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）

タイマ・アレイ・ユニットを使用した9ビットUART通信編

この資料は，サンプル・プログラムの動作概要や使用方法，およびタイマ・アレイ・ユニットのインターバル・タイマ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，タイマ・アレイ・ユニットのインターバル・タイマ機能を使用し，9ビット・データの送受信を行うプログラムです。UART機能を持つハードウエア・マクロと同様の送受信動作をインターバル・タイマ機能と端子機能のみで実現します。

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　U19606JJ1V0AN00（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，タイマ・アレイ・ユニットのインターバル・タイマ機能を使用し，9ビット・データの送受信を行うプログラムです。UART機能を持つハードウエア・マクロと同様の送受信動作をインターバル・タイマ機能と端子機能のみで実現します。

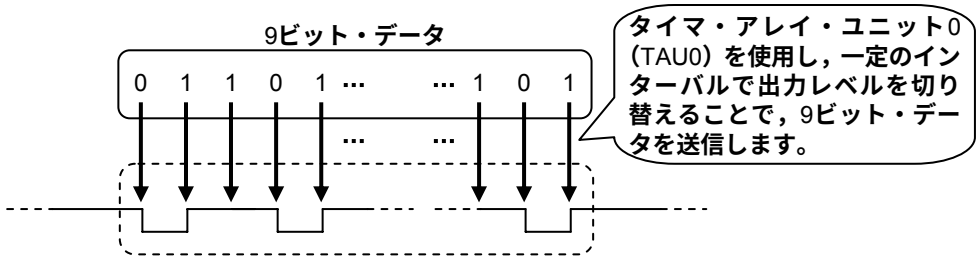
受信の場合，端子入力エッジ割り込みによりスタート・ビットを検出し，インターバル・タイマ動作を開始します。インターバル・タイマ割り込みではUARTの1ビット時間ごとに受信端子のサンプリングを行い，スタート・ビット，データ・ビット，パリティ・ビット，ストップ・ビットを取り込みます。ストップ・ビット取り込み後，エラーチェックを行い，受信完了通知を発行します。また送信の場合，インターバル・タイマ割り込みを1ビット時間ごとに呼び出し，スタート・ビット，送信データ，パリティ・ビット，ストップ・ビットを送信端子に出力します。

なお，このサンプル・プログラムで実現するUART機能は，データ長が9ビット，データ転送方向がLSBファースト，データ位相が正転出力，ストップ・ビットが1ビットで固定となります。また，転送レートは300～19200bps，パリティ・ビットは有または無（有の場合，ゼロ，偶数，奇数）から選択可能です。以上の設定で全2重での送受信を行います。また，送受信するデータは0～511の9ビットの数値です。

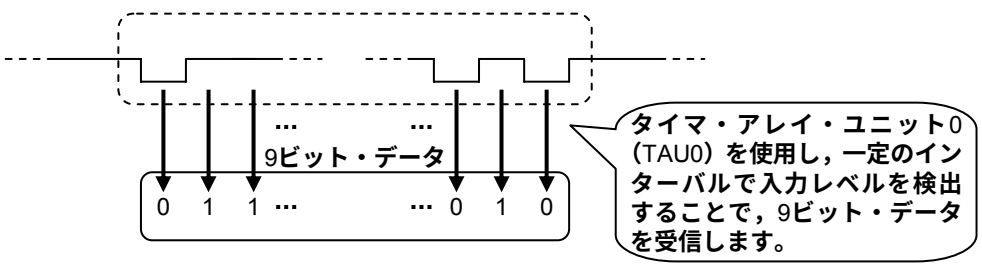
【動作概要】

サンプル・プログラムの動作概要を示します。

・送信動作



・受信動作



(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／周辺ハードウエア・クロックを高速システム・クロック（20MHz使用）の動作に設定
●ポートの設定
●タイマ・アレイ・ユニット0（TAU0）の設定
　・チャネル0をUART受信用に設定
　・チャネル1をUART送信用に設定
●INTP0をUART受信のスタート・ビット検出用に設定
●割り込みの許可

(2) メイン処理の内容

メイン処理の主な内容は，次のとおりです。

・UART受信動作の許可（INTP0割り込み許可）
・UART送信動作の開始（INTTM01割り込み許可，タイマ・チャネル1のカウント開始）

(3) INTP0割り込み処理（INTP0の立ち下がりエッジ割り込み使用）

INTP0割り込み処理の主な内容は，次のとおりです。

・INTP0割り込み禁止
・UART受信動作の開始（タイマ・チャネル0のカウント開始，INTTM00割り込み許可）

(4) INTTM00割り込み処理（タイマ・チャネル0のカウント完了割り込み使用）

INTTM00割り込み処理の主な内容は，次のとおりです。

・スタート・ビットのロウ・レベル確認
・データ・ビット，パリティ・ビットの取り込み
・パリティ・チェック
・UART受信動作の停止（タイマ・チャネル0のカウント停止，INTTM00割り込み禁止）

(5) INTTM01割り込み処理（タイマ・チャネル1のカウント完了割り込み使用）

INTTM01割り込み処理の主な内容は，次のとおりです。

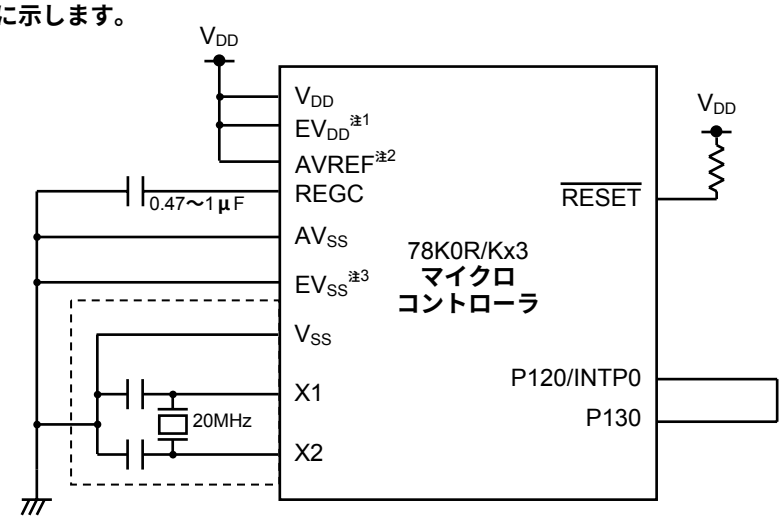
・データ・ビット，パリティ・ビット，およびストップ・ビットの出力
・UART送信動作の停止（タイマ・チャネル1のカウント停止，INTTM01割り込み禁止）
・UART送信データの作成

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}とAV_{REF1}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.　2.7 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（0.47～1μF）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.　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）メイン・システム・クロック（X1，X2）

X1，X2端子に20MHzの発振子を接続します。

(2) 送信端子（P130），受信端子（P120/INTP0）

送信端子（P130）と受信端子（P120/INTP0）を直接接続します。

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，フロー・チャート，およびタイミング・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_9bit_SW.prw	統合開発環境 PM+用ワーク・スペース・ファイル		●
Uart_9bit_SW.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
　　：タイマ・アレイ・ユニット0（TAU0）のチャネル0をインターバル・タイマ・モードで使用

・UART受信用（データ・ビット，パリティ・ビット，およびストップ・ビットの受信）
　　：タイマ・アレイ・ユニット0（TAU0）のチャネル1をインターバル・タイマ・モードで使用

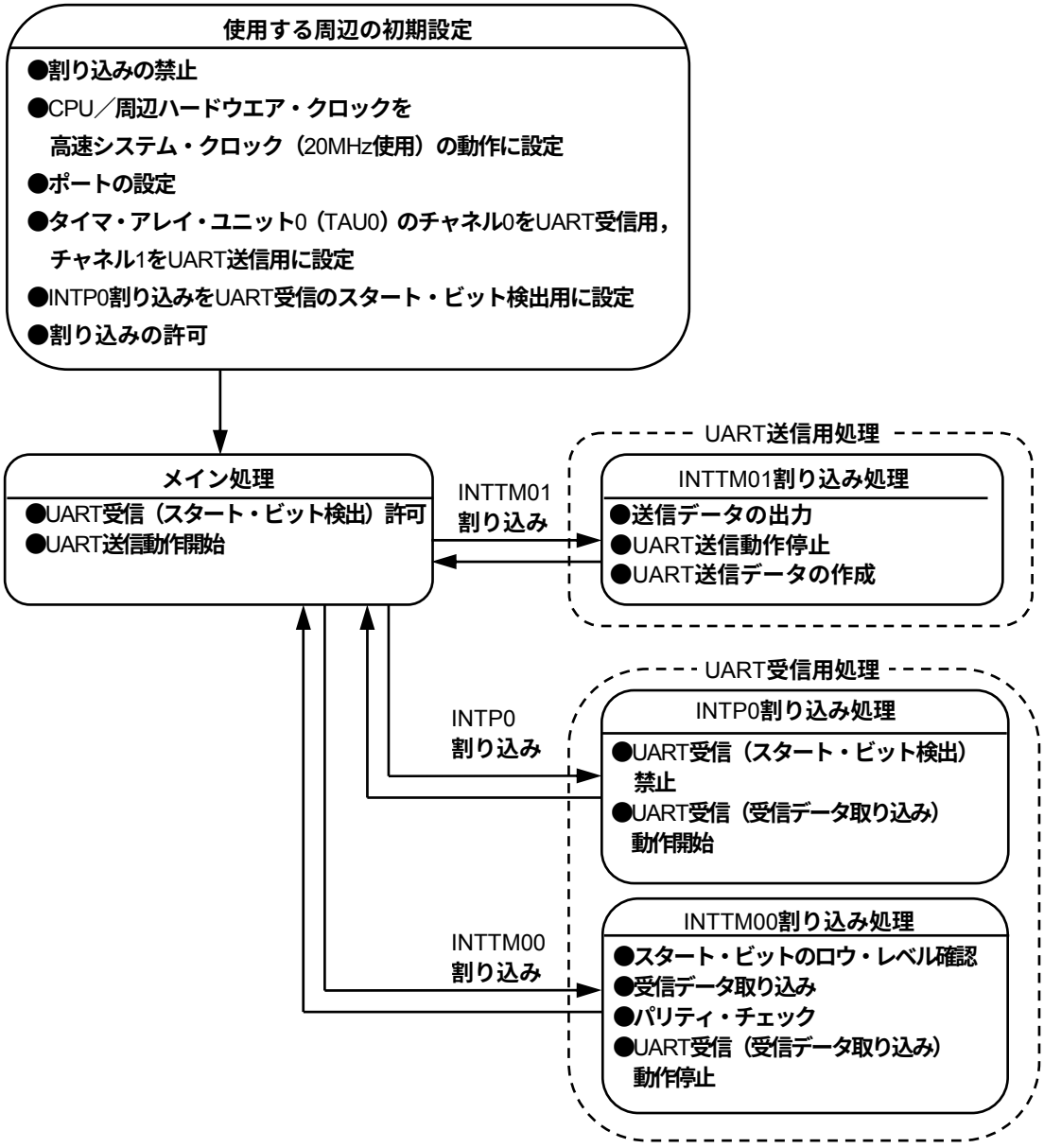
・UART受信用（スタート・ビット検出）
　　：INTP0割り込みを立ち下がりエッジ入力で使用

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，タイマ・アレイ・ユニット0（TAU0）を使用したUART通信のための設定などを行います。なお，このサンプル・プログラムで実現するUART機能は，データ長が9ビット，データ転送方向がLSBファースト，データ位相が正転出力，ストップ・ビットが1ビットで固定となります。また，転送レートは300～19200bps，パリティ・ビットは有または無（有の場合，ゼロ，偶数，奇数）から選択可能です。以上の設定で全2重での送受信を行います。また，送受信するデータは0～511の9ビットの数値です。

使用する周辺の初期設定完了後，メイン処理にてUART受信動作およびUART送信動作を行う割り込み処理が許可され，UART通信動作が開始されます。UART送信動作はINTTM01割り込み処理で行われます。タイマ・アレイ・ユニット0（TAU0）のチャネル0のインターバルを使用して送信データを出力し，全ビット出力した時点で動作は停止します。また，UART受信動作はINTTM00割り込み処理で行われます。INTP0割り込みを使用したスタート・ビット検出によりINTTM00割り込みが許可された後，タイマ・アレイ・ユニット0（TAU0）のチャネル0のインターバルを使用して受信データの取り込みを行います。全ビット取り込んだ時点で動作は停止します。UART送信動作，UART受信動作ともに停止した後，再びメイン処理でUART動作通信が開始されます。

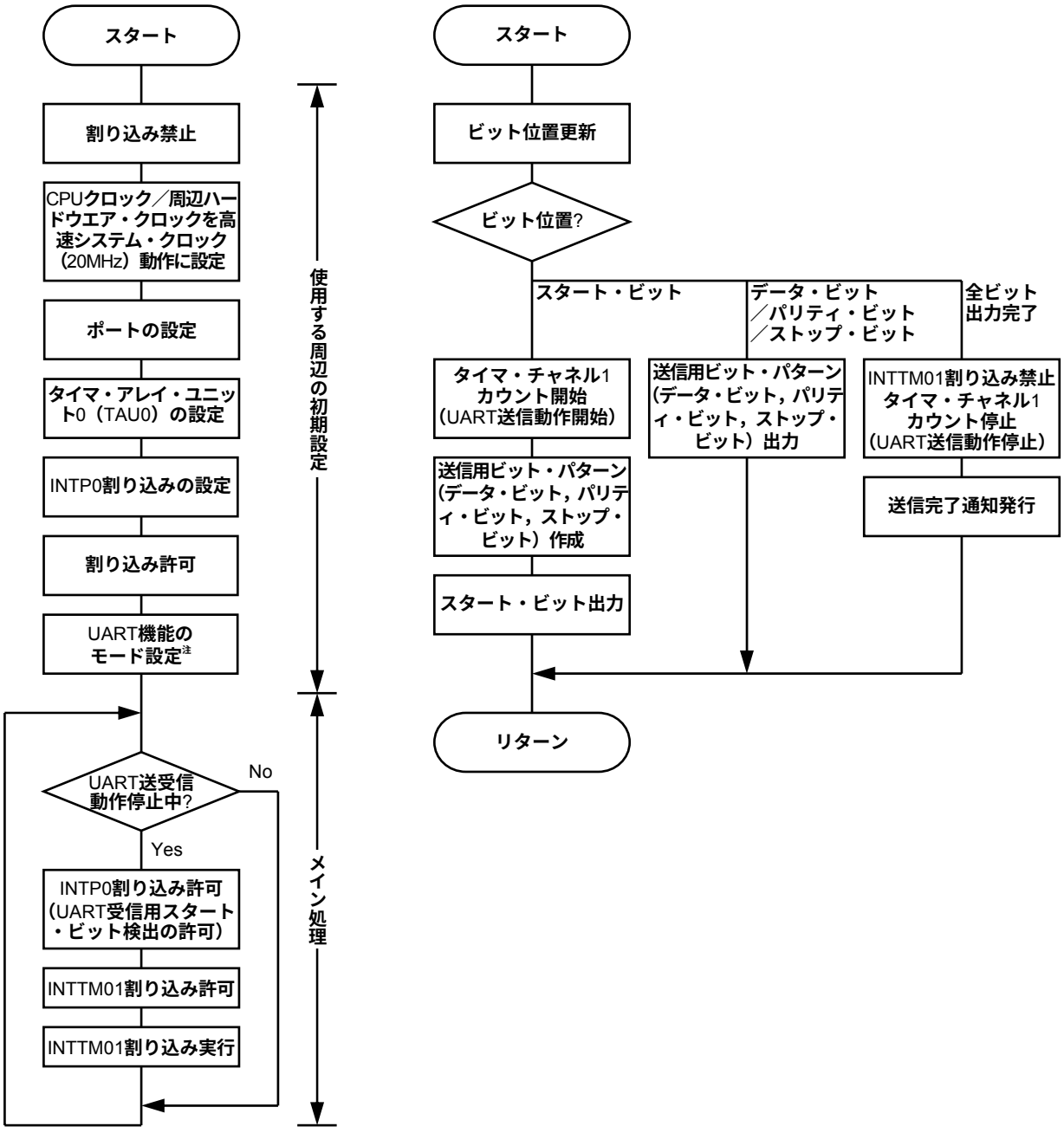
詳細については，次の状態遷移図（ステート・チャート）に示します。



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。

＜リセット解除後の使用する周辺の初期設定＞　＜INTTM01割り込み処理＞



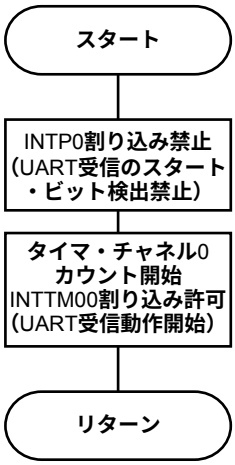
注．　ボー・レートの選択とパリティ・ビットの設定を行います。

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。

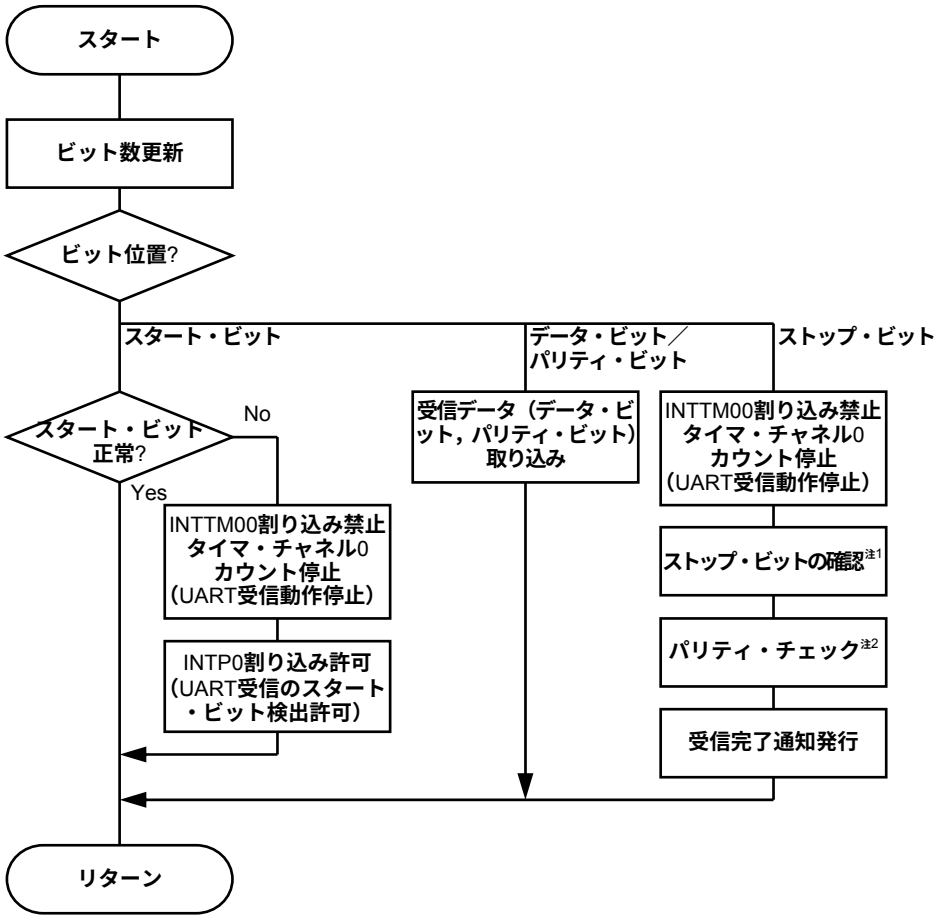
オプション・バイトは，次の内容が設定されます。

- 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

＜INTP0割り込み処理＞



＜INTTM00割り込み処理＞



注1．ストップ・ビットが正常かどうかを判定し，正常でない場合はフレーミング・エラー通知を発行します。

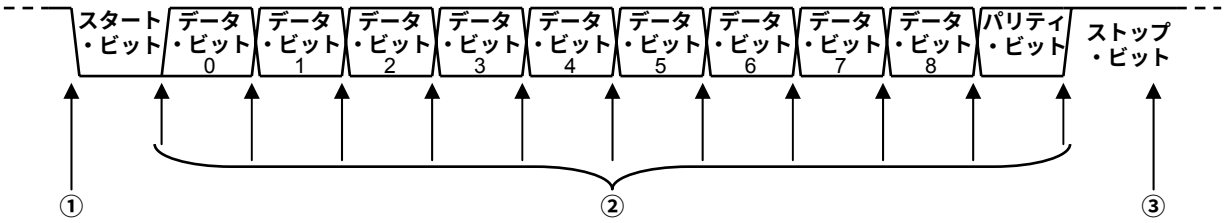
注2．パリティ・ビットの取得，およびパリティ・エラーの判定を行います。

3. 5　タイミング・チャート

このサンプル・プログラムで行うUART通信動作のタイミング・チャートを次に示します。

(1) UART送信動作のタイミング

UART送信動作のタイミングは次のとおりです。



① タイマ・チャネル1によるインターバル動作の開始とINTTM01割り込みの許可により，UART送信動作を開始する。インターバルは1ビット分に設定する。また，このときにスタート・ビットを出力する。

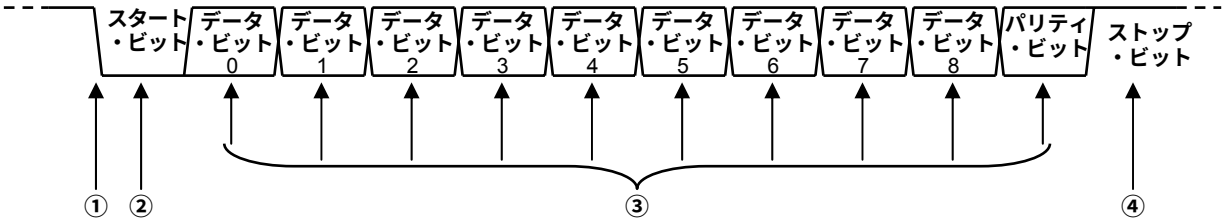
② 1ビット分のインターバルでデータ・ビット，パリティ・ビット，およびストップ・ビットを出力する。

③ インターバル動作の停止，およびINTTM00割り込みの禁止により，UART送信動作を停止する。

(2) UART受信動作のタイミング

UART受信動作のタイミングは次のとおりです。

【パリティ・ビットありの場合】



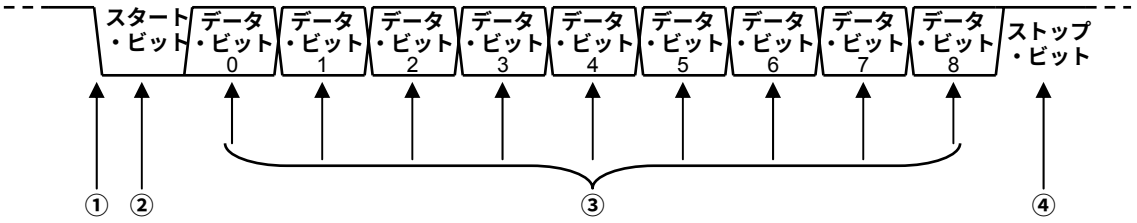
① INTP0の立ち下がりエッジ入力によりスタート・ビットが検出される。タイマ・チャネル0によるインターバル動作の開始とINTTM00割り込みの許可により，UART受信動作が開始される。なお，インターバル周期は半ビット分に設定される。

② スタート・ビット検出から半ビット分が経過し，正常なスタート・ビットが受信されたかどうかの判定が行われる。正常であればインターバル周期を1ビット分に変更した上でUART受信動作を継続する。また，正常でなければインターバル動作の停止とINTTM00割り込みの禁止により，UART受信動作を停止する。

③ 1ビット分のインターバルでデータ・ビットおよびパリティ・ビットの取り込みを行う。

④ インターバル動作の停止とINTTM00割り込みの禁止により，UART受信動作を停止する。また，このときにストップ・ビットの確認とパリティ・チェックを行う。

【パリティ・ビットなしの場合】



① INTP0の立ち下がりエッジ入力によりスタート・ビットが検出される。タイマ・チャネル0によるインターバル動作の開始とINTTM00割り込みの許可により，UART受信動作が開始される。なお，インターバル周期は半ビット分に設定される。

② スタート・ビット検出から半ビット分が経過し，正常なスタート・ビットが受信されたかどうかの判定が行われる。正常であればインターバル周期を1ビット分に変更した上でUART受信動作を継続する。また，正常でなければインターバル動作の停止とINTTM00割り込みの禁止により，UART受信動作を停止する。

③ 1ビット分のインターバルでデータ・ビットの取り込みを行う。

④ インターバル動作の停止とINTTM00割り込みの禁止により，UART受信動作を停止する。また，このときにストップ・ビットの確認を行う。

第4章　設定方法について

この章では，UART通信を行うための設定，ボー・レートについて，使用する周辺の初期設定，メイン処理，INTP0割り込み処理，INTTM00割り込み処理，およびINTTM01割り込み処理について説明します。

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（使用する周辺の初期設定）　LED点灯のスイッチ制御偏　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　UART通信を行うための設定

本サンプル・プログラムでは，UART通信を行うためにタイマ・アレイ・ユニット0（TAU0）のチャネル0，1，およびINTP0割り込みを使用します。

それぞれの機能の設定内容は次のとおりです。

【タイマ・アレイ・ユニット0（TAU0）の設定】

（1）タイマ・アレイ・ユニット0（TAU0）の入力クロックの制御

タイマ・アレイ・ユニット0（TAU0）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の周辺イネーブル・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
注								

タイマ・アレイ・ユニット（TAU0EN）の入力クロックの制御

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

注　78K0R/KE3, 78K0R/KF3, 78K0R/KG3マイクロコントローラの場合、PER0レジスタのビット1には必ず“0”を設定してください。

注意　タイマ・アレイ・ユニットの設定をする際には，必ず最初にTAU0EN = 1の設定を行ってください。TAU0EN = 0の場合は，タイマ・アレイ・ユニットの制御レジスタへの書き込みは無視され，読み出し値もすべて初期値となります。（タイマ入力選択レジスタ0（TIS0），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1（NFEN1），ポート・モード・レジスタ0, 1, 3, 4, 13, 14（PM0, PM1, PM3, PM4, PM13, PM14），ポート・レジスタ0, 1, 3, 4, 13,14（P0, P1, P3, P4, P13, P14）は除く）。

(2) 動作クロック（CK00）の選択

タイマ・アレイ・ユニット0（TAU0）のチャネル0，1，2に供給される2種類の動作クロック（CK00, CK01）を選択するため，タイマ・クロック選択レジスタ0（TPS0）の設定を行います。TPS0のビット7-4でCK01を，ビット3-0でCK00を選択します。

本サンプル・プログラムでは，チャネル0，1ともに動作クロックCK00を使用します。

図4－1－2　タイマ・クロック選択レジスタ0（TPS0）のフォーマット

アドレス：F01B6H，F01B7H

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	0	0	$f_{CLK}/2^{12}$
1	1	0	1	$f_{CLK}/2^{13}$
1	1	1	0	$f_{CLK}/2^{14}$
1	1	1	1	$f_{CLK}/2^{15}$

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	0	0	$f_{CLK}/2^{12}$
1	1	0	1	$f_{CLK}/2^{13}$
1	1	1	0	$f_{CLK}/2^{14}$
1	1	1	1	$f_{CLK}/2^{15}$

注意　ビット15-8には，必ず0を設定してください。

備考　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

タイマ・アレイ・ユニット0（TAU0）で使用する動作クロック（MCK）の選択，タイマ・クロック（TCLK）入力の使用可否，タイマ入力端子の有効エッジの設定，動作モード設定をタイマ・モード・レジスタ00（TMR00）およびタイマ・モード・レジスタ01（TMR01）にて行います。

図4－1－3－1　タイマ・モード・レジスタ00（TMR00）のフォーマット

アドレス：F0190H，F0191H

CKS00	0	0	CCS00	MASTER00	STS002	STS001	STS000	CIS001	CIS000	0	0	MD003	MD002	MD001	MD000

チャネル0の動作モードの設定

MD 003-000				チャネル0の動作モードの設定	TCRのカウント動作	単体動作
0	0	0	1/0	インターバル・タイマ・モード	ダウン・カウント	可
0	1	0	1/0	キャプチャ・モード	アップ・カウント	可
0	1	1	0	イベント･カウンタ・モード	ダウン・カウント	不可
1	0	0	1/0	ワンカウント・モード	ダウン・カウント	可
1	1	0	0	キャプチャ&ワンカウント・モード	アップ・カウント	可

TI00端子の有効エッジ選択

0	0	立ち下がりエッジ
0	1	立ち上がりエッジ
1	0	両エッジ（ロウ・レベル幅測定時） スタート・トリガ：立ち下がりエッジ キャプチャ・トリガ：立ち上がりエッジ
1	1	両エッジ（ハイ・レベル幅測定時） スタート・トリガ：立ち上がりエッジ キャプチャ・トリガ：立ち下がりエッジ

チャネル0のスタート・トリガ，キャプチャ・トリガの設定

0	0	0	ソフトウエア・トリガ・スタートのみ有効 （他のトリガ要因を非選択にする）
0	0	1	TI00端子入力の有効エッジを，スタート・トリガ，キャプチャ・トリガの両方に使用
0	1	0	TI00端子入力の両エッジを，スタート・トリガとキャプチャ・トリガに分けて使用
1	0	0	マスタ・チャネルの割り込み信号を使用 （連動動作機能のスレーブ・チャネル時）
その他			設定禁止

チャネルnの単体動作機能，連動動作機能のスレーブ／連動動作機能のマスタの選択

0	単体動作機能，または連動動作機能でスレーブ・チャネルとして動作
1	連動動作機能でマスタ・チャネルとして動作

チャネル0のカウント・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
1	TI00端子からの入力信号の有効エッジ

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動作クロックCK00
1	PRSレジスタで設定した動作クロックCK01

注意　　ビット14, 13, 5, 4には，必ず0を設定してください。

図4－1－3－2　タイマ・モード・レジスタ01（TMR01）のフォーマット

アドレス：F0192H，F0193H

CKS01	0	0	CCS01	MASTER01	STS012	STS011	STS010	CIS011	CIS010	0	0	MD013	MD012	MD011	MD010

チャネル1の動作モードの設定

MD 013-010				チャネル1の動作モードの設定	TCRのカウント動作	単体動作
0	**0**	**0**	1/**0**	インターバル・タイマ・モード	ダウン・カウント	可
0	1	0	1/0	キャプチャ・モード	アップ・カウント	可
0	1	1	0	イベント･カウンタ・モード	ダウン・カウント	不可
1	0	0	1/0	ワンカウント・モード	ダウン・カウント	可
1	1	0	0	キャプチャ&ワンカウント・モード	アップ・カウント	可

TI01端子の有効エッジ選択

0	**0**	立ち下がりエッジ
0	1	立ち上がりエッジ
1	0	両エッジ（ロウ・レベル幅測定時） スタート・トリガ：立ち下がりエッジ キャプチャ・トリガ：立ち上がりエッジ
1	1	両エッジ（ハイ・レベル幅測定時） スタート・トリガ：立ち上がりエッジ キャプチャ・トリガ：立ち下がりエッジ

チャネル1のスタート・トリガ，キャプチャ・トリガの設定

0	**0**	**0**	ソフトウエア・トリガ・スタートのみ有効 （他のトリガ要因を非選択にする）
0	0	1	TI01端子入力の有効エッジを，スタート・トリガ，キャプチャ・トリガの両方に使用
0	1	0	TI01端子入力の両エッジを，スタート・トリガとキャプチャ・トリガに分けて使用
1	0	0	マスタ・チャネルの割り込み信号を使用 （連動動作機能のスレーブ・チャネル時）
その他			設定禁止

チャネルnの単体動作機能，連動動作機能のスレーブ／連動動作機能のマスタの選択

0	単体動作機能，または連動動作機能でスレーブ・チャネルとして動作
1	連動動作機能でマスタ・チャネルとして動作

チャネル1のカウント・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
1	TI01端子からの入力信号の有効エッジ

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動作クロックCK00
1	PRSレジスタで設定した動作クロックCK01

注意　ビット14, 13, 5, 4には，必ず0を設定してください。

（4）カウント動作開始の許可

タイマのカウント動作開始の許可はチャネルごとに設定するタイマ・チャネル開始レジスタ0（TS0）で行います。タイマ・チャネル開始レジスタ0（TS0）は，トリガ・レジスタですので，タイマのカウント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，タイマ・アレイ・ユニット0（TAU0）のチャネル0のカウント動作開始時にTS00ビットを1，チャネル1のカウント動作開始時にTS01ビットを1にします。

図4－1－4　タイマ・チャネル開始レジスタ0（TS0）のフォーマット

アドレス：F01B2H，F01B3H

0	0	0	0	0	0	0	0	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

チャネルnの動作開始（スタート）トリガ

0	トリガ動作しない
1	カウント動作許可状態になる。

注意　TS0のビット15-8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）

【INTP0割り込みの設定】

（5）INTP0有効エッジの設定

INTP0割り込みをUART受信時のスタート・ビット検出に使用できるように，INTP0の有効エッジを設定します。

本サンプル・プログラムでは立ち下がりエッジを有効エッジとして設定します。

図4－1－5　外部割り込み立ち上がりエッジ許可レジスタ（EGP0），
外部割り込み立ち下がりエッジ許可レジスタ（EGN0）のフォーマット

アドレス：FFF38H

EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

アドレス：FFF39H

EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

INTP0端子の有効エッジの選択

EGP0	EGN0	INTP0端子の有効エッジの選択
0	0	エッジ検出禁止
0	1	立ち下がりエッジ
1	0	立ち上がりエッジ
1	1	立ち上がり，立ち下がりの両エッジ

4.2　ボー・レートについて

このサンプル・プログラムでは，タイマ・アレイ・ユニット0（TAU0）の割り込み周期を選択することにより，UART通信におけるボー・レートを300～19200bpsから選択することができます。割り込み周期の設定は，タイマ・データ・レジスタ00（TDR00）およびタイマ・データ・レジスタ01（TDR01）で行います。

各ボー・レートの設定に必要となるタイマ・データ・レジスタ00（TDR00）およびタイマ・データ・レジスタ01（TDR01）の設定値を次に示します。なお，このサンプル・プログラムでは，タイマ・アレイ・ユニット0（TAU0）のチャネル0およびチャネル1の動作クロック（CK00）をf_{CLK}/2（=10MHz）としています。

(1)　送信用ボー・レートの設定

タイマ・データ・レジスタ00（TDR01）	ボー・レート
33332	300bps
16665	600bps
8332	1200bps
4165	2400bps
2082	4800bps
1387	7200bps
1040	9600bps
693	14400bps
519	19200bps

注意　　動作クロック（CK00）がf_{CLK}/2（=10MHz）の場合の設定です。

(2)　受信用ボー・レートの設定

タイマ・データ・レジスタ01（TDR00）	ボー・レート
33332	300bps
16665	600bps
8332	1200bps
4165	2400bps
2082	4800bps
1387	7200bps
1040	9600bps
693	14400bps
519	19200bps

注意　　動作クロック（CK00）がf_{CLK}/2（=10MHz）の場合の設定です。

4. 3　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／X1クロック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 タイマ・アレイ・ユニット0（TAU0）のチャネル0をUART受信用に設定します。
⑦ タイマ・アレイ・ユニット0（TAU0）のチャネル1をUART送信用に設定します。
⑧ INTP0割り込みをUART受信のスタート・ビット検出用に設定します。
⑨ 割り込みを許可します。

```
;*******************************************************************************
;
;       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
;-------------------------------------------------------------------------------
;       20MHzのX1発振回路で動作が行えるように設定します
;-------------------------------------------------------------------------------
        MOV     CMC,    #01000001B      ;クロック動作モード
                        ;|||||||+--------- AMPH: 10MHz＜fMX≦20MHz
                        ;|||+++----------- <000>
                        ;||+-------------- OSCSELS: P123/P124端子を入力ポート
                        ;||+--------------- <0>
                        ;++---------------- EXCLK/OSCSEL: X1発振モード(20MHz)

        MOV     CSC,    #01000000B      ;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発振回路停止
                        ;+----------------- MSTOP: X1発振回路動作

        MOV     OSMC,   #00000001B      ;動作スピード・モード
④                       ;|||||||+--------- FSEL: 10MHzを越える周波数で動作
                        ;+++++++---------- <00000>

        MOV     OSTS,   #00000101B      ;発振安定時間: 2^15/fX

HRST300:
        NOP
        BF      OSTC.2, $HRST300        ;クロック発振安定待ち

        MOV     CKC,    #00011000B      ;クロック選択
                        ;|||||+++---------MDIV2-0:CPU/周辺ハードウエア・クロック(fCLK)=fMX
                        ;|||||+---------- <1>
                        ;|||+------------ MCM0: 高速システム・クロック（fMX）
                        ;||+------------- <R>
                        ;|+---------------- CSS: メイン・システム・クロック(fMAIN)=fCLK
                        ;+----------------- <R>

;-------------------------------------------------------------------------------
;       ポート0の設定
;-------------------------------------------------------------------------------
⑤       MOV     P0,     #00000000B      ;P00-P07の出力ラッチLow
        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
        MOV     PM1,    #00000000B      ;P10-P17を出力ポートに設定

;-------------------------------------------------------------------------------
;       ポート2の設定
;-------------------------------------------------------------------------------
        MOV     P2,     #00000000B      ;P20-P27の出力ラッチLow
        MOV     PM2,    #00000000B      ;P20-P27を出力ポートに設定

;-------------------------------------------------------------------------------
;       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,   #00011111B      ;P125-P127を出力ポートに設定
                                        ;[後でP120(INTP0)の設定を行う]

;-------------------------------------------------------------------------------
;       ポート13の設定
;-------------------------------------------------------------------------------
        MOV     P13,    #00000000B      ;P130-P137の出力ラッチLow
        MOV     PM13,   #00000000B      ;P131-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


;-------------------------------------------------------------------------------
;       タイマの設定
;-------------------------------------------------------------------------------

        ;-----------------------------------------
        ; TDR00設定値
        ;-----------------------------------------

;;      MOVW    RTDRDATARX,#33332       ;ボー・レート   300bps@20MHz/2^1
;;      MOVW    RTDRDATARX,#16665       ;ボー・レート   600bps@20MHz/2^1
;;      MOVW    RTDRDATARX,#8332        ;ボー・レート  1200bps@20MHz/2^1
;;      MOVW    RTDRDATARX,#4165        ;ボー・レート  2400bps@20MHz/2^1
;;      MOVW    RTDRDATARX,#2082        ;ボー・レート  4800bps@20MHz/2^1
;;      MOVW    RTDRDATARX,#1387        ;ボー・レート  7200bps@20MHz/2^1
;;      MOVW    RTDRDATARX,#1040        ;ボー・レート  9600bps@20MHz/2^1
;;      MOVW    RTDRDATARX,#693         ;ボー・レート 14400bps@20MHz/2^1
        MOVW    RTDRDATARX,#519         ;ボー・レート 19200bps@20MHz/2^1

        ;-----------------------------------------
⑥      ; TM00の設定
        ;-----------------------------------------
        SET1    !TAU0EN                 ;TAU0使用許可

        MOV     TPS0L,  #00000001B      ;タイマ・クロック選択
                        ;||||++++--------- PRS003-000: fCLK/2^1(シリアル・クロックと同じにする)
                        ;++++------------- PRS013-010: 未使用

        MOVW    AX,     #0000000000000000B;動作モード設定
                        ;||||||||||||++++- MD003-000: インターバル・タイマ・モード
                        ;||||||||||++------ <00>
                        ;||||||||++-------- CIS001-000:
                        ;||||+++---------- STS002-000: ソフトウエア・トリガ・スタートのみ有効
                        ;|||+------------- 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)
        MOVW    TMR00,  AX

        CLR1    TMIF00                  ;割り込み要求クリア
;;      CLR1    TMMK00                  ;割り込み許可

;;      SET1    TS0L.0                  ;TS00:TAU0のチャネル0のカウント動作開始(トリガ動作)

        ;-----------------------------------------
        ; TDR01設定値
        ;-----------------------------------------

;;      MOVW    RTDRDATATX,#33332       ;ボー・レート   300bps@20MHz/2^1
;;      MOVW    RTDRDATATX,#16665       ;ボー・レート   600bps@20MHz/2^1
;;      MOVW    RTDRDATATX,#8332        ;ボー・レート  1200bps@20MHz/2^1
;;      MOVW    RTDRDATATX,#4165        ;ボー・レート  2400bps@20MHz/2^1
;;      MOVW    RTDRDATATX,#2082        ;ボー・レート  4800bps@20MHz/2^1
;;      MOVW    RTDRDATATX,#1387        ;ボー・レート  7200bps@20MHz/2^1
;;      MOVW    RTDRDATATX,#1040        ;ボー・レート  9600bps@20MHz/2^1
;;      MOVW    RTDRDATATX,#693         ;ボー・レート 14400bps@20MHz/2^1
        MOVW    RTDRDATATX,#519         ;ボー・レート 19200bps@20MHz/2^1

        ;-----------------------------------------
        ; TM01の設定
        ;-----------------------------------------
⑦ ;;    SET1    !TAU0EN                 ;TAU0使用許可

;;      MOV     TPS0L,  #00000001B      ;タイマ・クロック選択
;;                      ;||||++++--------- PRS003-000: fCLK/2^1(シリアル・クロックと同じにする)
;;                      ;++++------------- PRS013-010: 未使用

        MOVW    AX,     #0000000000000000B;動作モード設定
                        ;||||||||||++++- MD013-010: インターバル・タイマ・モード
                        ;||||||||++------ <00>
                        ;||||||++-------- CIS011-010:
                        ;||||+++---------- STS012-010: ソフトウエア・トリガ・スタートのみ有効
                        ;|||+------------- MASTER01: 単体動作
                        ;|||+------------- CCS01: CKS00ビットで指定した動作クロックMCK
                        ;|++--------------- <00>
                        ;+----------------- CKS01: PRSで選択した動作クロックCK00(MCK)
        MOVW    TMR01,  AX

        CLR1    TMIF01                  ;割り込み要求クリア
```

```
;-------------------------------------------------------------------------------
;         INTP0(UART RX)の設定
;-------------------------------------------------------------------------------

⑧       CLR1     EGP0.0
        SET1     EGN0.0                 ;立ち下がりエッジ有効

        CLR1     PIF0                   ;割り込み要求クリア

⑨       EI                              ;割り込み許可
```

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)
{
    DI();                                   /* 割り込み禁止 */
/*--------------------------------------------------------------------------------
    クロック周波数の設定
--------------------------------------------------------------------------------
    20MHzのX1発振回路で動作が行えるように設定します
--------------------------------------------------------------------------------*/
    CMC = 0b01000001;                       /* クロック動作モード */
          /*|||||||+---------------- AMPH: 10MHz＜fMX≦20MHz */
          /*|||+++------------------ <000> */
          /*||+-------------------- OSCSELS: P123/P124端子を入力ポート */
          /*|+---------------------- <0> */
          /*++---------------------- EXCLK/OSCSEL: X1発振モード(20MHz) */

    CSC = 0b01000000;                       /* 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発振回路停止 */
          /*+----------------------- MSTOP: X1発振回路動作 */

    OSMC = 0b00000001;                      /* 動作スピード・モード */
           /*|||||||+---------------- FSEL: 10MHzを越える周波数で動作 */
           /*+++++++------------------ <00000> */

    OSTS = 0b00000101;                      /* 発振安定時間: 2^15/fX */

    while( (OSTC && 0b00000010) == 0);      /* クロック発振安定待ち */

    CKC = 0b00011000;                       /*クロック選択 */
          /*|||||+++---------------- MDIV2-0: CPU/周辺ハードウエア・クロック(fCLK)=fMX */
          /*|||+-------------------- <1> */
          /*||+---------------------- MCM0: 高速システム・クロック（fMX） */
          /*|+----------------------- <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 */
    PM1 = 0b00000000;                       /* P10-P17を出力ポートに設定 */

/*--------------------------------------------------------------------------------
    ポート2の設定
--------------------------------------------------------------------------------*/
    P2  = 0b00000000;                       /* P20-P27の出力ラッチLow */
    PM2 = 0b00000000;                       /* P20-P27を出力ポートに設定 */

/*--------------------------------------------------------------------------------
    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 */
                                                /* [後でP120(INTP0)の設定を行う] */

/*-------------------------------------------------------------------------------
    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 */


/*-------------------------------------------------------------------------------
    タイマの設定
-------------------------------------------------------------------------------*/


    /*-----------------------------------------
      TM00の設定
    -----------------------------------------*/
    TAU0EN = 1;                                         /* TAU0使用許可 */

    TPS0L = 0b00000001;                                 /* タイマ・クロック選択 */
            /*||||++++---------------- PRS003-000: fCLK/2^1(シリアル・クロックと同じにする)*/
            /*++++-------------------- PRS013-010: 未使用 */

    TMR00 = 0b0000000000000000;     /* 動作モード設定 */
            /*||||||||||||++++-------- MD003-000: インターバル・タイマ・モード */
            /*||||||||||++------------ <00> */
            /*||||||||++-------------- CIS001-000: */
            /*||||+++----------------- STS002-000: ソフトウエア・トリガ・スタートのみ有効 */
            /*|||+-------------------- 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 */
```

```
    /*-----------------------------------------
      TM01の設定
    -----------------------------------------*/

    TMR01 = 0b0000000000000000;             /* 動作モード設定 */
            /*||||||||||||++++-------- MD013-010: インターバル・タイマ・モード */
            /*||||||||||||++---------- <00> */
            /*||||||||||++------------ CIS011-010: */
            /*||||||||+++-------------- STS012-010: ソフトウエア・トリガ・スタートのみ有効 */
            /*||||+++------------------ MASTER01: 単体動作 */
            /*|||+-------------------- CCS01: CKS00ビットで指定した動作クロックMCK */
            /*||++---------------------- <00> */
            /*+------------------------ CKS01: PRSで選択した動作クロックCK00(MCK) */

    TMIF01 = 0;                                         /* 割り込み要求クリア */

/*-------------------------------------------------------------------------------
    INTP0(UART RX)の設定
-------------------------------------------------------------------------------*/

    EGP0.0 = 0;                                         /* 立ち上がりエッジ無効 */
    EGN0.0 = 1;                                         /* 立ち下がりエッジ有効 */

    PIF0 = 0;                                           /* 割り込み要求クリア */

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

}
```

4.4　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 UARTの送受信動作に使用する初期値を設定します。

② UART受信モードを設定します。なお，パリティ・ビットは次のような組み合わせで設定します。

FUARTRXPEN	FUARTRXPZERO	FUARTRXPODD	パリティ・ビット
1	1	無効	パリティ判定を行わない
	0	1	奇数パリティとして判定を行う
		0	偶数パリティとして判定を行う
0	無効	無効	パリティなしで受信

③ UART送信モードを設定します。なお，パリティ・ビットは次のような組み合わせで設定します。

FUARTTXPEN	FUARTTXPZERO	FUARTTXPODD	パリティ・ビット
1	1	無効	ゼロ・パリティを出力
	0	1	奇数パリティを出力
		0	偶数パリティを出力
0	無効	無効	パリティ・ビットを出力しない

④ UARTの送受信動作が停止しているか否かを確認します。停止している場合，⑦の処理に分岐します。

⑤ UART受信動作を開始するための設定を行います。

⑥ UART送信動作を開始するための設定を行います。また，INTTM01割り込みを実行し，送信動作を開始します。

⑦ UARTの送受信動作が完了したタイミングで送受信用のデータを初期化します。

⑧ ④の処理へ分岐します。

```
;*******************************************************************************
;
;       メイン処理
;
;*******************************************************************************
        ;===============================================================================
        ; データ初期化
        ;===============================================================================

①      MOVW    RTXBUF,#0000H                   ;送信データ初期化

        CLR1    FUARTRXEN                       ;受信中解除
        CLR1    FUARTTXEN                       ;送信中解除


        ;===============================================================================
        ; UART受信モードの設定
        ;===============================================================================


        SET1    FUARTRXPEN                      ; 受信パリティ有設定
;;      CLR1    FUARTRXPEN                      ; 受信パリティ無設定

        CLR1    FUARTRXPZERO                    ; 受信パリティをゼロ・パリティ以外に設定
②      SET1    FUARTRXPZERO                    ; 受信パリティをゼロ・パリティに設定

        SET1    FUARTRXPODD                     ; 受信パリティを奇数パリティに設定
;;      CLR1    FUARTRXPODD                     ; 受信パリティを偶数パリティに設定
```

```
            ;=============================================================================
            ; UART送信モードの設定
            ;=============================================================================

            SET1      FUARTTXPEN                        ; 送信パリティ有設定
;;          CLR1      FUARTTXPEN                        ; 送信パリティ無設定

            SET1      FUARTTXPODD                       ; 送信パリティを奇数パリティに設定
;;          CLR1      FUARTTXPODD                       ; 送信パリティを偶数パリティに設定
③
;;          SET1      FUARTTXPZERO                      ; 送信パリティをゼロ・パリティに設定
            CLR1      FUARTTXPZERO                      ; 送信パリティをゼロ・パリティ以外に設定

            SET1      P13.0                             ;データ出力端子初期化
            SET1      PM12.0                            ;データ入力端子設定

            ;=============================================================================
            ; UART送受信実行
            ;=============================================================================
MAIN_LOOP:
            BT        FUARTRXEN,$LMAIN320               ;受信中？, Yes,
            BT        FUARTTXEN,$LMAIN220               ;送信中？, Yes,
④           BT        FUARTRXCMP,$LMAIN320              ;受信完了？,Yes,
            BT        FUARTTXCMP,$LMAIN220              ;送信完了？, Yes,

            ;**************************************
            ; データ受信開始
            ;**************************************

            CLR1      FUARTFRM                          ;フレーミング・エラー クリア
            CLR1      FUARTPRT                          ;パリティ・エラー クリア

            CLR1      FUARTRXCMP                        ;受信完了クリア
⑤           SET1      FUARTRXEN                         ;受信動作開始
            MOVW      RRXREG,#0000H                     ;受信レジスタ クリア
            CLR1      PIF0                              ;スタート･ビット割り込み要求クリア
            CLR1      PMK0                              ;スタート･ビット割り込み許可

            ;**************************************
            ; データ送信開始
            ;**************************************

            MOVW      AX,RTXBUF                         ;送信データ取得
            INCW      AX                                ;送信データ更新
            AND       A,#1H                             ;9ビットに丸める
            MOVW      RTXBUF,AX

            SET1      P13.0                             ;データ出力端子初期化

            MOVW      AX,RTXBUF                         ;送信データ取得
⑥           MOVW      RTXREG,AX                         ;送信データを送信バッファに設定

            CLR1      FUARTTXCMP                        ;送信完了クリア
            SET1      FUARTTXEN                         ;送信中設定
            MOV       RTXCOUNT,#(9 + 4)                 ;ビット・カウント初期化
            CLR1      TMIF01                            ;割り込み要求クリア
            CLR1      TMMK01                            ;割り込み許可
            SET1      TMIF01                            ;割り込み要求セット（送信開始）

            BR        LMAINRET

            ;**************************************
            ; データ送信完了
            ;**************************************
LMAIN220:
            BF        FUARTTXCMP,$LMAIN320              ;送信完了？, No,

            CLR1      FUARTTXCMP                        ;送信完了クリア
            CLR1      FUARTTXEN                         ;送信中解除


            ;**************************************
            ; データ受信完了
            ;**************************************
⑦
LMAIN320:
            BF        FUARTRXCMP,$LMAINRET              ;受信完了？,No,

            MOVW      AX,RRXREG
            MOVW      RRXBUF,AX                         ;受信データ退避

            CLR1      FUARTRXCMP                        ;受信完了クリア
            CLR1      FUARTRXEN                         ;受信動作終了
            CLR1      FUARTFRM                          ;フレーミング・エラー クリア

LMAINRET:
⑧           BR        MAIN_LOOP                         ; MAIN_LOOPへ
```

INTTM01割り込みの実行により，UART送信動作を開始します。

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)
{
/*-----------------------------------------------------------------------------------
    データ初期化
-----------------------------------------------------------------------------------*/

    /*-----------------------------------------
      TDR00設定値
    -----------------------------------------*/

/*    usTdrRx = 33332;      /* ボー・レート   300bps@20MHz/2^1 */
/*    usTdrRx = 16665;      /* ボー・レート   600bps@20MHz/2^1 */
/*    usTdrRx = 8332;       /* ボー・レート  1200bps@20MHz/2^1 */
/*    usTdrRx = 4165;       /* ボー・レート  2400bps@20MHz/2^1 */
/*    usTdrRx = 2082;       /* ボー・レート  4800bps@20MHz/2^1 */
/*    usTdrRx = 1387;       /* ボー・レート  7200bps@20MHz/2^1 */
/*    usTdrRx = 1040;       /* ボー・レート  9600bps@20MHz/2^1 */
/*    usTdrRx = 693;        /* ボー・レート 14400bps@20MHz/2^1 */
      usTdrRx = 519;        /* ボー・レート 19200bps@20MHz/2^1 */


    /*-----------------------------------------
      TDR01設定値
    -----------------------------------------*/

/*    usTdrTx = 33332;      /* ボー・レート   300bps@20MHz/2^1 */
/*    usTdrTx = 16665;      /* ボー・レート   600bps@20MHz/2^1 */
/*    usTdrTx = 8332;       /* ボー・レート  1200bps@20MHz/2^1 */
/*    usTdrTx = 4165;       /* ボー・レート  2400bps@20MHz/2^1 */
/*    usTdrTx = 2082;       /* ボー・レート  4800bps@20MHz/2^1 */
/*    usTdrTx = 1387;       /* ボー・レート  7200bps@20MHz/2^1 */
/*    usTdrTx = 1040;       /* ボー・レート  9600bps@20MHz/2^1 */
/*    usTdrTx = 693;        /* ボー・レート 14400bps@20MHz/2^1 */
      usTdrTx = 519;        /* ボー・レート 19200bps@20MHz/2^1 */


    /*=================================================================
      データ初期化
    =================================================================*/

    usRxBuffer = 0x0000;            /* 受信データ・バッファ */

    usTxBuffer = 0x0000;            /* 送信データ・バッファ */

    usRxReg = 0x0000;               /* 受信データ・レジスタ */

    usTxReg = 0x0000;               /* 送信データ・レジスタ */

    ucRxBitCounter = 0;             /* 受信ビット・カウンタ */
    ucTxBitCounter = 0;             /* 送信ビット・カウンタ */

    bUartRxEnable = 0;              /* UART受信動作許可フラグ */
    bUartTxEnable = 0;              /* UART送信動作許可フラグ */
    bUartRxComp = 0;                /* UART受信動作完了フラグ */
    bUartTxComp = 0;                /* UART送信動作完了フラグ */

    bUartFrameErr = 0;              /* UARTフレーミング・エラー フラグ */
    bUartParityErr = 0;             /* UARTパリティ・エラー フラグ */

    PM12.0 = 1;                     /* データ入力端子 */
    P13.0 = 1;                      /* データ出力端子 */


    /*=================================================================
      UART受信モードの設定
    =================================================================*/

    bUartRxParityEnable = 1;        /* UART受信パリティ有 */
  /*bUartRxParityEnable = 0;        /* UART受信パリティ無 */

    bUartRxParityOdd = 1;           /* UART受信奇数パリティ */
  /*bUartRxParityOdd = 0;           /* UART受信偶数パリティ */

    bUartRxParityZero = 0;          /* UART受信ゼロ・パリティ無効 */
  /*bUartRxParityZero = 1;          /* UART受信ゼロ・パリティ有効 */
```

```
  /*=========================================================================
    UART送信モードの設定
  =========================================================================*/

  bUartTxParityEnable = 1;          /* UART送信パリティ有 */
/*bUartTxParityEnable = 0;          /* UART送信パリティ無 */

  bUartTxParityOdd = 1;             /* UART送信奇数パリティ */
/*bUartTxParityOdd = 0;             /* UART送信偶数パリティ */

  bUartTxParityZero = 0;            /* UART送信ゼロ・パリティ無効 */
/*bUartTxParityZero = 1;            /* UART送信ゼロ・パリティ有効 */



/*=========================================================================
  メイン・ルーチン
=========================================================================*/

    while (1)
    {
       if((bUartRxEnable == 0) && (bUartTxEnable == 0)){   /* 送受信動作停止 */
            if(bUartRxComp == 0) && (bUartTxComp == 0))   {  /* 送受信完了要求無 */

          /**************************************
           データ受信開始
          **************************************/

                   usRxReg = 0x0000;                          /* 受信データ・レジスタ */

                   bUartRxEnable = 1;                         /* UART受信動作許可 */
                   bUartRxComp = 0;                           /* UART受信動作完了フラグ クリア*/

                   bUartFrameErr = 0;                         /* UARTフレーミング・エラー クリア */
                   bUartParityErr = 0;                        /* UARTパリティ・エラー クリア */

                   PIF0 = 0;                                  /* スタート･ビット割り込み要求クリア */
                   PMK0 = 0;                                  /* スタート･ビット割り込み許可 */

          /**************************************
           データ送信開始
          **************************************/

                   usTxBuffer++;                              /* 送信データ下位更新 */
                   usTxBuffer &= 0x1ff;                       /* 9ビットに丸める */

                   P13.0 = 1;                                 /* データ出力端子初期化 */

                   usTxReg = usTxBuffer;                      /* 送信データ設定 */

                   ucTxBitCounter = (9 + 4);                  /* 送信ビット・カウンタ 初期化 */

                   bUartTxEnable = 1;                         /* UART送信動作中設定 */
                   bUartTxComp = 0;                           /* UART送信動作完了クリア */

                   TMIF01 = 0;                                /* 割り込み要求クリア */
                   TMMK01 = 0;                                /* 割り込み許可 */
                   TMIF01 = 1;                                /* 割り込み要求セット（送信開始）*/
          }
       }

       /**************************************
        データ送信完了
       **************************************/

       if(bUartTxComp == 1){                    /* 送信完了要求有 */
           bUartTxComp = 0;                     /* UART送信動作完了クリア */
       }

       /**************************************
        データ受信完了
       **************************************/

       if(bUartRxComp == 1){                    /* 受信完了要求有 */
           usRxBuffer = usRxReg;                /* 受信データ・バッファに受信データを保存 */
           bUartRxEnable = 0;                   /* UART受信動作停止 */
           bUartRxComp = 0;                     /* UART受信動作完了フラグ クリア*/

           bUartFrameErr = 0;                   /* UARTフレーミング・エラー クリア */
           bUartParityErr = 0;                  /* UARTパリティ・エラー クリア */
       }
    }
}
```

4. 5　INTP0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 この立ち下がりがスタート・ビット検出によるものか否かを判定します。受信レベルがハイ・レベルである場合，ノイズによる立ち下がりであったと判定し，UART受信動作を終了します。

③ INTP0割り込みの禁止により，UART受信のスタート・ビット検出を禁止します。

④ 受信用のボー・レートを設定します。なお，詳細は4. 2　ボー・レートについて　を参照してください。

⑤ UART受信モードを設定します。

⑥ タイマ・アレイ・ユニット0のチャネル0によるUART受信動作を開始します。

```
;*******************************************************************************
;
;       INTP0割り込み処理（INTP0の立ち下がりエッジ割り込み使用）
;
;*******************************************************************************
INT_RX_START:
①       SEL     RB1                     ;レジスタ・バンク切り替え
②       BT      P12.0,$HINTPRXRET       ;ノイズ・チェック

③       SET1    PMK0                    ;スタート･ビット割り込み禁止

        MOVW    AX,RTDRDATARX           ;1ビット時間取得
④       SHRW    AX,1                    ;半ビット時間作成
        MOVW    TDR00,AX                ;データ・サンプリング・タイマ設定

        MOV     RRXCOUNT,#(9 + 2)       ;ビット・カウンタ初期化
⑤       BF      FUARTRXPEN,$HINTPRX320  ;パリティ・ビット有？, No,
        INC     RRXCOUNT                ;パリティ・ビット分のカウンタ調整

HINTPRX320:
        SET1    TS0L.0                  ;サンプリング・タイマのカウント動作開始(トリガ動作)
⑥       CLR1    TMIF00                  ;割り込み要求クリア
        CLR1    TMMK00                  ;サンプリング・タイマ割り込み許可

HINTPRXRET:
        RETI
```

C言語の割り込み処理も，アセンブリ言語と同様な動作を行います。

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

    INTP0割り込み処理（INTP0の立ち下がりエッジ割り込み使用）

*******************************************************************************/
__interrupt void fn_intp0(void)
{

  NOP();
  NOP();
  if(P12.0 == 0){                       /* ノイズ(High)チェック */
        PMK0 = 1;                       /* スタート・ビット検出割り込み（この割り込み）禁止 */
        TDR00 = usTdrRx / 2;            /* 半ビット時間設定 */

        ucRxBitCounter = (9 + 2);       /* 受信ビット・カウンタ設定 */
        if(bUartRxParityEnable == 1){
                ucRxBitCounter++;       /* パリティ・ビット分のカウンタ調整 */
        }
        TS0L.0 = 1;                     /* サンプリング・タイマのカウント動作開始(トリガ動作) */
        TMIF00 = 0;                     /* サンプリング・タイマ割り込み要求クリア */
        TMMK00 = 0;                     /* サンプリング・タイマ割り込み許可 */

  }
}
```

●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        INTP0    fn_intp0
```

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_intp0(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 を参照してください。

4. 6　INTTM00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 受信したビット数を更新します。ストップ・ビットまで受信した場合，⑤の処理に分岐します。
③ スタート・ビットを受信したタイミングで，スタート・ビットが正常かどうかを判定します。スタート・ビットが正常でない場合，⑦の処理に分岐します。
④ 受信したデータ・ビット，およびパリティ・ビットを取り込みます。
⑤ ストップビットを受信したタイミングで，タイマ・アレイ・ユニット0のチャネル0によるUART受信動作を停止し，受信完了通知を発行します。また，受信したストップ・ビットからフレーミング・エラーを判定します。
⑥ 受信したパリティ・ビットからパリティ・エラーを判定します。
⑦ タイマ・アレイ・ユニット0のチャネル0によるUART受信動作を停止します。

なお，受信動作中の処理の流れについては　3. 5　タイミング・チャート　を参照してください。

```
;*****************************************************************************
;
;       INTTM00割り込み処理（タイマ・チャネル0のカウント完了割り込み使用）
;
;*****************************************************************************
INT_TM_RX:
①       SEL     RB1                     ;レジスタ・バンク切り替え

②       DEC     RRXCOUNT                ;ビット・カウント
        BZ      $HINTTR620              ;ストップ・ビット タイミング？, YES,

        ;-----------------------------------------
        ; スタート・ビットの確認
        ;-----------------------------------------

        MOV     A,#(9 + 1)              ;パリティ・ビット無のスタート・ビット位置
        BF      FUARTRXPEN,$HINTTR120   ;パリティ・ビット有？, No,
        MOV     A,#(9 + 2)              ;パリティ・ビット有のスタート・ビット位置

HINTTR120:
        CMP     A,RRXCOUNT              ;スタート・ビット タイミング？
③       BNZ     $HINTTR220              ; NO,

        BT      P12.0,$HINTTR820        ;ノイズ ?, YES,

        MOVW    AX,RTDRDATARX           ;1ビット時間
        MOVW    TDR00,AX                ;データ・サンプリング・タイマ設定
        BR      HINTTRRET

        ;-----------------------------------------
        ; データ・ビットの取り込み
        ;-----------------------------------------
HINTTR220:
        MOVW    AX,RRXREG
        SHRW    AX,1                    ;前回までの受信データを1ビット シフト
        MOV1    CY,P12.0                ;受信データ１ビット取り込み
        MOV1    A.1,CY                  ;今回の1ビットデータを前回までの受信データに結合
④       BT      FUARTRXPEN,$HINTTR260   ;パリティ・ビット有？, Yes,
        MOV1    A.0,CY                  ;パリティ・ビット無のビット位置補正

HINTTR260:
        MOVW    RRXREG,AX               ;データ格納
        BR      HINTTRRET

        ;-----------------------------------------
        ; ストップ・ビットの確認
        ;-----------------------------------------
HINTTR620:
        CLR1    FUARTRXEN               ;受信動作中解除
        SET1    FUARTRXCMP              ;1フレーム受信完了通知

        SET1    TMMK00                  ;サンプリング・タイマ割り込み禁止
⑤       SET1    TT0L.0                  ;サンプリング・タイマのカウント動作停止
        CLR1    TMIF00                  ;割り込み要求クリア

        BT      P12.0,$HINTTR720        ;ストップ・ビットＯＫ ?, Yes,
        SET1    FUARTFRM                ; No, フレーミングエラー
```

```
          ;-----------------------------------------
          ; パリティ・チェック
          ;-----------------------------------------
HINTTR720:
          BF        FUARTRXPEN,$HINTTR780       ;パリティ・チェック有？, No,
          MOV1      CY,(RRXREG + 1).1           ;パリティ・ビット読込
          BT        FUARTRXPZERO,$HINTTR760     ;ゼロ・パリティ？, Yes,

          XOR1      CY,(RRXREG + 1).0
          XOR1      CY,RRXREG.7
          XOR1      CY,RRXREG.6
          XOR1      CY,RRXREG.5
          XOR1      CY,RRXREG.4
          XOR1      CY,RRXREG.3
          XOR1      CY,RRXREG.2
⑥         XOR1      CY,RRXREG.1
          XOR1      CY,RRXREG.0                 ;データ中のビット=1の数が奇数か偶数か調べる
          XOR1      CY,FUARTRXPODD              ;パリティ・モード　チェック
HINTTR760:
          BNC       $HINTTR780                  ;パリティ正常受信？,Yes,

HINTTR770:
          SET1      FUARTPRT                    ;パリティ・エラー設定

HINTTR780:
          AND       (RRXREG + 1),#1             ;データのみとする

          BR        HINTTRRET

          ;-----------------------------------------
          ; 受信待機
          ;-----------------------------------------
HINTTR820:
          SET1      TMMK00                      ;サンプリング・タイマ割り込み禁止
          SET1      TT0L.0                      ;サンプリング・タイマのカウント動作停止
          CLR1      TMIF00                      ;割り込み要求クリア
⑦         SET1      PMK0                        ;スタート･ビット割り込み禁止

          BF        FUARTRXEN,$HINTTRRET        ;受信動作中 ?, No,

          CLR1      PMK0                        ;スタート･ビット割り込み許可

HINTTRRET:

          RETI
```

C言語の割り込み処理も，アセンブリ言語と同様な動作を行います。

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

  INTTM00割り込み処理（タイマ・チャネル0のカウント完了割り込み使用）

*****************************************************************************/
 __interrupt void fn_inttm00(void)
{
  Static unsigned char sreg ucRxParityWork;
  Static unsigned char sreg ucRxRegWork;

  ucRxBitCounter--;                             /* 受信ビット位置カウント */
  if(ucRxBitCounter != 0){                      /* ストップ・ビット位置以外の場合 */
    /*-----------------------------------------
      スタート・ビットの確認
    -----------------------------------------*/
    if( ((ucRxBitCounter == (9 + 1)) && (bUartRxParityEnable == 0)) ||
        ((ucRxBitCounter == (9 + 2)) && (bUartRxParityEnable == 1))   ){
      if(P12.0 == 0){                                   /* スタート・ビットのレベルOKの場合 */
        TDR00 = usTdrRx;                                /* １ビット時間設定 */
      }
      else{                                             /* スタート・ビットのレベルNGの場合 */
        TMMK00 = 1;                                     /* サンプリング・タイマ割り込み禁止 */
        TT0L.0 = 1;                                     /* サンプリング・タイマのカウント動作停止 */
        TMIF00 = 0;                                     /* 割り込み要求クリア */

        if(bUartRxEnable == 1){                         /* 受信動作中の場合 */
        PMK0 = 0;                                       /* スタート･ビット割り込み許可 */
        }

      }
    }
```

```
      /*-----------------------------------------
        データ・ビット,パリティ・ビットの取り込み
      -----------------------------------------*/
      else{                                          /* データ・ビット位置の場合 */
        usRxReg >>= 1;                               /* 前回までの受信データを1ビット シフト */
        if(P12.0 == 1){                              /* 今回の受信データが１の場合 */
          if(bUartRxParityEnable == 1){              /* パリティ・ビット有の場合 */
          usRxReg |= 0b1000000000;                   /* 今回のデータ(1)を設定 */
          }
          else{
          usRxReg |= 0b100000000;                    /* 今回のデータ(1)を設定 */
          }
        }
      }
    }
    /*-----------------------------------------
      ストップ・ビットの確認
    -----------------------------------------*/
    else{
      bUartRxEnable = 0;                             /* UART受信動作中解除 */
      bUartRxComp = 1;                               /* UART受信動作完了通知 */

      TMMK00 = 1;                                    /* サンプリング・タイマ割り込み禁止 */
      TT0L.0 = 1;                                    /* サンプリング・タイマのカウント動作停止 */
      TMIF00 = 0;                                    /* 割り込み要求クリア */

      if(P12.0 == 0){                                /* ストップ・ビットNG */
        bUartFrameErr = 1;                           /* フレーミング・エラー通知 */
      }
      /*-----------------------------------------
        パリティ・チェック
      -----------------------------------------*/

      if(bUartRxParityEnable == 1){                  /* パリティ・ビット有の場合 */

        ucRxRegWork = (unsigned char)(usRxReg >> 8); /* 受信データbit8、パリティ・ビット取得 */
        ucRxParityWork = ucRxRegWork;
        ucRxRegWork >>= 1;
        if(bUartRxParityZero == 1){                  /* ゼロ・パリティの場合 */
          ucRxParityWork = ucRxRegWork;              /* パリティ・ビット取得 */
        }
        else{                                        /* 偶数/奇数パリティの場合 */
          ucRxParityWork ^= ucRxRegWork;             /* パリティ・ビット ^ bit8 */

          ucRxRegWork = (unsigned char)usRxReg;      /* 受信データbit7～bit0取得 */
          ucRxParityWork ^= ucRxRegWork;             /* ^ bit0 */
          ucRxRegWork >>= 1;
          ucRxParityWork ^= ucRxRegWork;             /* ^ bit1 */
          ucRxRegWork >>= 1;
          ucRxParityWork ^= ucRxRegWork;             /* ^ bit2 */
          ucRxRegWork >>= 1;
          ucRxParityWork ^= ucRxRegWork;             /* ^ bit3 */
          ucRxRegWork >>= 1;
          ucRxParityWork ^= ucRxRegWork;             /* ^ bit4 */
          ucRxRegWork >>= 1;
          ucRxParityWork ^= ucRxRegWork;             /* ^ bit5 */
          ucRxRegWork >>= 1;
          ucRxParityWork ^= ucRxRegWork;             /* ^ bit6 */
          ucRxRegWork >>= 1;
          ucRxParityWork ^= ucRxRegWork;             /* ^ bit7 */

          ucRxParityWork &= 0x0001;                  /* 1となっているビットの数が偶数か奇数か */
          if(bUartRxParityOdd == 1){                 /* 奇数パリティの場合 */
          ucRxParityWork ^= 0x0001;                  /* 奇数・偶数に関わらず０の時OKとする */
          }
        }
        if((ucRxParityWork & 0x01) != 0){            /* パリティNGの場合 */
          bUartParityErr = 1;                        /* パリティ・エラー */
        }

      }
      usRxReg &= 0x1ff;                              /* データのみとする */
    }
}
```

4. 7　INTTM01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 出力したビット数を更新します。全ビットの送信が完了した場合，⑤の処理に分岐します。
③ 送信動作開始のタイミングで，データ長の設定，パリティ・ビットの付加，スタート・ビットの出力を行います。
④ ビット数に従って，データ・ビット，パリティ・ビット，およびストップ・ビットを出力します。
⑤ タイマ・アレイ・ユニット0のチャネル1によるUART送信動作を停止します。

なお，送信動作中の処理の流れについては　3. 5　タイミング・チャート　を参照してください。

```
;*******************************************************************************
;
;         INTTM01割り込み処理（タイマ・チャネル1のカウント完了割り込み使用）
;
;*******************************************************************************
INT_TM_TX:
①        SEL      RB1                          ;レジスタ・バンク切り替え

②        DEC      RTXCOUNT                     ;ビット・カウント
         BZ       $HINTTTX820                  ;ストップ・ビット終了？, YES,

         CMP      RTXCOUNT,#9 + 3              ;スタート・ビット 出力タイミング ?
         BNZ      $HINTTTX520                  ; NO,

         BT        FUARTTXPEN,$HINTTTX140      ;送信パリティ有？,Yes,

         DEC      RTXCOUNT                     ;ビット・カウント(パリティ・ビット分)補正

         SET1     (RTXREG + 1).1               ;出力データにストップ・ビット付加
         BR       HINTTTX220

         ;-----------------------------------------
         ; スタート・ビットの出力
         ;-----------------------------------------
HINTTTX140:
         ;パリティ・ビット付加
         CLR1     CY
         BT       FUARTTXPZERO,$HINTTTX180     ;ゼロ・パリティ？, Yes,

         MOV1     CY,FUARTTXPODD               ;パリティ・モード（偶数=0・奇数=1）
         XOR1     CY,(RTXREG + 1).0
③        XOR1     CY,RTXREG.7
         XOR1     CY,RTXREG.6
         XOR1     CY,RTXREG.5
         XOR1     CY,RTXREG.4
         XOR1     CY,RTXREG.3
         XOR1     CY,RTXREG.2
         XOR1     CY,RTXREG.1
         XOR1     CY,RTXREG.0                  ;データ中のビット=1の数が奇数か偶数か調べる

HINTTTX180:
         MOV1     (RTXREG + 1).1,CY            ;パリティ・ビット設定
         SET1     (RTXREG + 1).2               ;出力データにストップ・ビット付加

         ;スタート・ビット出力
HINTTTX220:
         MOVW     AX,RTDRDATATX
         MOVW     TDR01,AX                     ;データ出力タイミング設定

         SET1     TS0L.1                       ;データ出力タイマのカウント動作開始
         CLR1     TMIF01                       ;割り込み要求クリア

         CLR1     P13.0                        ;スタート・ビット出力
         BR       HINTTTXRET
```

```
          ;-----------------------------------------
          ; データ・ビット、パリティ・ビット、
          ; ストップ・ビットの出力
          ;-----------------------------------------
HINTTTX520:

          MOVW      AX,RTXREG
          SHRW      AX,1                    ;データ・ビットをCYに取得
④         MOVW      RTXREG,AX
          MOV1      P13.0,CY                ;データ出力
          BR        HINTTTXRET

          ;-----------------------------------------
          ; ストップ・ビット出力終了
          ; (9ビット送信完了)
          ;-----------------------------------------
          ;-----------------------------------------
          ; 全データ送信完了
          ;-----------------------------------------
HINTTTX820:
          CLR1      FUARTTXEN               ;送信中状態解除
          SET1      TMMK01                  ;送信タイマ割り込み禁止
⑤         SET1      TT0L.1                  ;データ出力タイマのカウント動作停止

          SET1      FUARTTXCMP              ;1フレーム送信完了通知

HINTTTXRET:

          RETI
```

C言語の割り込み処理も，アセンブリ言語と同様な動作を行います。

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

  INTTM01割り込み処理（タイマ・チャネル1のカウント完了割り込み使用）

*****************************************************************************/
__interrupt void fn_inttm01(void)
{
  static unsigned char sreg ucTxParityWork;
  static unsigned char sreg ucTxRegWork;

  ucTxBitCounter--;                                     /* 受信ビット位置カウント */

  /*-----------------------------------------
    送信開始（スタート・ビット出力開始）
  -----------------------------------------*/
  if(ucTxBitCounter == (9+3)){                          /* 送信開始の場合 */
    if( bUartTxParityEnable == 0){                      /* 送信パリティ・ビット無の場合 */
       ucTxBitCounter--;                                /* ビット位置カウンタ補正 */
       usTxReg |= 0x200;                                /* ストップ・ビット付加 */
    }
    else{                                               /* 送信パリティ・ビット有の場合 */
      usTxReg &= 0xdff;                                 /* パリティ・ビットを0に仮設定 */
      if(bUartTxParityZero == 0){                       /* ゼロ・パリティ以外(偶数/奇数)の場合 */
        ucTxParityWork = (unsigned char)(usTxReg >> 8);         /* bit8 */
        ucTxRegWork = (unsigned char)(usTxReg & 0xff);          /* bit0 ～ bit7取得 */
        ucTxParityWork ^= ucTxRegWork;                  /* bit8 ^ bit0 */
        ucTxRegWork >>= 1;
        ucTxParityWork ^= ucTxRegWork;                  /* ^ bit1 */
        ucTxRegWork >>= 1;
        ucTxParityWork ^= ucTxRegWork;                  /* ^ bit2 */
        ucTxRegWork >>= 1;
        ucTxParityWork ^= ucTxRegWork;                  /* ^ bit3 */
        ucTxRegWork >>= 1;
        ucTxParityWork ^= ucTxRegWork;                  /* ^ bit4 */
        ucTxRegWork >>= 1;
        ucTxParityWork ^= ucTxRegWork;                  /* ^ bit5 */
        ucTxRegWork >>= 1;
        ucTxParityWork ^= ucTxRegWork;                  /* ^ bit6 */
        ucTxRegWork >>= 1;
        ucTxParityWork ^= ucTxRegWork;                  /* ^ bit7 */

        ucTxParityWork &= 0x0001;                       /* パリティ・ビット(bit0)を取得 */
        if(bUartTxParityOdd == 1){                      /* 奇数パリティの場合 */
                   ucTxParityWork ^= 0x0001;            /* パリティ・ビット(bit0)を反転 */
        }
        if(ucTxParityWork == 0x0001){                   /* パリティ・ビットが１の場合 */

                  usTxReg |= 0x0200;                    /* パリティ・ビットに１を設定 */
        }
      }
      usTxReg |= 0x400;                                 /* ストップ・ビット付加 */
    }
```

```
        TDR01 = usTdrTx;                                    /* 1ビット時間(データ出力タイミング)設定 */

        TS0L.1 = 1;                                         /* データ出力タイマのカウント動作開始 */
        TMIF01 = 0;                                         /* 割り込み要求クリア */

        P13.0 = 0;                                          /* スタート・ビット出力 */
    }
    else{
        /*-----------------------------------------
          データ・ビット、パリティ・ビット、
          ストップ・ビットの出力
        -----------------------------------------*/
        if(ucTxBitCounter != 0){

           if((usTxReg & 0x0001) == 0){                     /* 出力データが0の場合 */
             P13.0 = 0;                                     /* 送信データ端子にLow出力 */
           }
           else{
             P13.0 = 1;                                     /* 送信データ端子にHigh出力 */
           }
           usTxReg >>= 1;                                   /* 次回出力データ設定 */
        }
        else{
        /*-----------------------------------------
          ストップ・ビット出力終了
          (9ビット送信完了)
        -----------------------------------------*/

          bUartTxEnable = 0;                                /* UART送信動作中解除 */
          bUartTxComp = 1;                                  /* UART送信動作完了通知 */
          TMMK01 = 1;                                       /* 送信タイマ割り込み禁止 */
          TT0L.1 = 1;                                       /* データ出力タイマのカウント動作停止 */

        }
    }
}
```

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/KJ3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/KJ3シリーズ
;
;*******************************************************************************
;     78K0R/KJ3シリーズ        サンプル・プログラム
;*******************************************************************************
;     タイマ・アレイ・ユニットを使用した9ビットUART通信
;*******************************************************************************
;【履歴】
;     2008.10.--       新規作成
;*******************************************************************************
;
;【概要】
;
;このサンプル・プログラムは，タイマ・アレイ・ユニットのインターバル・タイマ機能を
;使用し，9ビット・データの送受信を行うプログラムです。UART機能を持つハードウエア・
;マクロと同様の送受信動作をインターバル・タイマ機能と端子機能のみで実現します。
;受信の場合，端子入力エッジ割り込みによりスタート・ビットを検出し，インターバル・
;タイマ動作を開始します。インターバル・タイマ割り込みではUARTの1ビット時間ごとに
;受信端子のサンプリングを行い，スタート・ビット，データ・ビット，パリティ・ビット，
;ストップ・ビットを取り込みます。ストップ・ビット取り込み後，エラーチェックを行い，
;受信完了通知を発行します。また送信の場合，インターバル・タイマ割り込みを1ビット
;時間ごとに呼び出し，スタート・ビット，送信データ，パリティ・ビット，ストップ・
;ビットを送信端子に出力します。
;なお，このサンプル・プログラムで実現するUART機能は，データ長が9ビット，データ転送方向
;がLSBファースト，データ位相が正転出力，ストップ・ビットが1ビットで固定となります。
;また，ボー・レートは300～19200bps，パリティ・ビットは有または無（有の場合，ゼロ，偶数，
;奇数）から選択可能です。以上の設定で全2重での送受信を行います。また，送受信するデータ
;は0～511の9ビットの数値です。
;
;
;＜使用する周辺の初期設定の主な内容＞
;
;   ・割り込みの禁止
```

```
;   ・CPU／周辺ハードウエア・クロックを高速システム・クロック(20MHz使用)の動作に設定
;   ・ポートの設定
;   ・タイマ・アレイ・ユニット0（TAU0）のチャネル1をUART送信用，チャネル0をUART受信用
;     に設定
;   ・INTP0をUART受信のスタート・ビット検出用に設定
;   ・割り込みの許可
;
;
;＜メイン処理の主な内容＞
;
;   ・UART送信データの作成
;   ・UART受信動作の許可（INTP0割り込み許可）
;   ・UART送信動作の開始（INTTM01割り込み許可，タイマ・チャネル1のカウント開始）
;
;
;＜INTP0割り込み処理（INTP0の立ち下がりエッジ割り込み使用）＞
;
;   ・UART受信動作の開始（INTTM00割り込み許可，タイマ・チャネル0のカウント開始）
;   ・INTP0割り込み禁止
;
;
;＜INTTM00割り込み処理（タイマ・チャネル0のカウント完了割り込み使用）＞
;
;   ・スタート・ビットのロウ・レベル確認
;   ・データ・ビット，パリティ・ビットの取り込み
;   ・パリティ・チェック
;   ・UART受信動作の停止（INTTM00割り込み禁止，タイマ・チャネル0のカウント停止）
;
;
;＜INTTM01割り込み処理（タイマ・チャネル1のカウント完了割り込み使用）＞
;
;   ・UART送信動作の開始（タイマ・チャネル1のカウント開始）
;   ・データ・ビット，パリティ・ビット，およびストップ・ビットの出力
;   ・UART送信動作の停止（INTTM01割り込み禁止，タイマ・チャネル1のカウント停止）
;
;
;*******************************************************************************


;===============================================================================
;
;      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      INT_RX_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      RESET_START                 ;(1E)   INTST0/INTCSI00
      DW      RESET_START                 ;(20)   INTSR0/INTCSI01
      DW      RESET_START                 ;(22)   INTSRE0
      DW      RESET_START                 ;(24)   INTST1/INTCSI10/INTIIC10
      DW      RESET_START                 ;(26)   INTSR1/INTCSI11/INTIIC11
      DW      RESET_START                 ;(28)   INTSRE1
      DW      RESET_START                 ;(2A)   INTIIC0
      DW      INT_TM_RX                   ;(2C)   INTTM00
      DW      INT_TM_TX                   ;(2E)   INTTM01
      DW      RESET_START                 ;(30)   INTTM02
      DW      RESET_START                 ;(32)   INTTM03
      DW      RESET_START                 ;(34)   INTAD
      DW      RESET_START                 ;(36)   INTRTC
      DW      RESET_START                 ;(38)   INTRTCI
      DW      RESET_START                 ;(3A)   INTKR
      DW      RESET_START                 ;(3C)   INTST2/INTCSI20/INTIIC20
      DW      RESET_START                 ;(3E)   INTSR2/INTCSI21/INTIIC21
      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
        DW      RESET_START                     ;(56)   INTTM10
        DW      RESET_START                     ;(58)   INTTM11
        DW      RESET_START                     ;(5A)   INTTM12
        DW      RESET_START                     ;(5C)   INTTM13

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

;=============================================================================
;
;       スタック領域の確保
;
;=============================================================================
DSTK    DSEG    AT      0FFEC0H
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＝FFEE0H

;=============================================================================
;
;       ＲＡＭの定義
;
;=============================================================================
DMAIN   DSEG    SADDRP
RTDRDATATX:     DS      2                       ;TDRの設定値(送信ボー・レート)を保持する
RTDRDATARX:     DS      2                       ;TDRの設定値(受信ボー・レート)を保持する

RRXBUF:                 DS      2                       ;受信データ・バッファ

RRXREG:                 DS      2                       ;受信データ・レジスタ
RTXREG:                 DS      2                       ;送信データ・レジスタ


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


RRXCOUNT:       DS      1                       ;受信ビット・カウンタ
RTXCOUNT:       DS      1                       ;送信ビット・カウンタ


RUARTRXST:      DS      1                       ;UART受信状態フラグ
```

```
 FUARTRXEN    EQU     RUARTRXST.7            ; 受信許可
 FUARTRXCMP   EQU     RUARTRXST.6            ; 受信完了
 FUARTFRM     EQU     RUARTRXST.5            ; フレーミング・エラー
 FUARTPRT     EQU     RUARTRXST.4            ; パリティ・エラー

 FUARTRXPEN   EQU     RUARTRXST.3            ; 受信パリティ設定(1:パリティ有)
 FUARTRXPODD  EQU     RUARTRXST.2            ; 受信パリティモード(1:奇数)
 FUARTRXPZERO EQU     RUARTRXST.1            ; 受信パリティモード(1:0パリティ)


RUARTTXST:    DS      1                      ;UART送信状態フラグ
 FUARTTXEN    EQU     RUARTTXST.7            ; 送信許可
 FUARTTXCMP   EQU     RUARTTXST.6            ; 送信完了

 FUARTTXPEN   EQU     RUARTTXST.3            ; 送信パリティ設定(1:パリティ有)
 FUARTTXPODD  EQU     RUARTTXST.2            ; 送信パリティモード(1:奇数)
 FUARTTXPZERO EQU     RUARTTXST.1            ; 送信パリティモード(0:0パリティ)



;*****************************************************************************
;
;     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
;-----------------------------------------------------------------------------
;     20MHzのX1発振回路で動作が行えるように設定します
;-----------------------------------------------------------------------------
```

```
        MOV     CMC,    #01000001B              ;クロック動作モード
                        ;|||||||+--------------- AMPH: 10MHz＜fMX≦20MHz
                        ;||||+++---------------- <000>
                        ;|||+------------------- OSCSELS: P123/P124端子を入力ポート
                        ;||+-------------------- <0>
                        ;++--------------------- EXCLK/OSCSEL: X1発振モード(20MHz)

        MOV     CSC,    #01000000B              ;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発振回路停止
                        ;+----------------------- MSTOP: X1発振回路動作

        MOV     OSMC,   #00000001B              ;動作スピード・モード
                        ;|||||||+--------------- FSEL: 10MHzを越える周波数で動作
                        ;+++++++---------------- <00000>

        MOV     OSTS,   #00000101B              ;発振安定時間: 2^15/fX

HRST300:
        NOP
        BF      OSTC.2, $HRST300                ;クロック発振安定待ち

        MOV     CKC,    #00011000B              ;クロック選択
                        ;|||||+++--------------- MDIV2-0: CPU/周辺ハードウエア・クロック(fCLK)=fMX
                        ;||||+------------------ <1>
                        ;|||+------------------- MCM0: 高速システム・クロック（fMX）
                        ;||+-------------------- <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
        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
        MOV     PM1,    #00000000B              ;P10-P17を出力ポートに設定
```

```
;-------------------------------------------------------------------------------
;     ポート2の設定
;-------------------------------------------------------------------------------
      MOV     P2,     #00000000B              ;P20-P27の出力ラッチLow
      MOV     PM2,    #00000000B              ;P20-P27を出力ポートに設定

;-------------------------------------------------------------------------------
;     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,   #00011111B              ;P125-P127を出力ポートに設定
                                              ;[後でP120(INTP0)の設定を行う]

;-------------------------------------------------------------------------------
;     ポート13の設定
;-------------------------------------------------------------------------------
      MOV     P13,    #00000000B              ;P130-P137の出力ラッチLow
      MOV     PM13,   #00000000B              ;P131-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


;--------------------------------------------------------------------------------
;     タイマの設定
;--------------------------------------------------------------------------------

      ;----------------------------------------
      ; TDR00設定値
      ;----------------------------------------

;;    MOVW    RTDRDATARX,#33332               ;ボー・レート    300bps@20MHz/2^1
;;    MOVW    RTDRDATARX,#16665               ;ボー・レート    600bps@20MHz/2^1
;;    MOVW    RTDRDATARX,#8332                ;ボー・レート   1200bps@20MHz/2^1
;;    MOVW    RTDRDATARX,#4165                ;ボー・レート   2400bps@20MHz/2^1
;;    MOVW    RTDRDATARX,#2082                ;ボー・レート   4800bps@20MHz/2^1
;;    MOVW    RTDRDATARX,#1387                ;ボー・レート   7200bps@20MHz/2^1
;;    MOVW    RTDRDATARX,#1040                ;ボー・レート   9600bps@20MHz/2^1
;;    MOVW    RTDRDATARX,#693                 ;ボー・レート  14400bps@20MHz/2^1
      MOVW    RTDRDATARX,#519                 ;ボー・レート  19200bps@20MHz/2^1

      ;----------------------------------------
      ; TM00の設定
      ;----------------------------------------
      SET1    !TAU0EN                         ;TAU0使用許可

      MOV     TPS0L,  #00000001B              ;タイマ・クロック選択
                      ;||||++++-------------- PRS003-000: fCLK/2^1(シリアル・クロックと同じにする)
                      ;++++------------------ PRS013-010: 未使用

      MOVW    AX,     #0000000000000000B      ;動作モード設定
                      ;||||||||||||++++------- MD003-000: インターバル・タイマ・モード
                      ;||||||||||++----------- <00>
                      ;||||||||++------------- CIS001-000:
                      ;|||||+++--------------- STS002-000: ソフトウエア・トリガ・スタートのみ有効
                      ;||||+------------------ 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)
      MOVW    TMR00,  AX
```

```
      CLR1    TMIF00                          ;割り込み要求クリア
;;    CLR1    TMMK00                          ;割り込み許可

;;    SET1    TS0L.0                          ;TS00:TAU0のチャネル0のカウント動作開始(トリガ動作)

      ;----------------------------------------
      ; TDR01設定値
      ;----------------------------------------

;;    MOVW    RTDRDATATX,#33332               ;ボー・レート    300bps@20MHz/2^1
;;    MOVW    RTDRDATATX,#16665               ;ボー・レート    600bps@20MHz/2^1
;;    MOVW    RTDRDATATX,#8332                ;ボー・レート   1200bps@20MHz/2^1
;;    MOVW    RTDRDATATX,#4165                ;ボー・レート   2400bps@20MHz/2^1
;;    MOVW    RTDRDATATX,#2082                ;ボー・レート   4800bps@20MHz/2^1
;;    MOVW    RTDRDATATX,#1387                ;ボー・レート   7200bps@20MHz/2^1
;;    MOVW    RTDRDATATX,#1040                ;ボー・レート   9600bps@20MHz/2^1
;;    MOVW    RTDRDATATX,#693                 ;ボー・レート  14400bps@20MHz/2^1
      MOVW    RTDRDATATX,#519                 ;ボー・レート  19200bps@20MHz/2^1

      ;----------------------------------------
      ; TM01の設定
      ;----------------------------------------
;;    SET1    !TAU0EN                         ;TAU0使用許可

;;    MOV     TPS0L,  #00000001B              ;タイマ・クロック選択
;;                    ;||||++++-------------- PRS003-000: fCLK/2^1(シリアル・クロックと同じにする)
;;                    ;++++------------------ PRS013-010: 未使用

      MOVW    AX,     #0000000000000000B      ;動作モード設定
                      ;||||||||||||++++------- MD013-010: インターバル・タイマ・モード
                      ;||||||||||++----------- <00>
                      ;||||||||++------------- CIS011-010:
                      ;|||||+++--------------- STS012-010: ソフトウエア・トリガ・スタートのみ有効
                      ;||||+------------------ MASTER01: 単体動作
                      ;|||+------------------- CCS01: CKS00ビットで指定した動作クロックMCK
                      ;|++-------------------- <00>
                      ;+---------------------- CKS01: PRSで選択した動作クロックCK00(MCK)
      MOVW    TMR01,  AX

      CLR1    TMIF01                          ;割り込み要求クリア

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

```
;     INTP0(UART RX)の設定
;-------------------------------------------------------------------------

      CLR1    EGP0.0
      SET1    EGN0.0                        ;立ち下がりエッジ有効

      CLR1    PIF0                          ;割り込み要求クリア

      EI                                    ;割り込み許可


;*************************************************************************
;
;     メイン処理
;
;*************************************************************************
      ;=====================================================================
      ; データ初期化
      ;=====================================================================

      MOVW    RTXBUF,#0000H                 ;送信データ初期化

      CLR1    FUARTRXEN                     ;受信中解除
      CLR1    FUARTTXEN                     ;送信中解除


      ;=====================================================================
      ; UART受信モードの設定
      ;=====================================================================


      SET1    FUARTRXPEN                    ; 受信パリティ有設定
;;    CLR1    FUARTRXPEN                    ; 受信パリティ無設定

      CLR1    FUARTRXPZERO                  ; 受信パリティをゼロ・パリティ以外に設定
;;    SET1    FUARTRXPZERO                  ; 受信パリティをゼロ・パリティに設定

      SET1    FUARTRXPODD                   ; 受信パリティを奇数パリティに設定
;;    CLR1    FUARTRXPODD                   ; 受信パリティを偶数パリティに設定

      ;=====================================================================
      ; UART送信モードの設定
      ;=====================================================================
```

```
        SET1    FUARTTXPEN                  ; 送信パリティ有設定
;;      CLR1    FUARTTXPEN                  ; 送信パリティ無設定

        SET1    FUARTTXPODD                 ; 送信パリティを奇数パリティに設定
;;      CLR1    FUARTTXPODD                 ; 送信パリティを偶数パリティに設定

;;      SET1    FUARTTXPZERO                ; 送信パリティをゼロ・パリティに設定
        CLR1    FUARTTXPZERO                ; 送信パリティをゼロ・パリティ以外に設定

        SET1    P13.0                       ;データ出力端子初期化
        SET1    PM12.0                      ;データ入力端子設定

        ;=====================================================================
        ; UART送受信実行
        ;=====================================================================
MAIN_LOOP:
        BT      FUARTRXEN,$LMAIN320         ;受信中？, Yes,
        BT      FUARTTXEN,$LMAIN220         ;送信中？, Yes,
        BT      FUARTRXCMP,$LMAIN320        ;受信完了？,Yes,
        BT      FUARTTXCMP,$LMAIN220        ;送信完了？, Yes,

        ;**************************************
        ; データ受信開始
        ;**************************************

        CLR1    FUARTFRM                    ;フレーミング・エラー クリア
        CLR1    FUARTPRT                    ;パリティ・エラー クリア

        CLR1    FUARTRXCMP                  ;受信完了クリア
        SET1    FUARTRXEN                   ;受信動作開始
        MOVW    RRXREG,#0000H               ;受信レジスタ クリア
        CLR1    PIF0                        ;スタート･ビット割り込み要求クリア
        CLR1    PMK0                        ;スタート･ビット割り込み許可

        ;**************************************
        ; データ送信開始
        ;**************************************

        MOVW    AX,RTXBUF                   ;送信データ取得
        INCW    AX                          ;送信データ更新
        AND     A,#1H                       ;9ビットに丸める
        MOVW    RTXBUF,AX
```

```
        SET1    P13.0                   ;データ出力端子初期化

        MOVW    AX,RTXBUF               ;送信データ取得
        MOVW    RTXREG,AX               ;送信データを送信バッファに設定

        CLR1    FUARTTXCMP              ;送信完了クリア
        SET1    FUARTTXEN               ;送信中設定
        MOV     RTXCOUNT,#(9 + 4)       ;ビット・カウント初期化
        CLR1    TMIF01                  ;割り込み要求クリア
        CLR1    TMMK01                  ;割り込み許可
        SET1    TMIF01                  ;割り込み要求セット（送信開始）

        BR      LMAINRET


        ;***************************************
        ; データ送信完了
        ;***************************************
LMAIN220:
        BF      FUARTTXCMP,$LMAIN320    ;送信完了？, No,

        CLR1    FUARTTXCMP              ;送信完了クリア
        CLR1    FUARTTXEN               ;送信中解除


        ;***************************************
        ; データ受信完了
        ;***************************************
LMAIN320:
        BF      FUARTRXCMP,$LMAINRET    ;受信完了？,No,

        MOVW    AX,RRXREG
        MOVW    RRXBUF,AX               ;受信データ退避

        CLR1    FUARTRXCMP              ;受信完了クリア
        CLR1    FUARTRXEN               ;受信動作終了
        CLR1    FUARTFRM                ;フレーミング・エラー クリア

LMAINRET:
        BR      MAIN_LOOP               ; MAIN_LOOPへ
```

```
;*******************************************************************************
;
;     INTP0割り込み処理（INTP0の立ち下がりエッジ割り込み使用）
;
;*******************************************************************************
INT_RX_START:
      SEL     RB1                         ;レジスタ・バンク切り替え
      BT      P12.0,$HINTPRXRET           ;ノイズ・チェック

      SET1    PMK0                        ;スタート･ビット割り込み禁止

      MOVW    AX,RTDRDATARX               ;1ビット時間取得
      SHRW    AX,1                        ;半ビット時間作成
      MOVW    TDR00,AX                    ;データ・サンプリング・タイマ設定

      MOV     RRXCOUNT,#(9 + 2)           ;ビット・カウンタ初期化
      BF      FUARTRXPEN,$HINTPRX320      ;パリティ・ビット有？, No,
      INC     RRXCOUNT                    ;パリティ・ビット分のカウンタ調整

HINTPRX320:
      SET1    TS0L.0                      ;サンプリング・タイマのカウント動作開始(トリガ動作)
      CLR1    TMIF00                      ;割り込み要求クリア
      CLR1    TMMK00                      ;サンプリング・タイマ割り込み許可

HINTPRXRET:
      RETI



;*******************************************************************************
;
;     INTTM00割り込み処理（タイマ・チャネル0のカウント完了割り込み使用）
;
;*******************************************************************************
INT_TM_RX:
      SEL     RB1                         ;レジスタ・バンク切り替え

      DEC     RRXCOUNT                    ;ビット・カウント
      BZ      $HINTTR620                  ;ストップ・ビット タイミング？, YES,

      ;----------------------------------------
      ; スタート・ビットの確認
      ;----------------------------------------
```

```
        MOV     A,#(9 + 1)                  ;パリティ・ビット無のスタート・ビット位置
        BF      FUARTRXPEN,$HINTTR120       ;パリティ・ビット有？, No,
        MOV     A,#(9 + 2)                  ;パリティ・ビット有のスタート・ビット位置

HINTTR120:
        CMP     A,RRXCOUNT                  ;スタート・ビット タイミング？
        BNZ     $HINTTR220                  ; NO,

        BT      P12.0,$HINTTR820            ;ノイズ ?, YES,

        MOVW    AX,RTDRDATARX               ;1ビット時間
        MOVW    TDR00,AX                    ;データ・サンプリング・タイマ設定
        BR      HINTTRRET

        ;----------------------------------------
        ; データ・ビットの取り込み
        ;----------------------------------------
HINTTR220:
        MOVW    AX,RRXREG
        SHRW    AX,1                        ;前回までの受信データを1ビット シフト
        MOV1    CY,P12.0                    ;受信データ１ビット取り込み
        MOV1    A.1,CY                      ;今回の1ビットデータを前回までの受信データに結合
        BT      FUARTRXPEN,$HINTTR260       ;パリティ・ビット有？, Yes,
        MOV1    A.0,CY                      ;パリティ・ビット無のビット位置補正

HINTTR260:
        MOVW    RRXREG,AX                   ;データ格納
        BR      HINTTRRET

        ;----------------------------------------
        ; ストップ・ビットの確認
        ;----------------------------------------
HINTTR620:
        CLR1    FUARTRXEN                   ;受信動作中解除
        SET1    FUARTRXCMP                  ;1フレーム受信完了通知

        SET1    TMMK00                      ;サンプリング・タイマ割り込み禁止
        SET1    TT0L.0                      ;サンプリング・タイマのカウント動作停止
        CLR1    TMIF00                      ;割り込み要求クリア

        BT      P12.0,$HINTTR720            ;ストップ・ビットＯＫ ?, Yes,
        SET1    FUARTFRM                    ; No, フレーミングエラー
```

```
        ;----------------------------------------
        ; パリティ・チェック
        ;----------------------------------------
HINTTR720:
        BF      FUARTRXPEN,$HINTTR780           ;パリティ・チェック有？, No,
        MOV1    CY,(RRXREG + 1).1               ;パリティ・ビット読込
        BT      FUARTRXPZERO,$HINTTR760         ;ゼロ・パリティ？, Yes,

        XOR1    CY,(RRXREG + 1).0
        XOR1    CY,RRXREG.7
        XOR1    CY,RRXREG.6
        XOR1    CY,RRXREG.5
        XOR1    CY,RRXREG.4
        XOR1    CY,RRXREG.3
        XOR1    CY,RRXREG.2
        XOR1    CY,RRXREG.1
        XOR1    CY,RRXREG.0                     ;データ中のビット=1の数が奇数か偶数か調べる
        XOR1    CY,FUARTRXPODD                  ;パリティ・モード　チェック
HINTTR760:
        BNC     $HINTTR780                      ;パリティ正常受信？,Yes,

HINTTR770:
        SET1    FUARTPRT                        ;パリティ・エラー設定

HINTTR780:
        AND     (RRXREG + 1),#1                 ;データのみとする

        BR      HINTTRRET

        ;----------------------------------------
        ; 受信待機
        ;----------------------------------------
HINTTR820:
        SET1    TMMK00                          ;サンプリング・タイマ割り込み禁止
        SET1    TT0L.0                          ;サンプリング・タイマのカウント動作停止
        CLR1    TMIF00                          ;割り込み要求クリア

        SET1    PMK0                            ;スタート･ビット割り込み禁止

        BF      FUARTRXEN,$HINTTRRET            ;受信動作中 ?, No,

        CLR1    PMK0                            ;スタート･ビット割り込み許可
```

```
HINTTRRET:

      RETI



;*****************************************************************************
;
;     INTTM01割り込み処理（タイマ・チャネル1のカウント完了割り込み使用）
;
;*****************************************************************************
INT_TM_TX:
      SEL     RB1                         ;レジスタ・バンク切り替え

      DEC     RTXCOUNT                    ;ビット・カウント
      BZ      $HINTTTX820                 ;ストップ・ビット終了？, YES,

      CMP     RTXCOUNT,#9 + 3             ;スタート・ビット 出力タイミング ?
      BNZ     $HINTTTX520                 ; NO,

      BT       FUARTTXPEN,$HINTTTX140     ;送信パリティ有？,Yes,

      DEC     RTXCOUNT                    ;ビット・カウント(パリティ・ビット分)補正

      SET1    (RTXREG + 1).1              ;出力データにストップ・ビット付加
      BR      HINTTTX220

      ;-----------------------------------------
      ; スタート・ビットの出力
      ;-----------------------------------------
HINTTTX140:
      ;パリティ・ビット付加
      CLR1    CY
      BT      FUARTTXPZERO,$HINTTTX180    ;ゼロ・パリティ？, Yes,

      MOV1    CY,FUARTTXPODD              ;パリティ・モード（偶数=0・奇数=1）
      XOR1    CY,(RTXREG + 1).0
      XOR1    CY,RTXREG.7
      XOR1    CY,RTXREG.6
      XOR1    CY,RTXREG.5
      XOR1    CY,RTXREG.4
      XOR1    CY,RTXREG.3
      XOR1    CY,RTXREG.2
```

```
        XOR1    CY,RTXREG.1
        XOR1    CY,RTXREG.0                 ;データ中のビット=1の数が奇数か偶数か調べる

HINTTTX180:
        MOV1    (RTXREG + 1).1,CY           ;パリティ・ビット設定
        SET1    (RTXREG + 1).2              ;出力データにストップ・ビット付加

        ;スタート・ビット出力
HINTTTX220:
        MOVW    AX,RTDRDATATX
        MOVW    TDR01,AX                    ;データ出力タイミング設定

        SET1    TS0L.1                      ;データ出力タイマのカウント動作開始
        CLR1    TMIF01                      ;割り込み要求クリア

        CLR1    P13.0                       ;スタート・ビット出力
        BR      HINTTTXRET

        ;-----------------------------------------
        ; データ・ビット、パリティ・ビット、
        ; ストップ・ビットの出力
        ;-----------------------------------------
HINTTTX520:

        MOVW    AX,RTXREG
        SHRW    AX,1                        ;データ・ビットをCYに取得
        MOVW    RTXREG,AX
        MOV1    P13.0,CY                    ;データ出力
        BR      HINTTTXRET

        ;-----------------------------------------
        ; ストップ・ビット出力終了
        ; (9ビット送信完了)
        ;-----------------------------------------
        ;-----------------------------------------
        ; 全データ送信完了
        ;-----------------------------------------
HINTTTX820:
        CLR1    FUARTTXEN                   ;送信中状態解除
        SET1    TMMK01                      ;送信タイマ割り込み禁止
        SET1    TT0L.1                      ;データ出力タイマのカウント動作停止

        SET1    FUARTTXCMP                  ;1フレーム送信完了通知
```

```
HINTTTXRET:

        RETI


end
```

● main.c （C言語版）

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

      NEC Electronics      78K0R/KJ3シリーズ

************************************************************************************
      78K0R/KJ3シリーズ          サンプル・プログラム
************************************************************************************
      タイマ・アレイ・ユニットを使用した9ビットUART通信
************************************************************************************
【履歴】
      2008.10.--        新規作成
************************************************************************************

【概要】

このサンプル・プログラムは，タイマ・アレイ・ユニットのインターバル・タイマ機能を
使用し，9ビット・データの送受信を行うプログラムです。UART機能を持つハードウエア・
マクロと同様の送受信動作をインターバル・タイマ機能と端子機能のみで実現します。
受信の場合，端子入力エッジ割り込みによりスタート・ビットを検出し，インターバル・
タイマ動作を開始します。インターバル・タイマ割り込みではUARTの1ビット時間ごとに
受信端子のサンプリングを行い，スタート・ビット，データ・ビット，パリティ・ビット，
ストップ・ビットを取り込みます。ストップ・ビット取り込み後，エラーチェックを行い，
受信完了通知を発行します。また送信の場合，インターバル・タイマ割り込みを1ビット
時間ごとに呼び出し，スタート・ビット，送信データ，パリティ・ビット，ストップ・
ビットを送信端子に出力します。
なお，このサンプル・プログラムで実現するUART機能は，データ長が9ビット，データ転送方向
がLSBファースト，データ位相が正転出力，ストップ・ビットが1ビットで固定となります。
また，ボー・レートは300～19200bps，パリティ・ビットは有または無（有の場合，ゼロ，偶数，
奇数）から選択可能です。以上の設定で全2重での送受信を行います。また，送受信するデータ
は0～511の9ビットの数値です。


＜使用する周辺の初期設定の主な内容＞

   ・割り込みの禁止
   ・CPU／周辺ハードウエア・クロックを高速システム・クロック(20MHz使用)の動作に設定
   ・ポートの設定
   ・タイマ・アレイ・ユニット0（TAU0）のチャネル1をUART送信用，チャネル0をUART受信用
     に設定
   ・INTP0をUART受信のスタート・ビット検出用に設定
   ・割り込みの許可
```

＜メイン処理の主な内容＞

・UART送信データの作成
・UART受信動作の許可（INTP0割り込み許可）
・UART送信動作の開始（INTTM01割り込み許可，タイマ・チャネル1のカウント開始）

＜INTP0割り込み処理（INTP0の立ち下がりエッジ割り込み使用）＞

・UART受信動作の開始（INTTM00割り込み許可，タイマ・チャネル0のカウント開始）
・INTP0割り込み禁止

＜INTTM00割り込み処理（タイマ・チャネル0のカウント完了割り込み使用）＞

・スタート・ビットのロウ・レベル確認
・データ・ビット，パリティ・ビットの取り込み
・パリティ・チェック
・UART受信動作の停止（INTTM00割り込み禁止，タイマ・チャネル0のカウント停止）

＜INTTM01割り込み処理（タイマ・チャネル1のカウント完了割り込み使用）＞

・UART送信動作の開始（タイマ・チャネル1のカウント開始）
・データ・ビット，パリティ・ビット，およびストップ・ビットの出力
・UART送信動作の停止（INTTM01割り込み禁止，タイマ・チャネル1のカウント停止）

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

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

      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 */
```

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

      関数プロトタイプ宣言

=============================================================================*/
#pragma interrupt INTTM00 fn_inttm00 RB1      /* 割り込み関数宣言:INTTM00 */
#pragma interrupt INTTM01 fn_inttm01 RB1      /* 割り込み関数宣言:INTTM01 */
#pragma interrupt INTP0         fn_intp0 RB1          /* 割り込み関数宣言:INTP0 */


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

      ＲＡＭの定義

;=============================================================================*/
static unsigned short sreg usTdrRx;          /* 受信ボー・レート（TDR00設定値）*/
static unsigned short sreg usTdrTx;          /* 送信ボー・レート（TDR01設定値）*/

static unsigned short sreg usRxBuffer;               /* 受信データ・バッファ */
static unsigned short sreg usTxBuffer;               /* 送信データ・バッファ */

static unsigned short sreg usRxReg;          /* 受信データ・レジスタ */
static unsigned short sreg usTxReg;          /* 送信データ・レジスタ */

static unsigned char sreg ucRxBitCounter;    /* 受信ビット・カウンタ */
static unsigned char sreg ucTxBitCounter;    /* 送信ビット・カウンタ */


static boolean bUartRxEnable;                /* UART受信動作許可フラグ */
static boolean bUartTxEnable;                /* UART送信動作許可フラグ */
static boolean bUartRxComp;                  /* UART受信動作完了フラグ */
static boolean bUartTxComp;                  /* UART送信動作完了フラグ */

static boolean bUartFrameErr;                /* UARTフレーミング・エラー フラグ */
static boolean bUartParityErr;                       /* UARTパリティ・エラー フラグ */

static boolean bUartRxParityEnable;          /* UART受信パリティ有効フラグ */
static boolean bUartRxParityOdd;             /* UART受信奇数パリティ・フラグ (1=奇数：0=偶数) */
static boolean bUartRxParityZero;            /* UART受信ゼロ・パリティ・フラグ */

static boolean bUartTxParityEnable;          /* UART送信パリティ有効フラグ */
static boolean bUartTxParityOdd;             /* UART送信奇数パリティ・フラグ (1=奇数：0=偶数) */
static boolean bUartTxParityZero;            /* UART送信ゼロ・パリティ・フラグ */
```

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

      使用する周辺の初期設定

*****************************************************************************/
void hdwinit(void)
{

      DI();                                 /* 割り込み禁止 */
/*---------------------------------------------------------------------------
      クロック周波数の設定
---------------------------------------------------------------------------
      20MHzのX1発振回路で動作が行えるように設定します
---------------------------------------------------------------------------*/
      CMC = 0b01000001;                     /* クロック動作モード */
            /*|||||||+--------------- AMPH: 10MHz＜fMX≦20MHz */
            /*||||+++---------------- <000> */
            /*|||+------------------- OSCSELS: P123/P124端子を入力ポート */
            /*||+-------------------- <0> */
            /*++--------------------- EXCLK/OSCSEL: X1発振モード(20MHz) */

      CSC = 0b01000000;                     /* 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発振回路停止 */
            /*+----------------------- MSTOP: X1発振回路動作 */

      OSMC = 0b00000001;                    /* 動作スピード・モード */
             /*|||||||+--------------- FSEL: 10MHzを越える周波数で動作 */
             /*+++++++---------------- <00000> */

      OSTS = 0b00000101;                    /* 発振安定時間: 2^15/fX */

      while( (OSTC && 0b00000010) == 0);    /* クロック発振安定待ち */

      CKC = 0b00011000;                     /*クロック選択 */
            /*|||||+++--------------- MDIV2-0: CPU/周辺ハードウエア・クロック(fCLK)=fMX */
            /*||||+------------------ <1> */
            /*|||+------------------- MCM0: 高速システム・クロック（fMX） */
            /*||+-------------------- <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 */
      PM1 = 0b00000000;                 /* P10-P17を出力ポートに設定 */

/*-----------------------------------------------------------------------------
      ポート2の設定
-----------------------------------------------------------------------------*/
      P2  = 0b00000000;                 /* P20-P27の出力ラッチLow */
      PM2 = 0b00000000;                 /* P20-P27を出力ポートに設定 */

/*-----------------------------------------------------------------------------
      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 */
                                        /* [後でP120(INTP0)の設定を行う] */

/*-------------------------------------------------------------------------
      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 */


/*-------------------------------------------------------------------------
      タイマの設定
-------------------------------------------------------------------------*/


      /*---------------------------------------
        TM00の設定
      ---------------------------------------*/
      TAU0EN = 1;                                   /* TAU0使用許可 */

      TPS0L = 0b00000001;                           /* タイマ・クロック選択 */
            /*||||++++--------------- PRS003-000: fCLK/2^1(シリアル・クロックと同じにする)*/
            /*++++------------------- PRS013-010: 未使用 */
```

```
    TMR00 = 0b0000000000000000;     /* 動作モード設定 */
            /*||||||||||||++++------- MD003-000: インターバル・タイマ・モード */
            /*||||||||||++----------- <00> */
            /*||||||||++------------- CIS001-000: */
            /*|||||+++--------------- STS002-000: ソフトウエア・トリガ・スタートのみ有効 */
            /*||||+------------------ 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 */



    /*-----------------------------------------
      TM01の設定
    -----------------------------------------*/

    TMR01 = 0b0000000000000000;             /* 動作モード設定 */
            /*||||||||||||++++------- MD013-010: インターバル・タイマ・モード */
            /*||||||||||++----------- <00> */
            /*||||||||++------------- CIS011-010: */
            /*|||||+++--------------- STS012-010: ソフトウエア・トリガ・スタートのみ有効 */
            /*||||+------------------ MASTER01: 単体動作 */
            /*|||+------------------- CCS01: CKS00ビットで指定した動作クロックMCK */
            /*|++-------------------- <00> */
            /*+---------------------- CKS01: PRSで選択した動作クロックCK00(MCK) */

    TMIF01 = 0;                                 /* 割り込み要求クリア */

/*--------------------------------------------------------------------------
    INTP0(UART RX)の設定
--------------------------------------------------------------------------*/

    EGP0.0 = 0;                                 /* 立ち上がりエッジ無効 */
    EGN0.0 = 1;                                 /* 立ち下がりエッジ有効 */

    PIF0 = 0;                                   /* 割り込み要求クリア */

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

}
```

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

      メイン処理

************************************************************************************/
void main(void)
{
/*--------------------------------------------------------------------------------
      データ初期化
--------------------------------------------------------------------------------*/

      /*---------------------------------------
        TDR00設定値
      ---------------------------------------*/

/*    usTdrRx = 33332;              /* ボー・レート    300bps@20MHz/2^1 */
/*    usTdrRx = 16665;              /* ボー・レート    600bps@20MHz/2^1 */
/*    usTdrRx = 8332;               /* ボー・レート   1200bps@20MHz/2^1 */
/*    usTdrRx = 4165;               /* ボー・レート   2400bps@20MHz/2^1 */
/*    usTdrRx = 2082;               /* ボー・レート   4800bps@20MHz/2^1 */
/*    usTdrRx = 1387;               /* ボー・レート   7200bps@20MHz/2^1 */
/*    usTdrRx = 1040;               /* ボー・レート   9600bps@20MHz/2^1 */
/*    usTdrRx = 693;                /* ボー・レート  14400bps@20MHz/2^1 */
      usTdrRx = 519;                /* ボー・レート  19200bps@20MHz/2^1 */


      /*---------------------------------------
        TDR01設定値
      ---------------------------------------*/

/*    usTdrTx = 33332;              /* ボー・レート    300bps@20MHz/2^1 */
/*    usTdrTx = 16665;              /* ボー・レート    600bps@20MHz/2^1 */
/*    usTdrTx = 8332;               /* ボー・レート   1200bps@20MHz/2^1 */
/*    usTdrTx = 4165;               /* ボー・レート   2400bps@20MHz/2^1 */
/*    usTdrTx = 2082;               /* ボー・レート   4800bps@20MHz/2^1 */
/*    usTdrTx = 1387;               /* ボー・レート   7200bps@20MHz/2^1 */
/*    usTdrTx = 1040;               /* ボー・レート   9600bps@20MHz/2^1 */
/*    usTdrTx = 693;                /* ボー・レート  14400bps@20MHz/2^1 */
      usTdrTx = 519;                /* ボー・レート  19200bps@20MHz/2^1 */


      /*====================================================================
        データ初期化
```

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

    usRxBuffer = 0x0000;                /* 受信データ・バッファ */

    usTxBuffer = 0x0000;                /* 送信データ・バッファ */

    usRxReg = 0x0000;                   /* 受信データ・レジスタ */

    usTxReg = 0x0000;                   /* 送信データ・レジスタ */

    ucRxBitCounter = 0;                 /* 受信ビット・カウンタ */
    ucTxBitCounter = 0;                 /* 送信ビット・カウンタ */

    bUartRxEnable = 0;                  /* UART受信動作許可フラグ */
    bUartTxEnable = 0;                  /* UART送信動作許可フラグ */
    bUartRxComp = 0;                    /* UART受信動作完了フラグ */
    bUartTxComp = 0;                    /* UART送信動作完了フラグ */

    bUartFrameErr = 0;                  /* UARTフレーミング・エラー フラグ */
    bUartParityErr = 0;                 /* UARTパリティ・エラー フラグ */

    PM12.0 = 1;                         /* データ入力端子 */
    P13.0 = 1;                          /* データ出力端子 */


    /*================================================================
      UART受信モードの設定
    ================================================================*/

    bUartRxParityEnable = 1;            /* UART受信パリティ有 */
/*  bUartRxParityEnable = 0;            /* UART受信パリティ無 */

    bUartRxParityOdd = 1;               /* UART受信奇数パリティ */
/*  bUartRxParityOdd = 0;               /* UART受信偶数パリティ */

    bUartRxParityZero = 0;              /* UART受信ゼロ・パリティ無効 */
/*  bUartRxParityZero = 1;              /* UART受信ゼロ・パリティ有効 */

    /*================================================================
      UART送信モードの設定
    ================================================================*/

    bUartTxParityEnable = 1;            /* UART送信パリティ有 */
```

```
/*    bUartTxParityEnable = 0;              /* UART送信パリティ無 */

      bUartTxParityOdd = 1;                 /* UART送信奇数パリティ */
/*    bUartTxParityOdd = 0;                 /* UART送信偶数パリティ */

      bUartTxParityZero = 0;                /* UART送信ゼロ・パリティ無効 */
/*    bUartTxParityZero = 1;                /* UART送信ゼロ・パリティ有効 */



/*====================================================================
  メイン・ルーチン
====================================================================*/

      while (1)
      {
            if((bUartRxEnable == 0) && (bUartTxEnable == 0)){       /* 送受信動作停止 */
                  if((bUartRxComp == 0) && (bUartTxComp == 0)){     /* 送受信完了要求無 */

                  /*************************************
                   データ受信開始
                  *************************************/

                        usRxReg = 0x0000;                           /* 受信データ・レジスタ */

                        bUartRxEnable = 1;                          /* UART受信動作許可 */
                        bUartRxComp = 0;                            /* UART受信動作完了フラグ クリ
ア*/

                        bUartFrameErr = 0;                          /* UARTフレーミング・エラー ク
リア */
                        bUartParityErr = 0;                         /* UARTパリティ・エラー クリア
*/

                        PIF0 = 0;                                   /* スタート･ビット割り込み要求
クリア */
                        PMK0 = 0;                                   /* スタート･ビット割り込み許可
*/

                  /*************************************
                    データ送信開始
                  *************************************/
```

```
                        usTxBuffer++;                   /* 送信データ下位更新 */
                        usTxBuffer &= 0x1ff;            /* 9ビットに丸める */

                        P13.0 = 1;                      /* データ出力端子初期化 */

                        usTxReg = usTxBuffer;           /* 送信データ設定 */

                        ucTxBitCounter = (9 + 4);       /* 送信ビット・カウンタ 初期化
*/

                        bUartTxEnable = 1;              /* UART送信動作中設定 */
                        bUartTxComp = 0;                /* UART送信動作完了クリア */

                        TMIF01 = 0;                     /* 割り込み要求クリア */
                        TMMK01 = 0;                     /* 割り込み許可 */
                        TMIF01 = 1;                     /* 割り込み要求セット（送信開
始）*/
                }
        }

        /***************************************
         データ送信完了
        ***************************************/

        if(bUartTxComp == 1){                   /* 送信完了要求有 */
                bUartTxComp = 0;                /* UART送信動作完了クリア */
        }

        /***************************************
         データ受信完了
        ***************************************/

        if(bUartRxComp == 1){                   /* 受信完了要求有 */
                usRxBuffer = usRxReg;           /* 受信データ・バッファに受信データを保
存 */
                bUartRxEnable = 0;              /* UART受信動作停止 */
                bUartRxComp = 0;                /* UART受信動作完了フラグ クリア*/

                bUartFrameErr = 0;              /* UARTフレーミング・エラー クリア */
                bUartParityErr = 0;             /* UARTパリティ・エラー クリア */
        }
    }
}
```

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

      INTP0割り込み処理（INTP0の立ち下がりエッジ割り込み使用）

*****************************************************************************/
__interrupt void fn_intp0(void)
{

      NOP();
      NOP();
      if(P12.0 == 0){                                   /* ノイズ(High)チェック */
            PMK0 = 1;                                   /* スタート・ビット検出割り込み
(この割り込み）禁止 */
            TDR00 = usTdrRx / 2;                        /* 半ビット時間設定 */

            ucRxBitCounter = (9 + 2);                   /* 受信ビット・カウンタ設定 */
            if(bUartRxParityEnable == 1){
                  ucRxBitCounter++;                     /* パリティ・ビット分のカウンタ
調整 */
            }
            TS0L.0 = 1;                                 /* サンプリング・タイマのカウン
ト動作開始(トリガ動作) */
            TMIF00 = 0;                                 /* サンプリング・タイマ割り込み
要求クリア */
            TMMK00 = 0;                                 /* サンプリング・タイマ割り込み
許可 */

      }
}



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

      INTTM00割り込み処理（タイマ・チャネル0のカウント完了割り込み使用）

*****************************************************************************/
__interrupt void fn_inttm00(void)
{
      static  unsigned char sreg ucRxParityWork;
      static  unsigned char sreg ucRxRegWork;

      ucRxBitCounter--;                                 /* 受信ビット位置カウント */
```

```
    if(ucRxBitCounter != 0){                                    /* ストップ・ビット位置以外の場
合 */
        /*----------------------------------------
          スタート・ビットの確認
        ----------------------------------------*/
        if( ((ucRxBitCounter == (9 + 1)) && (bUartRxParityEnable == 0)) ||
            ((ucRxBitCounter == (9 + 2)) && (bUartRxParityEnable == 1))   ){
            if(P12.0 == 0){                                     /* スタート・ビットのレベルOKの
場合 */
                TDR00 = usTdrRx;                                /* １ビット時間設定 */
            }
            else{                                               /* スタート・ビットのレベルNGの
場合 */
                TMMK00 = 1;                                     /* サンプリング・タイマ割り込み
禁止 */
                TT0L.0 = 1;                                     /* サンプリング・タイマのカウン
ト動作停止 */
                TMIF00 = 0;                                     /* 割り込み要求クリア */

                if(bUartRxEnable == 1){                         /* 受信動作中の場合 */
                    PMK0 = 0;                                   /* スタート･ビット割り込み許可
*/
                }

            }
        }
        /*----------------------------------------
          データ・ビット,パリティ・ビットの取り込み
        ----------------------------------------*/
        else{                                                   /* データ・ビット位置の場合 */
            usRxReg >>= 1;                                      /* 前回までの受信データを1ビッ
ト シフト */
            if(P12.0 == 1){                                     /* 今回の受信データが１の場合
*/
                if(bUartRxParityEnable == 1){                   /* パリティ・ビット有の場合 */
                    usRxReg |= 0b1000000000;                    /* 今回のデータ(1)を設定 */
                }
                else{
                    usRxReg |= 0b100000000;                     /* 今回のデータ(1)を設定 */
                }
            }
        }
    }
```

```
        /*----------------------------------------
          ストップ・ビットの確認
        ----------------------------------------*/
        else{
                bUartRxEnable = 0;                                  /* UART受信動作中解除 */
                bUartRxComp = 1;                                    /* UART受信動作完了通知 */

                TMMK00 = 1;                                         /* サンプリング・タイマ割り込み
禁止 */
                TT0L.0 = 1;                                         /* サンプリング・タイマのカウン
ト動作停止 */
                TMIF00 = 0;                                         /* 割り込み要求クリア */

                if(P12.0 == 0){                                     /* ストップ・ビットNG */
                        bUartFrameErr = 1;                          /* フレーミング・エラー通知 */
                }
                /*----------------------------------------
                  パリティ・チェック
                ----------------------------------------*/

                if(bUartRxParityEnable == 1){                       /* パリティ・ビット有の場合 */

                        ucRxRegWork = (unsigned char)(usRxReg >> 8);    /* 受信データbit8、パリティ・ビ
ット取得 */
                        ucRxParityWork = ucRxRegWork;
                        ucRxRegWork >>= 1;
                        if(bUartRxParityZero == 1){                 /* ゼロ・パリティの場合 */
                                ucRxParityWork = ucRxRegWork;       /* パリティ・ビット取得 */
                        }
                        else{                                       /* 偶数/奇数パリティの場合 */
                                ucRxParityWork ^= ucRxRegWork;      /* パリティ・ビット ^ bit8 */

                                ucRxRegWork = (unsigned char)usRxReg;   /* 受信データbit7～bit0取得 */
                                ucRxParityWork ^= ucRxRegWork;      /* ^ bit0 */
                                ucRxRegWork >>= 1;
                                ucRxParityWork ^= ucRxRegWork;      /* ^ bit1 */
                                ucRxRegWork >>= 1;
                                ucRxParityWork ^= ucRxRegWork;      /* ^ bit2 */
                                ucRxRegWork >>= 1;
                                ucRxParityWork ^= ucRxRegWork;      /* ^ bit3 */
                                ucRxRegWork >>= 1;
                                ucRxParityWork ^= ucRxRegWork;      /* ^ bit4 */
                                ucRxRegWork >>= 1;
```

```
                        ucRxParityWork ^= ucRxRegWork;          /* ^ bit5 */
                        ucRxRegWork >>= 1;
                        ucRxParityWork ^= ucRxRegWork;          /* ^ bit6 */
                        ucRxRegWork >>= 1;
                        ucRxParityWork ^= ucRxRegWork;          /* ^ bit7 */

                        ucRxParityWork &= 0x0001;               /* 1となっているビットの数が偶
数か奇数か */
                        if(bUartRxParityOdd == 1){              /* 奇数パリティの場合 */
                                ucRxParityWork ^= 0x0001;       /* 奇数・偶数に関わらず０の時OK
とする */
                        }
                }
                if((ucRxParityWork & 0x01) != 0){               /* パリティNGの場合 */
                        bUartParityErr = 1;                     /* パリティ・エラー */
                }

        }
        usRxReg &= 0x1ff;                                       /* データのみとする */
    }
}


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

    INTTM01割り込み処理（タイマ・チャネル1のカウント完了割り込み使用）

*****************************************************************************/
__interrupt void fn_inttm01(void)
{
    static  unsigned char sreg ucTxParityWork;
    static  unsigned char sreg ucTxRegWork;

    ucTxBitCounter--;                                           /* 受信ビット位置カウント */

    /*-----------------------------------------
      送信開始（スタート・ビット出力開始）
    -----------------------------------------*/
    if(ucTxBitCounter == (9+3)){                                /* 送信開始の場合 */
        if( bUartTxParityEnable == 0){                          /* 送信パリティ・ビット無の場合
*/
            ucTxBitCounter--;                                   /* ビット位置カウンタ補正 */
            usTxReg |= 0x200;                                   /* ストップ・ビット付加 */
```

```
        }
        else{                                           /* 送信パリティ・ビット有の場合
*/
                usTxReg &= 0xdff;                       /* パリティ・ビットを0に仮設定
*/
                if(bUartTxParityZero == 0){             /* ゼロ･パリティ以外(偶数/奇数)
の場合 */
                        ucTxParityWork = (unsigned char)(usTxReg >> 8); /* bit8 */
                        ucTxRegWork = (unsigned char)(usTxReg & 0xff);  /* bit0 ～ bit7取得 */
                        ucTxParityWork ^= ucTxRegWork;  /* bit8 ^ bit0 */
                        ucTxRegWork >>= 1;
                        ucTxParityWork ^= ucTxRegWork;  /* ^ bit1 */
                        ucTxRegWork >>= 1;
                        ucTxParityWork ^= ucTxRegWork;  /* ^ bit2 */
                        ucTxRegWork >>= 1;
                        ucTxParityWork ^= ucTxRegWork;  /* ^ bit3 */
                        ucTxRegWork >>= 1;
                        ucTxParityWork ^= ucTxRegWork;  /* ^ bit4 */
                        ucTxRegWork >>= 1;
                        ucTxParityWork ^= ucTxRegWork;  /* ^ bit5 */
                        ucTxRegWork >>= 1;
                        ucTxParityWork ^= ucTxRegWork;  /* ^ bit6 */
                        ucTxRegWork >>= 1;
                        ucTxParityWork ^= ucTxRegWork;  /* ^ bit7 */

                        ucTxParityWork &= 0x0001;       /* パリティ・ビット(bit0)を取得
*/
                        if(bUartTxParityOdd == 1){      /* 奇数パリティの場合 */
                                ucTxParityWork ^= 0x0001;       /* パリティ・ビット(bit0)を反転
*/
                        }
                        if(ucTxParityWork == 0x0001){   /* パリティ･ビットが１の場合 */

                                usTxReg |= 0x0200;      /* パリティ･ビットに１を設定 */
                        }
                }
                usTxReg |= 0x400;                       /* ストップ・ビット付加 */
        }

        TDR01 = usTdrTx;                                /* 1ビット時間(データ出力タイミ
ング)設定 */

        TSOL.1 = 1;                                     /* データ出力タイマのカウント動
```

```
作開始 */
            TMIF01 = 0;                                     /* 割り込み要求クリア */

            P13.0 = 0;                                      /* スタート・ビット出力 */
      }
      else{
            /*----------------------------------------
              データ・ビット、パリティ・ビット、
              ストップ・ビットの出力
            ----------------------------------------*/
            if(ucTxBitCounter != 0){

                  if((usTxReg & 0x0001) == 0){              /* 出力データが０の場合 */
                        P13.0 = 0;                          /* 送信データ端子にLow出力 */
                  }
                  else{
                        P13.0 = 1;                          /* 送信データ端子にHigh出力 */
                  }
                  usTxReg >>= 1;                            /* 次回出力データ設定 */
            }
            else{
            /*----------------------------------------
              ストップ・ビット出力終了
              (9ビット送信完了)
            ----------------------------------------*/

                  bUartTxEnable = 0;                        /* UART送信動作中解除 */
                  bUartTxComp = 1;                          /* UART送信動作完了通知 */
                  TMMK01 = 1;                               /* 送信タイマ割り込み禁止 */
                  TT0L.1 = 1;                               /* データ出力タイマのカウント動
作停止 */

            }
      }
}
```

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。