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

78K0R/Kx3

サンプル・プログラム（UART）

サブシステム・クロックを使用したボー・レート補正付きUART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，およびシリアル・アレイ・ユニットのUARTの機能の設定方法や活用方法について説明したものです。サンプル・プログラムは，サブシステム・クロックを使用し，高速内蔵発振クロックを動作クロックとしているシリアル・アレイ・ユニットの転送ボー・レートのキャリブレーションを行った後，シリアル・アレイ・ユニットのUARTの機能を使用したUART送受信を行うプログラムです。UART送受信は，マスタ機器からA/Dコンバータのチャネルと分解能の指定値を受信し，指定されたA/Dコンバータのチャネルで変換したA/D値を指定された分解能でマスタ機器へ送信を行う半2重での送受信としています。

対象デバイス

78K0R/KE3マイクロコントローラ
78K0R/KF3マイクロコントローラ
78K0R/KG3マイクロコントローラ
78K0R/KH3マイクロコントローラ
78K0R/KJ3マイクロコントローラ

目　次

資料番号　U19222JJ1V0AN00（第1版）
発行年月　January 2009 NS

- **本資料に記載されている内容は2009年1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

このサンプル・プログラムは，シリアル・アレイ・ユニットのUARTの機能を使用したUART送受信を行うプログラムです。まず，UART送受信を行う前にキャリブレーション開始処理の呼び出しにより，キャリブレーションを行います。キャリブレーションは，水晶振動子による周波数精度の良いサブシステム・クロックで動作するリアルタイム・カウンタの定周期割り込みを用い精度の良い0.5sの時間を計測し，タイマ・アレイ・ユニットのキャプチャ・モードでその間の高速内蔵発振クロックでのカウント数を計測します。計測したタイマ・カウント値を高速内蔵発振クロックを動作クロックとしているシリアル・アレイ・ユニットの転送ボー・レート用に計算し，設定します。キャリブレーション後，マスタ機器からA/Dコンバータのチャネルと分解能の指定値を受信し，指定されたA/Dコンバータのチャネルで変換したA/D値を指定された分解能でマスタ機器へ送信します。

このサンプルプログラムのUARTは，データ長が8ビット，転送レートが9600bps，データ位相が正転出力，パリティ・ビットなし，ストップ・ビットが1ビット付加，データ方向がLSBファースト転送となるように設定し，半2重での送受信としています。

(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
- ●CPU／周辺ハードウェア・クロックを高速内蔵発振クロックの動作に設定
- ●ポートの設定
- ●A/Dコンバータの設定
- ●シリアル・アレイ・ユニット0（SAU0）の設定
  - ・チャネル0をUART0送信用に設定
  - ・チャネル1をUART0受信用に設定
- ●割り込みの許可
- ●サブシステム・クロックの発振安定待ち。リアルタイム・カウンタ（RTC）を0.5sに一度の定周期割り込みに設定

(2) メイン処理の主な内容

メイン処理の主な内容は，次のとおりです。

- ・UART0のエラー・チェック
- ・UART0受信データの解析
- ・A/D値の取り込み
- ・UART0送信データの作成
- ・UART0送受信動作の開始

(3) キャリブレーション開始処理の内容

キャリブレーション開始処理の主な内容は，次のとおりです。

- タイマ・アレイ・ユニット0（TAU0）のチャネル0をキャプチャ・モードに設定
- リアルタイム・カウンタ（RTC）を0.5sに一度の定周期割り込みに設定
- タイマ・アレイ・ユニット0（TAU0）のチャネル0のキャプチャ動作とリアルタイム・カウンタ（RTC）の定周期割り込み動作の開始

(4) INTST0割り込み処理（UART0の送信完了割り込み）

INTST0割り込み処理の主な内容は，次のとおりです。

- UART0送信データのカウント
- UART0送信データの設定
- UART0送信動作の停止

(5) INTSR0割り込み処理（UART0受信完了割り込み）

INTSR0割り込み処理の主な内容は，次のとおりです。

- UART0受信データのカウント
- UART0受信データの保存
- UART0受信動作の停止

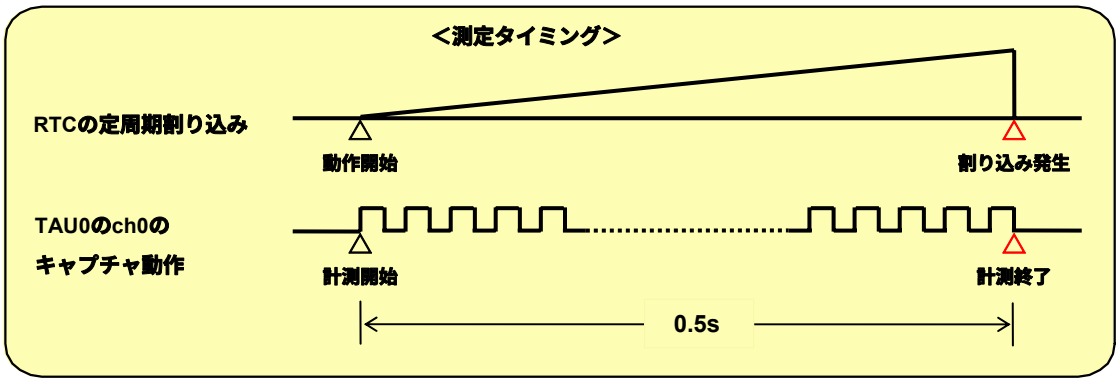
(6) INTRTC割り込み処理（リアルタイム・カウンタの定周期割り込み）

キャリブレーション開始処理の呼び出しから0.5s後に発生する割り込み処理です。
INTRTC割り込み処理の主な内容は，次のとおりです。

- タイマ・アレイ・ユニット0（TAU0）のチャネル0のキャプチャ動作とリアルタイム・カウンタ（RTC）の定周期割り込み動作の停止
- タイマ・アレイ・ユニット0（TAU0）のチャネル0のキャプチャ･モードで計測した0.5s間の高速内蔵発振クロックのタイマ･カウント値からUART0の転送ボー・レートを計算し，設定
- UART0受信動作の開始

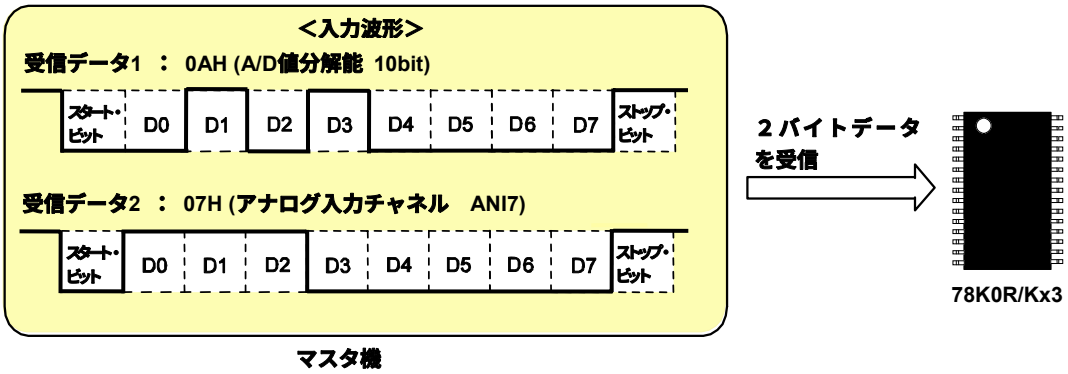
【動作概要】

- 計測方法



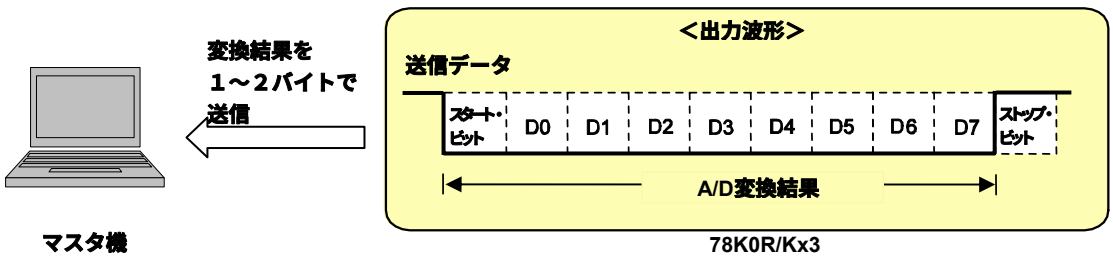
受信データよりUART転送ボー・レートを計算し設定

- マスタ機　→　78K0R/Kx3



受信データの設定に変更後、A/D変換を実行

- マスタ機　←　78K0R/Kx3

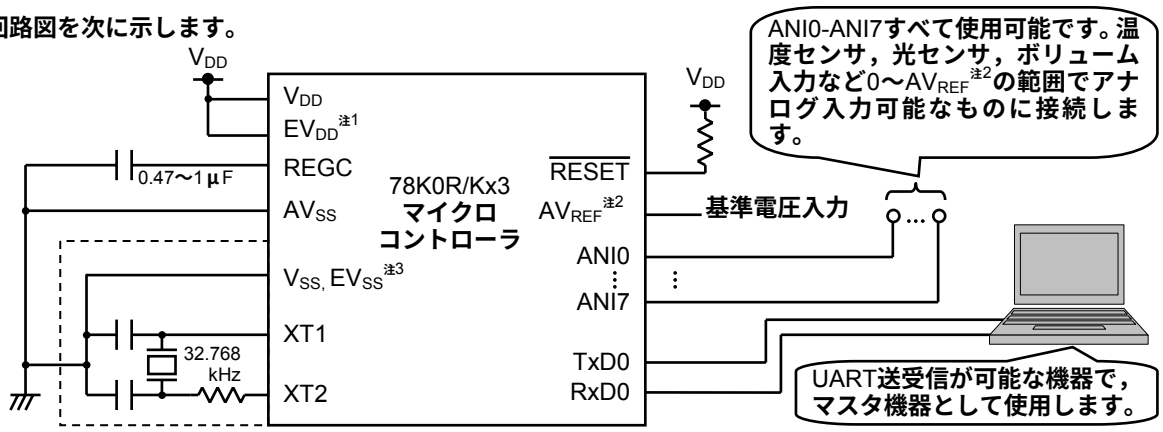


第2章　回路図

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

2. 1　回路図

回路図を次に示します。



注1.　78K0R/KG3，78K0R/KH3，78K0R/KJ3マイクロコントローラはEV_{DD0}とEV_{DD1}になります。

注2.　78K0R/KF3，78K0R/KG3，78K0R/KH3，78K0R/KJ3マイクロコントローラはAV_{REF0}になります。

注3.　78K0R/KG3, 78K0R/KH3, 78K0R/KJ3マイクロコントローラはEV_{SS0}とEV_{SS1}になります。

注意1.　4.5 V≦V_{DD}≦5.5 Vの電圧範囲で使用してください。

2.　AV_{SS}端子はEV_{SS}, V_{SS}と同電位にし，GNDに直接接続してください。

3.　EV_{DD}は，V_{DD}と同電位にしてください。

4.　REGCはコンデンサ（0.47～1 μF）を介し，V_{SS}に接続してください。

5.　回路図中の端子以外の未使用のポート機能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。

6.　ANI0-ANI7の端子はすべて使用可能です。温度センサ，光センサ，ボリューム入力など0～AV_{REF}（上記注２参照）の範囲でアナログ入力可能なものに接続します。

7.　TxD0，RxD0端子は，UART送受信が可能な機器に接続します。

8.　X1発振回路およびXT1発振回路を使用する場合は，配線容量などの影響を避けるために，図の破線の部分を次のように配線してください。

- 配線は極力短くする。
- 他の信号線と交差させない。また，変化する大電流が流れる線と接近させない。
- 発振回路のコンデンサの接地点は，常にV_{SS}と同電位となるようにする。大電流が流れるグランド・パターンに接地しない。
- 発振回路から信号を取り出さない。

特に，XT1発振回路は，低消費電力にするために増幅度の低い回路になっていますのでご注意ください。

備考　発振子の選択および発振回路定数についてはお客様において発振評価していただくか，発振子メーカに評価を依頼してください。

2. 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）サブシステム・クロック（XT1，XT2）

XT1，XT2端子に32.768kHzの発振子を接続します。

（2）UART通信用機器（TxD0，RxD0）

TxD0，RxD0端子にUART送受信用の機器を接続します。

（3）A/Dコンバータのアナログ入力端子（ANI0-ANI7）

ANI0-ANI7端子に温度センサ，光センサ，ボリューム入力など0～AV_{REF}[注]の範囲でアナログ入力可能なものに接続します。

注.　78K0R/KF3，78K0R/KG3，78K0R/KH3，78K0R/KJ3マイクロコントローラはAV_{REF0}になります。

第3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･プログラムの使用する周辺の初期設定と動作概要，およびフロー･チャートを説明します。

3. 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） main.c （C言語版）	マイコンのハードウエア初期化処理，メイン処理と割り込み処理のソース･ファイル	●注	●注
Uart_SubClock.prw	統合開発環境 PM+用ワーク･スペース･ファイル		●
Uart_SubClock.prj	統合開発環境 PM+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+で使用するファイルを同封

3. 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。

・UART送信用：　シリアル･アレイ･ユニット0（SAU0）のチャネル0をUART0送信として使用
・UART受信用：　シリアル･アレイ･ユニット0（SAU0）のチャネル1をUART0受信として使用
・マスタ機器への送信データとして使用する電圧値の計測用：
　　A/Dコンバータのチャネル0～7を使用
・サブシステム・クロックでの0.5sの時間計測用：
　　リアルタイム・カウンタ（RTC）を0.5sに一度の定周期割り込みで使用
・高速内蔵発振クロックのカウント数の計測用：
　　タイマ・アレイ・ユニット0（TAU0）のチャネル0をキャプチャ・モードで使用

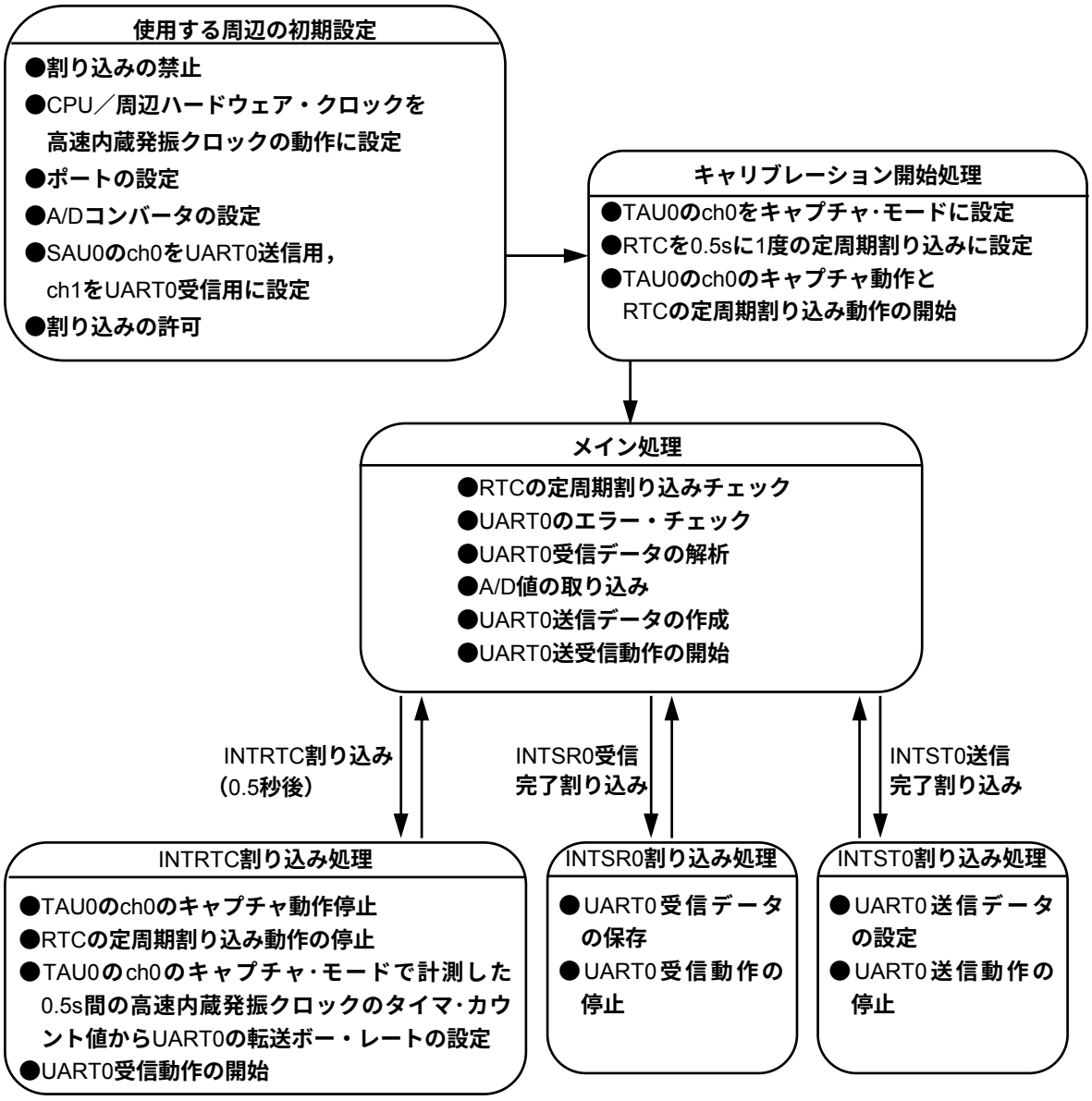
3. 3　使用する周辺の初期設定と動作概要

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，クロック周波数の選択や，A/Dコンバータ，シリアル・アレイ・ユニットを使用したUART，リアルタイム・カウンタの定周期割り込みの設定などを行います。

使用する周辺の初期設定完了後，最初のUARTを受信する前に一度だけキャリブレーション開始処理を呼び出します。このサンプル・プログラムでは最初のUART受信を行う前にキャリブレーションを一度行うだけですが，一定周期でキャリブレーションを行うなど，セットにおいて必要な精度を満たせる十分な評価のもと調整してください。

キャリブレーション開始処理では，サブシステム・クロックで動作するリアルタイム・カウンタ（RTC）の定周期割り込みにて精度の良い0.5sで，割り込みを発生させます。また，その0.5s間の高速内蔵発振クロックのカウント数を，タイマ・アレイ・ユニット0（TAU0）のチャネル0のキャプチャ・モードにて計測します。キャリブレーション開始処理で開始した0.5s後の割り込み処理では，タイマ・アレイ・ユニット0（TAU0）のチャネル0のキャプチャ・モードで計測したカウント値をもとに高速内蔵発振クロックを動作クロックとしているシリアル・アレイ・ユニット0（SAU0）のUART0送受信の転送ボー・レートを計算し，設定します。キャリブレーション後，マスタ機器からA/Dコンバータのチャネルと分解能の指定値を受信し，指定されたA/Dコンバータのチャネルで変換したA/D値を指定された分解能でマスタ機器へ送信します。

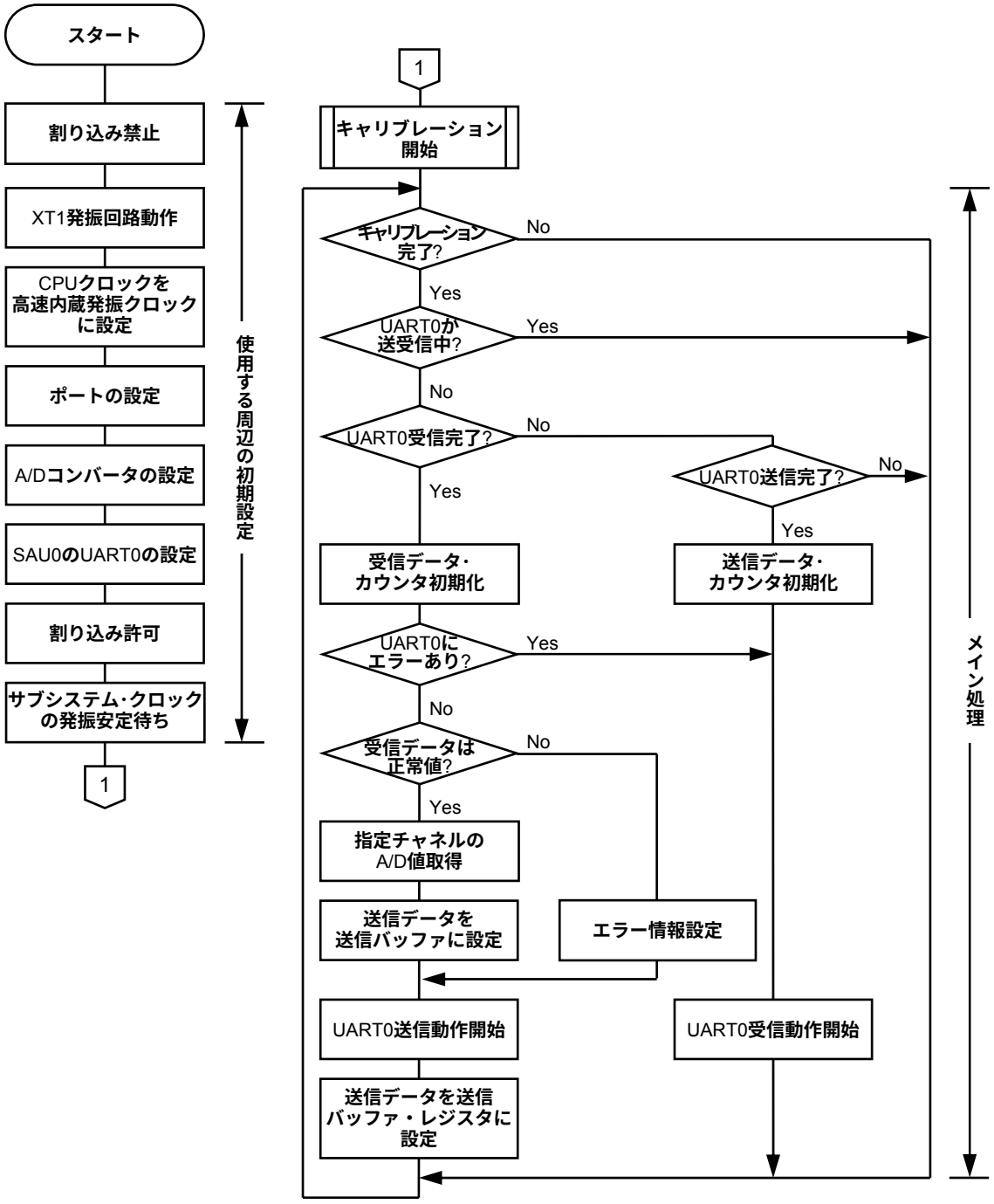
詳細については，次の状態遷移図（ステート・チャート）に示します。



3. 4　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。

＜リセット解除後の初期化処理＞

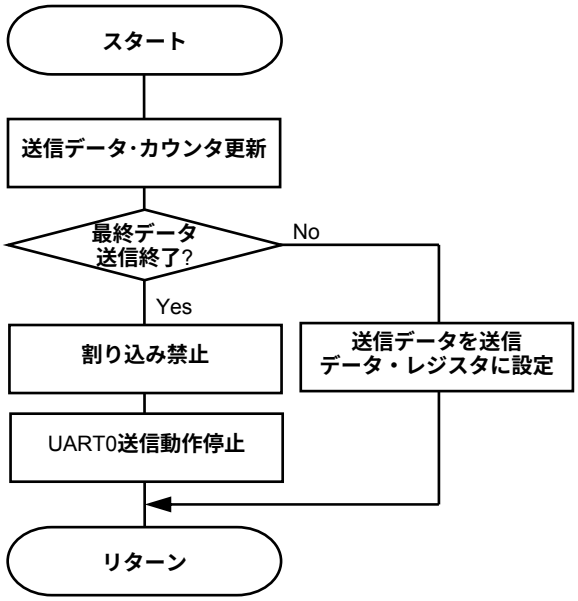


注意　**オプション・バイトは，**RA78K0R**のリンカ・オプションにて設定してください。設定の仕方については，**RA78K0R **アセンブラ・パッケージ ユーザーズ・マニュアルを参照してください。**

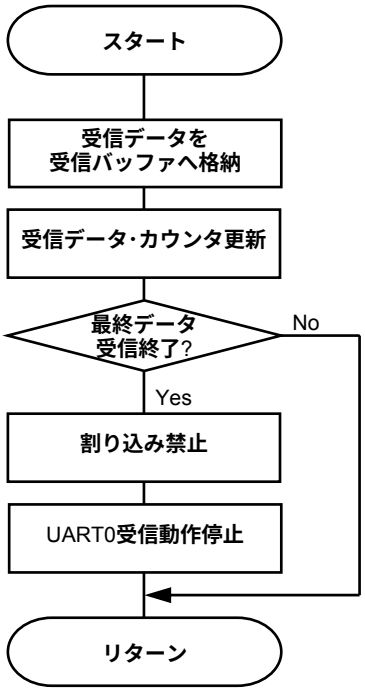
オプション・バイトは，次の内容が設定されます。

- **ウォッチドッグ・タイマの動作**
- **リセット解除時（電源立ち上げ時）の**LVI**の設定**
- **オンチップ・デバッグの動作制御**

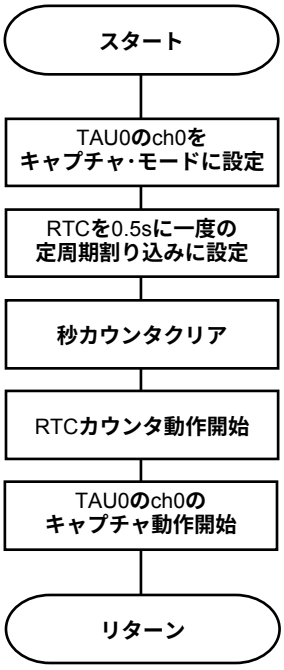
＜INTST0割り込み処理＞



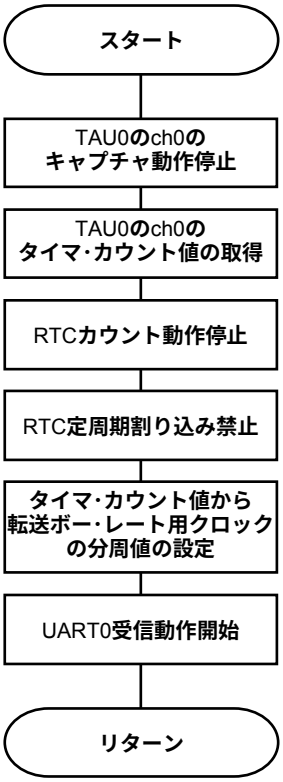
＜INTSR0割り込み処理＞



＜キャリブレーション開始処理＞



＜INTRTC割り込み処理＞

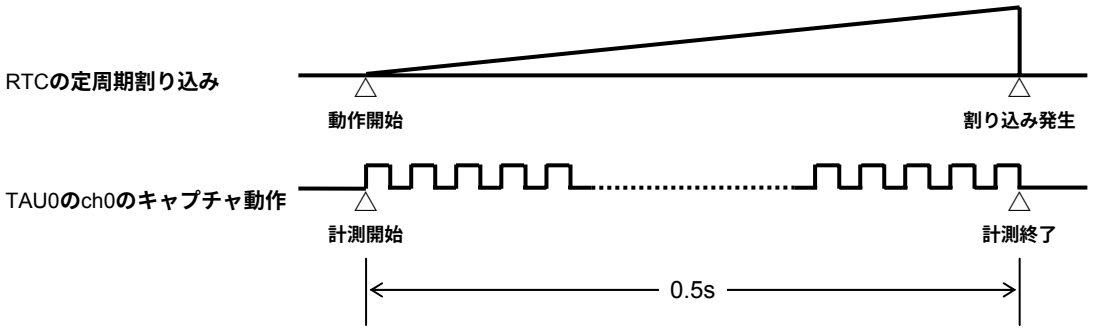


3. 5　キャリブレーション

水晶振動子による周波数精度の良いサブシステム・クロックを使用したリアルタイム・カウンタ（RTC）の定周期割り込みを用い精度の良い0.5sの時間を計測し，タイマ・アレイ・ユニット（TAU0）のチャネル0のキャプチャ・モードでその間の高速内蔵発振クロックでのカウント数を計測します。計測したタイマ・カウント値を高速内蔵発振クロックを動作クロックとしているシリアル・アレイ・ユニット0（SAU0）のUART0の転送ボー・レート用に計算し，設定することでキャリブレーションを行います。転送ボー・レートの計算は，UART0送受信の転送レートが9600bpsとなるように設定を行います。

計測のタイミングとボー・レートの算出は，次のとおりです。

【計測のタイミング】



【ボー・レートの算出】

ボー・レートの計算式は，

（ボー・レート）=｛対象チャネルの動作クロック（MCK）周波数｝ ÷ （SDRmn[15:9]＋1） ÷ 2 [bps] ･･･ ①

※ SDRmn[15:9]**は，**SDRmn**レジスタのビット**15-9**の値を示します。**

で算出できます。(詳細はユーザーズ・マニュアルを参照してください。)

・CPU／**周辺ハードウエア・クロック** ・・・ f_{CLK} [Hz]
・**ボー・レート** ・・・ 9600 [bps]

として，①式から，

（SDRmn[15:9]+1） = $\{f_{CLK}/2^2\}$ ÷ 9600 ÷ 2 ･･･ ②

と表せます。高速内蔵発振クロックの2^6**分周で動作するタイマ・アレイ・ユニット（**TAU0**）のチャネル**0**のキャプチャで計測した**0.5s**間のタイマ・カウント値（**TimerCount**）から，**

$\{2^6/f_{CLK}\}$ × TimerCount = 0.5 [s]

と表せますので，f_{CLK}**は，**

$f_{CLK} = 2^7$ × TimerCount ･･･ ③

となり，②，③式から，

SDRmn[15:9] = $\{2^7$ × TimerCount ÷ $2^2\}$ ÷ 9600 ÷ 2 － 1 = TimerCount ÷ 600 － 1

上記の式より，シリアル・データ・レジスタ（SDR0x**）の**UART0**転送ボー・レート用動作クロックの分周値の算出が行えます。**

【 例 】 TimerCount ・・・ 62,500 **の時**
・CPU／**周辺ハードウエア・クロック** ・・・ 8 [Hz]
・**ボー・レート** ・・・ 9600 [bps]

SDRmn[15:9] = TimerCount ÷ 600 － 1 =（62,500 ÷ 600）－ 1
= 103

SDRmn **には・・・ 「**CE00H**」 を設定。**
SDRmn**設定方法は、**P21（4）**動作クロックの分周設定 を参照してください。**

3. 6　UART送受信データのフォーマット

【受信データ】・・・2バイト

byte	+0								+1							
bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
data	A/D値分解能の指定 08H: 8ビット 0AH:10ビット								取得するA/Dの チャンネル指定 00H:CH0 01H:CH1 02H:CH2 03H:CH3 04H:CH4 05H:CH5 06H:CH6 07H:CH7							

【送信データ】・・・2バイト

8ビット分解能の場合

byte	+0								+1							
bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
data	エラー情報	未使用							8ビットA/D値							

10ビット分解能の場合

byte	+0								+1							
bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
data	エラー情報	未使用							10ビットA/D値							

注意　エラー情報は，0:エラーなし，1:エラーあり とし，受信データに誤りがあれば「エラーあり」とし，送信データに設定します。

第4章　設定方法について

この章では，シリアル・アレイ・ユニット0（SAU0）のUART0送受信用の設定，使用する周辺の初期設定，メイン処理，キャリブレーション開始処理，INTST0割り込み処理，INTSR0割り込み処理，およびINTRTC割り込み処理について説明します。

表の赤字部分がサンプル・プログラムでの設定値となります。

初期設定方法の詳細については，78K0R/Kx3　サンプル・プログラム（初期設定）　LED点灯のスイッチ制御編　アプリケーション・ノートを参照してください。

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（78K0R/KE3，78K0R/KF3，78K0R/KG3，78K0R/KH3，78K0R/KJ3）を参照してください。

アセンブラ命令については，78K0R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。

4. 1　シリアル・アレイ・ユニット0（SAU0）の設定

シリアル・データ送信（TxD）とシリアル・データ受信（RxD）の2本のラインによる，調歩同期式通信を行います。通信相手と非同期で（内部ボー・レートを使用して），データを送受信します。送信専用（偶数チャネル）と受信専用（奇数チャネル）の2チャネルを使用することで，全2重UART通信が実現できます。

本サンプル・プログラムでは，シリアル・アレイ・ユニット0（SAU0）のチャネル0，1を使用したUART0により，データ長が8ビット，転送レートが9600bps，データ位相が正転出力，パリティ・ビットなし，ストップ・ビットが1ビット付加，データ方向がLSBファースト転送，半2重でのUART送受信が行えるように設定を行います。

（1）シリアル・アレイ・ユニット0（SAU0）の入力クロックの制御

シリアル・アレイ・ユニット0（SAU0）を使用できるように，周辺ハードウエア・マクロの使用を可能にするために周辺イネーブル・レジスタ0（PER0）の設定を行います。使用しないハードウエアへはクロック供給も停止させることで，低消費電力化とノイズ低減をはかることができます。

図4－1－1　周辺イネーブル・レジスタ0（PER0）のフォーマット

アドレス：F00F0H

RTCEN	DACEN	ADCEN	IIC0EN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

シリアル・アレイ・ユニット0（SAU0EN）の入力クロックの制御

0	入力クロック供給停止 ・シリアル・アレイ・ユニット0で使用するSFRへのライト不可 ・シリアル・アレイ・ユニット0リセット状態
1	入力クロック供給 ・シリアル・アレイ・ユニット0で使用するSFRへのリード／ライト可

注意1. **シリアル・アレイ・ユニット**m**の設定をする際には，必ず最初に**SAUmEN = 1**の設定を行ってください。**SAUmEN = 0**の場合は，シリアル・アレイ・ユニット**m**の制御レジスタへの書き込みは無視され，読み出しても値はすべて初期値となります（入力切り替え制御レジスタ（**ISC**），ノイズフィルタ許可レジスタ（**NFEN0**），ポート入力モード・レジスタ（**PIM0**），ポート出力モード・レジスタ（**POM0**），ポート・モード・レジスタ（**PM0, PM1**），ポート・レジスタ（**P0, P1**）は除く）。**

2. PER0**レジスタを“**1**”に設定後に，**4**クロック以上間隔をあけてから**SPSm**レジスタを設定してください。**

備考　m：**ユニット番号（**m = 0, 1**）**

(2) 動作クロック（CK00）の選択

シリアル・アレイ・ユニット0（SAU0）のチャネル0，1で共通して供給される2種類の動作クロック（CK00, CK01）を選択するため，シリアル・クロック選択レジスタ0（SPS0）の設定を行います。SPS0のビット7-4でCK01を，ビット3-0でCK00を選択します。

図4－1－2　シリアル・クロック選択レジスタ0（SPS0）のフォーマット

アドレス：F0126H，F0127H

0	0	0	0	0	0	0	0	PRS013	PRS012	PRS011	PRS010	PRS003	PRS002	PRS001	PRS000

動作クロック（CK00）の選択

0	0	0	0	f_{CLK}
0	0	0	1	$f_{CLK}/2$
0	**0**	**1**	**0**	$f_{CLK}/2^2$
0	0	1	1	$f_{CLK}/2^3$
0	1	0	0	$f_{CLK}/2^4$
0	1	0	1	$f_{CLK}/2^5$
0	1	1	0	$f_{CLK}/2^6$
0	1	1	1	$f_{CLK}/2^7$
1	0	0	0	$f_{CLK}/2^8$
1	0	0	1	$f_{CLK}/2^9$
1	0	1	0	$f_{CLK}/2^{10}$
1	0	1	1	$f_{CLK}/2^{11}$
1	1	1	1	INTTM02
上記以外				設定禁止

動作クロック（CK01）の選択（本サンプル・プログラムでは未使用）

0	**0**	**0**	**0**	f_{CLK}
0	0	0	1	$f_{CLK}/2$
0	0	1	0	$f_{CLK}/2^2$
0	0	1	1	$f_{CLK}/2^3$
0	1	0	0	$f_{CLK}/2^4$
0	1	0	1	$f_{CLK}/2^5$
0	1	1	0	$f_{CLK}/2^6$
0	1	1	1	$f_{CLK}/2^7$
1	0	0	0	$f_{CLK}/2^8$
1	0	0	1	$f_{CLK}/2^9$
1	0	1	0	$f_{CLK}/2^{10}$
1	0	1	1	$f_{CLK}/2^{11}$
1	1	1	1	INTTM02
上記以外				設定禁止

注意1. ビット15-8には，必ず0を設定してください。

2. PER0レジスタを“1”に設定後に，4クロック以上間隔をあけてからSPS0レジスタを設定してください。

備考1. f_{CLK}：CPU／周辺ハードウエア・クロック周波数

(3) 動作モードなどの設定

シリアル・アレイ・ユニット0（SAU0）のUART0で使用する動作クロック（MCK）の選択，シリアル・クロック（SCK）入力の使用可否，スタート・トリガ設定，動作モード（CSI, UART, I²C）設定，割り込み要因の選択をシリアル・モード・レジスタ00（SMR00）およびシリアル・モード・レジスタ01（SMR01）にて行います。UARTモード時のみ，受信データのレベル反転の設定が行えます。シリアル・モード・レジスタ00（SMR00）がUART0送信，シリアル・モード・レジスタ01（SMR01）がUART0受信が可能となるように設定を行います。

図4－1－3　シリアル・モード・レジスタ00（SMR00）のフォーマット

アドレス：F0110H，F0111H

CKC00	CCS00	0	0	0	0	0	STS00	0	SIS000	1	0	0	MD002	MD001	MD000

チャネル0の割り込み要因の選択

0	転送完了割り込み
1	バッファ空き割り込み

チャネル0の動作モードの設定

0	0	CSIモード
0	1	UARTモード
1	0	簡易I²Cモード
1	1	設定禁止

UARTモードでのチャネル0の受信データのレベル反転の制御

0	立ち下がりエッジをスタートビットとして検出します。 入力される通信データは，そのまま取り込まれます。
1	立ち上がりエッジをスタートビットとして検出します。 入力される通信データは，反転して取り込まれます。

スタート・トリガ要因の選択

0	ソフトウエア・トリガのみ有効 （CSI, UART送信，簡易I²C時に選択）
1	RxD端子の有効エッジ（UART受信時に選択）

チャネル0の転送クロック（TCLK）の選択

0	CKS00ビットで指定した動作クロックMCKの分周クロック
1	SCK端子からの入力クロック（CSIモードのスレーブ転送）

チャネル0の動作クロック（MCK）の選択

0	PRSレジスタで設定したプリスケーラ出力クロックCK00
1	PRSレジスタで設定したプリスケーラ出力クロックCK01

注意　ビット13-9, 7, 4, 3には，必ず0を設定してください。ビット5には，必ず1を設定してください。

図4－1－4　シリアル・モード・レジスタ01（SMR01）のフォーマット

アドレス：F0112H，F0113H

CKC01	CCS01	0	0	0	0	0	STS01	0	SIS010	1	0	0	MD012	MD011	MD010

チャネル1の割り込み要因の選択

0	**転送完了割り込み**
1	**バッファ空き割り込み**

チャネル1の動作モードの設定

0	0	**CSIモード**
0	**1**	**UARTモード**
1	0	**簡易I^2Cモード**
1	1	**設定禁止**

UARTモードでのチャネル1の受信データのレベル反転の制御

0	**立ち下がりエッジをスタートビットとして検出します。 入力される通信データは，そのまま取り込まれます。**
1	**立ち上がりエッジをスタートビットとして検出します。 入力される通信データは，反転して取り込まれます。**

スタート・トリガ要因の選択

0	**ソフトウエア・トリガのみ有効 （CSI, UART送信，簡易I^2C時に選択）**
1	**RxD端子の有効エッジ（UART受信時に選択）**

チャネル1の転送クロック（TCLK）の選択

0	**CKS01ビットで指定した動作クロックMCKの分周クロック**
1	**SCK端子からの入力クロック（CSIモードのスレーブ転送）**

チャネル1の動作クロック（MCK）の選択

0	**PRSレジスタで設定したプリスケーラ出力クロックCK00**
1	**PRSレジスタで設定したプリスケーラ出力クロックCK01**

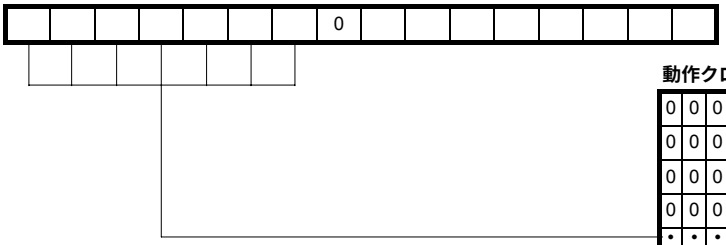
注意　ビット13-9, 7, 4, 3には，必ず0を設定してください。ビット5には，必ず1を設定してください。

(4) 動作クロックの分周設定

シリアル・アレイ・ユニット0（SAU0）のUART0で使用する動作クロック（MCK）の分周設定は，シリアル・データ・レジスタ00（SDR00）およびシリアル・データ・レジスタ01（SDR01）の16ビットのレジスタの上位7ビットで行います。本サンプル・プログラムでは，キャリブレーション終了後にUART0送受信の転送ボー・レート用に計算し，設定を行います。なお，シリアル・データ・レジスタ00（SDR00）の下位8ビットは送信データ・レジスタ（TxD0），シリアル・データ・レジスタ01（SDR01）の下位8ビットは受信データ・レジスタ（RxD0）となります。

図4－1－5　シリアル・データ・レジスタ00，01（SDR00，SDR01）のフォーマット

アドレス：FFF10H，FFF11H（SDR00），FFF12H，FFF13H（SDR01）



動作クロック（MCK）の分周設定（SDR00[15:9]，SDR01[15:9]）

0	0	0	0	0	0	0	UART使用時は設定禁止
0	0	0	0	0	0	1	UART使用時は設定禁止
0	0	0	0	0	1	0	MCK/6
0	0	0	0	0	1	1	MCK/8
・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・	・ ・ ・
1	1	1	1	1	1	0	MCK/254
1	1	1	1	1	1	1	MCK/256

注意　ビット8は，必ず0を設定してください。

(5) 通信動作の設定

通信動作の設定は，シリアル通信動作設定レジスタ00（SCR00）およびシリアル通信動作設定レジスタ01（SCR01）にて行います。データ送受信モード，データとクロックの位相，エラー信号のマスク可否，パリティ・ビット，先頭ビット，ストップ・ビット，データ長，などの設定を行います。シリアル通信動作設定レジスタ00（SCR00）にてUART0送信，シリアル通信動作設定レジスタ01（SCR01）にてUART0受信が可能となるように設定を行います。

図4－1－6　シリアル通信動作設定レジスタ00（SCR00）のフォーマット

アドレス：F0118H，F0119H

TXE00	RXE00	DAP00	CKP00	0	EOC00	PTC001	PTC000	DIR00	0	SLC001	SLC000	0	DLS002	DLS001	DLS000

UARTモードでのデータ長の設定

1	0	0	5ビット・データ長（SDR00レジスタのビット0-4に格納）
1	1	0	7ビット・データ長（SDR00レジスタのビット0-6に格納）
1	1	1	8ビット・データ長（SDR00レジスタのビット0-7に格納）
その他			設定禁止

UARTモードでのストップ・ビットの設定

0	0	ストップ・ビットなし
0	1	ストップ・ビット長 = 1ビット
1	0	ストップ・ビット長 = 2ビット
1	1	設定禁止

UARTモードでのデータ転送順序の選択

0	MSBファーストで入出力を行う
1	LSBファーストで入出力を行う

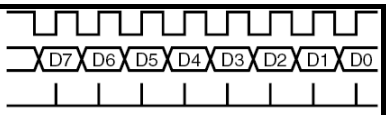
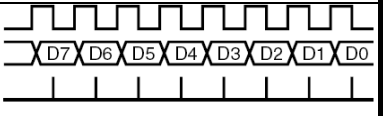
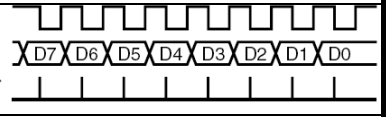
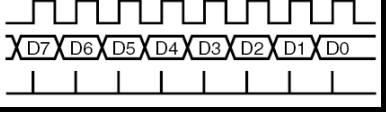
UARTモードでのストップ・ビットの設定

		送信動作	受信動作
0	0	パリティ・ビットを出力しない	パリティなしで受信
0	1	0パリティを出力	パリティ判定を行わない
1	0	偶数パリティを出力	偶数パリティとして判定を行う
1	1	奇数パリティを出力	奇数パリティとして判定を行う

エラー割り込み信号（INTSREx（x = 0-3））のマスク可否の選択

0	エラー割り込みINTSRExをマスクする （INTSRxはマスクされない）
1	エラー割り込みINTSRExの発生を許可する （エラー発生時にINTSRxはマスクされる）

CSIモードでのデータとクロックの位相選択

0	0		SCKp SOp SIp入力タイミング
0	1		SCKp SOp SIp入力タイミング
1	0		SCKp SOp SIp入力タイミング
1	1		SCKp SOp SIp入力タイミング

チャネル0の動作モードの設定

0	0	通信を開始しない
0	1	受信のみ
1	0	送信のみ
1	1	送受信を行う

注意　**ビット**11, 6, 3**には，必ず**0**を設定してください。ビット**2**には，必ず**1**を設定してください。**

備考　p：CSI**番号（**p = 00, 01, 10, 11, 20, 21**）**

図4－1－7　シリアル通信動作設定レジスタ01（SCR01）のフォーマット

アドレス：F011AH，F011BH

TXE01	RXE01	DAP01	CKP01	0	EOC01	PTC011	PTC010	DIR01	0	SLC011	SLC010	0	DLS012	DLS011	DLS010

UARTモードでのデータ長の設定

1	0	0	5ビット・データ長（SDR01レジスタのビット0-4に格納）
1	1	0	7ビット・データ長（SDR01レジスタのビット0-6に格納）
1	1	1	8ビット・データ長（SDR01レジスタのビット0-7に格納）
その他			設定禁止

UARTモードでのストップ・ビットの設定

0	0	ストップ・ビットなし
0	1	ストップ・ビット長 = 1ビット
1	0	ストップ・ビット長 = 2ビット
1	1	設定禁止

UARTモードでのデータ転送順序の選択

0	MSBファーストで入出力を行う
1	LSBファーストで入出力を行う

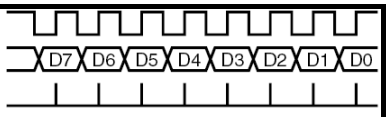
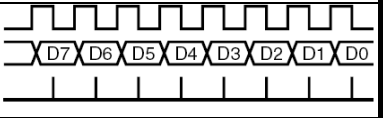
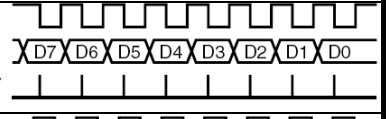
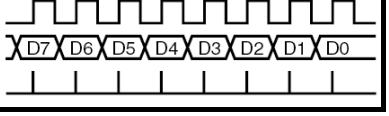
UARTモードでのストップ・ビットの設定

		送信動作	受信動作
0	0	パリティ・ビットを出力しない	パリティなしで受信
0	1	0パリティを出力	パリティ判定を行わない
1	0	偶数パリティを出力	偶数パリティとして判定を行う
1	1	奇数パリティを出力	奇数パリティとして判定を行う

エラー割り込み信号（INTSREx（x = 0-3））のマスク可否の選択

0	エラー割り込みINTSRExをマスクする （INTSRxはマスクされない）
1	エラー割り込みINTSRExの発生を許可する （エラー発生時にINTSRxはマスクされる）

CSIモードでのデータとクロックの位相選択

0	0	SCKp SOp SIp入力タイミング	
0	1	SCKp SOp SIp入力タイミング	
1	0	SCKp SOp SIp入力タイミング	
1	1	SCKp SOp SIp入力タイミング	

チャネル1の動作モードの設定

0	0	通信を開始しない
0	1	受信のみ
1	0	送信のみ
1	1	送受信を行う

注意　**ビット**11, 6, 3**には，必ず**0**を設定してください。ビット**2**には，必ず**1**を設定してください。**

備考　p：CSI**番号（**p = 00, 01, 10, 11, 20, 21**）**

(6) 初期出力レベルの設定

シリアル出力レジスタ0（SO0）にて，シリアル・アレイ・ユニット0（SAU0）のUART0送信での初期出力レベルの設定を行います。

本サンプル・プログラムでは，使用する周辺の初期設定にてシリアル・アレイ・ユニット0（SAU0）のUART0送信用にSO00ビットを1にします。

図4－1－8　シリアル出力レジスタ0（SO0）のフォーマット

アドレス：F0128H，F0129H

0	0	0	0	CKO03	CKO02	CKO01	CKO00	0	0	0	0	SO03	SO02	SO01	SO00

チャネルnのシリアル・クロック出力

0	シリアル・クロック出力値が“0”
1	シリアル・クロック出力値が“1”

チャネルnのシリアル・データ出力

0	シリアル・データ出力値が“0”
1	シリアル・データ出力値が“1”

注意　SO0のビット15-12, 7-4には，必ず0を設定してください。CKO03，SO03は，K0R/KJ3，K0R/KH3マイクロコントローラにのみ存在します。CKO01，SO01は，K0R/KJ3，K0R/KH3，K0R/KG3，K0R/KF3マイクロコントローラにのみ存在します。存在しないビットは，必ず1を設定してください。

備考　n：チャネル番号（n = 0-3）

(7) シリアル通信動作の出力許可／停止の分周設定

シリアル・アレイ・ユニット0（SAU0）のUART0で使用するシリアル通信動作の出力許可／停止は，シリアル出力許可レジスタ0（SOE0）の設定で行います。シリアル出力を許可したチャネルは，通信動作によって反映された値がシリアル・データ出力端子から出力されます。

本サンプル・プログラムでは，使用する周辺の初期設定にてシリアル・アレイ・ユニット0（SAU0）のUART0送信動作の出力許可時にSOE00ビットを1にします。

図4－1－9　シリアル出力許可レジスタ0（SOE0）のフォーマット

アドレス：F012AH，F012BH（SOE0）

0	0	0	0	0	0	0	0	0	0	0	0	SOE03	SOE02	SOE01	SOE00

チャネルnのシリアル出力許可／停止

0	シリアル通信動作による出力停止
1	シリアル通信動作による出力許可

注意　SOE0のビット15-4には，必ず0を設定してください。SOE03は，K0R/KJ3，K0R/KH3マイクロコントローラにのみ存在します。SOE01は，K0R/KJ3，K0R/KH3，K0R/KG3，K0R/KF3マイクロコントローラにのみ存在します。存在しないビットは，必ず0を設定してください。

備考　n：チャネル番号（n = 0-3）

(8) シリアル通信／カウント開始の許可

シリアル・アレイ・ユニット0（SAU0）のUART0で使用するシリアルの通信／カウント開始の許可はチャネルごとに設定するシリアル・チャネル開始レジスタ0（SS0）で行います。シリアル・チャネル開始レジスタ0（SS0）は，トリガ・レジスタですので，シリアル送受信の動作が許可状態になるとクリアされます。

本サンプル・プログラムでは，シリアル・アレイ・ユニット0（SAU0）のUART0送信動作開始時にSS00ビットを1，UART0受信動作開始時にSS01ビットを1にします。

図4－1－10　シリアル・チャネル開始レジスタ0（SS0）のフォーマット

アドレス：F0122H，F0123H（SS0）

0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

チャネルnの動作開始トリガ

0	トリガ動作せず
1	通信待機状態に遷移する （すでに通信動作中の場合は，通信動作を停止し，起動条件待ちの状態となる）

注意　SS0のビット15-4には，必ず0を設定してください。

備考　n：チャネル番号（n = 0-3）

(9) シリアル通信／カウント停止の許可

通信／カウント停止の許可はチャネルごとに設定するシリアル・チャネル停止レジスタ0（ST0）で行います。シリアル・チャネル停止レジスタ0（ST0）は，トリガ・レジスタですので，シリアル送受信の動作が停止状態になるとクリアされます。

本サンプル・プログラムでは，シリアル・アレイ・ユニット0（SAU0）のUART0送信動作停止時にST00ビットを1，UART0受信動作停止時にST01ビットを1にします。

図4－1－11　シリアル・チャネル停止レジスタ0（ST0）のフォーマット

アドレス：F0124H，F0125H（ST0）

0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00

チャネル0の動作停止トリガ

0	トリガ動作せず
1	通信動作を停止する （制御レジスタ，シフト・レジスタの値，およびシリアル・クロック入出力端子，シリアル・データ出力端子，FEF, PEF, OVFの各エラー・フラグの状態を保持したまま停止）

注意　ST0のビット15-4には，必ず0を設定してください。

備考　n：チャネル番号（n = 0-3）

【　本サンプル・プログラム・ソースからの抜粋　】

付録A　プログラム・リストから、シリアル・アレイ・ユニットの設定に関する抜粋部分を示します。

（1）アセンブリ言語

```
;-------------------------------------------------------------------------------
;       UART0の設定
;-------------------------------------------------------------------------------
        SET1    !SAU0EN                 ;UART0のあるSAU0使用許可

        NOP                             ;4クロック分のウェイト
        NOP                             ;（詳細はユーザーズ・マニュアルを参照してください。）
        NOP
        NOP

        MOV     !SPS0L,  #00000010B     ;プリスケーラ（動作クロック）の設定
                         ;||||++++---------- PRS003-000(CK00): fCLK/2^2
                         ;++++-------------- PRS013-010(CK01): 未使用

        ;---------------------------------------
        ; UART0送信設定（SAU0のチャネル0使用）
        ;---------------------------------------
        MOV     RTXBUFP,#0              ;送信データ・カウンタ初期化

        MOVW    AX,      #0000000000100010B;動作モードなどの設定
                         ;||||||||||||||+- MD000: 転送完了割り込み
                         ;||||||||||||++-- MD002-001: UARTモード
                         ;||||||||||++----- <100>
                         ;||||||||+-------- SIS000: 立ち下がりエッジをスタートビットとして検出
                         ;||||||+---------- <0>
                         ;|||||+----------- STS00: UART送信
                         ;||+++++---------- <00000>
                         ;|+---------------- CCS00: CKS00ビットで指定した動作クロックMCK
                         ;+----------------- CKS00: PRSで選択した動作クロックCK00(MCK)
        MOVW    !SMR00,  AX

        MOVW    AX,      #1000000010010111B;通信フォーマットの設定
                         ;||||||||||||+++- DLS002-000: 8ビット・データ長
                         ;|||||||||||+----- <0>
                         ;|||||||||++------ SLC001-000: ストップ・ビット長 = 1ビット
                         ;||||||||+-------- <0>
                         ;|||||||+--------- DIR00: LSBファーストで出力
                         ;|||||++---------- PTC001-000: パリティ・ビットを出力しない
                         ;||||+------------ EOC00: UART送信時はエラー割り込み未使用
                         ;|||+------------- <0>
                         ;||++-------------- DAP00/CKP00: UARTモードでは未使用
                         ;++---------------- TXE00/RXE00: 送信のみを行う
        MOVW    !SCR00,  AX

        ; SDR00 転送ボー・レートの設定はINTRTC割り込み処理にて設定

        MOV     SOL0L,   #00000000B     ;出力データのレベルの設定（反転なし）

        MOVW    AX,      #0000000000000001B;初期出力レベルの設定
                         ;||||||||||++++- SO03-0: チャネルnのシリアル・クロック出力
                         ;||||||||++++----- <0000>
                         ;||||++++---------- CKO03-0: チャネルnのシリアル・データ出力
                         ;++++-------------- <0000>
        MOVW    !SO0,    AX
        SET1    !SOE0L.0                ;SOE00:シリアル出力許可

        SET1    P1.2                    ;P12=TxD0

        ;---------------------------------------
        ; UART0受信設定（SAU0のチャネル1使用）
        ;---------------------------------------
        MOV     RRXBUFP,#0              ;受信データ・カウンタ初期化

        SET1    PM1.1                   ;P11=RxD0


        MOVW    AX,      #0000000100100010B;動作モードなどの設定
                         ;||||||||||||||+-- MD010: 転送完了割り込み
                         ;||||||||||||++--- MD012-011: UARTモード
                         ;||||||||||++------ <100>
                         ;||||||||+--------- SIS010: 立ち下がりエッジをスタートビットとして検出
                         ;||||||+---------- <0>
                         ;|||||+----------- STS01: RxD端子の有効エッジ（UART受信設定）
                         ;||+++++---------- <00000>
                         ;|+---------------- CCS01: CKS00ビットで指定した動作クロックMCK
                         ;+----------------- CKS01: PRSで選択した動作クロックCK00(MCK)
        MOVW    !SMR01,  AX
```

クロック供給開始

4クロックウエイト

ボー・レートは計算後設定

初期出力をHiに設定

ポート設定

チャネル1をUART受信に設定

```
        MOVW    AX,     #0100010010010111B;通信フォーマットの設定
                        ;|||||||||||+++- DLS012-010: 8ビット・データ長
                        ;|||||||||||+---- <0>
                        ;|||||||||++----- SLC011-010: ストップ・ビット長 = 1ビット
                        ;|||||||+-------- <0>
                        ;||||||+--------- DIR01: LSBファーストで入力を行う
                        ;||||++---------- PTC011-010: パリティなしで受信
                        ;||||+----------- EOC01: UART受信のエラー割り込み許可
                        ;|||+------------ <0>
                        ;||++------------- DAP01/CKP01: UARTモードでは未使用
                        ;++--------------- TXE01/RXE01: 受信のみを行う
        MOVW    !SCR01, AX

        ; SDR01 転送ボー・レートの設定はINTRTC割り込み処理にて設定


        EI                                  ;割り込み許可


;---------------------------------------------------------------------------
;       メイン処理
;---------------------------------------------------------------------------

        ;送信データ設定
LMAINRX700:
        MOV     RTXBUF,   A                 ;送信データを送信バッファに設定
        XCH     A,        X
        MOV     RTXBUF+1,A

        ;UART0送信動作開始
        CLR1    STIF0                       ;割り込み要求クリア
        CLR1    STMK0                       ;割り込み許可
        SET1    !SS0L.0                     ;SS00:UART0送信動作開始

        MOV     A,        RTXBUF            ;設定した送信データの第一バイト
        MOV     TxD0,     A                 ;データを送信データ・レジスタに設定
        BR      LMAINRET
```

マスク解除

TxD0送信

（2）C言語

```
/*-------------------------------------------------------------------------------
        UART0の設定
-------------------------------------------------------------------------------*/
        SAU0EN = 1;                 /* UART0のあるSAU0使用許可 */

        NOP();                      /* 4クロック分のウェイト */
        NOP();                      /* （詳細はユーザーズ・マニュアルを参照してください。） */
        NOP();
        NOP();

        SPS0L =  0b00000010;        /* プリスケーラ(動作クロック)の設定 */
                 /*||||++++---------- PRS003-000(CK00): fCLK/2^2 */
                 /*++++-------------- PRS013-010(CK01): 未使用 */

    /*--------------------------------------
          送信設定(チャネル0使用)
    --------------------------------------*/
        ucTxBufferCounter = 0;      /* 送信データ・カウンタのクリア */

        SMR00 =  0b0000000000100010;/* 動作モードなどの設定 */
                 /*||||||||||||||++--- MD000: 転送完了割り込み */
                 /*||||||||||||||||++--- MD002-001: UARTモード */
                 /*||||||||||+++------ <100> */
                 /*||||||||+---------- SIS000: 立ち下がりエッジをスタートビットとして検出 */
                 /*||||||+------------ <0> */
                 /*||||||+------------ STS00: UART送信 */
                 /*|||+++------------- <000> */
                 /*||+---------------- CCS00: CKS00ビットで指定した動作クロックMCK */
                 /*|++----------------- <00> */
                 /*+------------------- CKS00: PRSで選択した動作クロックCK00(MCK) */

        SCR00 =  0b1000000010010111;/* 通信フォーマットの設定 */
                 /*||||||||||||+++-- DLS002-000: 8ビット・データ長 */
                 /*||||||||||||+----- <0> */
                 /*||||||||||++------ SLC001-000: ストップ・ビット長 = 1ビット */
                 /*||||||||+---------- <0> */
                 /*||||||+------------ DIR00: LSBファーストで出力 */
                 /*||||++------------- PTC001-000: パリティ・ビットを出力しない */
                 /*||||+--------------- EOC00: UART送信時はエラー割り込み未使用 */
                 /*||++---------------- <0> */
                 /*||++---------------- DAP00/CKP00: UARTモードでは未使用 */
                 /*++------------------ TXE00/RXE00: 送信のみを行う */

        /* SDR00 転送ボー・レートの設定はINTRTC割り込み処理にて設定 */

        SOL0L = 0b00000000;         /* 出力データのレベルの設定(反転なし) */

        SO0   =  0b0000000000000001;/* 初期出力レベルの設定 */
                 /*||||||||||||++++-- SO03-0: チャネルnのシリアル・クロック出力 */
                 /*||||||||++++------ <0000> */
                 /*||||++++---------- CKO03-0: チャネルnのシリアル・データ出力 */
                 /*++++---------------- <0000> */

        SOE0L.0 = 1;                /* SOE00:シリアル出力許可 */

        P1.2  = 1;                  /* P12=TxD0 */

        /*--------------------------------------
          UART0受信設定(SAU0のチャネル1使用)
        --------------------------------------*/
        ucRxBufferCounter = 0;      /* 受信データ・カウンタのクリア */

        PM1.1 = 1;                  /* P11=RxD0 */

        SMR01 =  0b0000000100100010;/* 動作モード設定 */
                 /*||||||||||||||++--- MD010: 転送完了割り込み */
                 /*||||||||||||||++--- MD012-011: UARTモード */
                 /*||||||||||+++------ <100> */
                 /*||||||||+---------- SIS010: 立ち下がりエッジをスタートビットとして検出 */
                 /*||||||+------------ <0> */
                 /*||||||+------------ STS01: RxD端子の有効エッジ(UART受信設定) */
                 /*||+++++------------ <00000> */
                 /*|+----------------- CCS01: CKS00ビットで指定した動作クロックMCK */
                 /*+------------------ CKS01: PRSで選択した動作クロックCK00(MCK) */

        SCR01 =  0b0100010010010111;/* 通信フォーマットの設定 */
                 /*||||||||||||+++-- DLS012-010: 8ビット・データ長 */
                 /*||||||||||||+----- <0> */
                 /*||||||||||++------ SLC011-010: ストップ・ビット長 = 1ビット */
                 /*||||||||+---------- <0> */
                 /*||||||||+---------- DIR01: LSBファーストで入力を行う */
                 /*||||||++---------- PTC011-010: パリティなしで受信 */
                 /*||||+------------- EOC01: UART受信のエラー割り込み許可 */
                 /*|||+-------------- <0> */
                 /*||++---------------- DAP01/CKP01: UARTモードでは未使用 */
                 /*++------------------ TXE01/RXE01: 受信のみを行う */
```

```
        /* SDR01 転送ボー・レートの設定はINTRTC割り込み処理にて設定 */

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

/*--------------------------------------------------------------------------------
    メイン処理
--------------------------------------------------------------------------------*/

                  /* 受信データのチェック */
                if((ucRxBuffer[0]==10||ucRxBuffer[0]==8)&&ucRxBuffer[1]<=7){
                /* 受信したデータが正常な場合 */

                    /* A/D値の取得 */
                    ADS = ucRxBuffer[1];    /* A/D変換するチャネルの設定 */
                    ADCE = 1;       /* コンパレータの動作許可(変換待機) */

                    /* 1変換目(不正データ) */
                    /* 1us以上ウエイトしないでADCSに1を設定した場合は，
                       最初の変換データを無視します。
                       (詳細はユーザーズ・マニュアルを参照してください。) */
                    ADIF = 0;       /* 割り込み要求クリア */
                    ADCS = 1;       /* A/D変換動作許可 */
                    while(!ADIF){   /* 変換待ち */
                        NOP();
                    }
                    /* 2変換目(正規データ) */
                    ADIF = 0;       /* 割り込み要求クリア */
                    ADCS = 1;       /* 動作許可 */
                    while(!ADIF){   /* 変換待ち */
                        NOP();
                    }

                    ADCS = 0;       /* A/D変換動作停止 */
                    ADCE = 0;       /* コンパレータの動作停止 */

                    /* 送信データ設定 */
                    if(ucRxBuffer[0]==8){
                    /* 8ビット分解能 */
                        ucTxBuffer[0] = 0;
                        ucTxBuffer[1] = ADCRH;
                    }
                    else{
                    /* 10ビット分解能 */
                        ucTxBuffer[0] = (unsigned char)(ADCR >>14);
                        ucTxBuffer[1] = (unsigned char)(ADCR >>6);
                        /*データのシフト(10ビットA/D変換結果を下位ビット詰めを行っています。*/
                    }
                }
                else{
                /* 受信したデータが不正な場合 */
                    ucTxBuffer[0] =0x80;    /* エラー通知設定 */
                    ucTxBuffer[1] =0x00;
                }
                /* UART0送信動作開始 */
                STIF0  = 0;         /* 割り込み要求クリア */
                STMK0  = 0;         /* 割り込み許可 */
                SS0L.0 = 1;         /* SS00:UART0送信動作開始 */

                TxD0 = ucTxBuffer[0];    /* 送信データの第一バイトのデータを
                                            送信データ・レジスタに設定 */
            }
        }
        else if(ucTxBufferCounter >= sizeof(ucTxBuffer)){
        /* UART0送信完了 */

            ucTxBufferCounter = 0;        /* 送信データ・カウンタのクリア */

            /* UART0受信動作開始 */
            SRIF0  = 0;             /* 割り込み要求クリア */
            SRMK0  = 0;             /* 割り込み許可 */
            SS0L.1 = 1;             /* SS01:UART0受信動作開始 */
        }
    }
}
```

マスク解除

TxD0送信

受信動作許可

4.2　使用する周辺の初期設定

アセンブリ言語の使用する周辺の初期設定では，次の動作を行います。

① 割り込みを禁止します。
② レジスタバンクの設定を行います。
③ スタック・ポインタの設定を行います。
④ メイン・システム・クロックの設定／サブシステム・クロックの発振開始等のクロック設定を行います。
⑤ ポートの設定を行います。
⑥ A/Dコンバータの設定を行います。
⑦ シリアル・アレイ・ユニット0（SAU0）のチャネル0のUART0送信の設定を行います。
⑧ シリアル・アレイ・ユニット0（SAU0）のチャネル1のUART0受信の設定を行います。
⑨ 割り込みを許可します。
⑩ サブシステム・クロックの発振安定待ちを行います。リアルタイム・カウンタ（RTC）を0.5sに一度の定周期割り込みに設定
⑪ キャリブレーションの開始処理を呼び出します。
⑫ メイン処理へ続きます。

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

①       DI                                  ;割り込み禁止
;-------------------------------------------------------------------------------
;       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,     #LOWW STACKTOP  ;スタック・ポインタを設定

;-------------------------------------------------------------------------------
;       クロック周波数の設定
;-------------------------------------------------------------------------------
;       高速内蔵発振クロックで動作が行えるように設定します。
;-------------------------------------------------------------------------------
        MOV     CMC,    #00010000B      ;クロック動作モード
                        ;|||||||+--------- AMPH: 2MHz≦fMX≦10MHz
                        ;||||+++---------- <000>
                        ;|||+------------- OSCSELS: XT1発振モード
                        ;||+-------------- <0>
                        ;++---------------- EXCLK/OSCSEL: 入力ポート・モード

        MOV     CSC,    #10000000B      ;クロック動作ステータス制御
                        ;|||||||+--------- HIOSTOP: 高速内蔵発振回路動作
                        ;||+++++---------- <00000>
                        ;|+---------------- XTSTOP: XT1発振回路動作
④                       ;+----------------- MSTOP: X1発振回路停止

        MOV     OSMC,   #00000000B      ;動作スピード・モード
                        ;|||||||+--------- FSEL: 10MHz以下の周波数で動作
                        ;+++++++---------- <00000>

        MOV     CKC,    #00001000B      ;クロック選択
                        ;|||||+++--------- MDIV2-0: CPU/周辺ハードウエア・クロック(fCLK)=fIH
                        ;||||+------------ <1>
                        ;|||+------------- MCM0: 高速内蔵発振クロック(fIH)
                        ;||+-------------- <R>
                        ;|+---------------- CSS: メイン・システム・クロック(fMAIN)=fCLK
                        ;+----------------- <R>

;-------------------------------------------------------------------------------
;       ポート0の設定
;-------------------------------------------------------------------------------
        MOV     P0,     #00000000B      ;P00-P07の出力ラッチLow
⑤       MOV     PM0,    #00000000B      ;P00-P07を出力ポートに設定
```

```
;-------------------------------------------------------------------------------
;       ポート1の設定
;-------------------------------------------------------------------------------
        MOV     P1,     #00000000B      ;P10-P17の出力ラッチLow
                                        ;[後でP12(TxD0)の設定を行う]
        MOV     PM1,    #00000000B      ;P10-P17を出力ポートに設定
                                        ;[後でP11(RxD0)の設定を行う]

;-------------------------------------------------------------------------------
;       ポート2の設定
;-------------------------------------------------------------------------------
        MOV     P2,     #00000000B      ;P20-P27の出力ラッチLow
        MOV     PM2,    #11111111B      ;P20-P27(ANI0-ANI7)を入力ポートに設定

;-------------------------------------------------------------------------------
;       ポート3の設定
;-------------------------------------------------------------------------------
        MOV     P3,     #00000000B      ;P30-P37の出力ラッチLow
        MOV     PM3,    #00000000B      ;P30-P37を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート4の設定
;-------------------------------------------------------------------------------
        MOV     P4,     #00000000B      ;P40-P47の出力ラッチLow
        MOV     PM4,    #00000000B      ;P40-P47を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート5の設定
;-------------------------------------------------------------------------------
        MOV     P5,     #00000000B      ;P50-P57の出力ラッチLow
        MOV     PM5,    #00000000B      ;P50-P57を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート6の設定
;-------------------------------------------------------------------------------
        MOV     P6,     #00000000B      ;P60-P67の出力ラッチLow
        MOV     PM6,    #00000000B      ;P60-P67を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート7の設定
;-------------------------------------------------------------------------------
        MOV     P7,     #00000000B      ;P70-P77の出力ラッチLow
        MOV     PM7,    #00000000B      ;P70-P77を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート8の設定
;-------------------------------------------------------------------------------
        MOV     P8,     #00000000B      ;P80-P87の出力ラッチLow
        MOV     PM8,    #00000000B      ;P80-P87を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート9の設定
;-------------------------------------------------------------------------------
        MOV     P9,     #00000000B      ;P90-P97の出力ラッチLow
        MOV     PM9,    #00000000B      ;P90-P97を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート10の設定
;-------------------------------------------------------------------------------
        MOV     P10,    #00000000B      ;P100-P107の出力ラッチLow
        MOV     PM10,   #00000000B      ;P100-P107を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート11の設定
;-------------------------------------------------------------------------------
        MOV     P11,    #00000000B      ;P110-P117の出力ラッチLow
        MOV     PM11,   #00000000B      ;P110-P117を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート12の設定
;-------------------------------------------------------------------------------
        MOV     P12,    #00000000B      ;P120,P125-P127の出力ラッチLow
        MOV     PM12,   #00011110B      ;P120,P125-P127を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート13の設定
;-------------------------------------------------------------------------------
        MOV     P13,    #00000000B      ;P130-P137の出力ラッチLow
        MOV     PM13,   #00000000B      ;P130-P137を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート14の設定
;-------------------------------------------------------------------------------
        MOV     P14,    #00000000B      ;P140-P147の出力ラッチLow
        MOV     PM14,   #00000000B      ;P140-P147を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート15の設定
;-------------------------------------------------------------------------------
        MOV     P15,    #00000000B      ;P150-P157の出力ラッチLow
        MOV     PM15,   #00000000B      ;P150-P157を出力ポートに設定
```

```
;-------------------------------------------------------------------------------
;       ポート16の設定
;-------------------------------------------------------------------------------
        MOV     P16,    #00000000B      ;P160-P163の出力ラッチLow
        MOV     PM16,   #11110000B      ;P160-P163を出力ポートに設定

;-------------------------------------------------------------------------------
;       A/Dコンバータの設定
;-------------------------------------------------------------------------------
        SET1    !ADCEN                  ;ADコンバータ使用の許可

        MOV     ADPC,   #00000000B      ;A/Dポート・コンフィギュレーション
                        ;|||+++++--------- ADPC4-0: 使用するP20-P27を
                        ;|||                         すべてアナログ入力に切り替え
                        ;+++-------------- <000>
⑥       MOV     ADM,    #00100000B      ;変換時間の選択
                        ;|||||||+--------- ADCE: コンパレータの動作停止
                        ;|+++++----------- FR2-0,LV1-0: fCLK=8MHzで
                        ;|                            最速の変換時間(8.25us)を選択
                        ;|+---------------------- <0>
                        ;+----------------------- ADCS: 変換動作停止

;-------------------------------------------------------------------------------
;       UART0の設定
;-------------------------------------------------------------------------------
        SET1    !SAU0EN                 ;UART0のあるSAU0使用許可

        NOP                             ;4クロック分のウェイト
        NOP                             ;(詳細はユーザーズ・マニュアルを参照してください。)
        NOP
        NOP

        MOV     !SPS0L, #00000010B      ;プリスケーラ(動作クロック)の設定
                        ;||||++++--------- PRS003-000(CK00): fCLK/2^2
                        ;++++-------------- PRS013-010(CK01): 未使用

        ;---------------------------------------
        ; UART0送信設定(SAU0のチャネル0使用)
        ;---------------------------------------
        MOV     RTXBUFP,#0              ;送信データ・カウンタ初期化

        MOVW    AX,     #0000000000100010B;動作モードなどの設定
                        ;|||||||||||||||+- MD000: 転送完了割り込み
                        ;|||||||||||||++-- MD002-001: UARTモード
                        ;|||||||||++++----- <100>
                        ;|||||||+-------- SIS000: 立ち下がりエッジをスタートビットとして検出
                        ;|||||||+--------- <0>
                        ;||||||+---------- STS00: UART送信
                        ;|+++++----------- <00000>
                        ;+----------------- CCS00: CKS00ビットで指定した動作クロックMCK
                        ;+------------------ CKS00: PRSで選択した動作クロックCK00(MCK)
⑦       MOVW    !SMR00, AX

        MOVW    AX,     #1000000010010111B;通信フォーマットの設定
                        ;||||||||||||+++- DLS002-000: 8ビット・データ長
                        ;|||||||||||+----- <0>
                        ;|||||||||++------ SLC001-000: ストップ・ビット長 = 1ビット
                        ;|||||||||+-------- <0>
                        ;||||||||+--------- DIR00: LSBファーストで出力
                        ;||||||++---------- PTC001-000: パリティ・ビットを出力しない
                        ;||||+------------ EOC00: UART送信時はエラー割り込み未使用
                        ;|||+------------- <0>
                        ;||++-------------- DAP00/CKP00: UARTモードでは未使用
                        ;++---------------- TXE00/RXE00: 送信のみを行う
        MOVW    !SCR00, AX

        ; SDR00 転送ボー・レートの設定はINTRTC割り込み処理にて設定

        MOV     SOL0L,  #00000000B      ;出力データのレベルの設定(反転なし)

        MOVW    AX,     #0000000000000001B;初期出力レベルの設定
                        ;||||||||||||++++- SO03-0: チャネルnのシリアル・クロック出力
                        ;||||||||++++------ <0000>
                        ;||||++++---------- CKO03-0: チャネルnのシリアル・データ出力
                        ;++++--------------- <0000>
        MOVW    !SO0,   AX
        SET1    !SOE0L.0                ;SOE00:シリアル出力許可

        SET1    P1.2                    ;P12=TxD0

        ;---------------------------------------
        ; UART0受信設定(SAU0のチャネル1使用)
        ;---------------------------------------
        MOV     RRXBUFP,#0              ;受信データ・カウンタ初期化

⑧       SET1    PM1.1                   ;P11=RxD0
```

```
        MOVW    AX,     #0000000100100010B;動作モードなどの設定
                        ;||||||||||||||||+-- MD010: 転送完了割り込み
                        ;||||||||||||||++--- MD012-011: UARTモード
                        ;|||||||||||+++----- <100>
                        ;||||||||||+-------- SIS010: 立ち下がりエッジをスタートビットとして検出
                        ;|||||||+----------- <0>
                        ;||||||+------------ STS01: RxD端子の有効エッジ(UART受信設定)
                        ;||+++++------------ <00000>
                        ;|+------------------ CCS01: CKS00ビットで指定した動作クロックMCK
                        ;+------------------- CKS01: PRSで選択した動作クロックCK00(MCK)
        MOVW    !SMR01, AX

        MOVW    AX,     #0100010010010111B;通信フォーマットの設定
                        ;||||||||||||+++- DLS012-010: 8ビット・データ長
                        ;|||||||||||+---- <0>
                        ;||||||||||++----- SLC011-010: ストップ・ビット長 = 1ビット
                        ;|||||||||+------- <0>
                        ;||||||||+-------- DIR01: LSBファーストで入力を行う
                        ;|||||++---------- PTC011-010: パリティなしで受信
                        ;||||+------------ EOC01: UART受信のエラー割り込み許可
                        ;|||+------------- <0>
                        ;||++-------------- DAP01/CKP01: UARTモードでは未使用
                        ;++---------------- TXE01/RXE01: 受信のみを行う
        MOVW    !SCR01, AX

        ; SDR01 転送ボー・レートの設定はINTRTC割り込み処理にて設定

⑨       EI                              ;割り込み許可

;---------------------------------------------------------------------------
;       サブシステム・クロックの発振安定待ち(本例は約3秒ですが，実装回路上で評価したうえで，
;       発振安定まで待ってください)。
;---------------------------------------------------------------------------

        MOV     C, #0H
WAIT1:
        MOVW    AX, #0H
WAIT:
        INCW    AX
        CMPW    AX, #0FFFFH
        BC      $WAIT

        INC     C
        MOV     A,C
        CMP     A, #3CH
        BC      $WAIT1

;---------------------------------------------------------------------------
;       RTCを0.5sに1度の定周期割り込みに設定します。
;---------------------------------------------------------------------------
⑩
        SET1    !RTCEN                  ;RTC使用許可

        CLR1    RTCE                    ;カウンタ動作禁止
        MOV     RTCC0,  #00000001B      ;RTCコントロール
                        ;|||||+++---------- CT2-0: 0.5sに1度の定期割り込み
                        ;||||+------------- AMPM: 未使用
                        ;|||+-------------- RCLOE0: 未使用
                        ;||+--------------- RCLOE1: 未使用
                        ;|+---------------- <0>
                        ;+----------------- RTCE: カウンタ動作停止

        MOV     SEC,    #0              ;秒カウント(RSUBC)クリア

        CLR1    RTCIF                   ;割り込み要求クリア
        SET1    RTCMK                   ;割り込み禁止

;---------------------------------------------------------------------------
;       キャリブレーション開始
;---------------------------------------------------------------------------
⑪       CALL    !!SCALIBSTT
          ⑫
```

C言語の使用する周辺の初期設定も，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。

```
/************************************************************************
        使用する周辺の初期設定
************************************************************************/
void hdwinit(void)
{
        unsigned char i;
        unsigned int j;

        DI();                                   /* 割り込み禁止 */
/*------------------------------------------------------------------------
        クロック周波数の設定
------------------------------------------------------------------------
        高速内蔵発振クロックで動作が行えるように設定します。
------------------------------------------------------------------------*/
        CMC   =   0b00010000;                   /* クロック動作モード */
                  /*|||||||+---------------- AMPH: 2MHz≦fMX≦10MHz */
                  /*||||+++----------------- <000> */
                  /*|||+-------------------- OSCSELS: XT1発振モード */
                  /*||+---------------------- <0> */
                  /*++----------------------- EXCLK/OSCSEL: 入力ポート・モード */

        CSC   =   0b10000000;                   /* クロック動作ステータス制御 */
                  /*|||||||+----------------- HIOSTOP: 高速内蔵発振回路動作 */
                  /*||+++++------------------ <00000> */
                  /*|+------------------------ XTSTOP: XT1発振回路動作 */
                  /*+------------------------- MSTOP: X1発振回路動作 */

        OSMC  =   0b00000000;                   /* 動作スピード・モード */
                  /*|||||||+----------------- FSEL: 10MHz以下の周波数で動作 */
                  /*+++++++------------------ <00000> */

        CKC   =   0b00001000;                   /* クロック選択 */
                  /*|||||+++----------------- MDIV2-0: CPU/周辺ハードウエア・クロック */
                  /*|||||                              (fCLK)=fIH */
                  /*||||+---------------------- <1> */
                  /*|||+----------------------- MCM0: 高速内蔵発振クロック(fIH) */
                  /*||+------------------------ <R> */
                  /*|+------------------------- CSS: メイン・システム・クロック(fMAIN)=fCLK */
                  /*+-------------------------- <R> */

/*------------------------------------------------------------------------
        ポート0の設定
------------------------------------------------------------------------*/
        P0  = 0b00000000;           /* P00-P07の出力ラッチLow */
        PM0 = 0b00000000;           /* P00-P07を出力ポートに設定 */

/*------------------------------------------------------------------------
        ポート1の設定
------------------------------------------------------------------------*/
        P1  = 0b00000000;           /* P10-P17の出力ラッチLow   [後でP12(TxD0)の設定を行う] */
        PM1 = 0b00000000;           /* P10-P17を出力ポートに設定 [後でP11(RxD0)の設定を行う] */

/*------------------------------------------------------------------------
        ポート2の設定
------------------------------------------------------------------------*/
        P2  = 0b00000000;           /* P20-P27の出力ラッチLow */
        PM2 = 0b11111111;           /* P20-P27(ANI0-ANI7)を入力ポートに設定 */

/*------------------------------------------------------------------------
        ポート3の設定
------------------------------------------------------------------------*/
        P3  = 0b00000000;           /* P30-P37の出力ラッチLow */
        PM3 = 0b00000000;           /* P30-P37を出力ポートに設定 */

/*------------------------------------------------------------------------
        ポート4の設定
------------------------------------------------------------------------*/
        P4  = 0b00000000;           /* P40-P47の出力ラッチLow */
        PM4 = 0b00000000;           /* P40-P47を出力ポートに設定 */

/*------------------------------------------------------------------------
        ポート5の設定
------------------------------------------------------------------------*/
        P5  = 0b00000000;           /* P50-P57の出力ラッチLow */
        PM5 = 0b00000000;           /* P50-P57を出力ポートに設定 */

/*------------------------------------------------------------------------
        ポート6の設定
------------------------------------------------------------------------*/
        P6  = 0b00000000;           /* P60-P67の出力ラッチLow */
        PM6 = 0b00000000;           /* P60-P67を出力ポートに設定 */
```

```
/*-------------------------------------------------------------------------------
        ポート7の設定
-------------------------------------------------------------------------------*/
        P7  = 0b00000000;           /* P70-P77の出力ラッチLow */
        PM7 = 0b00000000;           /* P70-P77を出力ポートに設定 */

/*-------------------------------------------------------------------------------
        ポート8の設定
-------------------------------------------------------------------------------*/
        P8  = 0b00000000;           /* P80-P87の出力ラッチLow */
        PM8 = 0b00000000;           /* P80-P87を出力ポートに設定 */

/*-------------------------------------------------------------------------------
        ポート9の設定
-------------------------------------------------------------------------------*/
        P9  = 0b00000000;           /* P90-P97の出力ラッチLow */
        PM9 = 0b00000000;           /* P90-P97を出力ポートに設定 */

/*-------------------------------------------------------------------------------
        ポート10の設定
-------------------------------------------------------------------------------*/
        P10  = 0b00000000;          /* P100-P107の出力ラッチLow */
        PM10 = 0b00000000;          /* P100-P107を出力ポートに設定 */

/*-------------------------------------------------------------------------------
        ポート11の設定
-------------------------------------------------------------------------------*/
        P11  = 0b00000000;          /* P110-P117の出力ラッチLow */
        PM11 = 0b00000000;          /* P110-P117を出力ポートに設定 */

/*-------------------------------------------------------------------------------
        ポート12の設定
-------------------------------------------------------------------------------*/
        P12  = 0b00000000;          /* P120,P125-P127の出力ラッチLow */
        PM12 = 0b00011110;          /* P120,P125-P127を出力ポートに設定 */

/*-------------------------------------------------------------------------------
        ポート13の設定
-------------------------------------------------------------------------------*/
        P13  = 0b00000000;          /* P130-P137の出力ラッチLow */
        PM13 = 0b00000000;          /* P130-P137を出力ポートに設定 */

/*-------------------------------------------------------------------------------
        ポート14の設定
-------------------------------------------------------------------------------*/
        P14  = 0b00000000;          /* P140-P147の出力ラッチLow */
        PM14 = 0b00000000;          /* P140-P147を出力ポートに設定 */

/*-------------------------------------------------------------------------------
        ポート15の設定
-------------------------------------------------------------------------------*/
        P15  = 0b00000000;          /* P150-P157の出力ラッチLow */
        PM15 = 0b00000000;          /* P150-P157を出力ポートに設定 */

/*-------------------------------------------------------------------------------
        ポート16の設定
-------------------------------------------------------------------------------*/
        P16  = 0b00000000;          /* P160-P163の出力ラッチLow */
        PM16 = 0b11110000;          /* P160-P163を出力ポートに設定 */

/*-------------------------------------------------------------------------------
        A/Dコンバータの設定
-------------------------------------------------------------------------------*/
        ADCEN = 1;                  /* ADコンバータ使用の許可 */

        ADPC  =  0b00000000;        /* A/Dポート・コンフィギュレーション */
                 /*|||+++++----------- ADPC4-0: 使用するP20-P27をすべてアナログ入力に切り替え */
                 /*+++---------------- <000> */

        ADM   =  0b00100000;                            /* 変換時間の選択 */
                 /*|||||||+----------- ADCE: コンパレータの動作停止 */
                 /*||+++++------------ FR2-0,LV1-0: fCLK=8MHzで最速の変換時間(8.25us)を選択 */
                 /*||+----------------- <0> */
                 /*+------------------ ADCS: 変換動作停止 */

/*-------------------------------------------------------------------------------
        UART0の設定
-------------------------------------------------------------------------------*/
        SAU0EN = 1;                 /* UART0のあるSAU0使用許可 */

        NOP();                      /* 4クロック分のウェイト */
        NOP();                      /* (詳細はユーザーズ・マニュアルを参照してください。) */
        NOP();
        NOP();

        SPS0L =  0b00000010;        /* プリスケーラ(動作クロック)の設定 */
                 /*||||++++----------- PRS003-000(CK00): fCLK/2^2 */
                 /*++++--------------- PRS013-010(CK01): 未使用 */
```

```
        /*--------------------------------------
         送信設定(チャネル0使用)
        --------------------------------------*/
        ucTxBufferCounter = 0;     /* 送信データ・カウンタのクリア */

        SMR00 =  0b0000000000100010;/* 動作モードなどの設定 */
                 /*||||||||||||||||+-- MD000: 転送完了割り込み */
                 /*||||||||||||||++--- MD002-001: UARTモード */
                 /*|||||||||||+++----- <100> */
                 /*||||||||||+-------- SIS000: 立ち下がりエッジをスタートビットとして検出 */
                 /*|||||||||+--------- <0> */
                 /*||||||||+---------- STS00: UART送信 */
                 /*|||+++------------- <000> */
                 /*||+---------------- CCS00: CKS00ビットで指定した動作クロックMCK */
                 /*++----------------- <00> */
                 /*+------------------ CKS00: PRSで選択した動作クロックCK00(MCK) */

        SCR00 =  0b1000000010010111;/* 通信フォーマットの設定 */
                 /*|||||||||||||+++-- DLS002-000: 8ビット・データ長 */
                 /*||||||||||||+----- <0> */
                 /*|||||||||||++------ SLC001-000: ストップ・ビット長 = 1ビット */
                 /*||||||||||+-------- <0> */
                 /*|||||||||+--------- DIR00: LSBファーストで出力 */
                 /*|||||||++---------- PTC001-000: パリティ・ビットを出力しない */
                 /*||||||+------------ EOC00: UART送信時はエラー割り込み未使用 */
                 /*|||||+------------- <0> */
                 /*|||++-------------- DAP00/CKP00: UARTモードでは未使用 */
                 /*+++----------------- TXE00/RXE00: 送信のみを行う */

        /* SDR00 転送ボー・レートの設定はINTRTC割り込み処理にて設定 */

        SOL0L = 0b00000000;        /* 出力データのレベルの設定(反転なし) */

        SO0   =  0b0000000000000001;/* 初期出力レベルの設定 */
                 /*||||||||||||++++-- SO03-0: チャネルnのシリアル・クロック出力 */
                 /*||||||||++++------ <0000> */
                 /*||||++++---------- CKO03-0: チャネルnのシリアル・データ出力 */
                 /*++++-------------- <0000> */

        SOE0L.0 = 1;               /* SOE00:シリアル出力許可 */

        P1.2  = 1;                 /* P12=TxD0 */

        /*--------------------------------------
         UART0受信設定(SAU0のチャネル1使用)
        --------------------------------------*/
        ucRxBufferCounter = 0;     /* 受信データ・カウンタのクリア */

        PM1.1 = 1;                 /* P11=RxD0 */

        SMR01 =  0b0000000100100010;/* 動作モード設定 */
                 /*||||||||||||||||+-- MD010: 転送完了割り込み */
                 /*||||||||||||||++--- MD012-011: UARTモード */
                 /*|||||||||||+++----- <100> */
                 /*||||||||||+-------- SIS010: 立ち下がりエッジをスタートビットとして検出 */
                 /*|||||||||+--------- <0> */
                 /*||||||||+---------- STS01: RxD端子の有効エッジ(UART受信設定) */
                 /*|||+++++----------- <00000> */
                 /*||+---------------- CCS01: CKS00ビットで指定した動作クロックMCK */
                 /*++----------------- CKS01: PRSで選択した動作クロックCK00(MCK) */

        SCR01 =  0b0100010010010111;/* 通信フォーマットの設定 */
                 /*|||||||||||||+++-- DLS012-010: 8ビット・データ長 */
                 /*||||||||||||+----- <0> */
                 /*|||||||||||++------ SLC011-010: ストップ・ビット長 = 1ビット */
                 /*||||||||||+-------- <0> */
                 /*|||||||||+--------- DIR01: LSBファーストで入力を行う */
                 /*|||||||++---------- PTC011-010: パリティなしで受信 */
                 /*||||||+------------ EOC01: UART受信のエラー割り込み許可 */
                 /*|||||+------------- <0> */
                 /*|||++-------------- DAP01/CKP01: UARTモードでは未使用 */
                 /*+++----------------- TXE01/RXE01: 受信のみを行う */

        /* SDR01 転送ボー・レートの設定はINTRTC割り込み処理にて設定 */


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

/*-------------------------------------------------------------------------------
        サブシステム・クロックの発振安定待ち(本例は約3秒ですが，実装回路上で評価したうえで
        発振安定まで待ってください)。
-------------------------------------------------------------------------------*/

        for (i=0; i<=0x27; i++) {
                 for (j=0; j<=0x0fff0; j++) {
                 }
        }
```

```
/*-------------------------------------------------------------------------------
        RTCを0.5sに1度の定周期割り込みに設定します。
-------------------------------------------------------------------------------*/

        RTCEN = 1;                  /* RTC使用許可 */

        RTCE = 0;                   /* カウンタ動作禁止 */
        RTCC0 =  0b00000001;        /* RTCコントロール */
                 /*|||||+++---------- CT2-0: 0.5sに1度の定期割り込み */
                 /*|||||+------------- AMPM: 未使用 */
                 /*|||+--------------- RCLOE0: 未使用 */
                 /*||+---------------- RCLOE1: 未使用 */
                 /*|+----------------- <0> */
                 /*+------------------ RTCE: カウンタ動作停止 */

        SEC = 0;                    /* 秒カウント(RSUBC)クリア */

        RTCIF = 0;                  /* 割り込み要求クリア */
        RTCMK = 1;                  /* 割り込み禁止 */

/*-------------------------------------------------------------------------------
        キャリブレーション開始
-------------------------------------------------------------------------------*/
        fn_CalibrationStart();
}
```

4.3　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 キャリブレーションが完了したかのチェックを行います。キャリブレーションが完了していない場合は，何も行いません（⑩へ）。

② シリアル・チャネル許可ステータス・レジスタ0（SE0）にてUART0の送受信動作のチェックを行います。送受信動作中の場合は，何も行いません（⑩へ）。

③ 最終データの受信が完了して動作が停止中であるかのチェックを行います。受信が完了していない場合は，送信完了のチェックへ分岐します（⑧へ）。

④ シリアル・ステータス・レジスタ01（SSR01）によりエラーのチェックを行います。エラーがある場合は，再度，受信動作を開始します（⑨へ）。

⑤ UART0で受信したデータが受信データのフォーマットに対応しているかのチェックを行います。受信データのフォーマットに対応していない場合は，送信バッファにエラー情報を設定します。

⑥ ⑤のUART0で受信したデータが通信フォーマットに対応している場合， A/Dコンバータで A/D値を取得し，送信データ用に加工し，送信バッファへ設定します。A/D値を取得するチャネルとA/Dの分解能は，UART0の受信データにより決定します。

⑦ 送信バッファのデータを送信データ・レジスタに設定し，UART0送信動作を開始します。（⑩へ）

⑧ 最終データの送信が完了して動作が停止中であるかのチェックを行います。送信が完了していない場合は何も行いません（⑩へ）。

⑨ UART0受信動作を開始します。（⑩へ）

⑩ ①へ分岐します。

```
;*******************************************************************************
;
;       メイン処理
;
;*******************************************************************************
MAIN_LOOP:
        ;---------------------------------------
        ; 動作状態による分岐
        ;---------------------------------------
①      CMP     RCALIBST,#CCALIB_END        ;キャリブレーション完了?
       BZ      $LMAIN100                   ; Yes,
       BR      LMAINRET                    ; No,
LMAIN100:
       MOV     A,      SE0L                ;動作停止状態/停止状態を確認
②      BT      A.0,    $LMAINRET           ;SE00:UART0送信動作停止中? No,
       BT      A.1,    $LMAINRET           ;SE01:UART0受信動作停止中? No,

       ;*************************************
       ; UART0受信完了
       ;*************************************
③      CMP     RRXBUFP,#RRXBUFE-RRXBUF     ;最終データの受信完了?
       BC      $LMAINTX                    ; No,
       MOV     RRXBUFP,#0                  ;受信データ・カウンタ初期化

④      MOV     A,      SSR01L              ;UART0のステータスを確認
       AND     A,      #00000111B          ;(フレーミング,パリティ,オーバラン)エラーあり?
       BNZ     $LMAIN800                   ; Yes,

       ;=============================================================
       ; 送信データの作成
       ;=============================================================
       ;
       ;サンプル・プログラムで使用する送受信データは，下図に示す2バイトのフォー
       ;マットとしています。
       ;
       ;【受信データ】
       ;    byte|←      +0     →|←      +1     →|
       ;        +----------------+----------------+
       ;        |A/D値分解能の指定|A/Dチャネルの指定|
       ;        |  08H: 8ビット  |   0:ch0 ・・・  |
       ;        |  0AH:10ビット  |    ・・・ 7:ch7 |
       ;        +----------------+----------------+
       ;
       ;
```

```
            ;【送信データ】
            ;
            ;   byte|←      +0      →|←      +1      →|
            ;   bit |  7|6|5|4|3|2|1|0|  7|6|5|4|3|2|1|0|
            ;       +---+-------------+-----------------+
            ;       |   |             | 8ビットA/D値    |
            ;       |   |             | (8ビット指定時) |
            ;       +---+-------------+-----------------+
            ;       +---+---------+-------------------+
            ;       |   |         |    10ビットA/D値  |
            ;       |   |         |   (10ビット指定時)|
            ;       +---+---------+-------------------+
            ;         |
            ;         +-エラー情報：受信したデータにエラーあり(1)/エラーなし(0)
            ;
            ;=====================================================================
            ;--------------------------------------
            ; 受信データのチェック
            ;--------------------------------------
            CMP       RRXBUF,  #10              ;A/D値分解能の指定は10ビット?
            BZ        $LMAINRX300               ; Yes,
            CMP       RRXBUF,  #8               ;A/D値分解能の指定は8ビット?
            BNZ       $LMAINRX400               ; No,(上記以外はエラー情報設定)
LMAINRX300:
⑤           CMP       RRXBUF+1,#7+1             ;チャネル指定(0-7)は正常値?
            BC        $LMAINRX500               ; Yes,
LMAINRX400:
            ;エラー
            MOVW      AX,      #8000H           ;エラー情報設定
            BR        LMAINRX700
LMAINRX500:
            ;--------------------------------------
            ; A/D値の取得
            ;--------------------------------------
            MOV       A,       RRXBUF+1         ;A/D変換するチャネルの設定
            MOV       ADS,     A

            SET1      ADCE                      ;コンパレータの動作許可(変換待機)

            ;1変換目(不正データ)
            CLR1      ADIF                      ;割り込み要求クリア
            SET1      ADCS                      ;A/D変換動作許可
LMAINRX530:
            NOP
            BF        ADIF,    $LMAINRX530      ;1us以上ウエイトしないでADCSに1を設定した場合は,
                                                ;最初の変換データを無視します。
                                                ;(詳細はユーザーズ・マニュアルを参照してください。)
            ;2変換目(正規データ)
            CLR1      ADIF                      ;割り込み要求クリア
            SET1      ADCS                      ;A/D変換動作許可
LMAINRX550:
            NOP
⑥           BF        ADIF,    $LMAINRX550

            CLR1      ADCS                      ;A/D変換動作停止
            CLR1      ADCE                      ;コンパレータの動作停止

            MOVW      AX,      ADCR             ;10ビットA/D変換結果取得

            SHRW      AX,      6                ;データのシフト
                                                ;(10ビットA/D変換結果を下位ビット詰め)
            CMP       RRXBUF,  #10              ;A/D値分解能の指定は10ビット?
            BZ        $LMAINRX700               ; Yes,
            SHRW      AX,      2                ;データシフト
                                                ;(10ビットA/D変換結果をさらにシフトし,
                                                ;   8ビットA/D変換結果として扱う)
            ;送信データ設定
LMAINRX700:
            MOV       RTXBUF,   A               ;送信データを送信バッファに設定
            XCH       A,        X
            MOV       RTXBUF+1,A

            ;UART0送信動作開始
            CLR1      STIF0                     ;割り込み要求クリア
            CLR1      STMK0                     ;割り込み許可
⑦           SET1      !SS0L.0                   ;SS00:UART0送信動作開始

            MOV       A,       RTXBUF           ;設定した送信データの第一バイト
            MOV       TxD0,    A                ;データを送信データ・レジスタに設定
            BR        LMAINRET

LMAINTX:
            ;*************************************
            ; UART0送信完了
            ;*************************************
            CMP       RTXBUFP,#RTXBUFE-RTXBUF   ;最終データの送信完了?
⑧           BC        $LMAINRET                 ; No,
            MOV       RTXBUFP,#0                ;送信データ・カウンタ初期化
LMAIN800:
            ;UART0受信動作開始
⑨           CLR1      SRIF0                     ;割り込み要求クリア
```

安定したコンパレータのデータを取得するため,最初の変換データを無視します。

2度目の変換データを正規のデータとして使用します。

分解能が8ビットの場合はADCRHレジスタの値と同様です。

```
            CLR1    SRMK0                   ;割り込み許可
            SET1    !SS0L.1                 ;SS01:UART0受信動作開始
LMAINRET:
⑩           BR      MAIN_LOOP               ;MAIN_LOOPへ
```

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。

```
/************************************************************************************
    メイン処理
************************************************************************************/
void main(void)
{
    while (1){
        if((ucCalibrationStatus==CALIBRATION_STATUS_END)&&!(SE0L & 0b00000011)){
        /* キャリブレーションが完了し，UART0の送受信動作を行っていない状態 */
            if(ucRxBufferCounter >= sizeof(ucRxBuffer)){
            /* UART0受信完了 */

                ucRxBufferCounter = 0;        /* 受信データ・カウンタのクリア */

                if(SSR01L & 0b00000111){    /* UART0のステータスを確認 */
                /* エラーあり */
                    /*-- 受信動作開始 --*/
                    SRIF0  = 0;         /* 割り込み要求クリア */
                    SRMK0  = 0;         /* 割り込み許可 */
                    SS0L.1 = 1;         /* SS01:UART0受信動作許可 */
                }
                else{
                /* エラーなし */
                    /*==========================================================================
                     送信データの作成
                    ==========================================================================

                    サンプル・プログラムで使用する送受信データは，下図に示す2バイトのフォー
                    マットとしています。

                    【受信データ】
                      byte|←      +0      →|←      +1      →|
                          +----------------+----------------+
                          |A/D値分解能の指定|A/Dチャネルの指定|
                          |  08H: 8ビット  |   0:ch0 ・・・  |
                          |  0AH:10ビット  |    ・・・ 7:ch7 |
                          +----------------+----------------+

                    【送信データ】

                      byte|←      +0      →|←      +1      →|
                      bit |  7|6|5|4|3|2|1|0|  7|6|5|4|3|2|1|0|
                          +---+-------------+-----------------+
                          |   |             |  8ビットA/D値   |
                          |   |             | (8ビット指定時) |
                          +---+-------------+-----------------+
                          +---+---------+-------------------+
                          |   |         |    10ビットA/D値  |
                          |   |         |   (10ビット指定時)|
                          +---+---------+-------------------+
                            |
                            +-エラー情報：受信したデータにエラーあり(1)/エラーなし(0)

                    ==========================================================================*/
                    /* 受信データのチェック */
                    if((ucRxBuffer[0]==10||ucRxBuffer[0]==8)&&ucRxBuffer[1]<=7){
                    /* 受信したデータが正常な場合 */

                        /* A/D値の取得 */
                        ADS = ucRxBuffer[1];    /* A/D変換するチャネルの設定 */
                        ADCE = 1;           /* コンパレータの動作許可(変換待機) */

                        /* 1変換目(不正データ) */
                        /* 1us以上ウエイトしないでADCSに1を設定した場合は，
                           最初の変換データを無視します。
                           (詳細はユーザーズ・マニュアルを参照してください。) */
                        ADIF = 0;           /* 割り込み要求クリア */
                        ADCS = 1;           /* A/D変換動作許可 */
                        while(!ADIF){   /* 変換待ち */
                            NOP();
                        }
                        /* 2変換目(正規データ) */
                        ADIF = 0;           /* 割り込み要求クリア */
                        ADCS = 1;           /* 動作許可 */
                        while(!ADIF){   /* 変換待ち */
                            NOP();
                        }
```

安定したコンパレータのデータを取得するため，最初の変換データを無視します。

2度目の変換データを正規のデータとして使用します。

```
                    ADCS = 0;       /* A/D変換動作停止 */
                    ADCE = 0;       /* コンパレータの動作停止 */

                    /* 送信データ設定 */
                    if(ucRxBuffer[0]==8){
                    /* 8ビット分解能 */
                        ucTxBuffer[0] = 0;
                        ucTxBuffer[1] = ADCRH;
                    }
                    else{
                    /* 10ビット分解能 */
                        ucTxBuffer[0] = (unsigned char)(ADCR >>14);
                        ucTxBuffer[1] = (unsigned char)(ADCR >>6);
                        /*データのシフト(10ビットA/D変換結果を下位ビット詰めを行っています。*/
                    }
                }
                else{
                /* 受信したデータが不正な場合 */
                    ucTxBuffer[0] =0x80;    /* エラー通知設定 */
                    ucTxBuffer[1] =0x00;
                }
                /* UART0送信動作開始 */
                STIF0  = 0;       /* 割り込み要求クリア */
                STMK0  = 0;       /* 割り込み許可 */
                SS0L.0 = 1;       /* SS00:UART0送信動作開始 */

                TxD0 = ucTxBuffer[0];    /* 送信データの第一バイトのデータを
                                            送信データ・レジスタに設定 */
            }
        }
        else if(ucTxBufferCounter >= sizeof(ucTxBuffer)){
        /* UART0送信完了 */

            ucTxBufferCounter = 0;       /* 送信データ・カウンタのクリア */

            /* UART0受信動作開始 */
            SRIF0  = 0;           /* 割り込み要求クリア */
            SRMK0  = 0;           /* 割り込み許可 */
            SS0L.1 = 1;           /* SS01:UART0受信動作開始 */
        }
    }
  }
}
```

4.4　キャリブレーション開始処理

アセンブリ言語のキャリブレーション開始処理では，次の動作を行います。

① キャリブレーション状態を初期化します。

② タイマ・アレイ・ユニット0（TAU0）のチャネル0によるキャプチャ動作を行うための設定を行います。

③ リアルタイム・カウンタ（RTC）を0.5sに一度の定周期割り込みに設定します。

④ ②で設定したタイマ・アレイ・ユニット0（TAU0）のチャネル0によるキャプチャ動作と③で設定したリアルタイム・カウンタ（RTC）を0.5sに一度の定周期割り込みの動作を開始します。0.5s後にINTRTC割り込みが発生します。

```
;*****************************************************************************
;
;       キャリブレーション開始処理
;
;-----------------------------------------------------------------------------
;       精度の良い0.5sを計測するため，サブシステム・クロックを使用したRTCを定周
;       期割り込み機能を使用します。また，その0.5s間の高速内蔵発振クロックでの
;       カウント数を計測するため，TAU0のチャネル0をキャプチャモードに設定し，動
;       作を開始します。
;*****************************************************************************
SCALIBSTT:

①      MOV     RCALIBST,#CCALIB_RESET      ;キャリブレーション状態の初期化

        ;-------------------------------------------
        ; キャプチャの設定
        ;-------------------------------------------
        ; TAU0のチャネル0を使用し，0.5s間の高速内蔵発振
        ; クロックのカウント数の計測
        ;-------------------------------------------
        SET1    !TAU0EN                     ;TAU0使用許可

        MOV     TPS0L,  #00000110B          ;タイマ・クロック選択
                        ;||||++++---------- PRS003-000: fCLK/2^6 (キャプチャ動作で使用)
                        ;++++-------------- PRS013-010: 未使用

        MOVW    AX,     #0000000000000100B;動作モード設定
②                       ;||||||||||||++++- MD003-000: キャプチャ・モード，アップ・カウント
                        ;||||||||||++------ <0>
                        ;||||||||++-------- CIS001-000: 未使用
                        ;|||||+++---------- STS002-000: ソフトウェア・トリガ・スタート
                        ;||||+------------- MASTER00: 単体動作
                        ;|||+-------------- CCS00: CKS00ビットで指定した動作クロックMCK
                        ;||++--------------- <0>
                        ;++----------------- CKS00: PRSで選択した動作クロックCK00(MCK)
        MOVW    TMR00,  AX

        CLR1    TMIF00                      ;割り込み要求クリア
        SET1    TMMK00                      ;割り込み禁止

        ;-------------------------------------------
        ; 定周期割り込みの設定
        ;-------------------------------------------
        ; RTCを使用したサブシステム・クロックの安定待ち
        ; に使用した設定をそのまま使用します。
        ;-------------------------------------------
        MOV     SEC,    #0                  ;秒カウント(RSUBC)クリア
③       CLR1    RTCIF                       ;割り込み要求クリア
        CLR1    RTCMK                       ;割り込み許可

        ;-------------------------------------------
        ; 定周期割り込みとキャプチャ動作開始
        ;-------------------------------------------
④       SET1    RTCE            ;RTCカウンタ動作開始
        SET1    TS0L.0          ;TS00:TAU0のチャネル0のキャプチャ動作開始(トリガ動作)

        RET
```

C言語のキャリブレーション開始処理も，アセンブリ言語と同様な動作を行います。

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

        キャリブレーション開始処理

-------------------------------------------------------------------------------------
        精度の良い0.5sを計測するため，サブシステム・クロックを使用したRTCを定周
        期割り込み機能を使用します。また，その0.5s間の高速内蔵発振クロックでの
        カウント数を計測するため，TAU0のチャネル0をキャプチャモードに設定し，動
        作を開始します。
************************************************************************************/
static void fn_CalibrationStart(void)
{
        ucCalibrationStatus = CALIBRATION_STATUS_RESET;
                                /* キャリブレーション状態の初期化 */

        /*-----------------------------------------------
         キャプチャの設定
        -----------------------------------------------
         TAU0のチャネル0を使用し，0.5s間の高速内蔵発振
         クロックのカウント数の計測
        -----------------------------------------------*/
        TAU0EN = 1;             /* TAU0使用許可 */

        TPS0L =  0b00000110;    /* タイマクロック選択 */
                 /*||||++++----------- PRS003-000: fCLK/2^6 (キャプチャ動作で使用) */
                 /*++++--------------- PRS013-010: 未使用 */

        TMR00 =  0b0000000000000100;/* 動作モード設定 */
                 /*||||||||||||++++--  MD003-000: キャプチャ・モード，アップ・カウント */
                 /*||||||||||++------- <00> */
                 /*||||||||||--------- CIS001-000: 未使用 */
                 /*||||++++----------- STS002-000: ソフトウェア・トリガ・スタート */
                 /*|||+--------------- MASTER00: 単体動作 */
                 /*||+---------------- CCS00: CKS00ビットで指定した動作クロックMCK */
                 /*++----------------- <00> */
                 /*+------------------ CKS00: PRSで選択した動作クロックCK00(MCK) */

        TMIF00 = 0;             /* 割り込み要求クリア */
        TMMK00 = 1;             /* 割り込み禁止 */

        /*-----------------------------------------------
         定周期割り込みの設定
        -----------------------------------------------
         RTCを使用したサブシステム・クロックの安定待ち
         に使用した設定をそのまま使用します。
        -----------------------------------------------*/
        SEC = 0;                /* 秒カウント(RSUBC)クリア */
        RTCIF = 0;              /* 割り込み要求クリア */
        RTCMK = 0;              /* 割り込み許可 */

        /*-----------------------------------------------
         定周期割り込みとキャプチャ動作開始
        -----------------------------------------------*/
        RTCE   = 1;             /* RTCカウンタ動作開始 */
        TS0L.0 = 1;             /* TS00:TAU0のチャネル0のキャプチャ動作開始(トリガ動作) */
}
```

4. 5　INTST0割り込み処理

アセンブリ言語の割り込み処理では，次の動作を行います。

① 割り込み処理で汎用レジスタを使用するため，レジスタバンクを切り替えます。

② 送信データ・カウンタの更新と送信終了チェックを行います。送信が終了した場合は，④へ進みます。

③ ②のチェックにて送信が終了していない場合，送信データ・カウンタに応じた送信バッファ位置のデータを送信し，処理を抜けます。

④ ②のチェックで送信が終了した場合，UART0送信動作を停止します。

```
;*******************************************************************************
;
;         INTST0割り込み（UART0の送信完了割り込み）
;
;*******************************************************************************
INT_UARTST:

①        SEL       RB1                          ;レジスタバンク切り替え

②        INC       RTXBUFP                      ;送信データ・カウンタ更新
         CMP       RTXBUFP,#RTXBUFE-RTXBUF      ;最終データ送信終了?
         BNC       $HINTST800                   ; Yes,

③        MOV       A,        RTXBUFP            ;送信データ・カウンタ
         MOV       B,        A
         MOV       A,        RTXBUF[B]          ;今回の送信データを取得
         MOV       TxD0,     A                  ;データを送信データ・レジスタに設定
         BR        HINTSTRET
HINTST800:
         ;データ送信終了
④        SET1      STMK0                        ;割り込み禁止
         SET1      !ST0L.0                      ;ST00:UART0送信動作停止
HINTSTRET:
         RETI
```

●C言語での割り込み処理の設定●

C言語での割り込み処理の設定は，pragma指令により，指定された割り込み要求名に対応する割り込みベクタ・テーブルに登録します。記述の仕方を以下に示します。

・#pragma指令による割り込みベクタ・テーブルへの登録　（Cソースの先頭）

```
#pragma interrupt        INTRTC    fn_intrtc
```

関数名：割り込み処理を記述した関数名を記述します。

割り込み要求名：大文字で記述します。各製品のユーザーズ・マニュアルを参照してください。

・割り込み関数修飾子によるハードウエア割り込み関数であることの宣言　（Cソース部分）

```
__interrupt void fn_intrtc(void)
{
      …(略)…
}
```

詳細に関しては，ユーザーズ・マニュアル CC78K0R コンパイラ・パッケージ 言語編 を参照ください。

C言語のINTST0割り込み処理も，アセンブリ言語と同様な動作を行います。

```
/************************************************************************
        INTST0割り込み（UART0の送信完了割り込み）
************************************************************************/
__interrupt void fn_intst0(void)
{
        ucTxBufferCounter++;                            /* 送信データ・カウンタ更新 */

        if(ucTxBufferCounter < sizeof(ucTxBuffer)){
        /* 後続データあり */
                /* 今回の送信データを送信データ・レジスタに設定 */
                TxD0 = ucTxBuffer[ucTxBufferCounter];
                                                        /* TxD0: データ送信 */
        }
        else{
        /* データ送信終了 */
                STMK0 = 1;                              /* 割り込み禁止 */
                ST0L.0 = 1;                             /* ST00:UART0送信動作停止 */
        }
}
```

4. 6　INTSR0割り込み処理

アセンブリ言語の割り込み処理では，次の動作を行います。

① 割り込み処理で汎用レジスタを使用するため，レジスタバンクを切り替えます。

② 受信データ・カウンタに応じた受信バッファ位置に受信データを格納します。

③ 受信データ・カウンタの更新と受信終了チェックを行います。受信が終了した場合は，④へ進みます。受信が終了していない場合は，処理を抜けます。

④ ③のチェックで受信が終了した場合，UART0送信動作を停止します。

```
;*******************************************************************************
;
;         INTSR0割り込み（UART0受信完了割り込み）
;
;*******************************************************************************
INT_UARTSR:

①       SEL       RB1                         ;レジスタバンク切り替え

        MOV       A,        RRXBUFP           ;受信データ・カウンタ
②      MOV       B,        A
        MOV       A,        RxD0              ;受信データ取得
        MOV       RRXBUF[B],A                 ;今回の受信データを格納

        INC       RRXBUFP                     ;受信データ・カウンタ更新
③      CMP       RRXBUFP,#RRXBUFE-RRXBUF     ;データ受信終了?
        BC        $HINTSRRET                  ; No,
        ;データ受信終了
        SET1      SRMK0                       ;割り込み禁止
④       SET1      !ST0L.1                     ;ST01:UART0受信動作停止
        RETI
```

C言語のINTSR0割り込み処理も，アセンブリ言語と同様な動作を行います。

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

         INTSR0割り込み（UART0受信完了割り込み）

*******************************************************************************/
__interrupt void fn_intsr0(void)
{
         ucRxBuffer[ucRxBufferCounter] = (unsigned char)SDR01;
                                                    /* 今回の受信データを格納 */

         ucRxBufferCounter++;                       /* 受信データ・カウンタ更新 */

         if(ucRxBufferCounter >= sizeof(ucRxBuffer)){
         /* データ受信終了 */
                  SRMK0 = 1;                        /* 割り込み禁止 */
                  ST0L.1 = 1;                       /* ST01:通信動作停止,RxD0受信禁止 */
         }
}
```

4. 7　INTRTC割り込み処理

アセンブリ言語の割り込み処理では，次の動作を行います。

① 割り込み処理で汎用レジスタを使用するため，レジスタバンクを切り替えます。
② タイマ・アレイ・ユニット0（TAU0）のチャネル0のキャプチャの動作を停止します。
③ 0.5s間の高速内蔵発振クロックのタイマ・カウント値を取得します。
④ リアルタイム・カウンタ（RTC）の動作を停止します。
⑤ 転送ボー・レート用クロックの分周値が9600bpsとなるように設定値を算出します。詳細は，3.5 キャリブレーションを参照してください。
⑥ ⑤で算出した分周値をシリアル・データ・レジスタ00（SDR00）およびシリアル・データ・レジスタ01（SDR01）のフォーマットに合わせ，設定します。
⑦ シリアル・アレイ・ユニット0（SAU0）のチャネル1のUART0受信動作を開始します。
⑧ キャリブレーション状態をキャリブレーション完了に設定します。

```
;*******************************************************************************
;
;       INTRTC割り込み処理
;       (INTRTC使用，RTCの0.5sの1度の定周期割り込み)
;
;-------------------------------------------------------------------------------
;       キャリブレーション開始処理の呼び出しから0.5s後に発生する定周期割り込み処
;       理です。TAU0のチャネル0のキャプチャ・モードで計測した0.5s間の高速内蔵発振
;       クロックのタイマ・カウント値からUART0の転送ボー・レートの設定を行います。
;*******************************************************************************
INT_RTC:
①       SEL     RB1             ;レジスタバンク切り替え

②       SET1    TT0L.0          ;TAU0のチャネル0のキャプチャ動作停止(トリガ動作)

③       MOVW    AX,     TCR00   ;0.5s間の高速内蔵発振クロックでのタイマ・カウント値を取得

④       CLR1    RTCE            ;RTCカウンタ動作停止
        SET1    RTCMK           ;割り込み禁止

        ;-------------------------------------------------------------------------------
        ; ボー・レートの算出式
        ;-------------------------------------------------------------------------------
        ; 取得したタイマ・カウント値から転送ボー・レートが9600bpsとなるように
        ; 動作クロックの分周値用に変換
        ;-------------------------------------------------------------------------------
        ;
        ; ボー・レートの算出式は，
        ;
        ;   (ボー・レート) =
        ;         {対象チャネルの動作クロック(MCK)周波数}/(SDRmn[15:9]+1)/2 [bps] ・・・①
        ;
        ;   ※ SDRmn[15:9]は，SDRmnレジスタのビット15-9の値を示します。
        ;
        ; で表されます。(詳細はユーザーズ・マニュアルを参照してください。)
        ;
        ;  ・CPU/周辺ハードウエア・クロック・・・fCLK [Hz]
        ;  ・ボー・レート・・・・・・・・・・・・・9600 [bps]
        ;
        ; として，①式から，
        ;
        ;   (SDRmn[15:9]+1) = {fCLK/2^2}/9600/2 ・・・②
        ;
        ; と表せます。
        ; また，高速内蔵発振クロックの2^6分周で動作するTAU0のチャネル0のキャプチャで
        ; 計測した0.5s間のタイマ・カウント値(AX)から，
        ;
        ;   {2^6/fCLK}*AX = 0.5 [s]
        ;
        ; と表せますので，fCLKは，
        ;
        ;   fCLK=2^7*AX ・・・③
        ;
        ; となり，②，③式から，
        ;
        ;  SDRmn[15:9] = {2^7*AX/2^2}/9600/2-1 = AX/600-1
        ;
        ; 上記の式より，シリアル・データ・レジスタ(SDR0x)のUART0転送ボー・レート用
        ; 動作クロックの分周値の算出が行えます。
        ;
        ;-------------------------------------------------------------------------------
        ; AX/600-1の計算
        MOV     B,      #0              ;B=AX/600の計算結果
HIRTC300:
        SUBW    AX,     #600
        BC      $HIRTC500
        INC     B
        BR      $HIRTC300
⑤HIRTC500:
        DEC     B                       ;B=(AX/600の計算結果)-1

        ;シリアル・データ・レジスタ(SDR0x)のビット15-9に合わせる
        MOV     A,      B               ;B=AX/600-1の計算結果(SDRmn[15:9])
        ROL     A,      1               ;ビット15-9に合わせる
        AND     A,      #11111110B      ;ビット8は0固定
        MOV     X,      #0              ;SDRmnの下位7-0ビット用の設定

⑥       MOVW    SDR00,  AX              ;UART0送信用転送ボー・レートの設定
        MOVW    SDR01,  AX              ;UART0受信用転送ボー・レートの設定

        ;---------------------------------------
        ; UART0受信動作開始
        ;---------------------------------------
        CLR1    SRIF0                   ;割り込み要求クリア
⑦       CLR1    SRMK0                   ;割り込み許可
        SET1    !SS0L.1                 ;SS01:UART0受信動作許可

⑧       MOV     RCALIBST,#CCALIB_END    ;キャリブレーション完了

        RETI
```

C言語のINTRTC割り込み処理も，アセンブリ言語と同様な動作を行います。

```
/************************************************************************
          INTRTC割り込み処理
          (INTRTC使用，RTCの0.5sの1度の定周期割り込み)

------------------------------------------------------------------------
          キャリブレーション開始処理の呼び出しから0.5s後に発生する定周期割り込み処
          理です。TAU0のチャネル0のキャプチャ･モードで計測した0.5s間の高速内蔵発振
          クロックのタイマ･カウント値からUART0の転送ボー・レートの設定を行います。
************************************************************************/
__interrupt void fn_intrtc(void)
{

          register unsigned short ushnTimeCnt;          /* タイマ・カウンタ値 */

          TT0L.0 = 1;                /* TAU0のチャネル0のキャプチャ動作停止(トリガ動作) */

          ushnTimeCnt = TCR00;       /* 0.5s間の高速内蔵発振クロックでのタイマ・カウント値を取得 */

          RTCE  = 0;                 /* RTCカウンタ動作停止 */
          RTCMK = 1;                 /* 割り込み禁止 */

          /*-------------------------------------------------------------------------------
           ボー・レートの算出式
          -------------------------------------------------------------------------------
           取得したタイマ・カウント値から転送ボー・レートが9600bpsとなるように
           動作クロックの分周値用に変換
          -------------------------------------------------------------------------------

           ボー・レートの算出式は，

             (ボー・レート) =
                    {対象チャネルの動作クロック(MCK)周波数}/(SDRmn[15:9]+1)/2 [bps] ･･･①

             ※ SDRmn[15:9]は，SDRmnレジスタのビット15-9の値を示します。

           で表されます。(詳細はユーザーズ・マニュアルを参照してください。)

             ・CPU/周辺ハードウエア・クロック・・・fCLK [Hz]
             ・ボー・レート・・・・・・・・・・・・9600 [bps]

           として，①式から，

             (SDRmn[15:9]+1) = {fCLK/2^2}/9600/2 ･･･②

           と表せます。
           また，高速内蔵発振クロックの2^6分周で動作するTAU0のチャネル0のキャプチャで
           計測した0.5s間のタイマ・カウント値(ushnTimeCnt)から，

             {2^6/fCLK}*ushnTimeCnt = 0.5 [s]

           と表せますので，fCLKは，

             fCLK=2^7*ushnTimeCnt ･･･③

           となり，②，③式から，

            SDRmn[15:9] = {2^7*ushnTimeCnt/2^2}/9600/2-1 = ushnTimeCnt/600-1

           上記の式より，シリアル・データ・レジスタ(SDR0x)のUART0転送ボー・レート用
           動作クロックの分周値の算出が行えます。

          -------------------------------------------------------------------------------*/
          /* ushnTimeCnt/600-1の計算結果をボー・レート用クロックの分周値(SDR0x)に設定 */
          /* SDR0nレジスタのビット15-9に合わせこむように設定します。 */

          /* 送受信転送ボー・レートの設定 */
          SDR00 = SDR01 = (ushnTimeCnt/600 - 1)<<9 & 0b1111111000000000;

          /*--------------------------------------
           受信動作開始
          --------------------------------------*/
          SRIF0  = 0;                           /* 割り込み要求クリア */
          SRMK0  = 0;                           /* 割り込み許可 */
          SS0L.1 = 1;                           /* SS01:UART0受信動作許可 */

          ucCalibrationStatus = CALIBRATION_STATUS_END;         /* キャリブレーション完了 */
}
```

第5章　関連資料

資　料　名		和文／英文
78K0R/KE3　ユーザーズ・マニュアル		PDF
78K0R/KF3　ユーザーズ・マニュアル		PDF
78K0R/KG3　ユーザーズ・マニュアル		PDF
78K0R/KH3　ユーザーズ・マニュアル		PDF
78K0R/KJ3　ユーザーズ・マニュアル		PDF
78K0Rマイクロコントローラ　命令編　ユーザーズ・マニュアル		PDF
RA78K0R　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
CC78K0R　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，78K0R/KG3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     78K0R/KG3シリーズ
;
;*******************************************************************************
;     78K0R/KG3シリーズ        サンプル・プログラム
;*******************************************************************************
;     サブシステム・クロックを使用したボー・レート補正付きUART
;*******************************************************************************
;【履歴】
;     2007.11.--       新規作成
;*******************************************************************************
;
;【概要】
;
;このサンプル・プログラムは，シリアル・アレイ・ユニットのUARTの機能を使用したUART
;送受信を行うプログラムです。まず，UART送受信を行う前にキャリブレーション開始処理
;の呼び出しにより，キャリブレーションを行います。キャリブレーションは，水晶振動子
;による周波数精度の良いサブシステム・クロックで動作するリアルタイム・カウンタの定
;期割り込みを用い精度の良い0.5sの時間を計測し，タイマ・アレイ・ユニットのキャプチ
;ャ・モードでその間の高速内蔵発振クロックでのカウント数を計測します。計測したタイ
;マ・カウント値を高速内蔵発振クロックを動作クロックとしているシリアル・アレイ・ユ
;ニットの転送ボー・レート用に計算し，設定します。キャリブレーション後，マスタ機器
;からA/Dコンバータのチャネルと分解能の指定値を受信し，指定されたA/Dコンバータのチ
;ャネルで変換したA/D値を指定された分解能でマスタ機器へ送信します。
;このサンプルプログラムのUARTは，データ長が8ビット，転送レートが9600bps，データ位
;相が正転出力，パリティ・ビットなし，ストップ・ビットが1ビット付加，データ方向が
;LSBファースト転送となるように設定し，半2重での送受信としています。
;
;
;＜使用する周辺の初期設定の主な内容＞
;
;   ・割り込みの禁止
;   ・CPU／周辺ハードウェア・クロックを高速内蔵発振クロックの動作に設定
;   ・ポートの設定
```

```
;    ・A/Dコンバータの設定
;    ・SAU0のチャネル0をUART0送信用，チャネル1をUART0受信用に設定
;    ・割り込みの許可
;    ・サブシステム・クロックの発振安定待ち。リアルタイム・カウンタ（RTC）を0.5sに一度の定周期割り込み
       に設定
;
;
;＜メイン処理の主な内容＞
;
;    ・UART0のエラー・チェック
;    ・UART0受信データの解析
;    ・A/D値の取り込み
;    ・UART0送信データの作成
;    ・UART0送受信動作の開始
;
;
;＜キャリブレーション開始処理の内容＞
;
;    ・TAU0のチャネル0をキャプチャ・モードに設定
;    ・RTCを0.5sに1度の定周期割り込みに設定
;    ・TAU0のチャネル0をキャプチャ動作とRTCの定周期割り込み動作の開始
;
;
;＜INTST0割り込み処理（UART0の送信完了割り込み）＞
;    ・UART0送信データのカウント
;    ・UART0送信データの設定
;    ・UART0送信動作の停止
;
;
;＜INTSR0割り込み処理（UART0受信完了割り込み）＞
;
;    ・UART0受信データのカウント
;    ・UART0受信データの保存
;    ・UART0受信動作の停止
;
;
;＜INTRTC割り込み処理（リアルタイム・カウンタの定周期信号割り込み）＞
;
;    ・TAU0のチャネル0のキャプチャ動作とRTCの定周期割り込み動作の停止
;    ・TAU0のチャネル0のキャプチャ･モードで計測した0.5s間の高速内蔵発振クロックの
;      タイマ･カウント値からUART0の転送ボー・レートの設定
;    ・UART0受信動作の開始
;
```

```
;
;*******************************************************************************



;===============================================================================
;
;     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)   INTWDTI
      DW      RESET_START                 ;(06)   INTLVI
      DW      RESET_START                 ;(08)   INTP0
      DW      RESET_START                 ;(0A)   INTP1
      DW      RESET_START                 ;(0C)   INTP2
      DW      RESET_START                 ;(0E)   INTP3
      DW      RESET_START                 ;(10)   INTP4
      DW      RESET_START                 ;(12)   INTP5
      DW      RESET_START                 ;(14)   INTST3
      DW      RESET_START                 ;(16)   INTSR3
      DW      RESET_START                 ;(18)   INTSRE3
      DW      RESET_START                 ;(1A)   INTDMA0
      DW      RESET_START                 ;(1C)   INTDMA1
      DW      INT_UARTST                  ;(1E)   INTST0/INTCSI00
      DW      INT_UARTSR                  ;(20)   INTSR0/INTCSI01
      DW      RESET_START                 ;(22)   INTSRE0
      DW      RESET_START                 ;(24)   INTST1/INTCSI10/INTIIC10
      DW      RESET_START                 ;(26)   INTSR1
      DW      RESET_START                 ;(28)   INTSRE1
      DW      RESET_START                 ;(2A)   INTIIC0
      DW      RESET_START                 ;(2C)   INTTM00
      DW      RESET_START                 ;(2E)   INTTM01
      DW      RESET_START                 ;(30)   INTTM02
      DW      RESET_START                 ;(32)   INTTM03
      DW      RESET_START                 ;(34)   INTAD
      DW      INT_RTC                     ;(36)   INTRTC
      DW      RESET_START                 ;(38)   INTRTCI
      DW      RESET_START                 ;(3A)   INTKR
      DW      RESET_START                 ;(3C)   INTST2/INTCSI20/INTIIC20
      DW      RESET_START                 ;(3E)   INTSR2
      DW      RESET_START                 ;(40)   INTSRE2
```

```
        DW      RESET_START                     ;(42)   INTTM04
        DW      RESET_START                     ;(44)   INTTM05
        DW      RESET_START                     ;(46)   INTTM06
        DW      RESET_START                     ;(48)   INTTM07
        DW      RESET_START                     ;(4A)   INTP6
        DW      RESET_START                     ;(4C)   INTP7
        DW      RESET_START                     ;(4E)   INTP8
        DW      RESET_START                     ;(50)   INTP9
        DW      RESET_START                     ;(52)   INTP10
        DW      RESET_START                     ;(54)   INTP11

TBRK    CSEG    AT      00007EH
        DW      RESET_START                     ;(7E)   BRK

;=============================================================================
;
;       スタック領域の確保
;
;=============================================================================
DSTK    DSEG    AT      0FEFC0H
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＝FEFE0H

;=============================================================================
;
;       ＲＡＭの定義
;
;=============================================================================
DMAIN DSEG    SADDRP
RRXBUF:         DS      2                       ;受信データ・バッファ
RRXBUFE:

RRXBUFP:        DS      1                       ;受信データ・カウンタ

RTXBUF:         DS      2                       ;送信データ・バッファ
RTXBUFE:

RTXBUFP:        DS      1                       ;送信データ・カウンタ

RCALIBST:       DS      1                       ;キャリブレーション状態
 CCALIB_RESET EQU       0               ; キャリブレーション未終了状態
 CCALIB_END             EQU     1               ; キャリブレーション完了状態
```

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

      DI                                    ;割り込み禁止
;-------------------------------------------------------------------------------
;     レジスタバンク設定
;-------------------------------------------------------------------------------
      SEL     RB0
;-------------------------------------------------------------------------------
;     スタック・ポインタの設定
;-------------------------------------------------------------------------------
      MOVW    SP,     #LOWW STACKTOP  ;スタック・ポインタを設定

;-------------------------------------------------------------------------------
;     クロック周波数の設定
;-------------------------------------------------------------------------------
;     高速内蔵発振クロックで動作が行えるように設定します。
;-------------------------------------------------------------------------------
      MOV     CMC,    #00010000B            ;クロック動作モード
                      ;|||||||+--------------- AMPH: 2MHz≦fMX≦10MHz
                      ;||||+++---------------- <000>
                      ;|||+------------------- OSCSELS: XT1発振モード
                      ;||+-------------------- <0>
                      ;++--------------------- EXCLK/OSCSEL: 入力ポート・モード

      MOV     CSC,    #10000000B            ;クロック動作ステータス制御
                      ;|||||||+--------------- HIOSTOP: 高速内蔵発振回路動作
                      ;||+++++---------------- <00000>
                      ;|+---------------------- XTSTOP: XT1発振回路動作
                      ;+----------------------- MSTOP: X1発振回路停止

      MOV     OSMC,   #00000000B            ;動作スピード・モード
                      ;|||||||+--------------- FSEL: 10MHz以下の周波数で動作
                      ;+++++++---------------- <00000>
```

```
      MOV     CKC,    #00001000B              ;クロック選択
                      ;|||||+++-------------- MDIV2-0: CPU/周辺ハードウエア・クロック(fCLK)=fIH
                      ;||||+----------------- <1>
                      ;|||+------------------ MCM0: 高速内蔵発振クロック(fIH)
                      ;||+------------------- <R>
                      ;|+-------------------- CSS: メイン・システム・クロック(fMAIN)=fCLK
                      ;+--------------------- <R>

;-------------------------------------------------------------------------------
;     ポート0の設定
;-------------------------------------------------------------------------------
      MOV     P0,     #00000000B              ;P00-P06の出力ラッチLow
      MOV     PM0,    #10000000B              ;P00-P06を出力ポートに設定

;-------------------------------------------------------------------------------
;     ポート1の設定
;-------------------------------------------------------------------------------
      MOV     P1,     #00000000B              ;P10-P17の出力ラッチLow
                                              ;[後でP12(TxD0)の設定を行う]
      MOV     PM1,    #00000000B              ;P10-P17を出力ポートに設定
                                              ;[後でP11(RxD0)の設定を行う]

;-------------------------------------------------------------------------------
;     ポート2の設定
;-------------------------------------------------------------------------------
      MOV     P2,     #00000000B              ;P20-P27の出力ラッチLow
      MOV     PM2,    #11111111B              ;P20-P27(ANI0-ANI7)を入力ポートに設定

;-------------------------------------------------------------------------------
;     ポート3の設定
;-------------------------------------------------------------------------------
      MOV     P3,     #00000000B              ;P30-P31の出力ラッチLow
      MOV     PM3,    #11111100B              ;P30-P31を出力ポートに設定

;-------------------------------------------------------------------------------
;     ポート4の設定
;-------------------------------------------------------------------------------
      MOV     P4,     #00000000B              ;P40-P47の出力ラッチLow
      MOV     PM4,    #00000000B              ;P40-P47を出力ポートに設定
```

```
;-------------------------------------------------------------------------------
;     ポート5の設定
;-------------------------------------------------------------------------------
      MOV     P5,     #00000000B              ;P50-P57の出力ラッチLow
      MOV     PM5,    #00000000B              ;P50-P57を出力ポートに設定

;-------------------------------------------------------------------------------
;     ポート6の設定
;-------------------------------------------------------------------------------
      MOV     P6,     #00000000B              ;P60-P67の出力ラッチLow
      MOV     PM6,    #00000000B              ;P60-P67を出力ポートに設定

;-------------------------------------------------------------------------------
;     ポート7の設定
;-------------------------------------------------------------------------------
      MOV     P7,     #00000000B              ;P70-P77の出力ラッチLow
      MOV     PM7,    #00000000B              ;P70-P77を出力ポートに設定

;-------------------------------------------------------------------------------
;     ポート8の設定
;-------------------------------------------------------------------------------
      MOV     P8,     #00000000B              ;P80-P87の出力ラッチLow
      MOV     PM8,    #00000000B              ;P80-P87を出力ポートに設定

;-------------------------------------------------------------------------------
;     ポート11の設定
;-------------------------------------------------------------------------------
      MOV     P11,    #00000000B              ;P110-P111の出力ラッチLow
      MOV     PM11,   #11111100B              ;P110-P111を出力ポートに設定

;-------------------------------------------------------------------------------
;     ポート12の設定
;-------------------------------------------------------------------------------
      MOV     P12,    #00000000B              ;P120の出力ラッチLow
      MOV     PM12,   #11111110B              ;P120を出力ポートに設定

;-------------------------------------------------------------------------------
;     ポート13の設定
;-------------------------------------------------------------------------------
      MOV     P13,    #00000000B              ;P130-P131の出力ラッチLow
      MOV     PM13,   #11111100B              ;P130-P131を出力ポートに設定
```

```
;-------------------------------------------------------------------------------
;     ポート14の設定
;-------------------------------------------------------------------------------
      MOV     P14,    #00000000B              ;P140-P145の出力ラッチLow
      MOV     PM14,   #11000000B              ;P140-P145を出力ポートに設定

;-------------------------------------------------------------------------------
;     ポート15の設定
;-------------------------------------------------------------------------------
      MOV     P15,    #00000000B              ;P150-P157の出力ラッチLow
      MOV     PM15,   #00000000B              ;P150-P157を出力ポートに設定

;-------------------------------------------------------------------------------
;     A/Dコンバータの設定
;-------------------------------------------------------------------------------
      SET1    !ADCEN                          ;ADコンバータ使用の許可

      MOV     ADPC,   #00000000B              ;A/Dポート・コンフィギュレーション
                      ;|||+++++--------------- ADPC4-0: 使用するP20-P27をすべてアナログ入力に切り替え
                      ;+++-------------------- <000>

      MOV     ADM,    #00100000B              ;変換時間の選択
                      ;|||||||+--------------- ADCE: コンパレータの動作停止
                      ;||+++++---------------- FR2-0,LV1-0: fCLK=8MHzで最速の変換時間(8.25us)を選択
                      ;|+--------------------- <0>
                      ;+---------------------- ADCS: 変換動作停止

;-------------------------------------------------------------------------------
;     UART0の設定
;-------------------------------------------------------------------------------
      SET1    !SAU0EN                         ;UART0のあるSAU0使用許可

      NOP                                     ;4クロック分のウェイト
      NOP                                     ;(詳細はユーザーズ・マニュアルを参照してください。)
      NOP
      NOP

      MOV     !SPS0L, #00000010B              ;プリスケーラ(動作クロック)の設定
                      ;||||++++--------------- PRS003-000(CK00): fCLK/2^2
                      ;++++------------------- PRS013-010(CK01): 未使用
```

```
;--------------------------------------
; UART0送信設定(SAU0のチャネル0使用)
;--------------------------------------
MOV     RTXBUFP,#0                      ;送信データ・カウンタ初期化

MOVW    AX,     #0000000000100010B      ;動作モードなどの設定
                ;|||||||||||||||+------- MD000: 転送完了割り込み
                ;|||||||||||||++-------- MD002-001: UARTモード
                ;||||||||||+++---------- <100>
                ;|||||||||+------------- SIS000: 立ち下がりエッジをスタートビットとして検出
                ;||||||||+-------------- <0>
                ;|||||||+--------------- STS00: UART送信
                ;||+++++---------------- <00000>
                ;|+---------------------- CCS00: CKS00ビットで指定した動作クロックMCK
                ;+----------------------- CKS00: PRSで選択した動作クロックCK00(MCK)
MOVW    !SMR00, AX

MOVW    AX,     #1000000010010111B      ;通信フォーマットの設定
                ;|||||||||||||+++------- DLS002-000: 8ビット・データ長
                ;||||||||||||+---------- <0>
                ;||||||||||++----------- SLC001-000: ストップ・ビット長 = 1ビット
                ;|||||||||+------------- <0>
                ;||||||||+-------------- DIR00: LSBファーストで出力
                ;||||||++---------------- PTC001-000: パリティ・ビットを出力しない
                ;|||||+------------------ EOC00: UART送信時はエラー割り込み未使用
                ;||||+------------------- <0>
                ;||++--------------------- DAP00/CKP00: UARTモードでは未使用
                ;++----------------------- TXE00/RXE00: 送信のみを行う
MOVW    !SCR00, AX

; SDR00 転送ボー・レートの設定はINTRTC割り込み処理にて設定

MOV     SOL0L,  #00000000B              ;出力データのレベルの設定(反転なし)

MOVW    AX,     #0000000000000001B      ;初期出力レベルの設定
                ;|||||||||||||+++------- SO02-0: チャネルnのシリアル・クロック出力
                ;||||||||+++++---------- <00001>
                ;|||||+++---------------- CKO02-0: チャネルnのシリアル・データ出力
                ;+++++------------------- <00001>
MOVW    !SO0,   AX
SET1    !SOE0L.0                        ;SOE00:シリアル出力許可
```

```
        SET1    P1.2                            ;P12=TxD0

        ;---------------------------------------
        ; UART0受信設定(SAU0のチャネル1使用)
        ;---------------------------------------
        MOV     RRXBUFP,#0                      ;受信データ・カウンタ初期化

        SET1    PM1.1                           ;P11=RxD0

        MOVW    AX,     #00000001001000010B     ;動作モードなどの設定
                        ;|||||||||||||||+------- MD010: 転送完了割り込み
                        ;|||||||||||||++-------- MD012-011: UARTモード
                        ;|||||||||+++----------- <100>
                        ;|||||||||+------------- SIS010: 立ち下がりエッジをスタートビットとして検出
                        ;||||||||+-------------- <0>
                        ;|||||||+--------------- STS01: RxD端子の有効エッジ(UART受信設定)
                        ;||+++++---------------- <00000>
                        ;|+---------------------- CCS01: CKS00ビットで指定した動作クロックMCK
                        ;+----------------------- CKS01: PRSで選択した動作クロックCK00(MCK)
        MOVW    !SMR01, AX

        MOVW    AX,     #0100010010010111B      ;通信フォーマットの設定
                        ;|||||||||||||+++------- DLS012-010: 8ビット・データ長
                        ;||||||||||||+---------- <0>
                        ;|||||||||++------------ SLC011-010: ストップ・ビット長 = 1ビット
                        ;|||||||||+------------- <0>
                        ;||||||||+-------------- DIR01: LSBファーストで入力を行う
                        ;||||||++--------------- PTC011-010: パリティなしで受信
                        ;|||||+----------------- EOC01: UART受信のエラー割り込み許可
                        ;||||+------------------ <0>
                        ;||++-------------------- DAP01/CKP01: UARTモードでは未使用
                        ;++---------------------- TXE01/RXE01: 受信のみを行う
        MOVW    !SCR01, AX

        ; SDR01 転送ボー・レートの設定はINTRTC割り込み処理にて設定


        EI                                      ;割り込み許可

;-------------------------------------------------------------------------------
;       サブシステム・クロックの発振安定待ち(本例は約3秒ですが，実装回路上で
;       評価したうえで，発振安定まで待ってください)。
;-------------------------------------------------------------------------------
```

```
      MOV     C, #0H
WAIT1:
      MOVW    AX, #0H
WAIT:
      INCW    AX
      CMPW    AX, #0FFFFH
      BC      $WAIT

      INC     C
      MOV     A,C
      CMP     A, #3CH
      BC      $WAIT1

;-------------------------------------------------------------------------------
;     RTCを0.5sに1度の周期割り込みに設定します。
;-------------------------------------------------------------------------------

      SET1    !RTCEN                          ;RTC使用許可

      CLR1    RTCE                            ;カウンタ動作禁止
      MOV     RTCC0,  #00000001B              ;RTCコントロール
                      ;|||||+++--------------- CT2-0: 0.5sに1度の定期割り込み
                      ;||||+------------------ AMPM: 未使用
                      ;|||+------------------- RCL0E0: 未使用
                      ;||+-------------------- RCL0E1: 未使用
                      ;|+--------------------- <0>
                      ;+---------------------- RTCE: カウンタ動作停止

      MOV     SEC,    #0                      ;秒カウント(RSUBC)クリア

      CLR1    RTCIF                           ;割り込み要求クリア
      SET1    RTCMK                           ;割り込み禁止

;-------------------------------------------------------------------------------
;     キャリブレーション開始
;-------------------------------------------------------------------------------
      CALL    !!SCALIBSTT
```

```
;*******************************************************************************
;
;     メイン処理
;
;*******************************************************************************
MAIN_LOOP:
      ;-------------------------------------
      ; 動作状態による分岐
      ;-------------------------------------
      CMP     RCALIBST,#CCALIB_END          ;キャリブレーション完了?
      BZ      $LMAIN100                     ; Yes,
      BR      LMAINRET                      ; No,
LMAIN100:
      MOV     A,      SE0L                  ;動作停止状態/停止状態を確認
      BT      A.0,    $LMAINRET             ;SE00:UART0送信動作停止中? No,
      BT      A.1,    $LMAINRET             ;SE01:UART0受信動作停止中? No,

      ;*************************************
      ; UART0受信完了
      ;*************************************
      CMP     RRXBUFP,#RRXBUFE-RRXBUF ;最終データの受信完了?
      BC      $LMAINTX                      ; No,
      MOV     RRXBUFP,#0                    ;受信データ・カウンタ初期化

      MOV     A,      SSR01L                ;UART0のステータスを確認
      AND     A,      #00000111B            ;(フレーミング,パリティ,オーバラン)エラーあり?
      BNZ     $LMAIN800                     ; Yes,

      ;=====================================================================
      ; 送信データの作成
      ;=====================================================================
      ;
      ;サンプル・プログラムで使用する送受信データは，下図に示す2バイトのフォー
      ;マットとしています。
      ;
      ;【受信データ】
      ;   byte|←      +0     →|←      +1     →|
      ;       +----------------+----------------+
      ;       |A/D値分解能の指定|A/Dチャネルの指定|
      ;       |  08H: 8ビット   |   0:ch0 ･･･     |
      ;       |  0AH:10ビット   |    ･･･ 7:ch7    |
      ;       +----------------+----------------+
      ;
```

```
;【送信データ】
;
;   byte|←      +0      →|←      +1      →|
;   bit |  7|6|5|4|3|2|1|0|  7|6|5|4|3|2|1|0|
;       +---+-------------+-----------------+
;       |   |             |  8ビットA/D値    |
;       |   |             | (8ビット指定時) |
;       +---+-------------+-----------------+
;       +---+---------+--------------------+
;       |   |         |     10ビットA/D値    |
;       |   |         |    (10ビット指定時) |
;       +---+---------+--------------------+
;         |
;         +-エラー情報：受信したデータにエラーあり(1)/エラーなし(0)
;
;=====================================================================
;-------------------------------------
; 受信データのチェック
;-------------------------------------
      CMP     RRXBUF, #10             ;A/D値分解能の指定は10ビット?
      BZ      $LMAINRX300                     ; Yes,
      CMP     RRXBUF, #8                      ;A/D値分解能の指定は8ビット?
      BNZ     $LMAINRX400                     ; No,(上記以外はエラー情報設定)
LMAINRX300:
      CMP     RRXBUF+1,#7+1                   ;チャネル指定(0-7)は正常値?
      BC      $LMAINRX500                     ; Yes,
LMAINRX400:
      ;エラー
      MOVW    AX,     #8000H                  ;エラー情報設定
      BR      LMAINRX700
LMAINRX500:
      ;-------------------------------------
      ; A/D値の取得
      ;-------------------------------------
      MOV     A,      RRXBUF+1                ;A/D変換するチャネルの設定
      MOV     ADS,    A

      SET1    ADCE                            ;コンパレータの動作許可(変換待機)

      ;1変換目(不正データ)
      CLR1    ADIF                            ;割り込み要求クリア
      SET1    ADCS                            ;A/D変換動作許可
LMAINRX530:
```

```
        NOP
        BF      ADIF,   $LMAINRX530             ;1us以上ウエイトしないでADCSに1を設定した場合は，
                                                ;最初の変換データを無視します。
                                                ;(詳細はユーザーズ・マニュアルを参照してください。)
        ;2変換目(正規データ)
        CLR1    ADIF                            ;割り込み要求クリア
        SET1    ADCS                            ;A/D変換動作許可
  LMAINRX550:
        NOP
        BF      ADIF,   $LMAINRX550

        CLR1    ADCS                            ;A/D変換動作停止
        CLR1    ADCE                            ;コンパレータの動作停止

        MOVW    AX,     ADCR                    ;10ビットA/D変換結果取得

        SHRW    AX,     6                       ;データのシフト(10ビットA/D変換結果を下位ビット詰め)
        CMP     RRXBUF,#10                      ;A/D値分解能の指定は10ビット?
        BZ      $LMAINRX700                     ; Yes,
        SHRW    AX,     2                       ;データシフト
                                                ;(10ビットA/D変換結果をさらにシフトし，8ビットA/D変換結
果として扱う)
        ;送信データ設定
  LMAINRX700:
        MOV     RTXBUF, A                       ;送信データを送信バッファに設定
        XCH     A,      X
        MOV     RTXBUF+1,A

        ;UART0送信動作開始
        CLR1    STIF0                           ;割り込み要求クリア
        CLR1    STMK0                           ;割り込み許可
        SET1    !SSOL.0                         ;SSOO:UART0送信動作開始

        MOV     A,      RTXBUF                  ;設定した送信データの第一バイト
        MOV     TxD0,   A                       ;データを送信データ・レジスタに設定
        BR      LMAINRET

  LMAINTX:
        ;**************************************
        ; UART0送信完了
        ;**************************************
        CMP     RTXBUFP,#RTXBUFE-RTXBUF ;最終データの送信完了?
        BC      $LMAINRET                       ; No,
```

```
      MOV     RTXBUFP,#0                    ;送信データ・カウンタ初期化
LMAIN800:
      ;UART0受信動作開始
      CLR1    SRIF0                         ;割り込み要求クリア
      CLR1    SRMK0                         ;割り込み許可
      SET1    !SS0L.1                       ;SS01:UART0受信動作開始
LMAINRET:
      BR      MAIN_LOOP                     ; MAIN_LOOPへ

;*******************************************************************************
;
;     INTST0割り込み（UART0の送信完了割り込み）
;
;*******************************************************************************
INT_UARTST:

      SEL     RB1                           ;レジスタバンク切り替え

      INC     RTXBUFP                 ;送信データ・カウンタ更新
      CMP     RTXBUFP,#RTXBUFE-RTXBUF ;最終データ送信終了?
      BNC     $HINTST800                    ; Yes,

      MOV     A,      RTXBUFP         ;送信データ・カウンタ
      MOV     B,      A
      MOV     A,      RTXBUF[B]             ;今回の送信データを取得
      MOV     TxD0,   A                     ;データを送信データ・レジスタに設定
      BR      HINTSTRET
HINTST800:
      ;データ送信終了
      SET1    STMK0                         ;割り込み禁止
      SET1    !ST0L.0                       ;ST00:UART0送信動作停止
HINTSTRET:
      RETI

;*******************************************************************************
;
;     INTSR0割り込み（UART0受信完了割り込み）
;
;*******************************************************************************
INT_UARTSR:

      SEL     RB1                           ;レジスタバンク切り替え
```

```
        MOV     A,      RRXBUFP         ;受信データ・カウンタ
        MOV     B,      A
        MOV     A,      RxD0                    ;受信データ取得
        MOV     RRXBUF[B],A                     ;今回の受信データを格納

        INC     RRXBUFP                 ;受信データ・カウンタ更新
        CMP     RRXBUFP,#RRXBUFE-RRXBUF ;データ受信終了?
        BC      $HINTSRRET                      ; No,
        ;データ受信終了
        SET1    SRMK0                           ;割り込み禁止
        SET1    !ST0L.1                         ;ST01:UART0受信動作停止
HINTSRRET:
        RETI

;*******************************************************************************
;
;       キャリブレーション開始処理
;
;-------------------------------------------------------------------------------
;       精度の良い0.5sを計測するため，サブシステム・クロックを使用したRTCを定周
;       期割り込み機能を使用します。また，その0.5s間の高速内蔵発振クロックでの
;       カウント数を計測するため，TAU0のチャネル0をキャプチャモードに設定し，動
;       作を開始します。
;*******************************************************************************
SCALIBSTT:

        MOV     RCALIBST,#CCALIB_RESET          ;キャリブレーション状態の初期化

        ;-----------------------------------------------
        ; キャプチャの設定
        ;-----------------------------------------------
        ; TAU0のチャネル0を使用し，0.5s間の高速内蔵発振
        ; クロックのカウント数の計測
        ;-----------------------------------------------
        SET1    !TAU0EN                         ;TAU0使用許可

        MOV     TPS0L,  #00000110B              ;タイマ・クロック選択
                        ;||||++++-------------- PRS003-000: fCLK/2^6 (キャプチャ動作で使用)
                        ;++++------------------ PRS013-010: 未使用
```

```
    MOVW    AX,     #0000000000000100B      ;動作モード設定
                    ;||||||||||||++++------- MD003-000: キャプチャ・モード，アップ・カウント
                    ;||||||||||++----------- <0>
                    ;||||||||++------------- CIS001-000: 未使用
                    ;|||||+++--------------- STS002-000: ソフトウェア・トリガ・スタート
                    ;||||+------------------ MASTER00: 単体動作
                    ;|||+------------------- CCS00: CKS00ビットで指定した動作クロックMCK
                    ;|++--------------------- <0>
                    ;+---------------------- CKS00: PRSで選択した動作クロックCK00(MCK)
    MOVW    TMR00,  AX

    CLR1    TMIF00                          ;割り込み要求クリア
    SET1    TMMK00                          ;割り込み禁止

    ;-----------------------------------------------
    ; 定周期割り込みの設定
    ;-----------------------------------------------
    ; RTCを使用したサブシステム・クロックの安定待ち
    ; に使用した設定をそのまま使用します。
    ;-----------------------------------------------
    MOV     SEC,    #0                      ;秒カウント(RSUBC)クリア
    CLR1    RTCIF                           ;割り込み要求クリア
    CLR1    RTCMK                           ;割り込み許可

    ;-----------------------------------------------
    ; 定周期割り込みとキャプチャ動作開始
    ;-----------------------------------------------
    SET1    RTCE                            ;RTCカウンタ動作開始
    SET1    TS0L.0                          ;TS00:TAU0のチャネル0のキャプチャ動作開始(トリガ動作)

    RET

;*******************************************************************************
;
;     INTRTC割り込み処理
;     (INTRTC使用，RTCの0.5sの1度の定周期割り込み)
;
;-------------------------------------------------------------------------------
;     キャリブレーション開始処理の呼び出しから0.5s後に発生する定周期割り込み処
;     理です。TAU0のチャネル0のキャプチャ･モードで計測した0.5s間の高速内蔵発振
;     クロックのタイマ･カウント値からUART0の転送ボー・レートの設定を行います。
;*******************************************************************************
INT_RTC:
```

```
        SEL     RB1                             ;レジスタバンク切り替え

        SET1    TT0L.0                          ;TAU0のチャネル0のキャプチャ動作停止(トリガ動作)

        MOVW    AX,     TCR00                   ;0.5s間の高速内蔵発振クロックでのタイマ・カウント値を取
得

        CLR1    RTCE                            ;RTCカウンタ動作停止
        SET1    RTCMK                           ;割り込み禁止

        ;-------------------------------------------------------------------------------
        ; ボー・レートの算出式
        ;-------------------------------------------------------------------------------
        ; 取得したタイマ・カウント値から転送ボー・レートが9600bpsとなるように
        ; 動作クロックの分周値用に変換
        ;-------------------------------------------------------------------------------
        ;
        ; ボー・レートの算出式は，
        ;
        ;   (ボー・レート) =
        ;         {対象チャネルの動作クロック(MCK)周波数}/(SDRmn[15:9]+1)/2 [bps] ･･･①
        ;
        ;   ※ SDRmn[15:9]は，SDRmnレジスタのビット15-9の値を示します。
        ;
        ; で表されます。(詳細はユーザーズ・マニュアルを参照してください。)
        ;
        ;   ・CPU/周辺ハードウエア・クロック・・・fCLK [Hz]
        ;   ・ボー・レート・・・・・・・・・・・・・9600 [bps]
        ;
        ; として，①式から，
        ;
        ;   (SDRmn[15:9]+1) = {fCLK/2^2}/9600/2 ･･･②
        ;
        ; と表せます。
        ; また，高速内蔵発振クロックの2^6分周で動作するTAU0のチャネル0のキャプチャで
        ; 計測した0.5s間のタイマ・カウント値(AX)から，
        ;
        ;   {2^6/fCLK}*AX = 0.5 [s]
        ;
        ; と表せますので，fCLKは，
        ;
        ;   fCLK=2^7*AX ･･･③
```

```
        ;
        ; となり，②，③式から，
        ;
        ;  SDRmn[15:9] = {2^7*AX/2^2}/9600/2-1 = AX/600-1
        ;
        ; 上記の式より，シリアル・データ・レジスタ(SDR0x)のUART0転送ボー・レート用
        ; 動作クロックの分周値の算出が行えます。
        ;
        ;-------------------------------------------------------------------------
        ; AX/600-1の計算
        MOV     B,      #0                      ;B=AX/600の計算結果
HIRTC300:
        SUBW    AX,     #600
        BC      $HIRTC500
        INC     B
        BR      $HIRTC300
HIRTC500:
        DEC     B                               ;B=(AX/600の計算結果)-1

        ;シリアル・データ・レジスタ(SDR0x)のビット15-9に合わせる
        MOV     A,      B                       ;B=AX/600-1の計算結果(SDRmn[15:9])
        ROL     A,      1                       ;ビット15-9に合わせる
        AND     A,      #11111110B              ;ビット8は0固定
        MOV     X,      #0                      ;SDRmnの下位7-0ビット用の設定

        MOVW    SDR00,  AX                      ;UART0送信用転送ボー・レートの設定
        MOVW    SDR01,  AX                      ;UART0受信用転送ボー・レートの設定

        ;--------------------------------------
        ; UART0受信動作開始
        ;--------------------------------------
        CLR1    SRIF0                           ;割り込み要求クリア
        CLR1    SRMK0                           ;割り込み許可
        SET1    !SS0L.1                         ;SS01:UART0受信動作許可

        MOV     RCALIBST,#CCALIB_END    ;キャリブレーション完了

        RETI

end
```

● main.c （C言語版）

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

    NEC Electronics      78K0R/KG3シリーズ

************************************************************************************
    78K0R/KG3シリーズ サンプル・プログラム
************************************************************************************
    サブシステム・クロックを使用したボー・レート補正付きUART
************************************************************************************
【履歴】
    2007.11.--          新規作成
************************************************************************************

【概要】

このサンプル・プログラムは，シリアル・アレイ・ユニットのUARTの機能を使用したUART
送受信を行うプログラムです。まず，UART送受信を行う前にキャリブレーション開始処理
の呼び出しにより，キャリブレーションを行います。キャリブレーションは，水晶振動子
による周波数精度の良いサブシステム・クロックで動作するリアルタイム・カウンタの定
期割り込みを用い精度の良い0.5sの時間を計測し，タイマ・アレイ・ユニットのキャプチ
ャ・モードでその間の高速内蔵発振クロックでのカウント数を計測します。計測したタイ
マ・カウント値を高速内蔵発振クロックを動作クロックとしているシリアル・アレイ・ユ
ニットの転送ボー・レート用に計算し，設定します。キャリブレーション後，マスタ機器
からA/Dコンバータのチャネルと分解能の指定値を受信し，指定されたA/Dコンバータのチ
ャネルで変換したA/D値を指定された分解能でマスタ機器へ送信します。
このサンプルプログラムのUARTは，データ長が8ビット，転送レートが9600bps，データ位
相が正転出力，パリティ・ビットなし，ストップ・ビットが1ビット付加，データ方向が
LSBファースト転送となるように設定し，半2重での送受信としています。


＜使用する周辺の初期設定の主な内容＞

    ・割り込みの禁止
    ・CPU／周辺ハードウェア・クロックを高速内蔵発振クロックの動作に設定
    ・ポートの設定
    ・A/Dコンバータの設定
    ・SAU0のチャネル0をUART0送信用，チャネル1をUART0受信用に設定
    ・割り込みの許可
    ・サブシステム・クロックの発振安定待ち。リアルタイム・カウンタ（RTC）を0.5sに一度の定周期割り込みに
      設定
```

＜メイン処理の主な内容＞

・UART0のエラー・チェック
・UART0受信データの解析
・A/D値の取り込み
・UART0送信データの作成
・UART0送受信動作の開始

＜キャリブレーション開始処理の内容＞

・TAU0のチャネル0をキャプチャ・モードに設定
・RTCを0.5sに1度の定周期割り込みに設定
・TAU0のチャネル0をキャプチャ動作とRTCの定周期割り込み動作の開始

＜INTST0割り込み処理（UART0の送信完了割り込み）＞
・UART0送信データのカウント
・UART0送信データの設定
・UART0送信動作の停止

＜INTSR0割り込み処理（UART0受信完了割り込み）＞

・UART0受信データのカウント
・UART0受信データの保存
・UART0受信動作の停止

＜INTRTC割り込み処理（リアルタイム・カウンタの定周期信号割り込み）＞

・TAU0のチャネル0のキャプチャ動作とRTCの定周期割り込み動作の停止
・TAU0のチャネル0のキャプチャ･モードで計測した0.5s間の高速内蔵発振クロックの
　タイマ･カウント値からUART0の転送ボー・レートの設定
・UART0受信動作の開始

**********/

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

　前処理指令（#pragma指令）

```
================================================================================*/
#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 */
#pragma interrupt INTST0 fn_intst0          /* 割り込み関数宣言:INTST0 */
#pragma interrupt INTSR0 fn_intsr0          /* 割り込み関数宣言:INTSR0 */
#pragma interrupt INTRTC fn_intrtc          /* 割り込み関数宣言:INTRTC */

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

    関数プロトタイプ宣言

================================================================================*/
static void fn_CalibrationStart(void);

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

    ＲＡＭの定義

================================================================================*/
static unsigned char ucRxBuffer[2];                 /* 受信データ・バッファ */
static unsigned char ucRxBufferCounter;                     /* 受信データ・カウンタ */

static unsigned char ucTxBuffer[2];                 /* 送信データ・バッファ */
static unsigned char ucTxBufferCounter;                     /* 送信データ・カウンタ */

static unsigned char ucCalibrationStatus;                   /* キャリブレーション状態 */
#define       CALIBRATION_STATUS_RESET      0       /*  キャリブレーション未終了状態 */
#define       CALIBRATION_STATUS_END        1       /*  キャリブレーション完了状態 */

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

    使用する周辺の初期設定

*******************************************************************************/
void hdwinit(void)
{
    unsigned char i;
    unsigned int j;

    DI();                                       /* 割り込み禁止 */
```

```
/*-------------------------------------------------------------------------
    クロック周波数の設定
-------------------------------------------------------------------------
    高速内蔵発振クロックで動作が行えるように設定します。
-------------------------------------------------------------------------*/
    CMC   = 0b00010000;                     /* クロック動作モード */
            /*|||||||+----------------------- AMPH: 2MHz≦fMX≦10MHz */
            /*||||+++------------------------ <000> */
            /*|||+-------------------------- OSCSELS: XT1発振モード */
            /*||+--------------------------- <0> */
            /*++---------------------------- EXCLK/OSCSEL: 入力ポート・モード */

    CSC   = 0b10000000;                       /* クロック動作ステータス制御 */
            /*|||||||+----------------------- HIOSTOP: 高速内蔵発振回路動作 */
            /*||+++++------------------------ <00000> */
            /*|+---------------------------- XTSTOP: XT1発振回路動作 */
            /*+----------------------------- MSTOP: X1発振回路動作 */

    OSMC  = 0b00000000;                     /* 動作スピード・モード */
            /*|||||||+----------------------- FSEL: 10MHz以下の周波数で動作 */
            /*+++++++------------------------ <00000> */

    CKC   = 0b00001000;                     /* クロック選択 */
            /*|||||+++----------------------- MDIV2-0: CPU/周辺ハードウエア・クロック(fCLK)=fIH */
            /*||||+------------------------- <1> */
            /*|||+-------------------------- MCM0: 高速内蔵発振クロック(fIH) */
            /*||+--------------------------- <R> */
            /*|+---------------------------- CSS: メイン・システム・クロック(fMAIN)=fCLK */
            /*+----------------------------- <R> */

/*-------------------------------------------------------------------------
     ポート0の設定
-------------------------------------------------------------------------*/
     P0   = 0b00000000;                   /* P00-P06の出力ラッチLow */
     PM0  = 0b10000000;                   /* P00-P06を出力ポートに設定 */

/*-------------------------------------------------------------------------
     ポート1の設定
-------------------------------------------------------------------------*/
     P1   = 0b00000000;                   /* P10-P17の出力ラッチLow */
                                          /* [後でP12(TxD0)の設定を行う] */
     PM1  = 0b00000000;                   /* P10-P17を出力ポートに設定 */
                                          /* [後でP11(RxD0)の設定を行う] */
```

```
/*-----------------------------------------------------------------------------
      ポート2の設定
-----------------------------------------------------------------------------*/
      P2  = 0b00000000;                 /* P20-P27の出力ラッチLow */
      PM2 = 0b11111111;                 /* P20-P27(ANI0-ANI7)を入力ポートに設定 */

/*-----------------------------------------------------------------------------
      ポート3の設定
-----------------------------------------------------------------------------*/
      P3  = 0b00000000;                 /* P30-P31の出力ラッチLow */
      PM3 = 0b11111100;                 /* P30-P31を出力ポートに設定 */

/*-----------------------------------------------------------------------------
      ポート4の設定
-----------------------------------------------------------------------------*/
      P4  = 0b00000000;                 /* P40-P47の出力ラッチLow */
      PM4 = 0b00000000;                 /* P40-P47を出力ポートに設定 */

/*-----------------------------------------------------------------------------
      ポート5の設定
-----------------------------------------------------------------------------*/
      P5  = 0b00000000;                 /* P50-P57の出力ラッチLow */
      PM5 = 0b00000000;                 /* P50-P57を出力ポートに設定 */

/*-----------------------------------------------------------------------------
      ポート6の設定
-----------------------------------------------------------------------------*/
      P6  = 0b00000000;                 /* P60-P67の出力ラッチLow */
      PM6 = 0b00000000;                 /* P60-P67を出力ポートに設定 */

/*-----------------------------------------------------------------------------
      ポート7の設定
-----------------------------------------------------------------------------*/
      P7  = 0b00000000;                 /* P70-P77の出力ラッチLow */
      PM7 = 0b00000000;                 /* P70-P77を出力ポートに設定 */

/*-----------------------------------------------------------------------------
      ポート8の設定
-----------------------------------------------------------------------------*/
      P8  = 0b00000000;                 /* P80-P87の出力ラッチLow */
      PM8 = 0b00000000;                 /* P80-P87を出力ポートに設定 */
```

```
/*-------------------------------------------------------------------------
      ポート11の設定
-------------------------------------------------------------------------*/
      P11  = 0b00000000;                    /* P110-P111の出力ラッチLow */
      PM11 = 0b11111100;                    /* P110-P111を出力ポートに設定 */

/*-------------------------------------------------------------------------
      ポート12の設定
-------------------------------------------------------------------------*/
      P12  = 0b00000000;                    /* P120の出力ラッチLow */
      PM12 = 0b11111110;                    /* P120を出力ポートに設定 */

/*-------------------------------------------------------------------------
      ポート13の設定
-------------------------------------------------------------------------*/
      P13  = 0b00000000;                    /* P130-P131の出力ラッチLow */
      PM13 = 0b11111100;                    /* P130-P131を出力ポートに設定 */

/*-------------------------------------------------------------------------
      ポート14の設定
-------------------------------------------------------------------------*/
      P14  = 0b00000000;                    /* P140-P145の出力ラッチLow */
      PM14 = 0b11000000;                    /* P140-P145を出力ポートに設定 */

/*-------------------------------------------------------------------------
      ポート15の設定
-------------------------------------------------------------------------*/
      P15  = 0b00000000;                    /* P150-P157の出力ラッチLow */
      PM15 = 0b00000000;                    /* P150-P157を出力ポートに設定 */

/*-------------------------------------------------------------------------
    A/Dコンバータの設定
-------------------------------------------------------------------------*/
    ADCEN = 1;                              /* ADコンバータ使用の許可 */

    ADPC  = 0b00000000;                     /* A/Dポート・コンフィギュレーション */
            /*|||+++++------------------------ ADPC4-0: 使用するP20-P27をすべてアナログ入力に切り替え */
            /*+++---------------------------- <000> */
```

```
    ADM   = 0b00100000;                     /* 変換時間の選択 */
            /*|||||||+----------------------- ADCE: コンパレータの動作停止 */
            /*||+++++------------------------ FR2-0,LV1-0: fCLK=8MHzで最速の変換時間(8.25us)を選択 */
            /*|+------------------------------ <0> */
            /*+------------------------------- ADCS: 変換動作停止 */

/*-------------------------------------------------------------------------
    UART0の設定
-------------------------------------------------------------------------*/
    SAU0EN = 1;                             /* UART0のあるSAU0使用許可 */

    NOP();                              /* 4クロック分のウェイト */
    NOP();                              /* (詳細はユーザーズ・マニュアルを参照してください。) */
    NOP();
    NOP();

    SPS0L = 0b00000010;                 /* プリスケーラ(動作クロック)の設定 */
            /*||||++++----------------------- PRS003-000(CK00): fCLK/2^2 */
            /*++++--------------------------- PRS013-010(CK01): 未使用 */

    /*-------------------------------------
     送信設定(チャネル0使用)
    -------------------------------------*/
    ucTxBufferCounter = 0;              /* 送信データ・カウンタのクリア */

    SMR00 = 0b0000000000100010;         /* 動作モードなどの設定 */
            /*|||||||||||||||+---------------- MD000: 転送完了割り込み */
            /*|||||||||||||++----------------- MD002-001: UARTモード */
            /*||||||||||+++------------------- <100> */
            /*|||||||||+---------------------- SIS000: 立ち下がりエッジをスタートビットとして検出 */
            /*||||||||+----------------------- <0> */
            /*|||||||+------------------------ STS00: UART送信 */
            /*||||+++------------------------- <000> */
            /*|||+--------------------------- CCS00: CKS00ビットで指定した動作クロックMCK */
            /*|++---------------------------- <00> */
            /*+----------------------------- CKS00: PRSで選択した動作クロックCK00(MCK) */
```

```
SCR00 = 0b1000000010010111;       /* 通信フォーマットの設定 */
        /*|||||||||||||+++---------- DLS002-000: 8ビット・データ長 */
        /*|||||||||||||+------------- <0> */
        /*||||||||||++--------------- SLC001-000: ストップ・ビット長 = 1ビット */
        /*||||||||+------------------ <0> */
        /*|||||||+------------------- DIR00: LSBファーストで出力 */
        /*||||||++-------------------- PTC001-000: パリティ・ビットを出力しない */
        /*|||||+--------------------- EOC00: UART送信時はエラー割り込み未使用 */
        /*||||+---------------------- <0> */
        /*||++----------------------- DAP00/CKP00: UARTモードでは未使用 */
        /*++------------------------- TXE00/RXE00: 送信のみを行う */

/* SDR00 転送ボー・レートの設定はINTRTC割り込み処理にて設定 */

SOL0L = 0b00000000;                   /* 出力データのレベルの設定(反転なし) */

S00 = 0b0000100000001001; /* 初期出力レベルの設定 */
      /*|||||||||||||+++---------- S002-0: チャネルnのシリアル・クロック出力 */
      /*||||||||+++++------------- <00001> */
      /*|||||+++------------------- CKO02-0: チャネルnのシリアル・データ出力 */
      /*+++++----------------------- <00001> */

SOE0L.0 = 1;                          /* SOE00:シリアル出力許可 */

P1.2  = 1;                            /* P12=TxD0 */

/*-------------------------------------
 UART0受信設定(SAU0のチャネル1使用)
-------------------------------------*/
ucRxBufferCounter = 0;            /* 受信データ・カウンタのクリア */

PM1.1 = 1;                            /* P11=RxD0 */
```

```
    SMR01 = 0b0000000100100010;       /* 動作モード設定 */
            /*|||||||||||||||+---------- MD010: 転送完了割り込み */
            /*|||||||||||||++----------- MD012-011: UARTモード */
            /*||||||||||+++------------- <100> */
            /*|||||||||+---------------- SIS010: 立ち下がりエッジをスタートビットとして検出 */
            /*||||||||+----------------- <0> */
            /*|||||||+------------------ STS01: RxD端子の有効エッジ(UART受信設定) */
            /*||+++++------------------- <00000> */
            /*|+---------------------------- CCS01: CKS00ビットで指定した動作クロックMCK */
            /*+----------------------------- CKS01: PRSで選択した動作クロックCK00(MCK) */

    SCR01 = 0b0100010010010111;       /* 通信フォーマットの設定 */
            /*|||||||||||||+++---------- DLS012-010: 8ビット・データ長 */
            /*||||||||||||+------------- <0> */
            /*||||||||||++-------------- SLC011-010: ストップ・ビット長 = 1ビット */
            /*|||||||||+---------------- <0> */
            /*||||||||+----------------- DIR01: LSBファーストで入力を行う */
            /*||||||++------------------ PTC011-010: パリティなしで受信 */
            /*|||||+-------------------- EOC01: UART受信のエラー割り込み許可 */
            /*||||+--------------------- <0> */
            /*||++---------------------- DAP01/CKP01: UARTモードでは未使用 */
            /*++----------------------- TXE01/RXE01: 受信のみを行う */

    /* SDR01 転送ボー・レートの設定はINTRTC割り込み処理にて設定 */


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

/*---------------------------------------------------------------------------
    サブシステム・クロックの発振安定待ち(本例は約3秒ですが，実装回路で評価した
    うえで，発振安定まで待ってください)。
---------------------------------------------------------------------------*/

      for (i=0; i<=0x27; i++) {
              for (j=0; j<=0xfff0; j++) {
              }
      }

/*---------------------------------------------------------------------------
    RTCを0.5sに1度の定周期割り込みに設定します。
---------------------------------------------------------------------------*/
    RTCEN = 1;                              /* RTC使用許可 */
```

```
    RTCE  = 0;                              /*カウンタ動作禁止 */
    RTCC0 = 0b00000001;                     /* RTCコントロール */
          /*|||||+++----------------------- CT2-0: 0.5sに1度の定期割り込み */
          /*||||+------------------------- AMPM: 未使用 */
          /*|||+-------------------------- RCLOE0: 未使用 */
          /*||+--------------------------- RCLOE1: 未使用 */
          /*|+---------------------------- <0> */
          /*+----------------------------- RTCE: カウンタ動作停止 */

    SEC = 0;                            /* 秒カウント(RSUBC)クリア */

    RTCIF = 0;                              /* 割り込み要求クリア */
    RTCMK = 1;                              /* 割り込み禁止 */

/*-------------------------------------------------------------------------------
    キャリブレーション開始
-------------------------------------------------------------------------------*/
    fn_CalibrationStart();
}

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

    メイン処理

*******************************************************************************/
void main(void)
{
    while (1){
        if((ucCalibrationStatus==CALIBRATION_STATUS_END)&&!(SEOL & 0b00000011)){
        /* キャリブレーションが完了し，UART0の送受信動作を行っていない状態 */
            if(ucRxBufferCounter >= sizeof(ucRxBuffer)){
            /* UART0受信完了 */

                ucRxBufferCounter = 0;      /* 受信データ・カウンタのクリア */

                if(SSR01L & 0b00000111){    /* UART0のステータスを確認 */
                /* エラーあり */
                    /*-- 受信動作開始 --*/
                    SRIF0  = 0;             /* 割り込み要求クリア */
                    SRMK0  = 0;             /* 割り込み許可 */
                    SSOL.1 = 1;             /* SS01:UART0受信動作許可 */
                }
                else{
```

```
/* エラーなし */
    /*====================================================================
     送信データの作成
    ====================================================================

    サンプル・プログラムで使用する送受信データは，下図に示す2バイトのフォー
    マットとしています。

    【受信データ】
      byte|←      +0     →|←      +1     →|
          +-----------------+-----------------+
          |A/D値分解能の指定|A/Dチャネルの指定|
          |  08H: 8ビット   |   0:ch0 ･･･     |
          |  0AH:10ビット   |    ･･･ 7:ch7    |
          +-----------------+-----------------+

    【送信データ】

      byte|←      +0     →|←      +1     →|
      bit |  7|6|5|4|3|2|1|0|  7|6|5|4|3|2|1|0|
          +---+-------------+-----------------+
          |   |             |  8ビットA/D値    |
          |   |             | (8ビット指定時) |
          +---+-------------+-----------------+
          +---+---------+--------------------+
          |   |         |     10ビットA/D値    |
          |   |         |    (10ビット指定時) |
          +---+---------+--------------------+
            |
            +-エラー情報：受信したデータにエラーあり(1)/エラーなし(0)

    ====================================================================*/
    /* 受信データのチェック */
    if((ucRxBuffer[0]==10||ucRxBuffer[0]==8)&&ucRxBuffer[1]<=7){
    /* 受信したデータが正常な場合 */

        /* A/D値の取得 */
        ADS = ucRxBuffer[1];  /* A/D変換するチャネルの設定 */
        ADCE = 1;             /* コンパレータの動作許可(変換待機) */

        /* 1変換目(不正データ) */
        /* 1us以上ウエイトしないでADCSに1を設定した場合は，
           最初の変換データを無視します。
```

```
                (詳細はユーザーズ・マニュアルを参照してください。) */
            ADIF = 0;           /* 割り込み要求クリア */
            ADCS = 1;           /* A/D変換動作許可 */
            while(!ADIF){       /* 変換待ち */
                NOP();
            }
            /* 2変換目(正規データ) */
            ADIF = 0;           /* 割り込み要求クリア */
            ADCS = 1;           /* 動作許可 */
            while(!ADIF){       /* 変換待ち */
                NOP();
            }
            ADCS = 0;           /* A/D変換動作停止 */
            ADCE = 0;           /* コンパレータの動作停止 */

            /* 送信データ設定 */
            if(ucRxBuffer[0]==8){
            /* 8ビット分解能 */
                ucTxBuffer[0] = 0;
                ucTxBuffer[1] = ADCRH;
            }
            else{
            /* 10ビット分解能 */
                ucTxBuffer[0] = (unsigned char)(ADCR >>14);
                ucTxBuffer[1] = (unsigned char)(ADCR >>6);
                /*データのシフト(10ビットA/D変換結果を下位ビット詰めを行っています。*/
            }
        }
        else{
        /* 受信したデータが不正な場合 */
            ucTxBuffer[0] =0x80;  /* エラー通知設定 */
            ucTxBuffer[1] =0x00;
        }
        /* UART0送信動作開始 */
        STIF0  = 0;             /* 割り込み要求クリア */
        STMK0  = 0;             /* 割り込み許可 */
        SS0L.0 = 1;             /* SS00:UART0送信動作開始 */

        TxD0 = ucTxBuffer[0];   /* 送信データの第一バイトのデータを送信データ・レジスタ
に設定 */
    }
}
else if(ucTxBufferCounter >= sizeof(ucTxBuffer)){
```

```
            /* UART0送信完了 */

                ucTxBufferCounter = 0;        /* 送信データ・カウンタのクリア */

                /* UART0受信動作開始 */
                SRIF0  = 0;                   /* 割り込み要求クリア */
                SRMK0  = 0;             /* 割り込み許可 */
                SSOL.1 = 1;                   /* SS01:UART0受信動作開始 */
            }
        }
    }
}

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

    INTST0割り込み（UART0の送信完了割り込み）

*****************************************************************************/
__interrupt void fn_intst0(void)
{
    ucTxBufferCounter++;                      /* 送信データ・カウンタ更新 */

    if(ucTxBufferCounter < sizeof(ucTxBuffer)){
    /* 後続データあり */
        /* 今回の送信データを送信データ・レジスタに設定 */
        TxD0 = ucTxBuffer[ucTxBufferCounter];
                        /* TxD0: データ送信 */
    }
    else{
    /* データ送信終了 */
        STMK0 = 1;                    /* 割り込み禁止 */
        STOL.0 = 1;                   /* ST00:UART0送信動作停止 */
    }
}

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

    INTSR0割り込み（UART0受信完了割り込み）

*****************************************************************************/
__interrupt void fn_intsr0(void)
{
    ucRxBuffer[ucRxBufferCounter] = (unsigned char)SDR01;
```

```
                        /* 今回の受信データを格納 */

    ucRxBufferCounter++;                        /* 受信データ・カウンタ更新 */

    if(ucRxBufferCounter >= sizeof(ucRxBuffer)){
    /* データ受信終了 */
        SRMK0 = 1;                      /* 割り込み禁止 */
        ST0L.1 = 1;                     /* ST01:通信動作停止,RxD0受信禁止 */
    }
}

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

    キャリブレーション開始処理

-----------------------------------------------------------------------------
    精度の良い0.5sを計測するため，サブシステム・クロックを使用したRTCを定周
    期割り込み機能を使用します。また，その0.5s間の高速内蔵発振クロックでの
    カウント数を計測するため，TAU0のチャネル0をキャプチャモードに設定し，動
    作を開始します。
*****************************************************************************/
static void fn_CalibrationStart(void)
{
    ucCalibrationStatus = CALIBRATION_STATUS_RESET;
                        /* キャリブレーション状態の初期化 */

    /*-------------------------------------------
     キャプチャの設定
    -------------------------------------------
     TAU0のチャネル0を使用し，0.5s間の高速内蔵発振
     クロックのカウント数の計測
    -------------------------------------------*/
    TAU0EN = 1;                                 /* TAU0使用許可 */
```

```
    TPS0L = 0b00000110;                         /* タイマクロック選択 */
            /*||||++++------------------------ PRS003-000: fCLK/2^6（キャプチャ動作で使用）*/
            /*++++--------------------------- PRS013-010: 未使用 */

    TMR00 = 0b0000000000000100;         /* 動作モード設定 */
            /*||||||||||||++++--------------- MD003-000: キャプチャ・モード，アップ・カウント */
            /*||||||||||++------------------- <00> */
            /*||||||||++--------------------- CIS001-000: 未使用 */
            /*|||||+++----------------------- STS002-000: ソフトウェア・トリガ・スタート */
            /*||||+------------------------- MASTER00: 単体動作 */
            /*|||+-------------------------- CCS00: CKS00ビットで指定した動作クロックMCK */
            /*|++---------------------------- <00> */
            /*+----------------------------- CKS00: PRSで選択した動作クロックCK00(MCK) */

    TMIF00 = 0;                                 /* 割り込み要求クリア */
    TMMK00 = 1;                         /* 割り込み禁止 */

    /*-----------------------------------------------
     定周期割り込みの設定
    -----------------------------------------------
     RTCを使用したサブシステム・クロックの安定待ち
     に使用した設定をそのまま使用します。
    -----------------------------------------------*/
    SEC = 0;                            /* 秒カウント(RSUBC)クリア */
    RTCIF = 0;                                  /* 割り込み要求クリア */
    RTCMK = 0;                                  /* 割り込み許可 */

    /*-----------------------------------------------
     定周期割り込みとキャプチャ動作開始
    -----------------------------------------------*/
    RTCE   = 1;                                 /* RTCカウンタ動作開始 */
    TS0L.0 = 1;                                 /* TS00:TAU0のチャネル0のキャプチャ動作開始(トリガ動作)
*/
}

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

    INTRTC割り込み処理
    (INTRTC使用，RTCの0.5sの1度の定周期割り込み)

------------------------------------------------------------------------------
    キャリブレーション開始処理の呼び出しから0.5s後に発生する定周期割り込み処
    理です。TAU0のチャネル0のキャプチャ･モードで計測した0.5s間の高速内蔵発振
```

```
    クロックのタイマ･カウント値からUART0の転送ボー・レートの設定を行います。
*******************************************************************************/
__interrupt void fn_intrtc(void)
{

    register unsigned short ushnTimeCnt;      /* タイマ・カウンタ値 */

    TT0L.0 = 1;                               /* TAU0のチャネル0のキャプチャ動作停止(トリガ動作) */

    ushnTimeCnt = TCR00;              /* 0.5s間の高速内蔵発振クロックでのタイマ・カウント値を取得 */

    RTCE  = 0;                                /* RTCカウンタ動作停止 */
    RTCMK = 1;                                /* 割り込み禁止 */

    /*-------------------------------------------------------------------------
     ボー・レートの算出式
    -------------------------------------------------------------------------
     取得したタイマ・カウント値から転送ボー・レートが9600bpsとなるように
     動作クロックの分周値用に変換
    -------------------------------------------------------------------------

     ボー・レートの算出式は，

       (ボー・レート) =
         {対象チャネルの動作クロック(MCK)周波数}/(SDRmn[15:9]+1)/2 [bps] ･･･①

       ※ SDRmn[15:9]は，SDRmnレジスタのビット15-9の値を示します。

     で表されます。(詳細はユーザーズ・マニュアルを参照してください。)

      ・CPU/周辺ハードウエア・クロック・・・fCLK [Hz]
      ・ボー・レート・・・・・・・・・・・・9600 [bps]

     として，①式から，

       (SDRmn[15:9]+1) = {fCLK/2^2}/9600/2 ･･･②

     と表せます。
     また，高速内蔵発振クロックの2^6分周で動作するTAU0のチャネル0のキャプチャで
     計測した0.5s間のタイマ・カウント値(ushnTimeCnt)から，

       {2^6/fCLK}*ushnTimeCnt = 0.5 [s]
```

```
    と表せますので，fCLKは，

      fCLK=2^7*ushnTimeCnt ･･･③

    となり，②，③式から，

     SDRmn[15:9] = {2^7*ushnTimeCnt/2^2}/9600/2-1 = ushnTimeCnt/600-1

    上記の式より，シリアル・データ・レジスタ(SDR0x)のUART0転送ボー・レート用
    動作クロックの分周値の算出が行えます。

    ------------------------------------------------------------------------*/
    /* ushnTimeCnt/600-1の計算結果をボー・レート用クロックの分周値(SDR0x)に設定 */
    /* SDR0nレジスタのビット15-9に合わせこむように設定します。 */

    /* 送受信転送ボー・レートの設定 */
    SDR00 = SDR01 = (ushnTimeCnt/600 - 1)<<9 & 0b1111111000000000;

    /*-------------------------------------
     受信動作開始
    -------------------------------------*/
    SRIF0  = 0;                             /* 割り込み要求クリア */
    SRMK0  = 0;                         /* 割り込み許可 */
    SSOL.1 = 1;                             /* SSO1:UART0受信動作許可 */

    ucCalibrationStatus = CALIBRATION_STATUS_END;     /* キャリブレーション完了 */
}
```

付録B　改版履歴

版　数	発行年月	改版箇所	改版内容
第1版	January 2009	－	－

【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。