戻り値を値をエラーにする
        BR      JGETP800
JGETP700:
        SET1    PM3.2               ;サーミスタでの放電用のポートを入力に戻す

JGETP800:
        MOV     TMC00,#0            ;16ビット・タイマ／イベント・カウンタ00動作停止
        CLR1    P3.3
        CLR1    PM3.3               ;TI000をL出力に戻す

        RET
```

```
;*******************************************************************************
;
;       温度取得処理
;
;-------------------------------------------------------------------------------
;       [   IN   ] RCALBCNT:キャリブレーション抵抗の放電パルス幅
;                  RTHERMCNT:サーミスタの放電パルス幅
;                  ROVFCNT:TM00オーバフロー回数(サーミスタの放電パルス幅測定時)
;       [   OUT ] RHEAT:温度(BCD)
;
;       測定したパルス幅から抵抗を算出し，
;       抵抗を温度に変換するテーブルから温度を取得します。
;
;       ◎パルス幅から下記の式にて抵抗値を算出します。
;
;                          Rc  ×  (CNTth +  オーバフロー回数  ×  10000H)
;               Rth = ----------------------------------------------
;                                              CNTc
;
;               Rth   :サーミスタの抵抗値[100Ω]
;               Rc    :キャリブレーション抵抗の抵抗値  = 330 [100Ω]
;               CNTth:サーミスタの放電パルス幅
;               CNTc :キャリブレーション抵抗の放電パルス幅
;
;
;       ◎Rthの測定範囲に対する相対値を下記の式で求め，
;         その値をオフセットとして抵抗を温度に変換するテーブルから取得します。
;
;               Rrel = Rth - Rmin
;
;               Rrel:Rthの測定範囲に対する相対値[100Ω]
;               Rmin:測定範囲の最小抵抗値  = 245 [100Ω]
;*******************************************************************************
SGETHEAT:
        CMP     RCALBCNT,#0             ;キャリブレーション抵抗の放電パルス幅測定でエラーが起きた?
        BZ      $JGETH800               ; →YES:抵抗値算出不可。抵抗値算出を行わない

        CMP     ROVFCNT,#2              ;パルス幅の測定でオーバフローが2回以上起きた?
        BZ      $JGETH800               ; →YES:既に抵抗値が測定範囲外。抵抗値算出を行わない

        ;--------------------------------;
        ;   パルス幅から抵抗値を算出する    ;
```

```
        ;----------------------------------;
        MOVW    RTEMP32,#0              ;計算用変数下位16bitに0を保存
        MOVW    AX,RTHERMCNT
        MOVW    (RTEMP32+2),AX          ;計算用変数上位16bitにCNTthを保存
        MOVW    AX,#330
        MOVW    RTEMP16A,AX             ;計算用変数にRc(330)[100Ω]を保存
        CALL    !SMULT16                ;(Rc × CNTth) を計算

        ;(Rc × CNTth) の計算結果に (Rc × オーバフロー回数 × 10000H)を加える
        CMP     ROVFCNT,#0              ;オーバフローが発生した?
        BZ      $JGETH300               ; →NO:10000Hの加算をせず抵抗値算出を続行
        MOV     A,ROVFCNT
        MOV     B,A                     ;オーバフロー回数をカウンタに設定
        MOVW    AX,(RTEMP32+2)          ;(Rc × CNTth)の結果の上位16bitに対してRcを足しこむ
JGETH200:
        ADDW    AX,#330                 ;Rcを加算
        BC      $JGETH800               ;加算結果がオーバフロー? →YES:温度は測定範囲外
        DBNZ    B,$JGETH200             ;オーバフロー回数分Rcを足しこんだ? →NO
        MOVW    (RTEMP32+2),AX

JGETH300:
        MOVW    AX,RCALBCNT
        MOVW    RTEMP16A,AX             ;キャリブレーション抵抗の放電パルス幅を除数に設定
        CALL    !SDIV32                 ;(Rc × (CNTth + オーバフロー回数 × 10000H))/CNTc の割り算を
行う

        ;-----------------------------------------------------------------;
        ;   サーミスタの抵抗値が測定範囲(24.5kΩ～37.0kΩ)内であるか判定   ;
        ;-----------------------------------------------------------------;
        MOVW    AX,(RTEMP32+2)          ;算出した抵抗値の上位16bitを取得
        CMPW    AX,#0000H               ;(370 = 172Hより)上位16bitを0と比較
        BNZ     $JGETH800               ;上位16bitが1以上の場合は，測定範囲外なので温度の値をエラーに
する
JGETH400:
        MOVW    AX,RTEMP32              ;算出した抵抗値の下位16bitを取得
        CMPW    AX,#371                 ;算出した抵抗値は37.0kΩ以下?
        BNC     $JGETH800               ; →NO:温度の値をエラーにする
        CMPW    AX,#245                 ;算出した抵抗値は24.5kΩ以上?
        BC      $JGETH800               ; →NO:温度の値をエラーにする

        ;-------------------------;
        ;   抵抗値を温度に変換    ;
        ;-------------------------;
```

```
JGETH500:          ;Rthの測定範囲に対する相対値を算出
        MOVW     AX,RTEMP32
        SUBW     AX,#245                ;Rrel = Rth - Rminを計算
        MOV      A,X                    ;下位8bit取得(測定範囲内であればRrelは8bitに収まる)
        ADD      A,A                    ;Rrelを2倍にし，
        MOV      B,A                    ;抵抗を温度に変換するテーブルのオフセットを取得
        MOVW     HL,#TR2HEAT            ;HLに抵抗を温度に変換するテーブルのアドレスを設定
        MOV      A,[HL+B]               ;温度(下位8bit)を取得
        MOV      X,A
        INC      B
        MOV      A,[HL+B]               ;温度(上位8bit)を取得
        MOVW     RHEAT,AX               ;温度の値を変数に保存
        BR       JGETH900

        ;---------------------------------------------;
        ;    温度測定でエラーが発生した場合の温度設定    ;
        ;---------------------------------------------;
JGETH800:
        MOVW     RHEAT,#0FFFFH          ;温度の値をエラーにする

JGETH900:
        RET

;*****************************************************************************
;
;       UART6送信データ作成＆データ送信
;
;-----------------------------------------------------------------------------
;       [   IN   ] RHEAT:温度(BCD)
;       [   OUT ] なし
;
;       測定した温度を，ASCIIコードに変換して送信バッファに設定し，
;       送信します。
;
;       ＜ 送信データの例 ＞
;               ◎測定結果が38.5℃場合
;                 0   1   2   3   4   5
;               ┌───┬───┬───┬───┬───┬───┐
;               │ 3 │ 8 │ . │ 5 │ ¥r │ ¥n │
;               └───┴───┴───┴───┴───┴───┘
;
;               ◎測定でエラーが発生した場合
;                 0   1   2   3   4   5
```

```
;                   ┌──┬──┬──┬──┬──┬──┐
;                   │ * │ * │ . │ * │¥r │¥n │
;                   └──┴──┴──┴──┴──┴──┘
;
;*******************************************************************************
SUART6TX:
        ;-----------------------------------------
        ;    送信バッファにUART6送信データ作成処理
        ;-----------------------------------------
        MOVW    AX,RHEAT
        CMPW    AX,#0FFFFH              ;温度測定が正常に終了した?
        BZ      $JU6TX100               ; →NO

        ;温度を送信バッファに設定
        AND     A,#0FH                  ;10の位の桁を取得
        ADD     A,#'0'                  ;ASCIIコードに変換
        MOV     RTXBUF,A                ;[0]温度10の位保存

        MOV     A,X                     ;1の位の桁，及び少数第一位を取得
        ROR     A,1                     ;上位4bitを下位4bitに落とす
        ROR     A,1
        ROR     A,1
        ROR     A,1
        AND     A,#0FH                  ;下位4ビット1の位の桁を取得
        ADD     A,#'0'                  ;ASCIIコードに変換
        MOV     (RTXBUF+1),A            ;[1]温度1の位保存

        MOV     (RTXBUF+2),#'.' ;[2]小数点保存

        MOV     A,X                     ;1の位の桁，及び少数第一位を取得
        AND     A,#0FH                  ;少数第一位を取得
        ADD     A,#'0'                  ;ASCIIコードに変換
        MOV     (RTXBUF+3),A            ;[3]温度小数第一位保存

        BR      JU6TX200

JU6TX100:
        ;「＊＊.＊」を送信バッファに設定
        MOV     RTXBUF,#'*'             ;[0]アスタリスク保存
        MOV     (RTXBUF+1),#'*' ;[1]アスタリスク保存
        MOV     (RTXBUF+2),#'.' ;[2]小数点保存
        MOV     (RTXBUF+3),#'*' ;[3]アスタリスク保存
```

```
JU6TX200:
        MOV     (RTXBUF+4),#0DH ;[4]キャリッジリターン保存
        MOV     (RTXBUF+5),#0AH ;[5]ラインフィード保存

        ;----------------------------------------
        ;               UART6データ送信
        ;----------------------------------------
JU6TX500:
        ;==============      送信動作開始      =============
        MOV     B,#6                    ;送信カウンタ設定
        MOVW    HL,#RTXBUF

JU6TX600:
        CLR1    STIF6                   ;割り込み要求クリア
        MOV     A,[HL]                  ;送信バッファから送信データを取得
        MOV     TXB6,A                  ;データ送信
JU6TX700:
        BF      STIF6,$JU6TX700         ;UART6 1Byte送信完了? →NO
        CLR1    STIF6                   ;割り込み要求クリア
        INCW    HL                      ;送信バッファの送信データ位置更新
        DBNZ    B,$JU6TX600             ;未送信データあり? →YES :次のデータを送信
JU6TX800:               ;送信終了
        RET

;*************************************************************************
;
;       掛け算を行う関数 (16bit * 16bit)
;
;-------------------------------------------------------------------------
;       [  IN  ] RTEMP16A:掛け算を行う値
;                RTEMP32:上位16bitに掛け算を行う数を,下位16bitは0を保存
;       [  OUT ] RTEMP32:演算結果
;*************************************************************************
SMULT16:
        MOV     B,#16                   ;ビットカウンタを設定
JMLT120:
        CLR1    CY
        MOV     A,RTEMP32
        ROLC    A,1
        MOV     RTEMP32,A
        MOV     A,(RTEMP32+1)
        ROLC    A,1
        MOV     (RTEMP32+1),A
```

```
        MOV     A,(RTEMP32+2)
        ROLC    A,1
        MOV     (RTEMP32+2),A
        MOV     A,(RTEMP32+3)
        ROLC    A,1
        MOV     (RTEMP32+3),A           ;演算結果(被乗数を含む)を1bit左シフト
        BNC     $JMLT220                ;MSB = 1 ? →NO

        MOV     A,RTEMP16A
        ADD     A,RTEMP32
        MOV     RTEMP32,A
        MOV     A,(RTEMP16A+1)
        ADDC    A,(RTEMP32+1)
        MOV     (RTEMP32+1),A
        MOV     A,#0
        ADDC    A,RTEMP16A
        MOV     RTEMP16A,A              ;被乗数を加える
JMLT220:
        DBNZ    B,$JMLT120              ;16bit分処理が完了した? →NO,

        RET
;*************************************************************************
;
;       割り算を行う関数 (32bit / 16bit)
;
;-------------------------------------------------------------------------
;       [   IN   ] RTEMP16A:除数
;                  RTEMP32:被除数
;       [   OUT ] RTEMP32:演算結果
;                  RTEMP16B:余り
;*************************************************************************
SDIV32:
        MOVW    RTEMP16B,#0             ;計算用変数を初期化

        MOV     B,#32                   ;ビットカウンタを設定
JDIV120:
        CLR1    CY
        MOV     A,RTEMP16B
        ROLC    A,1
        MOV     RTEMP16B,A
        MOV     A,(RTEMP16B+1)
        ROLC    A,1
        MOV     (RTEMP16B+1),A
```

```
        MOV     A,RTEMP32
        ROLC    A,1
        MOV     RTEMP32,A
        MOV     A,(RTEMP32+1)
        ROLC    A,1
        MOV     (RTEMP32+1),A
        MOV     A,(RTEMP32+2)
        ROLC    A,1
        MOV     (RTEMP32+2),A
        MOV     A,(RTEMP32+3)
        ROLC    A,1
        MOV     (RTEMP32+3),A           ;被除数を1bit左シフト
        MOV     A,#0
        ADDC    A,RTEMP16B
        MOV     RTEMP16B,A              ; MSB -> LSB

        SUB     A,RTEMP16A
        MOV     RTEMP16B,A
        MOV     A,(RTEMP16B+1)
        SUBC    A,(RTEMP16A+1)
        MOV     (RTEMP16B+1),A          ;RTEMP16B - RTEMP16A
        BT      RTEMP32.0,$JDIV220      ;桁借りできる? →YES
        BC      $JDIV180                ;RTEMP16B < RTEMP16A ? →YES

        SET1    RTEMP32.0               ;商をセット
        BR      JDIV220

JDIV180:
        MOV     A,RTEMP16B
        ADD     A,RTEMP16A
        MOV     RTEMP16B,A
        MOV     A,(RTEMP16B+1)
        ADDC    A,(RTEMP16A+1)
        MOV     (RTEMP16B+1),A
JDIV220:
        DBNZ    B,$JDIV120              ;32bit分処理が完了した? →NO

        RET

end
```

●main.c　（C言語版）

```
/************************************************************************************

        NEC Electronics        78K0/Lx3シリーズ

************************************************************************************
        78K0/LF3シリーズ            サンプル･プログラム
************************************************************************************
        ポート及びタイマ機能による温度測定プログラム編
************************************************************************************
【履歴】
        2008.5.--          新規作成
************************************************************************************

【概要】

 本サンプルプログラムは，外部に接続したサーミスタを使用し温度を測定するプログラムです。
 1秒毎に温度を測定し，シリアル・インタフェースUART6にて送信します。
 温度の測定は，まずキャリブレーション抵抗とサーミスタ，それぞれを使用してコンデンサの放電時間を測定し，放電時間と抵抗値の比よりサーミスタの抵抗値を算出します。算出したサーミスタの抵抗値を使用し，温度変換テーブルを使用て温度に変換します。
 放電時間の測定は16ビット・タイマ／イベント・カウンタ00のパルス幅測定機能を使用します。
 測定範囲は32.0℃から42.0℃です。測定範囲外の値を測定した場合は，UARTにてエラーを送信します。
 本サンプルプログラムのUARTは，送信のみを行います。


  ＜初期設定の主な設定内容＞

  ・ベクタ・テーブルの設定
  ・レジスタ・バンクの設定
  ・スタック・ポインタの設定
  ・ROM/RAMサイズの設定
  ・ポートの設定
  ・CPUクロック周波数を高速内蔵発振8MHz(TYP.)に設定
  ・16ビット・タイマ／イベント・カウンタ00の設定
  ・8ビット・タイマH2の設定
  ・シリアル・インタフェースUART6の設定
  ・割り込みマスクの設定
```

＜メイン処理の主な内容＞

・キャリブレーション抵抗の放電パルス幅取得処理
・サーミスタの放電パルス幅取得処理
・温度取得処理
・UART6の送信データ作成＆データ送信処理

＜放電パルス幅測定処理の主な内容＞

・コンデンサの充電
・コンデンサ放電開始
・TM00カウント値(パルス幅)取得

＜温度取得処理の主な内容＞

・パルス幅からサーミスタの抵抗値を算出
・サーミスタの抵抗値のエラー判定
・サーミスタの抵抗値から温度を取得

＜UART6の送信データ作成＆データ送信処理の主な内容＞

・送信データ作成
・通信動作の開始
・送信データのカウント
・送信データの設定

```
*****************************************************************************/
#pragma SFR                                    /* 特殊機能レジスタ(SFR)名を記述可能にする */
#pragma DI                                     /* DI命令を記述可能にする */
#pragma EI                                     /* EI命令を記述可能にする */
#pragma NOP                                    /* NOP命令を記述可能にする */

/*============================================================================

        関数プロトタイプ宣言

============================================================================*/
static short fn_GetPulseTime(unsigned char);       /* 放電パルス幅取得処理の主な内容 */
```

```
static short fn_GetHeatData(void);                          /* 温度取得処理 */
static void fn_UART6_Tx(void);                          /* UART6の送信データ作成＆データ送信処理 */

/*==========================================================================

        ＲＯＭの定義

==========================================================================*/
/*--------------------------------------------------------------------------
        抵抗を温度に変換するテーブル
--------------------------------------------------------------------------
        24.5 kΩを基準としたオフセット[100Ω]より，温度を参照します。
        参照される温度はBCD[0.1℃]です。
--------------------------------------------------------------------------*/
const unsigned short tR2Heat[] =
{
         0x0420          /* 24.5 kΩ → 42.0 */
        ,0x0419          /* 24.6 kΩ → 41.9 */
        ,0x0418          /* 24.7 kΩ → 41.8 */
        ,0x0417          /* 24.8 kΩ → 41.7 */
        ,0x0416          /* 24.9 kΩ → 41.6 */
        ,0x0415          /* 25.0 kΩ → 41.5 */
        ,0x0414          /* 25.1 kΩ → 41.4 */
        ,0x0413          /* 25.2 kΩ → 41.3 */
        ,0x0412          /* 25.3 kΩ → 41.2 */
        ,0x0411          /* 25.4 kΩ → 41.1 */
        ,0x0410          /* 25.5 kΩ → 41.0 */
        ,0x0409          /* 25.6 kΩ → 40.9 */
        ,0x0408          /* 25.7 kΩ → 40.8 */
        ,0x0407          /* 25.8 kΩ → 40.7 */
        ,0x0406          /* 25.9 kΩ → 40.6 */
        ,0x0405          /* 26.0 kΩ → 40.5 */
        ,0x0405          /* 26.1 kΩ → 40.5 */
        ,0x0404          /* 26.2 kΩ → 40.4 */
        ,0x0403          /* 26.3 kΩ → 40.3 */
        ,0x0402          /* 26.4 kΩ → 40.2 */
        ,0x0401          /* 26.5 kΩ → 40.1 */
        ,0x0400          /* 26.6 kΩ → 40.0 */
        ,0x0399          /* 26.7 kΩ → 39.9 */
        ,0x0398          /* 26.8 kΩ → 39.8 */
        ,0x0397          /* 26.9 kΩ → 39.7 */
        ,0x0396          /* 27.0 kΩ → 39.6 */
        ,0x0395          /* 27.1 kΩ → 39.5 */
```

```
,0x0394        /* 27.2 kΩ → 39.4 */
,0x0393        /* 27.3 kΩ → 39.3 */
,0x0392        /* 27.4 kΩ → 39.2 */
,0x0392        /* 27.5 kΩ → 39.2 */
,0x0391        /* 27.6 kΩ → 39.1 */
,0x0390        /* 27.7 kΩ → 39.0 */
,0x0389        /* 27.8 kΩ → 38.9 */
,0x0388        /* 27.9 kΩ → 38.8 */
,0x0387        /* 28.0 kΩ → 38.7 */
,0x0386        /* 28.1 kΩ → 38.6 */
,0x0385        /* 28.2 kΩ → 38.5 */
,0x0384        /* 28.3 kΩ → 38.4 */
,0x0384        /* 28.4 kΩ → 38.4 */
,0x0383        /* 28.5 kΩ → 38.3 */
,0x0382        /* 28.6 kΩ → 38.2 */
,0x0381        /* 28.7 kΩ → 38.1 */
,0x0380        /* 28.8 kΩ → 38.0 */
,0x0379        /* 28.9 kΩ → 37.9 */
,0x0378        /* 29.0 kΩ → 37.8 */
,0x0378        /* 29.1 kΩ → 37.8 */
,0x0377        /* 29.2 kΩ → 37.7 */
,0x0376        /* 29.3 kΩ → 37.6 */
,0x0375        /* 29.4 kΩ → 37.5 */
,0x0374        /* 29.5 kΩ → 37.4 */
,0x0373        /* 29.6 kΩ → 37.3 */
,0x0373        /* 29.7 kΩ → 37.3 */
,0x0372        /* 29.8 kΩ → 37.2 */
,0x0371        /* 29.9 kΩ → 37.1 */
,0x0370        /* 30.0 kΩ → 37.0 */
,0x0369        /* 30.1 kΩ → 36.9 */
,0x0368        /* 30.2 kΩ → 36.8 */
,0x0368        /* 30.3 kΩ → 36.8 */
,0x0367        /* 30.4 kΩ → 36.7 */
,0x0366        /* 30.5 kΩ → 36.6 */
,0x0365        /* 30.6 kΩ → 36.5 */
,0x0365        /* 30.7 kΩ → 36.5 */
,0x0364        /* 30.8 kΩ → 36.4 */
,0x0363        /* 30.9 kΩ → 36.3 */
,0x0362        /* 31.0 kΩ → 36.2 */
,0x0361        /* 31.1 kΩ → 36.1 */
,0x0361        /* 31.2 kΩ → 36.1 */
,0x0360        /* 31.3 kΩ → 36.0 */
,0x0359        /* 31.4 kΩ → 35.9 */
```

```
,0x0358         /* 31.5 kΩ → 35.8 */
,0x0358         /* 31.6 kΩ → 35.8 */
,0x0357         /* 31.7 kΩ → 35.7 */
,0x0356         /* 31.8 kΩ → 35.6 */
,0x0355         /* 31.9 kΩ → 35.5 */
,0x0354         /* 32.0 kΩ → 35.4 */
,0x0354         /* 32.1 kΩ → 35.4 */
,0x0353         /* 32.2 kΩ → 35.3 */
,0x0352         /* 32.3 kΩ → 35.2 */
,0x0351         /* 32.4 kΩ → 35.1 */
,0x0351         /* 32.5 kΩ → 35.1 */
,0x0350         /* 32.6 kΩ → 35.0 */
,0x0349         /* 32.7 kΩ → 34.9 */
,0x0348         /* 32.8 kΩ → 34.8 */
,0x0348         /* 32.9 kΩ → 34.8 */
,0x0347         /* 33.0 kΩ → 34.7 */
,0x0346         /* 33.1 kΩ → 34.6 */
,0x0346         /* 33.2 kΩ → 34.6 */
,0x0345         /* 33.3 kΩ → 34.5 */
,0x0344         /* 33.4 kΩ → 34.4 */
,0x0343         /* 33.5 kΩ → 34.3 */
,0x0343         /* 33.6 kΩ → 34.3 */
,0x0342         /* 33.7 kΩ → 34.2 */
,0x0341         /* 33.8 kΩ → 34.1 */
,0x0341         /* 33.9 kΩ → 34.1 */
,0x0340         /* 34.0 kΩ → 34.0 */
,0x0339         /* 34.1 kΩ → 33.9 */
,0x0338         /* 34.2 kΩ → 33.8 */
,0x0338         /* 34.3 kΩ → 33.8 */
,0x0337         /* 34.4 kΩ → 33.7 */
,0x0336         /* 34.5 kΩ → 33.6 */
,0x0336         /* 34.6 kΩ → 33.6 */
,0x0335         /* 34.7 kΩ → 33.5 */
,0x0334         /* 34.8 kΩ → 33.4 */
,0x0334         /* 34.9 kΩ → 33.4 */
,0x0333         /* 35.0 kΩ → 33.3 */
,0x0332         /* 35.1 kΩ → 33.2 */
,0x0332         /* 35.2 kΩ → 33.2 */
,0x0331         /* 35.3 kΩ → 33.1 */
,0x0330         /* 35.4 kΩ → 33.0 */
,0x0330         /* 35.5 kΩ → 33.0 */
,0x0329         /* 35.6 kΩ → 32.9 */
,0x0328         /* 35.7 kΩ → 32.8 */
```

```
        ,0x0328         /* 35.8 kΩ → 32.8 */
        ,0x0327         /* 35.9 kΩ → 32.7 */
        ,0x0326         /* 36.0 kΩ → 32.6 */
        ,0x0326         /* 36.1 kΩ → 32.6 */
        ,0x0325         /* 36.2 kΩ → 32.5 */
        ,0x0324         /* 36.3 kΩ → 32.4 */
        ,0x0324         /* 36.4 kΩ → 32.4 */
        ,0x0323         /* 36.5 kΩ → 32.3 */
        ,0x0322         /* 36.6 kΩ → 32.2 */
        ,0x0322         /* 36.7 kΩ → 32.2 */
        ,0x0321         /* 36.8 kΩ → 32.1 */
        ,0x0320         /* 36.9 kΩ → 32.0 */
        ,0x0320         /* 37.0 kΩ → 32.0 */
};

/*==========================================================================

        ＲＡＭの定義

==========================================================================*/
unsigned char uc1secCnt;                        /* 100ms(TMH2)をベース・タイマとして1秒をカウントする */
#define         TMH2_1SEC       (1000/100)      /* 1秒カウント用 */

unsigned short ushCalibrationCnt;               /* キャリブレーション用TI000パルス幅測定値取得用 */
unsigned short ushThermistorCnt;                /* サーミスタ用TI000パルス幅測定値取得用 */
unsigned char ucOVFcnt;                         /* TM00オーバフロー回数カウンタ */
unsigned short ushHeatData;                     /* 算出した温度を保存　※測定エラーの場合は"FFFFH"となる
*/

unsigned char ucTxBuffer[6];                    /* 送信データ･バッファ */
unsigned char ucTxBufferCounter;                /* 送信カウンタ */

/**************************************************************************

        リセット解除後の初期化処理

**************************************************************************/
void hdwinit(void)
{
        DI();                                   /* 割り込み禁止 */

/*--------------------------------------------------------------------------
        ROM/RAMサイズの設定
```

```
----------------------------------------------------------------------------/
        モデルにより設定値が異なるので注意してください。
        使用モデルの設定を有効にしてください。（デフォルトではuPD78F0485）
----------------------------------------------------------------------------*/
        /* uPD78F0471,uPD78F0481,uPD78F0491使用時の設定 */
        /*IMS = 0x04;                   /* ROMサイズの設定 */
        /*IXS = 0x0C;                   /*  内部拡張RAMサイズの設定 */

        /* uPD78F0472,uPD78F0482,uPD78F0492使用時の設定 */
        /*IMS = 0xC6;                    /* ROMサイズの設定 */
        /*IXS = 0x0C;                   /*  内部拡張RAMサイズの設定 */

        /* uPD78F0473,uPD78F0483,uPD78F0493使用時の設定 */
        /*IMS = 0xC8;                    /* ROMサイズの設定 */
        /*IXS = 0x0C;                   /*  内部拡張RAMサイズの設定 */

        /* uPD78F0474,uPD78F0484,uPD78F0494使用時の設定 */
        /*IMS = 0xCC;                    /* ROMサイズの設定 */
        /*IXS = 0x0A;                   /*  内部拡張RAMサイズの設定 */

        /* uPD78F0475,uPD78F0485,uPD78F0495使用時の設定 */
        IMS =    0xCF;                   /* ROMサイズの設定 */
        IXS =    0x0A;                   /*  内部拡張RAMサイズの設定 */

/*----------------------------------------------------------------------------
        ポートの設定 (未使用ポートはLow出力に設定)
----------------------------------------------------------------------------*/
        /*  ポート1 */
        P1 =     0b00000000;            /* P1初期値設定 */
                 /*++++++++---------- P17/P16/P15/P14/P13/P12/P11/P10:未使用(0) */
        PM1 =    0b00000000;            /* P1入出力設定 */
                 /*++++++++---------- PM17/PM16/PM15/PM14/PM13/PM12/PM11/PM10:未使用(0) */
        /*  ポート2 */
        P2 =     0b00000000;            /* P2初期値設定 */
                 /*++++++++---------- P27/P26/P25/P24/P23/P22/P21/P20:未使用(0) */
        PM2 =    0b00000000;            /* P1入出力設定 */
                 /*++++++++---------- PM27/PM26/PM25/PM24/PM23/PM22/PM21/PM20:未使用(0) */
        /*  ポート3 */
        P3 =     0b00000000;            /* P3初期値設定 */
                 /*|||+++++---------- P33/P32/P31/P34/P30:Lo(0) */
                 /*+++-------------------- <000固定> */
        PM3 =    0b11100110;            /* P3入出力設定 */
                 /*|||+|||+---------- PM34/PM30:未使用(0) */
```

```
                /*||| |++---------------- PM32/PM31:入力(1) 放電用ポートとして使用 */
                /*|||  +-----------------  PM33:出力(0) コンデンサ充電ポートとして使用(パルス幅測定時はTI000として使用) */
                /*+++-------------------- <111固定> */
        /*  ポート4 */
        P4 =    0b00000000;             /* P4初期値設定 */
                /*++++++++---------------- P47/P46/P45/P44/P43/P42/P41/P40:未使用(0) */
        PM4 =    0b00000000;             /* P4入出力設定 */
                /*++++++++---------------- PM47/PM46/PM45/PM44/PM43/PM42/PM41/PM40:未使用(0) */
        /*  ポート8 */
        P8 =    0b00000000;             /* P8初期値設定 */
                /*||||++++---------------- P83/P82/P81/P80:未使用(0) */
                /*++++-------------------- <0000固定> */
        PM8 =    0b11110000;             /* P8入出力設定 */
                /*||||++++---------------- PM83/PM82/PM81/PM80:未使用(0) */
                /*++++-------------------- <1111固定> */
        /*  ポート9 */
        P9 =    0b00000000;             /* P9初期値設定 */
                /*||||++++---------------- P93/P92/P91/P90:未使用(0) */
                /*++++-------------------- <0000固定> */
        PM9 =    0b11110000;             /* P9入出力設定 */
                /*||||++++---------------- PM93/PM92/PM91/PM90:未使用(0) */
                /*++++-------------------- <1111固定> */
        /*  ポート10 */
        P10 =    0b00000000;             /* P10初期値設定 */
                /*||||++++---------------- P103/P102/P101/P100:未使用(0) */
                /*++++-------------------- <0000固定> */
        PM10 =   0b11110000;             /* P10入出力設定 */
                /*||||++++---------------- PM103/PM102/PM101/PM100:未使用(0) */
                /*++++-------------------- <1111固定> */
        /*  ポート11 */
        P11   =  0b00000100;             /* P11初期値設定 */
                /*||||+|++---------------- P113/P111/P110:未使用(0) */
                /*|||| +------------------ P112:Hi(1)*/
                /*++++-------------------- <0000固定> */
        PM11 =   0b11110000;             /* P11入出力設定 */
                /*||||+|++---------------- PM113/PM111/PM110:未使用(0) */
                /*|||| +------------------ PM112:出力(0) TxD6として使用*/
                /*++++-------------------- <1111固定> */
        /*  ポート12 */
        P12 =    0b00000000;             /* P12初期値設定 */
                /*|||||||+---------------- P120:未使用(0) */
                /*|||++++------------------ P124/P123/P122/P121:Read Only */
```

```
            /*+++-------------------- <000固定> */
        PM12 =   0b11111110;                    /* P12入出力設定 */
            /*|||||||+--------------- PM120:未使用(0) */
            /*+++++++----------------- <1111111固定> */
        /* ポート13 */
        P13 =    0b00000000;                    /* P13初期値設定 */
            /*||||++++---------------- P133/P132/P131/P130:未使用(0) */
            /*++++-------------------- <0000固定> */
        PM13 =   0b11110000;                    /* P13入出力設定 */
            /*||||++++---------------- PM133/PM132/PM131/PM130:未使用(0) */
            /*++++-------------------- <1111固定> */
        /* ポート14 */
        P14 =    0b00000000;                    /* P14初期値設定 */
            /*||||++++---------------- P143/P142/P141/P140:未使用(0) */
            /*++++-------------------- <0000固定> */
        PM14 =   0b11110000;                    /* P14入出力設定 */
            /*||||++++---------------- PM143/PM142/PM141/PM140:未使用(0) */
            /*++++-------------------- <1111固定> */
        /* ポート15 */
        P15 =    0b00000000;                    /* P15初期値設定 */
            /*||||++++---------------- P153/P152/P151/P150:未使用(0) */
            /*++++-------------------- <0000固定> */
        PM15 =   0b11110000;                    /* P15入出力設定 */
            /*||||++++---------------- PM153/PM152/PM151/PM150:未使用(0) */
            /*++++-------------------- <1111固定> */

/*-------------------------------------------------------------------------------
        クロック周波数の設定
-------------------------------------------------------------------------------
        高速内蔵発振8MHz(TYP.)で動作を行うように設定します。
-------------------------------------------------------------------------------*/
        OSCCTL =0b00000000;                     /* クロック動作モード */
            /*||||++++---------------- <0000固定> */
            /*|||+-------------------- OSCSELS:入力ポート・モード */
            /*||+--------------------- <0固定> */
            /*++---------------------- EXCLK/OSCSEL: */
            /*                          高速システム・クロック端子の動作モード：入力ポート・モード */
            /*                          P121/X1,P122/X2/EXCLK：入力ポート */

        MOC = 0x80;                             /* X1発振回路停止,EXCLK端子からの外部クロック無効 */

        MCM =    0b00000000;                    /* 供給クロック選択 */
            /*|||||+|+---------------- XSEL/MCM0: */
```

```
                /*||||| |                       メイン・システム・クロック(fXP) =  高速内蔵発振クロック(fRH) */
                /*||||| |                       周辺ハードウエア・クロック(fPRS) =  高速内蔵発振クロック(fRH) */
                /*||||| +---------------- MCS: Read Only */
                /*+++++------------------- <00000固定> */

        PCC =    0b00000000;                  /* CPUクロック(fCPU)の選択 */
                /*|||+|+++---------------- CSS/PCC2/PCC1/PCC0: */
                /*||| |                                  CPUクロック(fCPU)=fXP */
                /*||| +------------------- <0固定> */
                /*||+---------------------- CLS:  メイン・システム・クロック */
                /*++----------------------- <00固定> */

        RCM =    0b00000001;                  /* CPUクロック(fCPU)の選択 */
                /*|||||||+---------------- LSRSTOP:低速内蔵発信器の停止 */
                /*||||||+----------------- RSTOP:高速内蔵発信器の発振 */
                /*|+++++------------------ <00000固定> */
                /*+----------------------- RSTS: Read Only */

/*-------------------------------------------------------------------------------
        8ビット・タイマH2
-------------------------------------------------------------------------------/
        100msのインターバル・タイマに設定し，
        温度測定及びUART送信(1s毎)のインターバル用に使用します。
-------------------------------------------------------------------------------*/
        TMHMD2 = 0b01100000;                  /*  タイマ･クロック選択レジスタ */
                /* |||||||+---------------- TOEN2:  タイマ出力禁止 */
                /* ||||||+----------------- TOLEV2:  タイマ出力レベル 未使用 */
                /* ||||++------------------ TMMD21/TMMD20:  タイマ動作=インターバル */
                /* |+++------------------- CKS22/CKS21/CKS20:  カウント･クロック fPRS/2^12 (fPRS = 8MHz より
1953.125Hz) */
                /* +----------------------- TMHE2:  タイマ動作禁止(タイマ設定完了後許可) */
        CMP02 = 195-1;                       /* 100msインターバル：(fPRS/2^12)*0.1[sec]=195.3125 */

        TMHE2 = 1;                           /*  タイマ動作開始 */
        TMHIF2 = 0;                          /*  割り込み要求クリア */
        uc1secCnt = TMH2_1SEC;               /* TMH0ベース・タイマの1秒カウンタ初期化 */

/*-------------------------------------------------------------------------------
        16ビット・タイマ／イベント・カウンタ00
-------------------------------------------------------------------------------/
        温度センサの抵抗値を測るため，コンデンサの放電時間(パルス幅)を測定します。
-------------------------------------------------------------------------------*/
        TMC00 = 0b00000000;                  /* 16ビット・タイマ・モード・コントロール・レジスタ00 */
```

```
                /*|||||||+---------------- OVF00:TM00のオーバフロー・フラグ クリア */
                /*||||||+----------------- TMC001:タイマ出力（TO00）反転条件はTM00とCR000の一致，TM00とCR010
の一致 */
                /*||||++------------------ TMC003/TMC002:16ビット・タイマ／イベント・カウンタ00動作禁止 */
                /*++++-------------------- <0固定> */
        CRC00 = 0b00000111;             /* キャプチャ／コンペア・コントロール・レジスタ00 */
                /*|||||||+---------------- CRC000:CR000をキャプチャ・レジスタとして動作 */
                /*||||||+----------------- CRC001:CR000のキャプチャ・トリガはTI000端子の有効エッジの逆相 */
                /*|||||+------------------ CRC002:CR010をキャプチャ・レジスタとして動作 */
                /*+++++------------------- <0固定> */
        TOC00 = 0b0000000;              /* 16ビット・タイマ出力コントロール・レジスタ00 */
                /*|||||||+---------------- TOE00:TO00出力禁止 */
                /*||||||+----------------- TOC001:CR000とTM00の一致によるTO00出力反転動作禁止 */
                /*||||++------------------ LVS00/LVR00:TO00端子出力の状態変化無し */
                /*|||+-------------------- TOC004:CR010とTM00の一致によるTO00出力反転動作禁止 */
                /*||+--------------------- OSPE00:ワンショット・パルス出力動作は連続パルス出力 */
                /*|+---------------------- OSPT00:ソフトウエアによるワンショット・パルス出力トリガなし */
                /*+----------------------- <0固定> */
        PRM00 = 0b00000000;             /* プリスケーラ・モード・レジスタ00 */
                /*|||||+++----------------- PRM002/PRM001/PRM000:fPRS=fRHより設定禁止 */
                /*||||+------------------- <0固定> */
                /*||++-------------------- ES001/ES000:TI000端子の有効エッジ 立下りエッジ */
                /*++---------------------- ES101/ES100:TI010端子の有効エッジ 立下りエッジ */


/*-------------------------------------------------------------------------------
        UART6の設定
-------------------------------------------------------------------------------/
        温度センサでの測定結果を送信します。
-------------------------------------------------------------------------------*/

        CKSR6 = 0b00000000;             /* UART6基本クロック選択 */
                /*||||++++---------------- TPS63-60:  基本クロック(fXCLK6) = fPRS */
                /*++++-------------------- <0固定> */

        /* ボー･レート用クロックの分周値設定 */
        BRGC6 = 35;       /* ボーレート = 8*10^6[Hz]/(2 * 115200[bps]) = 34.72 */
                          /* ※誤差を小さくするため，小数点以下を切り上げ */
                          /* ボーレート：115200bps ← 114285bps(ERR:-0.79%) */

        ASIM6 = 0b01000101;             /* UART6動作モード選択 */
                /*|||||||+---------------- ISRM6:  受信エラー発生時にINTSR6を割り込み */
                /*||||||+----------------- SL6:  ストップ･ビット数=1 */
                /*|||||+------------------ CL6:  データ長=8 */
```

```
                /*|||++------------------ PS61-60: パリティなし */
                /*||+------------------- RXE6: 受信動作禁止 */
                /*|+-------------------- TXE6: 送信動作許可 */
                /*+--------------------- POWER6: 内部動作クロックの動作禁止 */

        ASICL6 =0b00010110;                     /* 先頭ビット,TxD6出力反転選択 */
                /*|||||||+---------------- TXDLV6: TxD6通常出力 */
                /*||||||+----------------- DIR6: 先頭ビットLSB */
                /*|||+++------------------ SBL62-60: 未使用 */
                /*||+-------------------- SBTT6: 未使用 */
                /*|+--------------------- SBRT6: Read Only */
                /*+---------------------- SBRF6: 未使用 */

        ISC =   0b00001000;                     /* 入力切り替え制御 */
                /*|||||||+---------------- ISC0: 未使用 */
                /*||||||+----------------- ISC1: TI000への入力ソースにP33/TI000端子からの入力信号を選択 */
                /*|||||+------------------ ISC2: 未使用 */
                /*||||+------------------- ISC3: RxD6/P113入力許可 */
                /*||++-------------------- ISC5-4: TxD6=P112,RxD6=P113 */
                /*++---------------------- <0固定> */

        POWER6 =        1;                      /* 内部動作クロックの動作許可 */

/*---------------------------------------------------------------------------
        割り込みマスクの設定
---------------------------------------------------------------------------*/
        MK0 = 0x0FFFF;
        MK1 = 0x0FFFF;                          /* 全ての割り込みをマスク */

        EI();                                   /* 割り込み許可 */
}

/*****************************************************************************
        メイン・ループ

*****************************************************************************/
void main(void)
{
        while(1)
        {
                /*******************************************/
                /*                                         */
```

```
                /*                  測定温度送信処理                    */
                /*                                                      */
                /*******************************************/
                /*-----------------------------------------*/
                /*                タイミング作成処理                    */
                /*-----------------------------------------*/
                if(TMHIF2)
                {/* 100ms経過   */
                        TMHIF2 = 0;                 /* 割り込み要求クリア */
                        uc1secCnt--;                /* 1秒カウンタ更新 */
                }

                /*-----------------------------------------*/
                /*                  温度測定処理                        */
                /*-----------------------------------------*/
                if(uc1secCnt == 0)
                {/* 1秒経過 */
                        uc1secCnt = TMH2_1SEC;   /* 1秒カウンタクリア */

                        /* キャリブレーション抵抗の放電パルス幅測定 */
                        ushCalibrationCnt = fn_GetPulseTime(0);

                        /* サーミスタの放電パルス幅測定 */
                        ushThermistorCnt = fn_GetPulseTime(1);

                        /* 測定したパルス幅から抵抗を算出し，温度を取得 */
                        ushHeatData = fn_GetHeatData();

                        /* UART6送信データ作成＆データ送信 */
                        fn_UART6_Tx();
                }
                /*******************************************/
                /*                                                      */
                /*                  各種メイン処理                      */
                /*                                                      */
                /*******************************************/

                /* 各種メイン処理があればここで行います。 */
        }
}

/*****************************************************************************
```

コンデンサの放電時間測定 (TI000パルス幅測定)

```
--------------------------------------------------------------------------------
        [   IN   ] mode(0:キャリブレーション抵抗の放電パルス幅を測定
                        1:サーミスタの放電パルス幅を測定                )
        [   OUT ] 測定した放電パルス幅

        TI000のパルス幅測定機能を利用し，コンデンサの放電時間を測定します。
        キャリブレーション抵抗かサーミスタのどちらの放電パルス幅を測定するか
        引数で指定します。
        戻り値として測定した放電パルス幅を返します。
        パルス幅測定中にTM00がオーバフローした場合は，
        その回数をオーバフロー回数カウンタに設定します。
********************************************************************************/
static short fn_GetPulseTime(unsigned char mode)
{
        unsigned short ushRet;          /* 戻り値保存用 */
        unsigned short temp;            /* ワーク領域 */

        ucOVFcnt = 0;           /* オーバフロー回数カウンタクリア */

        /* コンデンサ充電 */
        P3.3 = 1;
        PM3.3 = 0;              /* コンデンサ充電開始 */
        for(temp = 224; temp > 0; temp--)
        {
                NOP();          /* 充電完了するまで2msec程度ウエイト */
        }
        PM3.3 = 1;              /* P33をTI000として使用 */
        TMIF010 = 0;            /* 割り込み要求クリア */

        /* コンデンサ放電開始 */
        if(mode == 0)
        {/* キャリブレーション抵抗の放電パルス幅を測定する */
                P3.1 = 0;               /* 放電準備 *//* P31をLo出力に設定すると放電が開始する */
                TMC00 = 0x08;           /* パルス幅測定開始 */
                PM3.1 = 0;              /* 放電開始 */
        }
        else
        {/* サーミスタの放電パルス幅を測定する */
                P3.2 = 0;               /* 放電準備 *//* P32をLo出力に設定すると放電が開始する */
                TMC00 = 0x08;           /* パルス幅測定開始 */
                PM3.2 = 0;              /* 放電開始 */
```

```
        }

        /* コンデンサ放電完了待ち */
        while(!TMIF010)
        {
                if(OVF00)
                {/* TM00のオーバーフロを検出ーした場合 */
                        OVF00 = 0;                  /* TM00オーバフロー・フラグをクリア */
                        ucOVFcnt++;                 /* オーバフロー回数を更新 */
                        if(ucOVFcnt >= 2)           /* オーバフロー回数が2回以上の場合は */
                                break;              /* 温度測定エラー，パルス幅の測定を中断する。 */
                }
        }
        TMIF010 = 0;                /* 割り込み要求クリア */
        ushRet = CR010;             /* 測定したパルス幅を取得 */

        /* コンデンサ放電完了 */
        if(mode == 0){              /* 放電用のポートを入力に戻す */
                PM3.1 = 1;          /* キャリブレーション抵抗を使用した場合 */
                if(ucOVFcnt > 0)
                        ushRet = 0;
        }
        else
        {
                PM3.2 = 1;          /* サーミスタを使用した場合 */
        }

        TMC00 = 0x00;               /* 16ビット・タイマ／イベント・カウンタ00動作停止 */
        P3.3 = 0;
        PM3.3 = 0;                  /* TI000をL出力にもどす */

        return ushRet;              /* パルス幅を返す */
}

/*****************************************************************************

        温度取得処理

-----------------------------------------------------------------------------
        [   IN   ] なし
        [   OUT ] 温度(BCD)

        測定したパルス幅から抵抗を算出し，
```

抵抗を温度に変換するテーブルから温度を取得します。

◎パルス幅から下記の式にて抵抗値を算出します。

(抵抗値とパルス幅は比例するより)

Rc : CNTc = Rth : CNTth

$$Rth = \frac{Rc \times (CNTth + \text{オーバフロー回数} \times 0x10000)}{CNTc}$$

Rth　:サーミスタの抵抗値[100Ω]

Rc　:キャリブレーション抵抗の抵抗値 = 330 [100Ω]

CNTth:サーミスタの放電パルス幅

CNTc :キャリブレーション抵抗の放電パルス幅

◎Rthの測定範囲に対する相対値を下記の式で求め，
その値をオフセットとして抵抗を温度に変換するテーブルから取得します。

Rrel = Rth - Rmin

Rrel:Rthの測定範囲に対する相対値[100Ω]

Rmin:測定範囲の最小抵抗値 = 245 [100Ω]

```
*********************************************************************************/
static short fn_GetHeatData(void)
{
    unsigned short ushRet;                  /* 戻り値保存用 */
    unsigned long int ulTemp1;              /* 計算用RAM */
    unsigned char ucTemp2;                  /* 計算用RAM */

    if((ushCalibrationCnt != 0) && (ucOVFcnt < 2))
    {/* キャリブレーション抵抗の放電パルス幅が正常に測定でき，*/
     /*かつ，サーミスタ抵抗の放電パルス幅測定でオーバフローの回数が2回以上でなければ，*/
     /* パルス幅から抵抗値を算出する */
        /* オーバフロー分を合わせ，測定したサーミスタのパルス幅を32bitに拡張 */
        ulTemp1 = (unsigned long)(ucOVFcnt * 0x10000) + ushThermistorCnt;
        /* サーミスタの抵抗値を計算 */
        ushRet = (unsigned short)((ulTemp1 * 330) / ushCalibrationCnt);

        /* サーミスタの抵抗値が測定範囲(24.5kΩ～37.0kΩ)内であるか判定 */
        if((ushRet <= 370)&&(ushRet >= 245))
        {/* 抵抗値が測定範囲に含まれている → 抵抗値から温度を取得 */
            ucTemp2 = (unsigned char)(ushRet - 245);
```

```
                ushRet = tR2Heat[ucTemp2];
            }
            else
            {/* 抵抗値が測定範囲外 → 温度の値をエラーにする */
                ushRet = 0xffff;
            }
        }
        else
        {/* サーミスタ放電パルス幅測定で，2回以上オーバフローが起こると */
        /* 既に抵抗値が測定範囲外となっている */
            ushRet = 0xffff;                    /* 温度の値をエラーにする */
        }

        return ushRet;              /* 温度を返す */
}

/*****************************************************************************

        送信データ作成＆UART6データ送信

-----------------------------------------------------------------------------
        [  IN  ] なし
        [  OUT ] なし

        測定した温度を，ASCIIコードに変換して送信バッファに設定し，
        送信します。

        ＜ 送信データの例 ＞
            ◎測定結果が38.5℃場合
              0   1   2   3   4   5
            ┌───┬───┬───┬───┬───┬───┐
            │ 3 │ 8 │ . │ 5 │ ¥r │ ¥n │
            └───┴───┴───┴───┴───┴───┘

            ◎測定でエラーが発生した場合
              0   1   2   3   4   5
            ┌───┬───┬───┬───┬───┬───┐
            │ * │ * │ . │ * │ ¥r │ ¥n │
            └───┴───┴───┴───┴───┴───┘

*****************************************************************************/
static void fn_UART6_Tx(void)
{
```

```
        /*******************************************/
        /*                                         */
        /*          UART6送信データ作成            */
        /*                                         */
        /*******************************************/
        if(ushHeatData != 0xFFFF)
        {/* 温度測定が正常に終了 → 温度を送信バッファに設定 */
                ucTxBuffer[0] = (unsigned char)(((ushHeatData >> 8) & 0x000f) + '0');    /* [0]温度10の位(ASCIIコ
ードに変換) */
                ucTxBuffer[1] = (unsigned char)(((ushHeatData >> 4) & 0x000f) + '0');    /* [1]温度1の位(ASCIIコ
ードに変換) */
                ucTxBuffer[2] = '.';                                                    /* [2]小数点 */
                ucTxBuffer[3] = (unsigned char)((ushHeatData & 0x000f) + '0');          /* [3]温度小数第一位
(ASCIIコードに変換) */
        }
        else
        {/* 測定エラーが発生 → 「＊＊.＊」を送信バッファに設定 */
                ucTxBuffer[0] = '*';    /* [0]アスタリスク保存 */
                ucTxBuffer[1] = '*';    /* [1]アスタリスク保存 */
                ucTxBuffer[2] = '.';    /* [2]小数点保存 */
                ucTxBuffer[3] = '*';    /* [3]アスタリスク保存 */
        }
        ucTxBuffer[4] = '¥r';           /* [4]キャリッジリターン */
        ucTxBuffer[5] = '¥n';           /* [5]ラインフィード */

        /*******************************************/
        /*                                         */
        /*          UART6データ送信                */
        /*                                         */
        /*******************************************/
        for(ucTxBufferCounter = 0; ucTxBufferCounter < sizeof(ucTxBuffer); ucTxBufferCounter++)
        {/* 全データ送信完了するまで，送信動作継続 */
                STIF6 = 0;                              /* 割り込み要求クリア */
                TXB6 = ucTxBuffer[ucTxBufferCounter];   /* データ送信 */
                while(!STIF6)/* UART6で1Byte送信完了待ち */
                        NOP();
        }
  }
```

付録B　改版履歴

版　数	発行年月	改版箇所	改版内容
第1版	November 2008	－	－

【発　行】
NECエレクトロニクス株式会社
〒211-8668　神奈川県川崎市中原区下沼部1753
電話（代表）：044(435)5111

お問い合わせ先

【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。
URL(アドレス)　http://www.necel.co.jp/

【営業関係，技術関係お問い合わせ先】

半導体ホットライン
（電話：午前 9:00～12:00，午後 1:00～5:00）
電　話　：044−435−9494
E-mail　：info@necel.com

【資料請求先】

NECエレクトロニクスのホームページよりダウンロードいただくか，NECエレクトロニクスの販売特約店へお申し付けください。