

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ひ申し上げます。

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のですが、誤りが無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、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と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。

アプリケーション・ノート

78K0R/Kx3

サンプル・プログラム (UART)

ロウ・レベルの入力パルスを使用したボー・レート補正付きUART編

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

対象デバイス

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

目次

第1章 概要 ...	3
第2章 回路図 ...	6
2.1 回路図 ...	6
2.2 周辺ハードウェア ...	7
第3章 ソフトウェアについて ...	8
3.1 ファイル構成 ...	8
3.2 使用する内蔵周辺機能 ...	9
3.3 使用する周辺の初期設定と動作概要 ...	10
3.4 フロー・チャート ...	11
3.5 キャリブレーション ...	13
3.6 UART送受信データのフォーマット ...	15
第4章 設定方法について ...	16
4.1 シリアル・アレイ・ユニット1 (SAU1) の設定 ...	17
4.2 使用する周辺の初期設定 ...	30
4.3 メイン処理 ...	37
4.4 キャリブレーション開始処理 ...	41
4.5 INTST3割り込み処理 ...	43
4.6 INTSR3割り込み処理 ...	45
4.7 INTTM07割り込み処理 ...	46
第5章 関連資料 ...	49
付録A プログラム・リスト ...	50
付録B 改版履歴 ...	85

資料番号 U19223JJ1V0AN00 (第1版)
 発行年月 December 2008 NS

- 本資料に記載されている内容は2008年12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とその総株主の議決権の過半数を直接または間接に保有する会社をいう。
- (2) 本事項において使用されている「当社製品」とは、(1)において定義された当社の開発、製造製品をいう。

M8E0710J

第1章 概 要

このサンプル・プログラムは、シリアル・アレイ・ユニットのUARTの機能を使用したUART送受信を行うプログラムです。まず、UART送受信を行う前にキャリブレーション開始処理の呼び出しにより、キャリブレーションを行います。キャリブレーションは、マスタ機器から送信されるデータ長が8ビット、転送レートが9600bps、データ位相が正転出力、パリティ・ビットなし、ストップ・ビットが1ビット付加、8ビットのデータが00Hのデータを受信します。受信したデータをタイマ入力端子へのロウ・レベルの入力パルスとして、立ち下がりエッジから立ち上がりエッジまでのロウ・レベル間的高速内蔵発振クロックでのカウント数を計測します。計測したタイマ・カウント値を高速内蔵発振クロックを動作クロックとしているシリアル・アレイ・ユニットの転送ボー・レート用に計算し、設定します。キャリブレーション後、マスタ機器からA/Dコンバータのチャンネルと分解能の指定値を受信し、指定されたA/Dコンバータのチャンネルで変換したA/D値を指定された分解能でマスタ機器へ送信します。

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

(1) 使用する周辺の初期設定の主な内容

使用する周辺の初期設定の主な内容は、次のとおりです。

割り込みの禁止

CPU / 周辺ハードウェア・クロックを高速内蔵発振クロックの動作に設定

ポートの設定

A/Dコンバータの設定

シリアル・アレイ・ユニット1 (SAU1) の設定

- ・チャンネル2をUART3送信用に設定
- ・チャンネル3をUART3受信用に設定

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。

- ・ UART3のエラー・チェック
- ・ UART3受信データの解析
- ・ A/D値の取り込み
- ・ UART3送信データの作成
- ・ UART3送受信動作の開始

(3) キャリブレーション開始処理の内容

キャリブレーション開始処理の主な内容は、次のとおりです。

- ・入力切り替え制御レジスタ (ISC) により, RxD3端子の入力信号をタイマ入力に設定
- ・タイマ・アレイ・ユニット0 (TAU0) のチャンネル7をキャプチャ・モードに設定
- ・タイマ・アレイ・ユニット0 (TAU0) のチャンネル7をTI07端子での両エッジ検出に設定
- ・タイマ・アレイ・ユニット0 (TAU0) のチャンネル7のキャプチャ動作開始

(4) INTST3割り込み処理 (UART3の送信完了割り込み)

INTST3割り込み処理の主な内容は、次のとおりです。

- ・UART3送信データのカウンタ
- ・UART3送信データの設定
- ・UART3送信動作の停止

(5) INTSR3割り込み処理 (UART3受信完了割り込み)

INTSR3割り込み処理の主な内容は、次のとおりです。

- ・UART3受信データのカウンタ
- ・UART3受信データの保存
- ・UART3受信動作の停止

(6) INTTM07割り込み処理 (TI07端子での両エッジ検出)

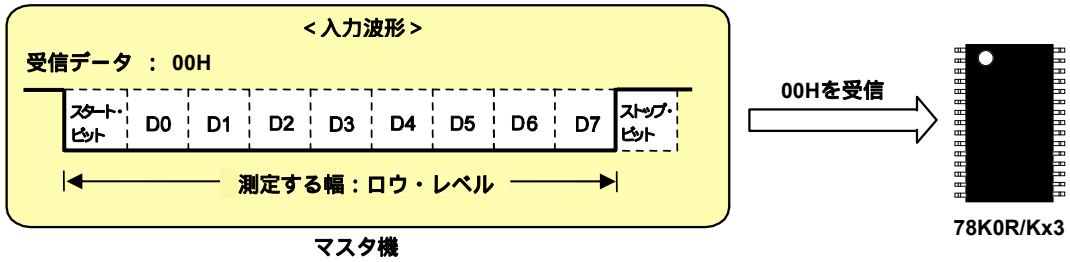
キャリブレーション開始処理の呼び出し後, マスタ機器から入力したパルスのロウ・レベル間の計測完了時に発生する割り込み処理です。

INTRTC割り込み処理の主な内容は、次のとおりです。

- ・タイマ・アレイ・ユニット0 (TAU0) のチャンネル7のキャプチャ動作停止
- ・キャプチャ測定値から高速内蔵発振クロックで動作するUART3用転送ポー・レートの設定
- ・タイマ・アレイ・ユニット0 (TAU0) のチャンネル7のキャプチャ・モードで計測したロウ・レベル間の高速内蔵発振クロックのタイマ・カウンタ値からUART3の転送ポー・レートの設定
- ・UART3の受信動作開始

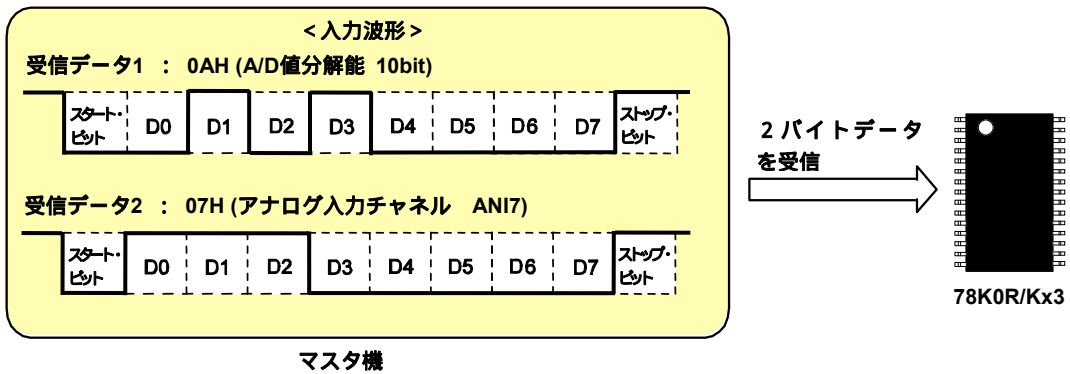
【動作概要】

- マスタ機 78K0R/Kx3



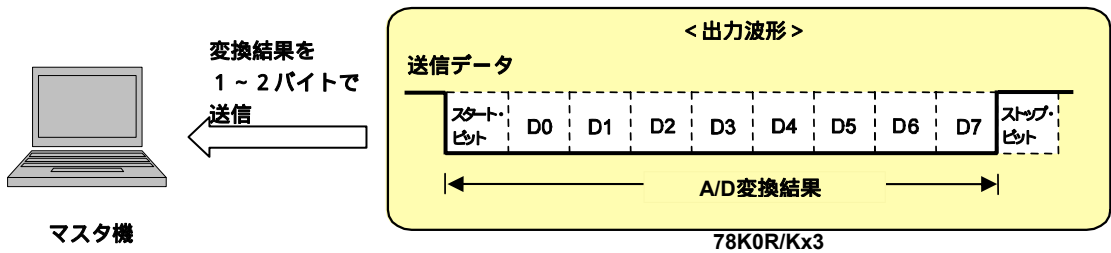
受信データよりUART転送ボー・レートを計算し設定

- マスタ機 78K0R/Kx3



受信データの設定に変更後、A/D変換を実行

- マスタ機 78K0R/Kx3

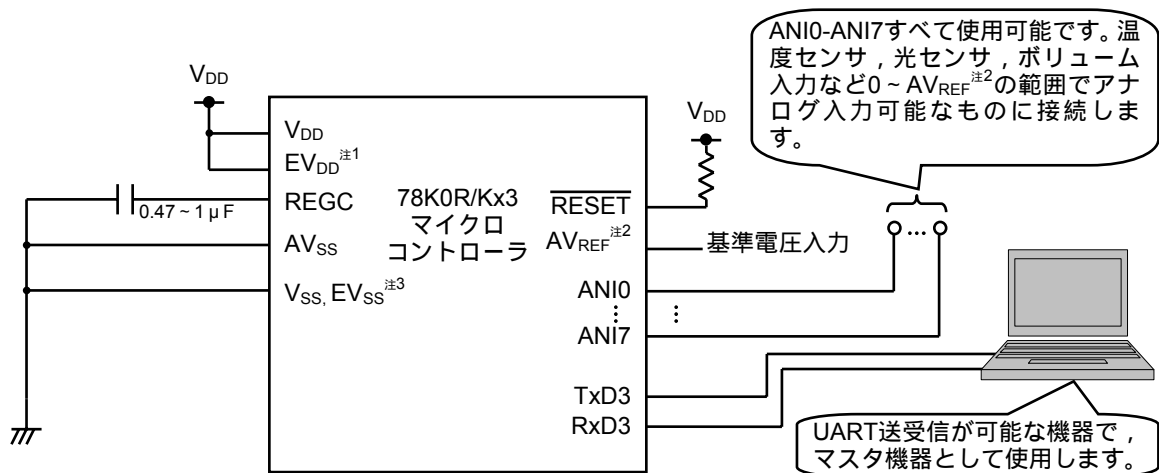


第2章 回路図

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

2.1 回路図

回路図を次に示します。



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

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

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

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

2. AV_{SS}端子はEV_{SS}, V_{SS}と同電位にし、GNDに直接接続してください。
3. EV_{DD}は、 V_{DD} と同電位にしてください。
4. REGCはコンデンサ(0.47～1 μ F)を介し、V_{SS}に接続してください。
5. 回路図中の端子以外の未使用のポート機能端子はすべて出力ポートのため、オープン(未接続)にしてください。
6. ANI0-ANI7の端子はすべて使用可能です。温度センサ、光センサ、ボリューム入力など0～AV_{REF}(上記注2参照)の範囲でアナログ入力可能なものに接続します。
7. TxD3, RxD3端子は、UART送受信が可能な機器に接続します。
8. RxD3端子は、入力切り替え制御レジスタ(ISC)の設定により、キャリブレーション時に入力信号をタイマ入力(TI07)として設定しています。

2.2 周辺ハードウェア

使用する周辺ハードウェアを次に示します。

(1) UART通信機器 (TxD3, RxD3)

TxD3, RxD3端子にUART送受信の機器を接続します。

(2) A/Dコンバータのアナログ入力端子 (ANI0-ANI7)

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



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

第3章 ソフトウェアについて

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

3.1 ファイル構成

ダウンロードする圧縮ファイルのファイル構成は、次のようになっています。

ファイル名	説明	同封圧縮 (*.zip) ファイル	
			
main.asm (アセンブリ言語版)	マイコンのハードウェア初期化処理、メイン処理と割り込み処理のソース・ファイル	注	注
main.c (C言語版)			
Uart_ZeroData.prw	統合開発環境 PM+用ワーク・スペース・ファイル		
Uart_ZeroData.prj	統合開発環境 PM+用プロジェクト・ファイル		

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

備考



: ソース・ファイルのみ同封



: 統合開発環境 PM+で使用するファイルを同封

3.2 使用する内蔵周辺機能

このサンプル・プログラムでは、マイコンに内蔵する次の周辺機能を使用します。

- ・ UART送信用： シリアル・アレイ・ユニット1 (SAU1) のチャンネル0をUART3送信として使用
- ・ UART受信用： シリアル・アレイ・ユニット1 (SAU1) のチャンネル1をUART3受信として使用
- ・ マスタ機器への送信データとして使用する電圧値の計測用：
 - A/Dコンバータのチャンネル0～7を使用
- ・ 高速内蔵発振クロックのカウント数の計測用：
 - 入力パルスのロウ・レベル幅を計測するためにタイマ・アレイ・ユニット0 (TAU0) のチャンネル7をキャプチャ・モードで使用

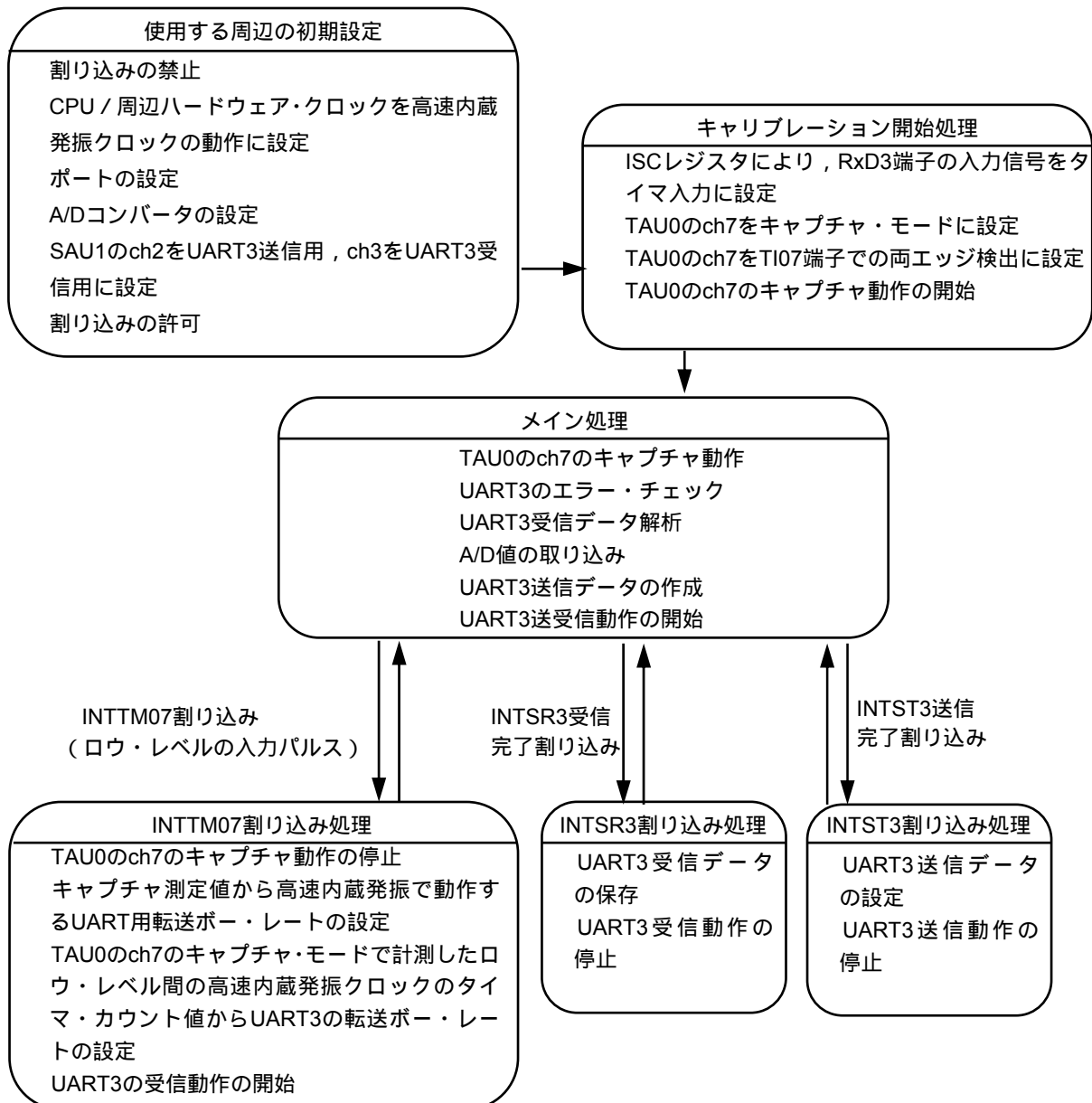
3.3 使用する周辺の初期設定と動作概要

このサンプル・プログラムでは、使用する周辺の初期設定にて、クロック周波数の選択や、A/Dコンバータ、シリアル・アレイ・ユニットを使用したUARTの設定などを行います。

使用する周辺の初期設定完了後、最初のUARTを受信する前に一度だけキャリブレーション開始処理を呼び出します。このサンプル・プログラムでは最初のUART受信を行う前にキャリブレーションを一度行うだけですが、一定周期でキャリブレーションを行うなど、セットにおいて必要な精度を満たせる十分な評価のもと調整してください。キャリブレーション開始処理では、マスタ機器から送信されるデータ長が8ビット、転送レートが9600bps、データ位相が正転出力、パリティ・ビットなし、ストップ・ビットが1ビット付加、8ビットのデータが00Hのデータを受信します。受信したデータをタイマ入力端子へのロウ・レベルの入力パルスとして、立ち下がりエッジから立ち上がりエッジまでのロウ・レベル間的高速内蔵発振クロックでのカウント数を計測します。計測したタイマ・カウント値を高速内蔵発振クロックを動作クロックとしているシリアル・アレイ・ユニットの転送ボー・レート用に計算し、設定します。

キャリブレーション後、マスタ機器からA/Dコンバータのチャンネルと分解能の指定値を受信し、指定されたA/Dコンバータのチャンネルで変換したA/D値を指定された分解能でマスタ機器へ送信します。

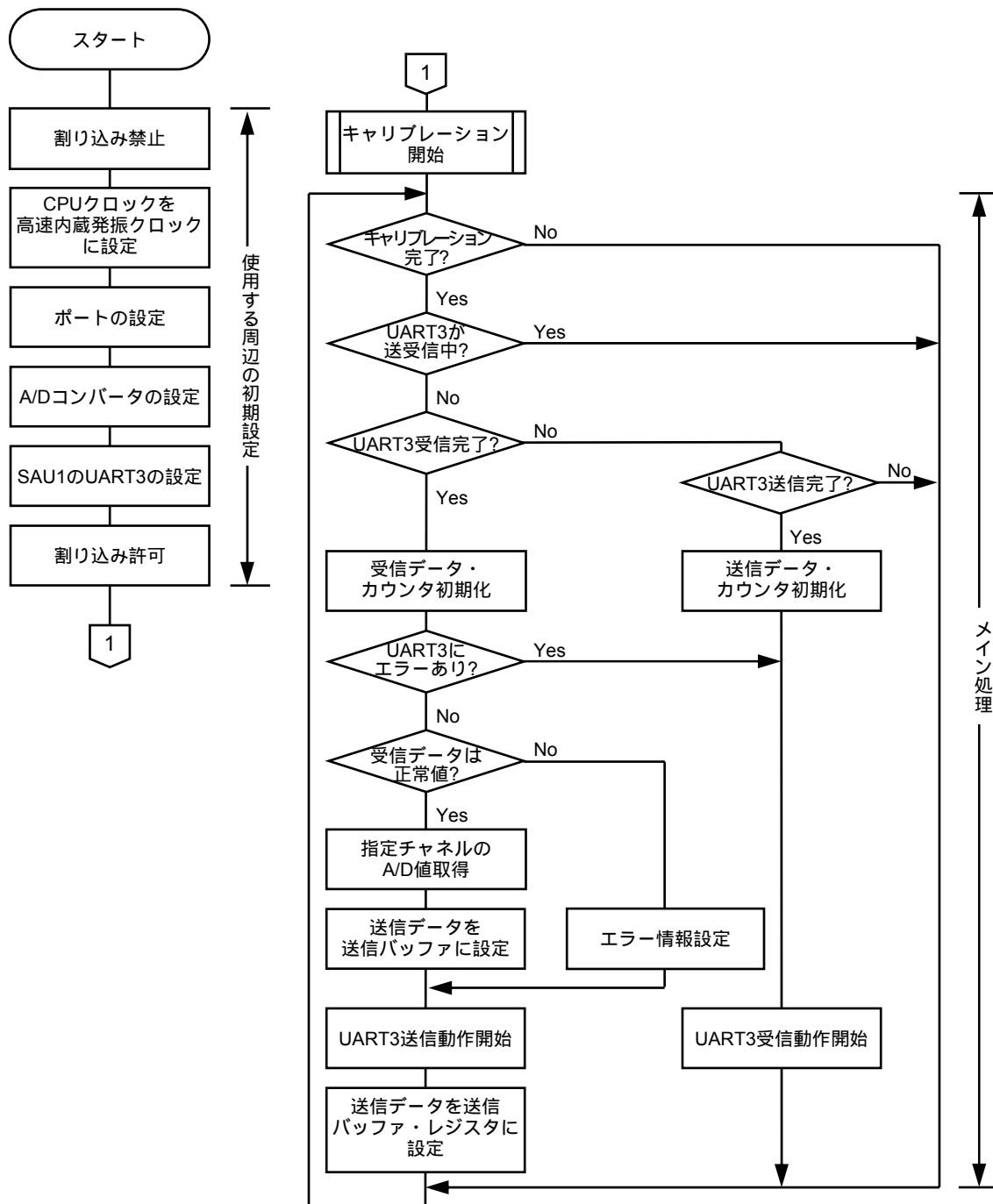
詳細については、次の状態遷移図（ステート・チャート）に示します。



3.4 フロー・チャート

このサンプル・プログラムのフロー・チャートを次に示します。

<リセット解除後の初期化処理>

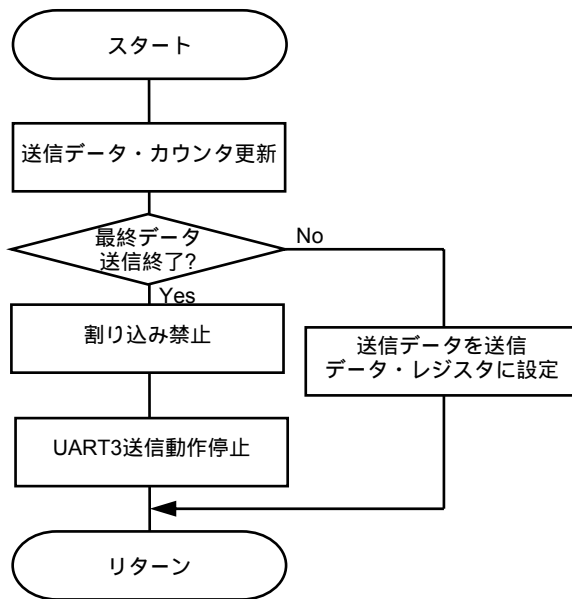


注意 オプション・バイトは、RA78K0Rのリンク・オプションにて設定してください。設定の仕方については、RA78K0R アセンブラ・パッケージ ユーザーズ・マニュアルを参照してください。

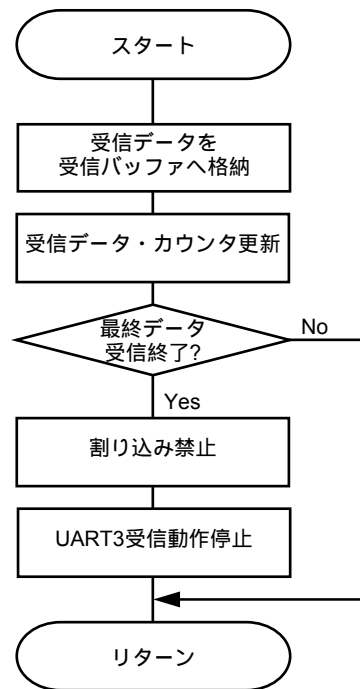
オプション・バイトは、次の内容が設定されます。

- ・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

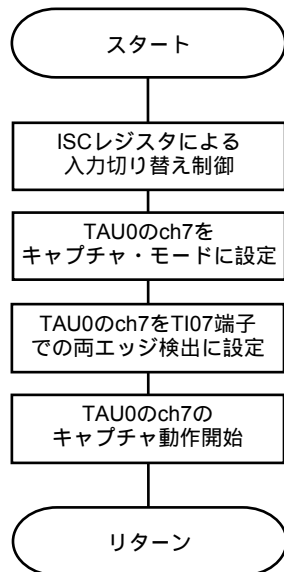
<INTST3割り込み処理>



<INTSR3割り込み処理>



<キャリブレーション開始処理>



<INTTM07割り込み処理>

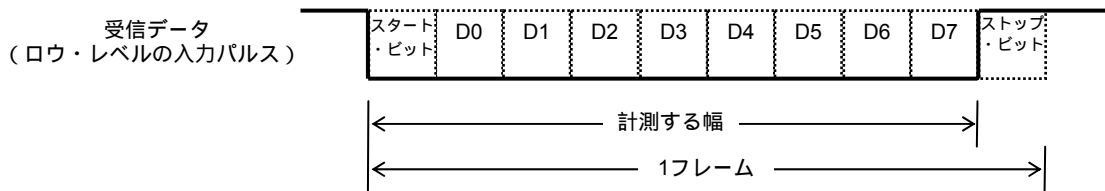


3.5 キャリブレーション

キャリブレーションは、マスタ機器から送信されるデータ長が8ビット、転送レートが9600bps、データ位相が正転出力、パリティ・ビットなし、ストップ・ビットが1ビット付加、8ビットのデータが00Hのデータを受信します。受信したデータをタイマ入力端子へのロウ・レベルの入力パルスとして、立ち下がりエッジから立ち上がりエッジまでのロウ・レベル間的高速内蔵発振クロックでのカウント数を計測します。計測したタイマ・カウント値を高速内蔵発振クロックを動作クロックとしているシリアル・アレイ・ユニットの転送ボー・レート用に計算し、設定します。

- 注意1. 本サンプル・プログラムでは、使用する周辺の初期設定完了後、最初に受信したデータを無条件にキャリブレーション用のデータとします。キャリブレーションが失敗した場合は、UARTの動作を行いません。
2. 本サンプル・プログラムでは最初のUART受信を行う前にキャリブレーションを一度行うだけですが、一定周期でキャリブレーションを行うなど、セットにおいて必要な精度を満たせる十分な評価のもと調整してください。

【キャリブレーション用データのフォーマット】



<通信フォーマット>

ボーレート	パリティ・ビット	データ長	ストップ・ビット	データ
9600bps	なし	8bit	1bit	00H

【ボー・レートの算出】

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

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

で表されます。(詳細はユーザーズ・マニュアルを参照してください。)

- ・CPU/周辺ハードウェア・クロック …… f_{CLK} [Hz]
- ・ボー・レート …… 9600 [bps]

として、式から、

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

と表せます。

また、高速内蔵発振クロックの22分周で動作するタイマ・アレイ・ユニット0 (TAU0) のチャンネル7のキャプチャで計測した時間と、転送レートが9600bpsのデータ9ビット分の時間は等しいことから、タイマ・カウント値 (TimerCount) は、

$$1 / 9600 \times 9 = \{ 2^2 / f_{\text{CLK}} \} \times (\text{TimerCount}) [\text{s}]$$

と表せますので、 f_{CLK} は、

$$f_{\text{CLK}} / 2^2 = (\text{TimerCount}) \times 9600 \div 9 \dots$$

となり、式から、

$$\text{SDRmn}[15:9] = \{ (\text{TimerCount}) \times 9600 \div 9 \} \div 9600 \div 2 - 1 = (\text{TimerCount}) \div 18 - 1$$

上記の式より、シリアル・データ・レジスタ(SDR1x)のUART3転送ボー・レート用動作クロックの分周値の算出が行えます。

【例】 TimerCount …… 1,800 の時

- ・CPU/周辺ハードウェア・クロック …… 8 [Hz]
- ・ボー・レート …… 9600 [bps]
- ・タイマ・アレイ・ユニット0 (TAU0) …… 2 [MHz]

$$\begin{aligned} \text{SDRmn}[15:9] &= \text{TimerCount} \div 18 - 1 = (1,800 \div 18) - 1 \\ &= 99 \end{aligned}$$

SDRmn には …… 「C600H」 を設定

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

3.6 UART送受信データのフォーマット

【受信データ】・・・2バイト

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

【送信データ】・・・2バイト

8ビット分解能の場合

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

10ビット分解能の場合

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

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

第4章 設定方法について

この章では、シリアル・アレイ・ユニット1 (SAU1) のUART3送受信の設定、使用する周辺の使用する周辺の初期設定、メイン処理、キャリブレーション開始処理、INTST3割り込み処理、INTSR3割り込み処理、およびINTTM07割り込み処理について説明します。

表の赤字部分がサンプル・プログラムでの設定値となります。

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

レジスタ設定方法の詳細については、各製品のユーザズ・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 ユーザズ・マニュアルを参照してください。

4.1 シリアル・アレイ・ユニット1 (SAU1) の設定

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

本サンプル・プログラムでは、シリアル・アレイ・ユニット1 (SAU1) のチャンネル2, 3を使用したUART3により、データ長が8ビット、転送レートが9600bps、データ位相が正転出力、パリティ・ビットなし、ストップ・ビットが1ビット付加、データ方向がLSBファースト転送、半2重でのUART送受信が行えるように設定を行います。

(1) シリアル・アレイ・ユニット1 (SAU1) の入力クロックの制御

シリアル・アレイ・ユニット1 (SAU1) を使用できるように、周辺ハードウェア・マクロの使用を可能にするために周辺イネーブル・レジスタ0 (PER0) の設定を行います。使用しないハードウェ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	SAU1EN	TAU1EN	TAU0EN
シリアル・アレイ・ユニット1 (SAU1EN) の入力クロックの制御							
0							
入力クロック供給停止							
・シリアル・アレイ・ユニット1で使用するSFRへのライト不可							
・シリアル・アレイ・ユニット1リセット状態							
1							
入力クロック供給							
・シリアル・アレイ・ユニット1で使用するSFRへのリード/ライト可							

- 注意1.** シリアル・アレイ・ユニットmの設定をする際には、必ず最初にSAUmEN = 1の設定を行ってください。SAUmEN = 0の場合は、シリアル・アレイ・ユニットmの制御レジスタへの書き込みは無視され、読み出しても値はすべて初期値となります (入力切り替え制御レジスタ (ISC)、ノイズフィルタ許可レジスタ (NFEN0)、ポート入力モード・レジスタ (PIM0)、ポート出力モード・レジスタ (POM0)、ポート・モード・レジスタ (PM0, PM1)、ポート・レジスタ (P0, P1) は除く)。
2. PER0レジスタを“1”に設定後に、4クロック以上間隔をあけてからSPSmレジスタを設定してください。

備考 m : ユニット番号 (m = 0, 1)

(2) 動作クロック (CK10) の選択

シリアル・アレイ・ユニット1 (SAU1) のチャンネル2, 3で共通して供給される2種類の動作クロック (CK10, CK11) を選択するため, シリアル・クロック選択レジスタ1 (SPS1) の設定を行います。SPS1のビット7-4でCK11を, ビット3-0でCK10を選択します。

図4 - 1 - 2 シリアル・クロック選択レジスタ1 (SPS1) のフォーマット

アドレス : F0166H, F0167H

0	0	0	0	0	0	0	0	PRS113	PRS112	PRS111	PRS110	PRS103	PRS102	PRS101	PRS100
---	---	---	---	---	---	---	---	--------	--------	--------	--------	--------	--------	--------	--------

												動作クロック (CK10) の選択				
												0	0	0	0	f_{CLK}
												0	0	0	1	$f_{CLK}/2$
												0	0	1	0	$f_{CLK}/2^2$
												0	0	1	1	$f_{CLK}/2^3$
												0	1	0	0	$f_{CLK}/2^4$
												0	1	0	1	$f_{CLK}/2^5$
												0	1	1	0	$f_{CLK}/2^6$
												0	1	1	1	$f_{CLK}/2^7$
												1	0	0	0	$f_{CLK}/2^8$
												1	0	0	1	$f_{CLK}/2^9$
												1	0	1	0	$f_{CLK}/2^{10}$
												1	0	1	1	$f_{CLK}/2^{11}$
												1	1	1	1	INTTM03
												上記以外		設定禁止		
												動作クロック (CK11) の選択 (本サンプル・プログラムでは未使用)				
												0	0	0	0	f_{CLK}
												0	0	0	1	$f_{CLK}/2$
												0	0	1	0	$f_{CLK}/2^2$
												0	0	1	1	$f_{CLK}/2^3$
												0	1	0	0	$f_{CLK}/2^4$
												0	1	0	1	$f_{CLK}/2^5$
												0	1	1	0	$f_{CLK}/2^6$
												0	1	1	1	$f_{CLK}/2^7$
												1	0	0	0	$f_{CLK}/2^8$
												1	0	0	1	$f_{CLK}/2^9$
												1	0	1	0	$f_{CLK}/2^{10}$
												1	0	1	1	$f_{CLK}/2^{11}$
												1	1	1	1	INTTM03
												上記以外		設定禁止		

- 注意1. ビット15-8には, 必ず0を設定してください。
2. PER0レジスタを “ 1 ” に設定後に, 4クロック以上間隔をあけてからSPS1レジスタを設定してください。

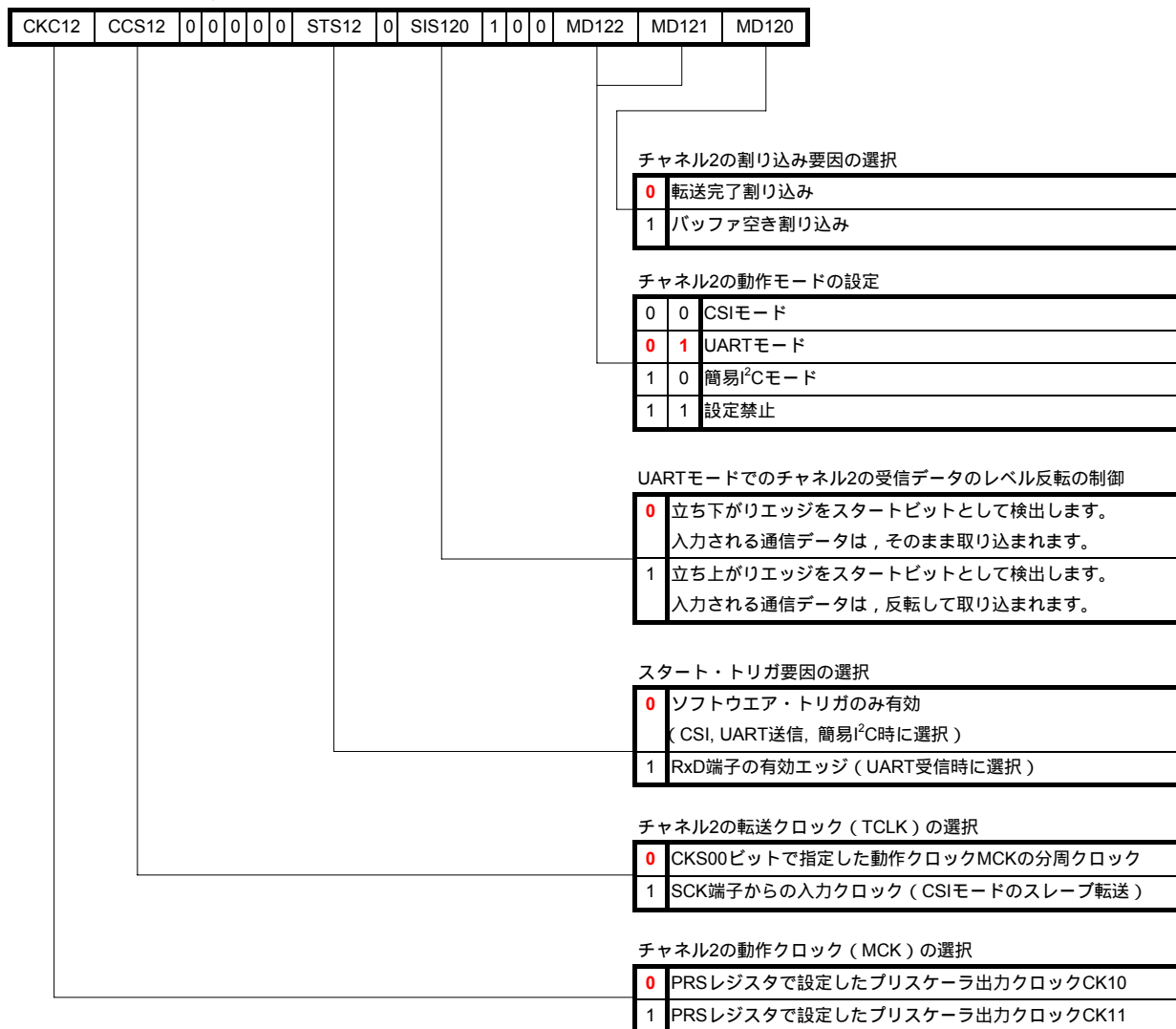
備考 f_{CLK} : CPU / 周辺ハードウェア・クロック周波数

(3) 動作モードなどの設定

シリアル・アレイ・ユニット1 (SAU1) のUART3で使用する動作クロック (MCK) の選択, シリアル・クロック (SCK) 入力の使用可否, スタート・トリガ設定, 動作モード (CSI, UART, I²C) 設定, 割り込み要因の選択をシリアル・モード・レジスタ12 (SMR12) およびシリアル・モード・レジスタ13 (SMR13) にて行います。UARTモード時のみ, 受信データのレベル反転の設定が行えます。シリアル・モード・レジスタ12 (SMR12) がUART3送信, シリアル・モード・レジスタ13 (SMR13) がUART3受信が可能となるように設定を行います。

図4 - 1 - 3 シリアル・モード・レジスタ12 (SMR12) のフォーマット

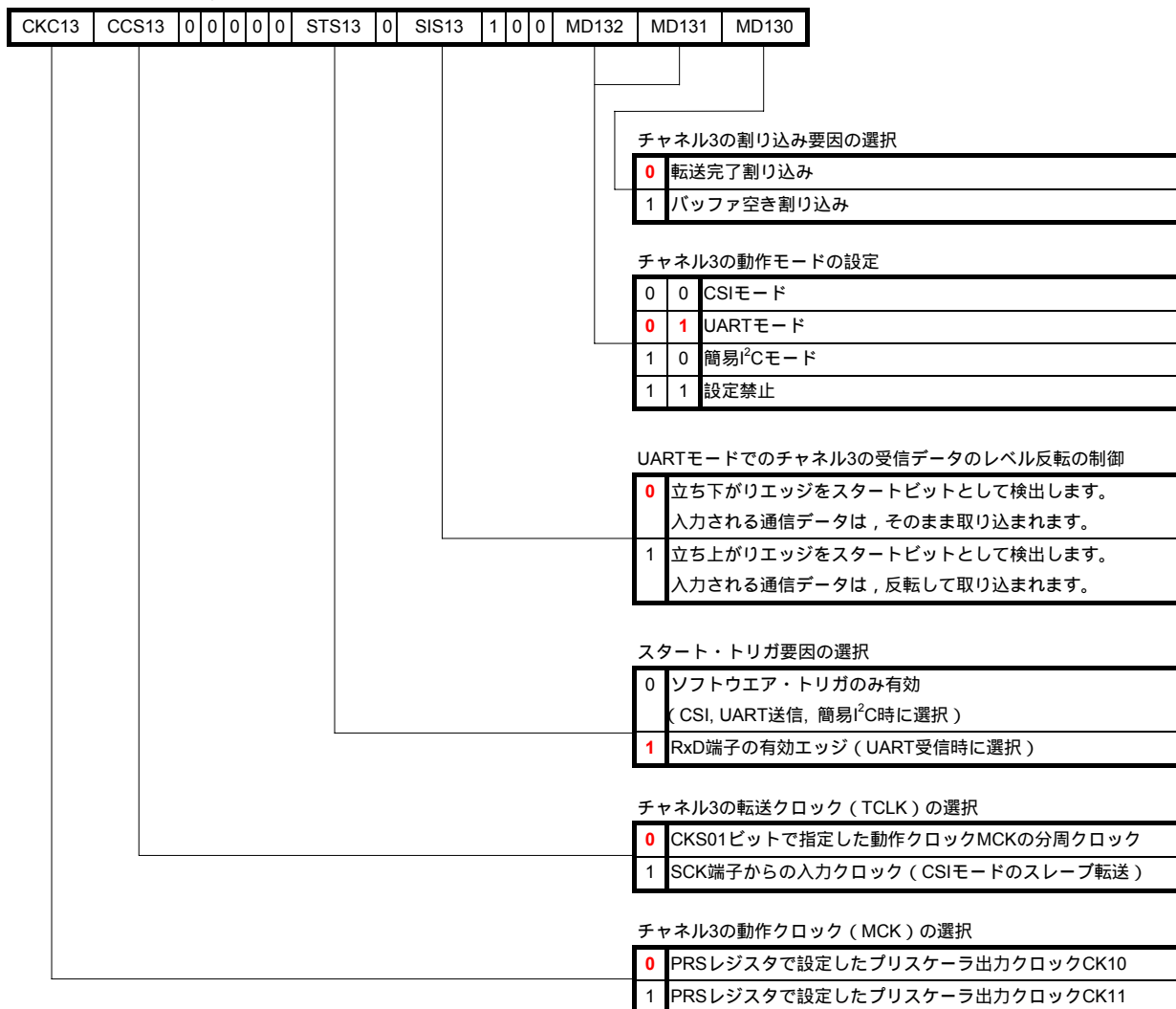
アドレス : F0154H, F0155H



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

図4 - 1 - 4 シリアル・モード・レジスタ13 (SMR13) のフォーマット

アドレス : F0156H , F0157H



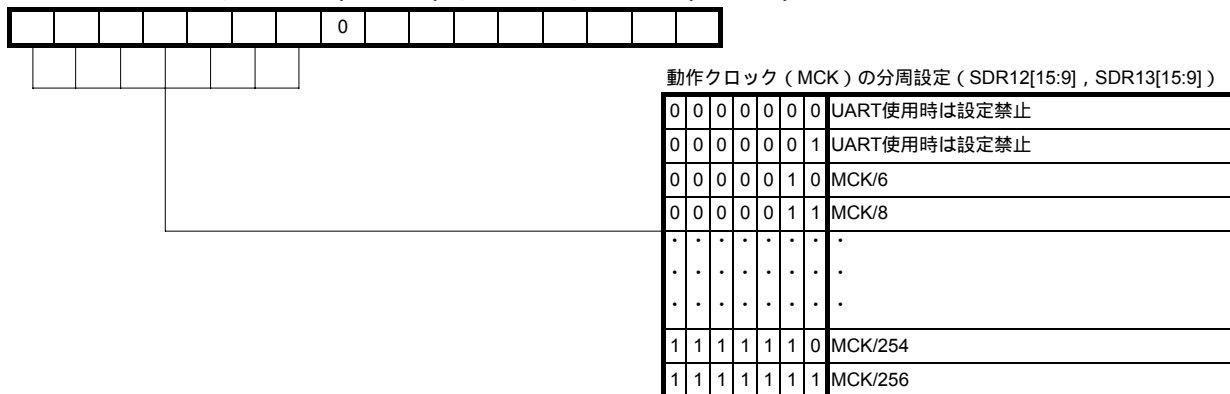
注意 ビット13-9, 7, 4, 3には、必ず0を設定してください。ビット5には、必ず1を設定してください。

(4) 動作クロックの分周設定

シリアル・アレイ・ユニット1 (SAU1) のUART3で使用する動作クロック (MCK) の分周設定は、シリアル・データ・レジスタ12 (SDR12) およびシリアル・データ・レジスタ13 (SDR13) の16ビットのレジスタの上位7ビットで行います。本サンプル・プログラムでは、キャリブレーション終了後にUART3送受信の転送ボー・レート用に計算し、設定を行います。なお、シリアル・データ・レジスタ12 (SDR12) の下位8ビットは送信データ・レジスタ (TxD3) , シリアル・データ・レジスタ13 (SDR13) の下位8ビットは受信データ・レジスタ (RxD3) となります。

図4 - 1 - 5 シリアル・データ・レジスタ12, 13 (SDR12, SDR13) のフォーマット

アドレス : FFF14H, FFF15H (SDR12) , FFF16H, FFF17H (SDR13)



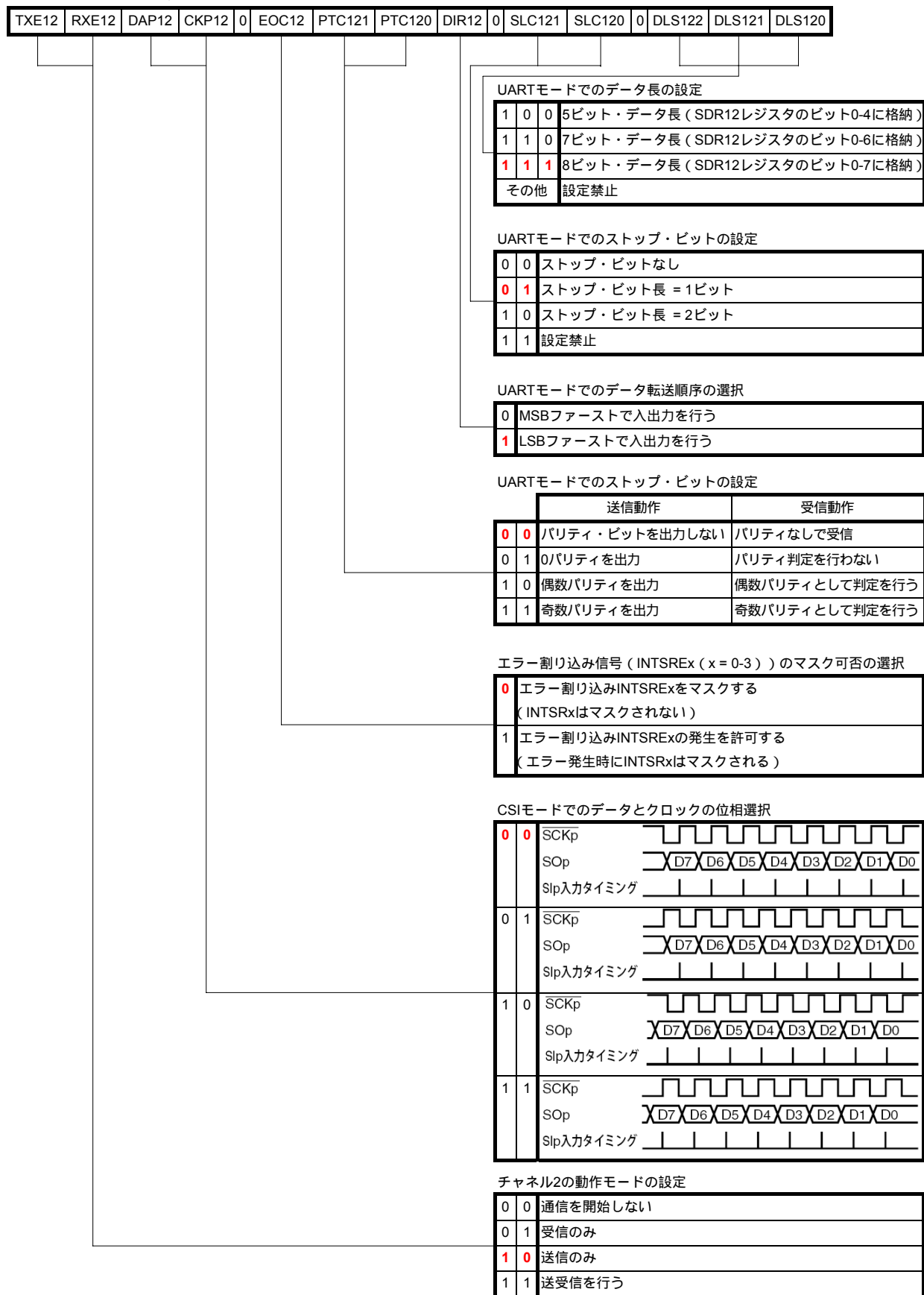
注意 ビット8は、必ず0を設定してください。

(5) 通信動作の設定

通信動作の設定は、シリアル通信動作設定レジスタ12 (SCR12) およびシリアル通信動作設定レジスタ13 (SCR13) にて行います。データ送受信モード, データとクロックの位相, エラー信号のマスク可否, パリティ・ビット, 先頭ビット, ストップ・ビット, データ長, などの設定を行います。シリアル通信動作設定レジスタ12 (SCR12) にてUART3送信, シリアル通信動作設定レジスタ13 (SCR13) にてUART3受信が可能となるように設定を行います。

図4 - 1 - 6 シリアル通信動作設定レジスタ12 (SCR12) のフォーマット

アドレス : F015CH , F015DH

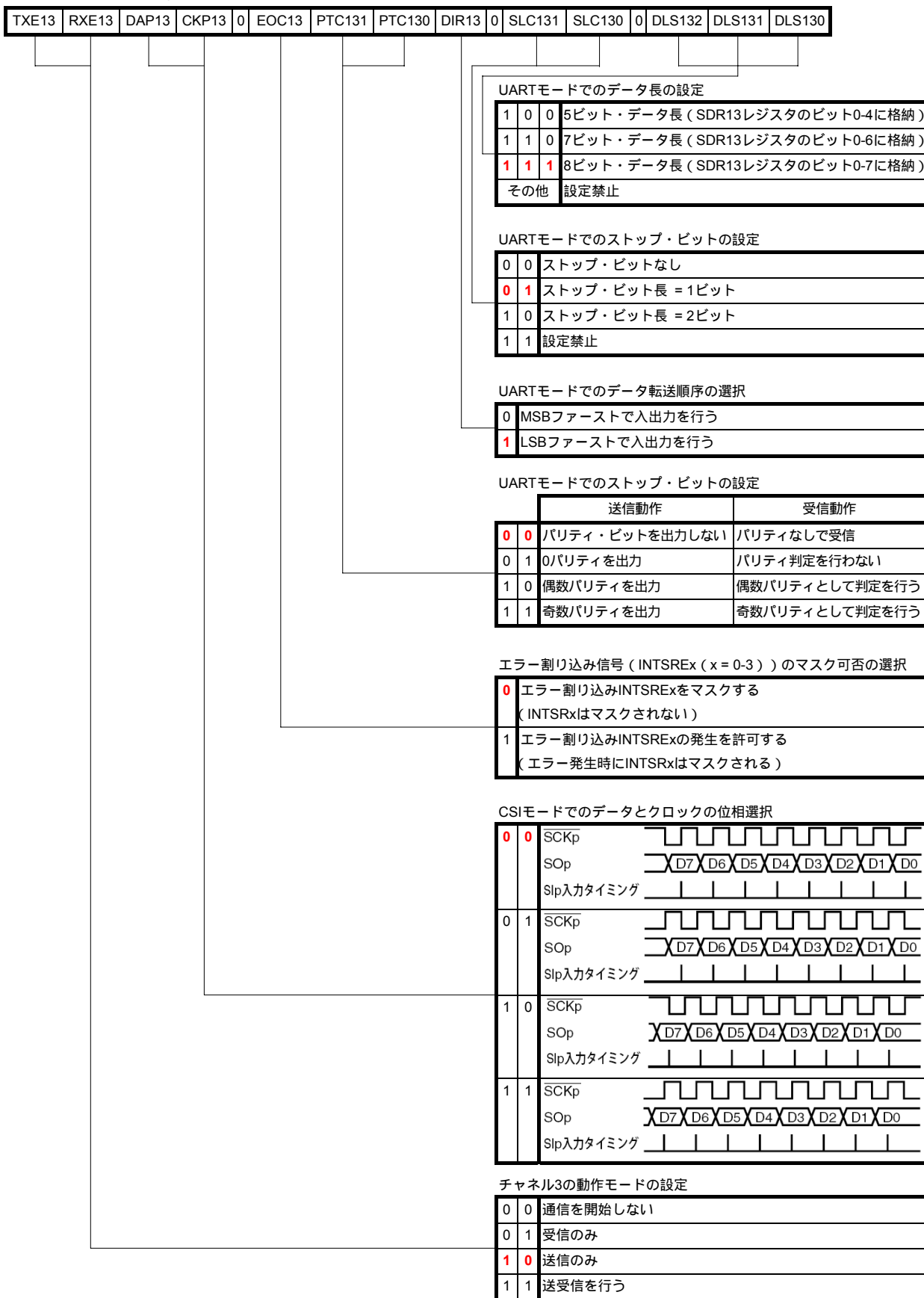


注意 ビット11, 6, 3には、必ず0を設定してください。ビット2には、必ず1を設定してください。

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

図4 - 1 - 7 シリアル通信動作設定レジスタ13 (SCR13) のフォーマット

アドレス : F015EH , F015FH



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

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

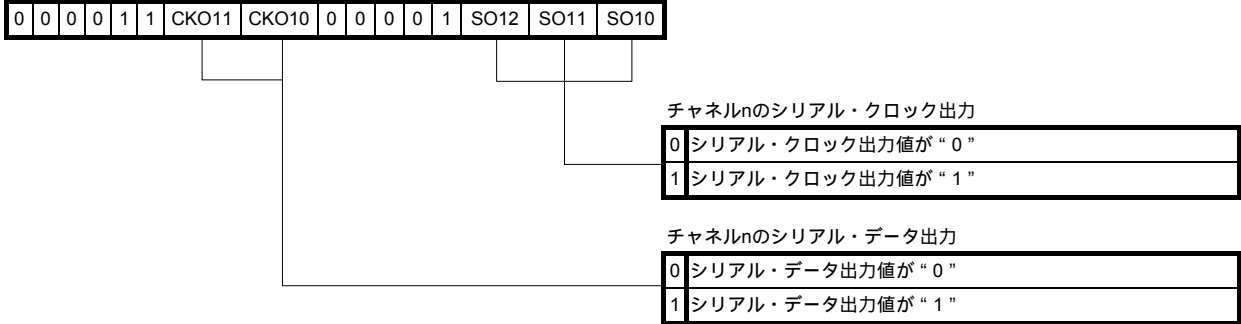
(6) 初期出力レベルの設定

シリアル出力レジスタ1 (SO1) にて, シリアル・アレイ・ユニット1 (SAU1) のUART3送信での初期出力レベルの設定を行います。

本サンプル・プログラムでは, 使用する周辺の初期設定にてシリアル・アレイ・ユニット1 (SAU1) のUART3送信用にSO12ビットを1にします。

図4 - 1 - 8 シリアル出力レジスタ1 (SO1) のフォーマット

アドレス : F0168H , F0169H



注意 SO1のビット11, 10, 3には, 必ず1を設定してください。SO1のビット15-12, 7-4には, 必ず0を設定してください。CKO11 ,SO11は ,K0R/KJ3 ,K0R/KH3マイクロコントローラにのみ存在します。CKO10 , SO10は , K0R/KJ3 , K0R/KH3 , K0R/KG3 , K0R/KF3マイクロコントローラにのみ存在します。存在しないビットは, 必ず1を設定してください。

備考 n : チャネル番号 (n = 0-3)

(7) シリアル通信動作の出力許可 / 停止の分周設定

シリアル・アレイ・ユニット1 (SAU1) のUART3で使用するシリアル通信動作の出力許可 / 停止は, シリアル出力許可レジスタ1 (SOE1) の設定で行います。シリアル出力を許可したチャンネルは, 通信動作によって反映された値がシリアル・データ出力端子から出力されます。

本サンプル・プログラムでは, 使用する周辺の初期設定にてシリアル・アレイ・ユニット1 (SAU1) のUART3送信動作の出力許可時にSOE12ビットを1にします。

図4 - 1 - 9 シリアル出力許可レジスタ1 (SOE1) のフォーマット

アドレス : F016AH , F016BH (SOE1)



注意 SOE1のビット15-4には, 必ず0を設定してください。SOE11は, K0R/KJ3 , K0R/KH3マイクロコントローラにのみ存在します。SOE10は, K0R/KJ3 , K0R/KH3 , K0R/KG3 , K0R/KF3マイクロコントローラにのみ存在します。存在しないビットは, 必ず0を設定してください。

備考 n : チャネル番号 (n = 0-3)

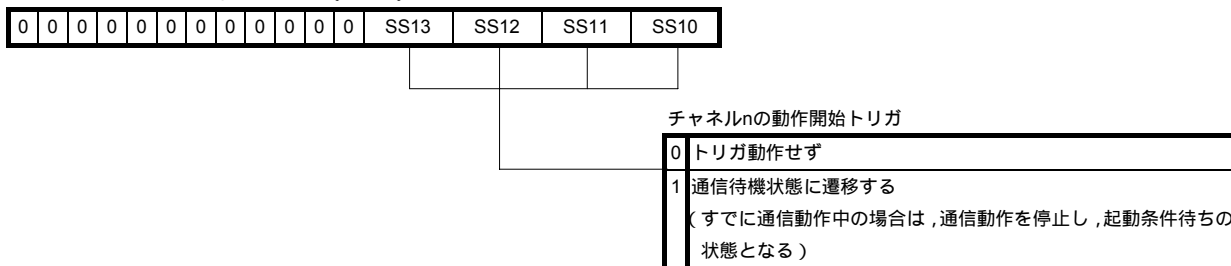
(8) シリアル通信 / カウント開始の許可

シリアル・アレイ・ユニット1 (SAU1) のUART3で使用するシリアルの通信 / カウント開始の許可はチャンネルごとに設定するシリアル・チャンネル開始レジスタ1 (SS1) で行います。シリアル・チャンネル開始レジスタ1 (SS1) は、トリガ・レジスタですので、シリアル送受信の動作が許可状態になるとクリアされます。

本サンプル・プログラムでは、シリアル・アレイ・ユニット1 (SAU1) のUART3送信動作開始時にSS12ビットを1, UART3受信動作開始時にSS13ビットを1にします。

図4 - 1 - 10 シリアル・チャンネル開始レジスタ1 (SS1) のフォーマット

アドレス : F0162H , F0163H (SS1)



注意 SS1のビット15-4には、必ず0を設定してください。

備考 n : チャンネル番号 (n = 0-3)

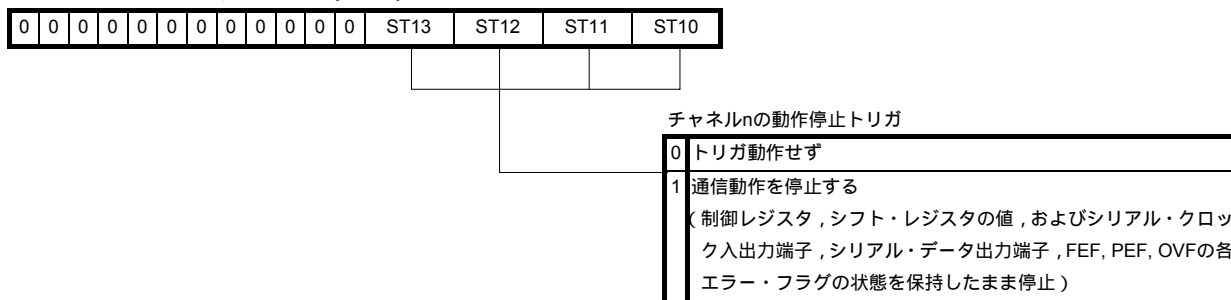
(9) シリアル通信 / カウント停止の許可

通信 / カウント停止の許可はチャンネルごとに設定するシリアル・チャンネル停止レジスタ1 (ST1) で行います。シリアル・チャンネル停止レジスタ1 (ST1) は、トリガ・レジスタですので、シリアル送受信の動作が停止状態になるとクリアされます。

本サンプル・プログラムでは、シリアル・アレイ・ユニット1 (SAU1) のUART3送信動作停止時にST12ビットを1, UART3受信動作停止時にST13ビットを1にします。

図4 - 1 - 11 シリアル・チャンネル停止レジスタ1 (ST1) のフォーマット

アドレス : F0164H , F0165H (ST1)



注意 ST1のビット15-4には、必ず0を設定してください。

備考 n : チャンネル番号 (n = 0-3)

【 本サンプル・プログラム・ソースからの抜粋 】

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

(1) アセンブリ言語

```

;-----
;
; UART3の設定
;-----
SET1    !SAU1EN                ;UART3のあるSAU1使用許可

; クロック供給開始
NOP
NOP
NOP
NOP

MOV     !SPS1L, #00000010B     ;プリスケアラ(動作クロック)の設定
; ||| |++++----- PRS103-100(CK10): fCLK/2^2
; +++----- PRS113-110(CK11): 未使用

;-----
; UART3送信設定(SAU1のチャンネル2使用)
;-----
MOV     RTXBUFF, #0            ;送信データ・カウンタ初期化

MOVW   AX, #000000000100010B;動作モードなどの設定
; ||| |++++----- MD120: 転送完了割り込み
; ||| |++++----- MD122-121: UARTモード
; | | | |++++----- <100>
; | | | |++++----- SIS120: 立ち下がりエッジをスタートビットとして検出
; | | | |++++----- <0>
; | | | |++++----- STS12: UART送信
; | | | |++++----- <00000>
; | | | |++++----- CCS12: CKS00ビットで指定した動作クロックMCK
; | | | |++++----- CKS12: PRSで選択した動作クロックCK10(MCK)

MOVW   !SMR12, AX

MOVW   AX, #1000000010010111B;通信フォーマットの設定
; ||| |++++----- DLS122-120: 8ビット・データ長
; | | | |++++----- <0>
; | | | |++++----- SLC121-120: ストップ・ビット長 = 1ビット
; | | | |++++----- <00>
; | | | |++++----- DIR12: LSBファーストで出力
; | | | |++++----- PTC121-120: パリティ・ビットを出力しない
; | | | |++++----- EOC12: UART送信時はエラー割り込み未使用
; | | | |++++----- <0>
; | | | |++++----- DAP12/CKP12: UARTモードでは未使用
; | | | |++++----- TXE12/RXE12: 送信のみを行う

MOVW   !SCR12, AX

; SDR12 転送ボー・レートの設定はINTTM07割り込み処理にて設定

MOV     SOL1L, #00000000B     ;出力データのレベルの設定(反転なし)

MOVW   AX, #0000111000001110B;初期出力レベルの設定
; ||| |++++----- SO10: チャンネル0のシリアル・クロック出力
; | | | |++++----- <1>
; | | | |++++----- SO12: チャンネル2のシリアル・クロック出力
; | | | |++++----- <00001>
; | | | |++++----- CKO10: チャンネル0のシリアル・データ出力
; | | | |++++----- <0000111>

MOVW   !SO1, AX
SET1   !SOE1L.2              ;SOE12:シリアル出力許可

;済
CLR1   PM1.3                 ;ポート1の設定にて設定済み
SET1   P1.3                  ;P13=TxD3

;-----
; UART3受信設定(SAU1のチャンネル3使用)
;-----
MOV     RRXBUFF, #0           ;受信データ・カウンタ初期化

SET1   PM1.4                 ;P14=RxD3

MOVW   AX, #0000000100100010B;動作モードなどの設定
; ||| |++++----- MD130: 転送完了割り込み
; ||| |++++----- MD132-131: UARTモード
; | | | |++++----- <100>
; | | | |++++----- SIS130: 立ち下がりエッジをスタートビットとして検出
; | | | |++++----- <0>
; | | | |++++----- STS13: RxD端子の有効エッジ(UART受信設定)
; | | | |++++----- <00000>
; | | | |++++----- CCS13: CKS13ビットで指定した動作クロックMCK
; | | | |++++----- CKS13: PRSで選択した動作クロックCK10(MCK)

MOVW   !SMR13, AX
    
```

クロック供給開始

4クロックウェイト

ボー・レートは計算後設定

初期出力をHiに設定

ポート設定

チャンネル3をUART受信に設定

(2) C言語

```

/*-----
UART3の設定
-----*/
SAU1EN = 1; /* UART3のあるSAU1使用許可 */

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

SPS1L = 0b00000010; /* プリスケラ(動作クロック)の設定 */
/* ||| ||++++----- PRS103-100(CK10): fCLK/2^2 */
/*++++----- PRS113-110(CK11): 未使用 */

/*-----
UART3送信設定(SAU1のチャンネル2使用)
-----*/
ucTxBufferCounter = 0; /* 送信データ・カウンタ初期化 */

SMR12 = 0b000000000100010; /* 動作モードなどの設定 */
/* ||| |||++++----- MD120: 転送完了割り込み */
/* ||| |||++++----- MD122-121: UARTモード */
/* ||| |||++++----- <100> */
/* ||| |||++++----- SIS120: 立ち上がりエッジをスタートビットとして検出 */
/* ||| |||++++----- <0> */
/* ||| |||++++----- STS12: UART送信 */
/* ||| |||++++----- <00000> */
/* ||| |||++++----- CCS12: CKS00ビットで指定した動作クロックMCK */
/* ||| |||++++----- CKS12: PRSで選択した動作クロックCK10(MCK) */

SCR12 = 0b1000000010010111; /* 通信フォーマットの設定 */
/* ||| |||++++----- DLS122-120: 8ビット・データ長 */
/* ||| |||++++----- <0> */
/* ||| |||++++----- SLC121-120: ストップ・ビット長 = 1ビット */
/* ||| |||++++----- <0> */
/* ||| |||++++----- DIR12: LSBファーストで出力 */
/* ||| |||++++----- PTC121-120: パリティ・ビットを出力しない */
/* ||| |||++++----- BOC12: UART送信時はエラー割り込み未使用 */
/* ||| |||++++----- <0> */
/* ||| |||++++----- DAP12/CKP12: UARTモードでは未使用 */
/* ||| |||++++----- TXE12/RXE12: 送信のみを行う */

/* SDR12 転送ボー・レートの設定はINTTM07割り込み処理にて設定 */

SOL1L = 0b00000000; /* 出力データのレベルの設定(反転なし) */

SOI = 0b0000111000001110; /* 初期出力レベルの設定 */
/* ||| |||++++----- SO10: チャンネル0のシリアル・クロック出力 */
/* ||| |||++++----- <1> */
/* ||| |||++++----- SO12: チャンネル2のシリアル・クロック出力 */
/* ||| |||++++----- <00001> */
/* ||| |||++++----- CKO10: チャンネル0のシリアル・データ出力 */
/* ||| |||++++----- <0000111> */

SOE1L.2 = 1; /* SOE12: シリアル出力許可 */

/*済PM1.3 = 0; /* ポート1の設定にて設定済み */
P1.3 = 1; /* P13=TxD3 */

/*-----
UART3受信設定(SAU1のチャンネル3使用)
-----*/
ucRxBufferCounter = 0; /* 受信データ・カウンタ初期化 */

PM1.4 = 1; /* P14=RxD3 */

SMR13 = 0b0000000100100010; /* 動作モードなどの設定 */
/* ||| |||++++----- MD130: 転送完了割り込み */
/* ||| |||++++----- MD132-131: UARTモード */
/* ||| |||++++----- <100> */
/* ||| |||++++----- SIS130: 立ち上がりエッジをスタートビットとして検出 */
/* ||| |||++++----- <0> */
/* ||| |||++++----- STS13: RxD端子の有効エッジ(UART受信設定) */
/* ||| |||++++----- <00000> */
/* ||| |||++++----- CCS13: CKS13ビットで指定した動作クロックMCK */
/* ||| |||++++----- CKS13: PRSで選択した動作クロックCK10(MCK) */

SCR13 = 0b0100000010010111; /* 通信フォーマットの設定 */
/* ||| |||++++----- DLS132-130: 8ビット・データ長 */
/* ||| |||++++----- <0> */
/* ||| |||++++----- SLC131-130: ストップ・ビット長 = 1ビット */
/* ||| |||++++----- <0> */
/* ||| |||++++----- DIR13: LSBファーストで入力を行う */
/* ||| |||++++----- PTC131-130: パリティなしで受信 */
/* ||| |||++++----- BOC13: UART受信のエラー割り込み許可 */
/* ||| |||++++----- <0> */
/* ||| |||++++----- DAP13/CKP13: UARTモードでは未使用 */
/* ||| |||++++----- TXE13/RXE13: 受信のみを行う */

```

ボー・レート
は計算後設定

初期出力を
Hiに設定

ポート設定

チャンネル3を
UART受信に設定

```

/* SDR13 転送ボー・レートの設定はINTTM07割り込み処理にて設定 */

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

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

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

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

/* 2変換目(正規データ) */
ADIF = 0;                             /* 割り込み要求クリア */
ADCS = 1;                             /* 動作許可 */
while(!ADIF){                         /* 変換待ち */
NOP();
}

ADCS = 0;                             /* 変換動作終了 */
ADCE = 0;

/* 送信用データの作成 */
if(ucRxBuffer[0]==8){
/* 8ビット分解能 */
ucTxBuffer[0] = 0;
ucTxBuffer[1] = ADCRH;
}
else{
/* 10ビット分解能 */
ucTxBuffer[0] = (unsigned char)(ADCR >>14);
ucTxBuffer[1] = (unsigned char)(ADCR >>6);
/* (10ビットA/D変換結果を下位ビット詰め) */
}
}
else{
/* 受信したデータが不正な場合 */
ucTxBuffer[0] = 0x80;                /* エラー情報設定 */
ucTxBuffer[1] = 0x00;
}
/* 送信動作開始 */
STIF3 = 0;                            /* 割り込み要求クリア */
STMK3 = 0;                            /* 割り込み許可 */
SS1L.2 = 1;                           /* SS12:UART3送信動作開始 */

TxD3 = ucTxBuffer[0];                 /* データを送信データ・レジスタに設定 */
}
}
else if(ucTxBufferCounter >= sizeof(ucTxBuffer)){
/* UART3送信完了 */

ucTxBufferCounter = 0;                /* 送信データ・カウンタ初期化 */

/* 受信動作開始 */
SRIF3 = 0;                            /* 割り込み要求クリア */
SRMK3 = 0;                            /* 割り込み許可 */
SS1L.3 = 1;                           /* SS13:UART3受信動作開始 */
}
}
}
}

```

マスク解除

TxD3送信

受信動作許可

4.2 使用する周辺の初期設定

アセンブリ言語の使用する周辺の初期設定では、次の動作を行います。

- 割り込みを禁止します。
- レジスタバンクの設定を行います。
- スタック・ポインタの設定を行います。
- メイン・システム・クロックの設定 / サブシステム・クロックの発振開始等のクロック設定を行います。
- ポートの設定を行います。
- A/Dコンバータの設定を行います。
- シリアル・アレイ・ユニット1 (SAU1) のチャンネル2のUART3送信の設定を行います。
- シリアル・アレイ・ユニット1 (SAU1) のチャンネル3のUART3受信の設定を行います。
- 割り込みを許可します。
- メイン処理へ続きます。

```

;*****
;
;      使用する周辺の初期設定
;
;*****
XMAIN   CSEG      UNIT
RESET_START:
-----
;      DI                      ; 割り込み禁止
;-----
;      レジスタバンク設定
;-----
;      SEL      RB0
;-----
;      スタック・ポインタの設定
;-----
MOVW    SP,      #LOWW STACKTOP ; スタック・ポインタを設定
;-----
;      クロック周波数の設定
;-----
;      高速内蔵発振クロックで動作が行えるように設定します。
;-----
MOV     CMC,     #00000000B      ; クロック動作モード
;      ; | | | | | | | | +----- AMPH: 2MHz fMX 10MHz
;      ; | | | | | | | | +----- <000>
;      ; | | | | | | | | +----- OSCSELS: P123/P124端子を入力ポート・モード
;      ; | | | | | | | | +----- <0>
;      ; ++----- EXCLK/OSCSEL: 入力ポート・モード

MOV     CSC,     #11000000B      ; 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発振回路停止

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

MOV     CKC,     #00001000B      ; クロック選択
;      ; | | | | | | | | +----- MDIV2-0: CPU/周辺ハードウェア・クロック (fCLK)=fIH
;      ; | | | | | | | | +----- <1>
;      ; | | | | | | | | +----- MCM0: 高速内蔵発振クロック (fIH)
;      ; | | | | | | | | +----- <R>
;      ; | | | | | | | | +----- CSS: メイン・システム・クロック (fMAIN)=fCLK
;      ; +----- <R>
;-----
;      ポート0の設定
;-----
MOV     P0,      #00000000B      ; P0-P07の出力ラッチLow
MOV     PM0,     #00000000B      ; P0-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	; [後でP13 (TxD3) の設定を行う]
			; P10-P17を出力ポートに設定
			; [後でP14 (RxD3) の設定を行う]
;-----			
; ポート2の設定			
;-----			
MOV	P2,	#00000000B	; P20-P27の出力ラッチ _{Low}
MOV	PM2,	#11111111B	; P20-P27 (ANI0-ANI7) を入力ポートに設定
;-----			
; ポート3の設定			
;-----			
MOV	P3,	#00000000B	; P30-P37の出力ラッチ _{Low}
MOV	PM3,	#00000000B	; P30-P37を出力ポートに設定
;-----			
; ポート4の設定			
;-----			
MOV	P4,	#00000000B	; P40-P47の出力ラッチ _{Low}
MOV	PM4,	#00000000B	; P40-P47を出力ポートに設定
;-----			
; ポート5の設定			
;-----			
MOV	P5,	#00000000B	; P50-P57の出力ラッチ _{Low}
MOV	PM5,	#00000000B	; P50-P57を出力ポートに設定
;-----			
; ポート6の設定			
;-----			
MOV	P6,	#00000000B	; P60-P67の出力ラッチ _{Low}
MOV	PM6,	#00000000B	; P60-P67を出力ポートに設定
;-----			
; ポート7の設定			
;-----			
MOV	P7,	#00000000B	; P70-P77の出力ラッチ _{Low}
MOV	PM7,	#00000000B	; P70-P77を出力ポートに設定
;-----			
; ポート8の設定			
;-----			
MOV	P8,	#00000000B	; P80-P87の出力ラッチ _{Low}
MOV	PM8,	#00000000B	; P80-P87を出力ポートに設定
;-----			
; ポート9の設定			
;-----			
MOV	P9,	#00000000B	; P90-P97の出力ラッチ _{Low}
MOV	PM9,	#00000000B	; P90-P97を出力ポートに設定
;-----			
; ポート10の設定			
;-----			
MOV	P10,	#00000000B	; P100-P107の出力ラッチ _{Low}
MOV	PM10,	#00000000B	; P100-P107を出力ポートに設定
;-----			
; ポート11の設定			
;-----			
MOV	P11,	#00000000B	; P110-P117の出力ラッチ _{Low}
MOV	PM11,	#00000000B	; P110-P117を出力ポートに設定
;-----			
; ポート12の設定			
;-----			
MOV	P12,	#00000000B	; P120, P125-P127の出力ラッチ _{Low}
MOV	PM12,	#00011110B	; P120, P125-P127を出力ポートに設定
;-----			
; ポート13の設定			
;-----			
MOV	P13,	#00000000B	; P130-P137の出力ラッチ _{Low}
MOV	PM13,	#00000000B	; P130-P137を出力ポートに設定
;-----			
; ポート14の設定			
;-----			
MOV	P14,	#00000000B	; P140-P147の出力ラッチ _{Low}
MOV	PM14,	#00000000B	; P140-P147を出力ポートに設定
;-----			
; ポート15の設定			
;-----			
MOV	P15,	#00000000B	; P150-P157の出力ラッチ _{Low}
MOV	PM15,	#00000000B	; P150-P157を出力ポートに設定
;-----			


```

MOVW    AX,          #0000000100100010B;動作モードなどの設定
;      | | | | | | | | | | | | | | | +- MD130: 転送完了割り込み
;      | | | | | | | | | | | | | | | +- MD132-131: UARTモード
;      | | | | | | | | | | | | | | | +++> <100>
;      | | | | | | | | | | | | | | | +----- SIS130: 立ち下がりエッジをスタートビットとして検出
;      | | | | | | | | | | | | | | | +----- <0>
;      | | | | | | | | | | | | | | | +----- STS13: RxD端子の有効エッジ (UART受信設定)
;      | | | | | | | | | | | | | | | +----- <00000>
;      | | | | | | | | | | | | | | | +----- CCS13: CKS13ビットで指定した動作クロックMCK
;      | | | | | | | | | | | | | | | +----- CKS13: PRSで選択した動作クロックCK10 (MCK)
MOVW    !SMR13, AX

MOVW    AX,          #0100000010010111B        ;通信フォーマットの設定
;      | | | | | | | | | | | | | | | ++++ DLS132-130: 8ビット・データ長
;      | | | | | | | | | | | | | | | +----- <0>
;      | | | | | | | | | | | | | | | +++> SLC131-130: ストップ・ビット長 = 1ビット
;      | | | | | | | | | | | | | | | +----- <0>
;      | | | | | | | | | | | | | | | +----- DIR13: LSBファーストで入力を行う
;      | | | | | | | | | | | | | | | ++----- PTC131-130: パリティなしで受信
;      | | | | | | | | | | | | | | | +----- EOC13: UART受信のエラー割り込み許可
;      | | | | | | | | | | | | | | | +----- <0>
;      | | | | | | | | | | | | | | | ++----- DAP13/CKP13: UARTモードでは未使用
;      | | | | | | | | | | | | | | | +++> TXE13/RXE13: 受信のみを行う
MOVW    !SCR13, AX

; SDR13 転送ポー・レートの設定はINTTM07割り込み処理にて設定

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
  -----
  高速内蔵発振クロックで動作が行えるように設定します。
  -----*/
  CMC = 0b00000000; /* クロック動作モード */
  /* | | | | | | | | +----- AMPH: 2MHz fMX 10MHz */
  /* | | | | | | | | +----- <000> */
  /* | | | | | | | | +----- OSCSELS: P123/P124端子を入力ポート・モード */
  /* | | | | | | | | +----- <0> */
  /* | | | | | | | | +----- EXCLK/OSCSEL: 入力ポート・モード */
  CSC = 0b11000000; /* クロック動作ステータス制御 */
  /* | | | | | | | | +----- HIOSTOP: 高速内蔵発振回路動作 */
  /* | | | | | | | | +----- <00000> */
  /* | | | | | | | | +----- XTSTOP: XT1発振回路動作 */
  /* | | | | | | | | +----- MSTOP: X1発振回路動作 */
  OSMC = 0b00000000; /* 動作スピード・モード */
  /* | | | | | | | | +----- FSEL: 10MHz以下の周波数で動作 */
  /* | | | | | | | | +----- <00000> */
  CKC = 0b00001000; /* クロック選択 */
  /* | | | | | | | | +----- MDIV2-0: CPU周辺ハードウェア・クロック (fCLK)=fIH */
  /* | | | | | | | | +----- <1> */
  /* | | | | | | | | +----- MCM0: 高速内蔵発振クロック (fIH) */
  /* | | | | | | | | +----- <R> */
  /* | | | | | | | | +----- CSS: メイン・システム・クロック (fMAIN)=fCLK */
  /* | | | | | | | | +----- <R> */
  /* -----
  ポート0の設定
  -----*/
  P0 = 0b00000000; /* P00-P07の出力ラッチLow */
  PM0 = 0b00000000; /* P00-P07を出力ポートに設定 */
  /* -----
  ポート1の設定
  -----*/
  P1 = 0b00000000; /* P10-P17の出力ラッチLow [後でP13 (TxD0)の設定を行う] */
  PM1 = 0b00000000; /* P10-P17を出力ポートに設定 [後でP14 (RxD0)の設定を行う] */
  /* -----
  ポート2の設定
  -----*/
  P2 = 0b00000000; /* P20-P27の出力ラッチLow */
  PM2 = 0b11111111; /* P20-P27 (ANI0-ANI7)を入力ポートに設定 */
  /* -----
  ポート3の設定
  -----*/
  P3 = 0b00000000; /* P30-P37の出力ラッチLow */
  PM3 = 0b00000000; /* P30-P37を出力ポートに設定 */
  /* -----
  ポート4の設定
  -----*/
  P4 = 0b00000000; /* P40-P47の出力ラッチLow */
  PM4 = 0b00000000; /* P40-P47を出力ポートに設定 */
  /* -----
  ポート5の設定
  -----*/
  P5 = 0b00000000; /* P50-P57の出力ラッチLow */
  PM5 = 0b00000000; /* P50-P57を出力ポートに設定 */
  /* -----
  ポート6の設定
  -----*/
  P6 = 0b00000000; /* P60-P67の出力ラッチLow */
  PM6 = 0b00000000; /* P60-P67を出力ポートに設定 */

```

```

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

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

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

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

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

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

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

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

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

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

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

ADPC = 0b00000000; /* A/Dポート・コンフィギュレーション */
/* | | | | | +----- ADPC4-0: すべてアナログ入力 */
/* +++----- <000> */

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

/*-----
   UART3の設定
   -----*/
SAU1EN = 1; /* UART3のあるSAU1使用許可 */

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

SPS1L = 0b00000010; /* プリスケアラ(動作クロック)の設定 */
/* | | | | | +----- PRS103-100(CK10): fCLK/2^2 */
/* +++----- PRS113-110(CK11): 未使用 */

```


4.3 メイン処理

アセンブリ言語のメイン処理では、次の動作を行います。

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

シリアル・チャンネル許可ステータス・レジスタ1（SE1）にてUART3の送受信動作のチェックを行います。送受信動作中の場合は、何も行いません（へ）。

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

シリアル・ステータス・レジスタ13（SSR13）によりエラーのチェックを行います。エラーがある場合は、再度、受信動作を開始します（へ）。

UART3で受信したデータが受信データのフォーマットに対応しているかのチェックを行います。受信データのフォーマットに対応していない場合は、送信バッファにエラー情報を設定します。

のUART3で受信したデータが通信フォーマットに対応している場合、A/DコンバータでA/D値を取得し、送信データ用に加工し、送信バッファへ設定します。A/D値を取得するチャンネルとA/Dの分解能は、UART3の受信データにより決定します。

送信バッファのデータを送信データ・レジスタに設定し、UART3送信動作を開始します。（へ）

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

UART3受信動作を開始します。（へ）

へ分岐します。

```

; *****
;
;      メイン処理
;
; *****
MAIN_LOOP:
; -----
;      動作状態による分岐
; -----
CMP     RCALIBST, #CCALIB_END      ; キャリブレーション完了?
BZ      $LMAIN100                 ; Yes,
BR      LMAINRET                   ; No,
LMAIN100:
MOV     A,      SE1L               ; 動作停止状態/停止状態を確認
BT      A.2,    $LMAINRET          ; SE12: UART3送信動作停止中? No,
BT      A.3,    $LMAINRET          ; SE13: UART3受信動作停止中? No,
; *****
;      UART3受信完了
; *****
CMP     RRXBUFFP, #RRXBUFFE-RRXBUF ; 最終データの受信完了?
BC      $LMAINTX                   ; No,
MOV     RRXBUFFP, #0               ; 受信データ・カウンタ初期化
;
MOV     A,      SSR13L              ; UART3のステータスを確認
AND     A,      #00000111B         ; (フレーミング, パリティ, オーバラン) エラーあり?
BNZ     $LMAIN800                  ; Yes,
; -----
;      送信データの作成
; -----
;
; サンプル・プログラムで使用する送受信データは、下図に示す2バイトのフォー
; マットとしています。
;
; 【受信データ】
;   byte |           +0           |           +1           |
;   -----+-----+-----+-----+
;   | A/D値分解能の指定 | A/Dチャンネルの指定 |
;   | 08H: 8ビット     | 0: ch0 ...         |
;   | 0AH: 10ビット    | ... 7: ch7         |
;   +-----+-----+-----+-----+
;
;
; *****

```

```

; 【送信データ】
;
; byte |          +0          |          +1          |
; bit  | 7|6|5|4|3|2|1|0 | 7|6|5|4|3|2|1|0 |
; -----+-----+-----+-----+-----+-----+-----+-----+
;                                     | 8ビットA/D値 |
;                                     | (8ビット指定時) |
; -----+-----+-----+-----+-----+-----+-----+-----+
;                                     | 10ビットA/D値 |
;                                     | (10ビット指定時) |
; -----+-----+-----+-----+-----+-----+-----+-----+
;                                     |
;                                     | +-エラー情報：受信したデータにエラーあり(1)/エラーなし(0)
;                                     |
; -----+-----+-----+-----+-----+-----+-----+-----+
;
; =====
; 受信データのチェック
;
; -----
CMP    RRXBUF, #10          ; A/D値分解能の指定は10ビット?
BZ     $LMAINRX300         ; Yes,
CMP    RRXBUF, #8          ; A/D値分解能の指定は8ビット?
BNZ    $LMAINRX400        ; No, (上記以外はエラー情報設定)
LMAINRX300:
CMP    RRXBUF+1, #7+1     ; チャンネル指定(0-7)は正常値?
BC     $LMAINRX500        ; Yes,
LMAINRX400:
; エラー
MOVW   AX, #8000H         ; エラー情報設定
BR     LMAINRX700
LMAINRX500:
; -----
; A/D値の取得
;
; -----
MOV    A, RRXBUF+1        ; A/D変換するチャンネルの設定
MOV    ADS, A
SET1   ADCE              ; コンパレータの動作許可(変換待機)
; 1変換目(不正データ)
CLR1   ADIF              ; 割り込み要求クリア
SET1   ADCS              ; A/D変換動作許可
LMAINRX530:
NOP
BF     ADIF, $LMAINRX530 ; 1us以上ウエイトしないでADCSに1を設定した場合は、
; 最初の交換データを無視します。
; (詳細はユーザーズ・マニュアルを参照してください。)
; 2変換目(正規データ)
CLR1   ADIF              ; 割り込み要求クリア
SET1   ADCS              ; A/D変換動作許可
LMAINRX550:
NOP
BF     ADIF, $LMAINRX550 ;
; -----
CLR1   ADCS              ; A/D変換動作停止
CLR1   ADCE              ; コンパレータの動作停止
MOVW   AX, ADCR          ; 10ビットA/D変換結果取得
SHRW   AX, 6             ; データのシフト
; (10ビットA/D変換結果を下位ビット詰め)
; A/D値分解能の指定は10ビット?
CMP    RRXBUF, #10       ; Yes,
BZ     $LMAINRX700
SHRW   AX, 2             ; データシフト
; (10ビットA/D変換結果をさらにシフトし、
; 8ビットA/D変換結果として扱う)
; -----
; 送信データ設定
LMAINRX700:
MOV    RTXBUF, A         ; 送信データを送信バッファに設定
XCH    A, X
MOV    RTXBUF+1, A
; -----
; UART3送信動作開始
CLR1   STIF3             ; 割り込み要求クリア
CLR1   STMK3             ; 割り込み許可
SET1   !SS1L.2          ; SS12: UART3送信動作開始
MOV    A, RTXBUF         ; 設定した送信データの第一バイト
MOV    TxD3, A           ; データを送信データ・レジスタに設定
BR     LMAINRET
LMAINTX:
; *****
; UART3送信完了
; *****
CMP    RTXBUFF, #RTXBUFFE-RTXBUF ; 最終データの送信完了?
BC     $LMAINRET         ; No,
MOV    RTXBUFF, #0       ; 送信データ・カウンタ初期化
LMAIN800:

```

安定したコンパレータのデータを取得するため、最初の交換データを無視します。

2度目の交換データを正規のデータとして使用します。

分解能が8ビットの場合はADCRHレジスタの値と同様です。


```

;UART3受信動作開始
CLR1    SRIF3          ;割り込み要求クリア
CLR1    SRMK3          ;割り込み許可
SET1    !SS1L.3       ;SS13:UART3受信動作開始
LMAINRET:
BR      MAIN_LOOP     ; MAIN_LOOPへ
    
```

C言語のメイン処理も、アセンブリ言語と同様な動作を行います。

```

/*****
メイン処理
*****/
void main(void)
{
    while (1)
    {
        if((ucCalibrationStatus==CALIBRATION_STATUS_END)&&!(SE1L & 0b00001100)){
            /* キャリブレーションが完了し、UART3送受信動作を行っていない状態 */
            if(ucRxBufferCounter >= sizeof(ucRxBuffer)){
                /* UART3受信完了 */

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

                if(SSR13L & 0b00000111){      /* UART3のステータスを確認 */
                    /* (フレーミング,パリティ,オーバーラン)エラーあり */
                    /*-- 受信動作開始 --*/
                    SRIF3 = 0;                /* 割り込み要求クリア */
                    SRMK3 = 0;                /* 割り込み許可 */
                    SS1L.3 = 1;               /* SS13:UART3受信動作開始 */
                }
                else{
                    /* エラーなし */
                    /*=====
                    送信データの作成
                    =====*/

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

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

                    【送信データ】

                    byte |          +0          |          +1          |
                    bit | 7|6|5|4|3|2|1|0 | 7|6|5|4|3|2|1|0 |
                    +-----+-----+
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    +-----+-----+
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    |          |          |          |          |
                    +-----+-----+
                    +-エラー情報：受信したデータにエラーあり(1)/エラーなし(0)

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

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

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

                        /* 2変換目 (正規データ) */
                        ADIF = 0;              /* 割り込み要求クリア */
                        ADCS = 1;              /* 動作許可 */
                        while(!ADIF){         /* 変換待ち */
                            NOP();
                        }
                    }
                }
            }
        }
    }
}

```

安定したコンバータのデータを取得するため、最初の変換データを無視します。

2度目の変換データを正規のデータとして使用します。

```
        NOP();
    }

    ADCS = 0;          /* 変換動作終了 */
    ADCE = 0;

    /* 送信データの作成 */
    if(ucRxBuffer[0]==8){
        /* 8ビット分解能 */
        ucTxBuffer[0] = 0;
        ucTxBuffer[1] = ADCRH;
    }
    else{
        /* 10ビット分解能 */
        ucTxBuffer[0] = (unsigned char)(ADCR >>14);
        ucTxBuffer[1] = (unsigned char)(ADCR >>6);
        /* (10ビットA/D変換結果を下位ビット詰め) */
    }
}
else{
    /* 受信したデータが不正な場合 */
    ucTxBuffer[0] =0x80;    /* エラー情報設定 */
    ucTxBuffer[1] =0x00;
}
/* 送信動作開始 */
STIF3 = 0;                /* 割り込み要求クリア */
STMK3 = 0;                /* 割り込み許可 */
SS1L.2 = 1;              /* SS12:UART3送信動作開始 */

TxD3 = ucTxBuffer[0];    /* データを送信データ・レジスタに設定 */
}
}
else if(ucTxBufferCounter >= sizeof(ucTxBuffer)){
    /* UART3送信完了 */

    ucTxBufferCounter = 0;    /* 送信データ・カウンタ初期化 */

    /* 受信動作開始 */
    SRIF3 = 0;                /* 割り込み要求クリア */
    SRMK3 = 0;                /* 割り込み許可 */
    SS1L.3 = 1;              /* SS13:UART3受信動作開始 */
}
}
}
```

4.4 キャリブレーション開始処理

アセンブリ言語のキャリブレーション開始処理では、次の動作を行います。

キャリブレーション状態を初期化します。

入力切り替え制御レジスタ (ISC) により、RxD3端子の入力信号をタイマ入力に設定します。

タイマ・アレイ・ユニット0 (TAU0) のチャンネル7をTI07端子での両エッジ検出に設定します。

で設定したタイマ・アレイ・ユニット0 (TAU0) のチャンネル7によるキャプチャ動作を開始します。ロウ・レベルのパルス入力の立ち上がりエッジのときにINTTM07割り込みが発生します。

```

;*****
;
;
;   キャリブレーション開始処理
;
;-----
;
; マスタ機器から入力したパルスのロウ・レベル間的高速内蔵発振クロックでの
; カウント数を計測するため、入力パルス間隔測定として動作させ、割り込みを
; 発生させます。
;*****
SCALIBSTT:
MOV     RCALIBST, #CCALIB_RESET      ;キャリブレーション状態の初期化
MOV     ISC,      #00000010B        ;入力切り替え制御
;      | | | | | | | | | | +----- ISC0: INTP0端子の入力信号を外部割り込み入力とする
;      | | | | | | | | | | (通常動作)
;      | | | | | | | | | | +----- ISC1: RxD3端子の入力信号をタイマ入力とする
;      | | | | | | | | | | (ウエイクアップ信号検出)
;      ++++++ <000000>
;
;      注意 RxD3端子は使用する周辺の初期設定にて入力に設定済み
;
;-----
;
;   キャプチャの設定
;-----
;
; マスタ機器から送信される
; ロウ・レベルの入力パルス幅を計測します。
;-----
SET1    !TAU0EN                      ;TAU0使用許可
MOV     TPSOL,    #00000010B          ;タイマ・クロック選択
;      | | | | | | | | | | +----- PRS003-000: fCLK/2^2
;      | | | | | | | | | | (シリアル・クロックと同じにしておく)
;      ++++++ <000000>
;      +----- PRS013-010: 未使用
MOVW    AX,      #0000001010001100B;動作モード設定
;      | | | | | | | | | | | | | | | | +----- MD073-070: キャプチャ&ワンカウント・モード
;      | | | | | | | | | | | | | | | | <00>
;      | | | | | | | | | | | | | | | | +----- CIS071-070: 両エッジ(ロウ・レベル幅測定時)
;      | | | | | | | | | | | | | | | | +----- STS072-070: TI07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
;      | | | | | | | | | | | | | | | | +----- MASTER07: 単体動作
;      | | | | | | | | | | | | | | | | +----- CCS07: CKS00ビットで指定した動作クロックMCK
;      | | | | | | | | | | | | | | | | <00>
;      +----- CKS07: PRSで選択した動作クロックCK00 (MCK)
MOVW    TMR07,   AX
CLR1    TMIF07                      ;割り込み要求クリア
CLR1    TMMK07                      ;割り込み許可
SET1    TSOL.7                       ;TS07:TAU0のチャンネル7のキャプチャ動作開始
; (トリガ動作)
RET

```

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

```

/*****
キャリブレーション開始処理
-----
マスタ機器から入力したパルスのロウ・レベル間的高速内蔵発振クロックでの
カウント数を計測するため、入力パルス間隔測定として動作させ、割り込みを
発生させます。
*****/
static void fn_CalibrationStart(void)
{
    ucCalibrationStatus = CALIBRATION_STATUS_RESET; /* キャリブレーション状態の初期化 */

    ISC = 0b00000010; /* 入力切り替え制御 */
    /* | | | | | | | | +----- ISC0: INTP0端子の入力信号を外部割り込み入力とする(通常動作) */
    /* | | | | | | | | +----- ISC1: RxD3端子の入力信号をタイマ入力とする(ウエイクアップ信号検出) */
    /* ++++++----- <000000> */
    /*
    /*          注意 RxD3端子は使用する周辺の初期設定にて入力に設定済み
    /*

    /*-----
    キャプチャの設定
    -----
    マスタ機器から送信される
    ロウ・レベルの入力パルス幅を計測します。
    -----*/
    TAU0EN = 1; /* TAU0使用許可 */

    TPS0L = 0b00000010; /* タイマクロック選択 */
    /* | | | | | +----- PRS003-000: fCLK(シリアル・クロックと同じにしてお
    /* | | | | | <> */
    /* ++++++----- PRS013-010: 未使用 */

    TMR07 = 0b0000001010001100; /* 動作モード設定 */
    /* | | | | | | | | | | +----- MD073-070: キャプチャ&ワンカウント・モード */
    /* | | | | | | | | | | +----- <00> */
    /* | | | | | | | | | | +----- CIS071-070: 両エッジ(ロウ・レベル幅測定時) */
    /* | | | | | | | | | | +----- STS072-070: TI07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 */
    /* | | | | | | | | | | +----- MASTER07: 単体動作 */
    /* | | | | | | | | | | +----- CCS07: CKS00ビットで指定した動作クロックMCK */
    /* | | | | | | | | | | +----- <0> */
    /* +----- CKS07: PRSで選択した動作クロックCK00(MCK) */

    TMIF07 = 0; /* 割り込み要求クリア */
    TMMK07 = 0; /* 割り込み許可 */

    TS0L.7 = 1; /* TS07:TAU0のチャンネル7のキャプチャ動作開始(トリガ動作) */
}

```

4.5 INTST3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。

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。

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

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

のチェックで送信が終了した場合、UART3送信動作を停止します。

```

;*****
;
;      INTST3割り込み (UART3の送信完了割り込み)
;
;*****
INT_UARTST:
-----
SEL      RB1                ;レジスタバンク切り替え
-----
INC      RTXBUFP            ;送信データ・カウンタ更新
CMP      RTXBUFP,#RTXBUFE-RTXBUF ;最終データ送信終了?
BNC     $HINTST800         ; Yes,
-----
MOV      A, RTXBUFP        ;送信データ・カウンタ
MOV      B, A
MOV      A, RTXBUF[B]     ;今回の送信データを取得
MOV      TxD3, A          ;データを送信データ・レジスタに設定
BR       HINTSTRET
-----
HINTST800:
;データ送信終了
SET1    STMK3              ;割り込み禁止
SET1    !ST1L.2           ;ST12:UART3送信動作停止
-----
HINTSTRET:
RETI

```

C言語での割り込み処理の設定

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

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

```
#pragma interrupt          INTRTC          fn_intrtc
```

関数名：割り込み処理を記述した関数名を記述します。

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

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

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

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

C言語のINTST3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。

```

/*****
INTST3割り込み (UART3の送信完了割り込み)
*****/
__interrupt void fn_intst3(void)
{
    ucTxBufferCounter++;          /* 送信データ・カウンタ更新 */

    if(ucTxBufferCounter < sizeof(ucTxBuffer)){
        /* 後続データあり */
        /* 今回の送信データを送信データ・レジスタに設定 */
        TxD3 = ucTxBuffer[ucTxBufferCounter];
    }
    else{
        /* 最終データ送信終了 */
        STMK3 = 1;                /* 割り込み禁止 */
        ST1L.2 = 1;              /* ST12:UART3送信動作停止 */
    }
}

```

4.6 INTSR3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。

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。

受信データ・カウンタに応じた受信バッファ位置に受信データを格納します。

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

のチェックで受信が終了した場合、UART3送信動作を停止します。

```

;*****
;
;      INTSR3割り込み (UART3受信完了割り込み)
;
;*****
INT_UARTSR:
-----
SEL      RB1                ;レジスタバンク切り替え
-----
MOV      A,      RRXBUIFP   ;受信データ・カウンタ
MOV      B,      A
MOV      A,      RxD3       ;受信データ取得
MOV      RRXBUIF[B],A      ;今回の受信データを格納
-----
INC      RRXBUIFP          ;受信データ・カウンタ更新
CMP      RRXBUIFP,#RRXBUIFE-RRXBUIF ;データ受信終了?
BC      $HINTSRRET        ; No,
;データ受信終了
-----
SET1    SRMK3             ;割り込み禁止
SET1    !ST1L.3          ;ST13:UART3受信動作停止
-----
HINTSRRET:
RETI

```

C言語のINTSR3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。

```

/*****
INTSR3割り込み (UART3受信完了割り込み)
*****/
__interrupt void fn_intsr3(void)
{
    ucRxBuffer[ucRxBufferCounter] = (unsigned char)SDR13;
    /* 今回の受信データを格納 */
    ucRxBufferCounter++;
    /* 受信データ・カウンタ更新 */
    if(ucRxBufferCounter == sizeof(ucRxBuffer)){
        /* データ受信終了 */
        SRMK3 = 1;
        /* 割り込み禁止 */
        ST1L.3 = 1;
        /* ST13:UART3受信動作停止 */
    }
}

```

4.7 INTTM07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。

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。

タイマ・アレイ・ユニット0 (TAU0) のチャンネル7のキャプチャの動作を停止します。

入力パルスのロウ・レベル間の高速内蔵発振クロックのタイマ・カウント値を取得します。

INTTM07割り込みを禁止します。

で得たタイマ・カウント値の範囲をチェックし、範囲外であれば、キャリブレーション状態をキャリブレーション失敗に設定し、処理を終了します。

で得たタイマ・カウント値から転送ボー・レート用クロックの分周値が9600bpsとなるように設定値を算出します。詳細は、3.5 キャリブレーションを参照してください。

で算出した分周値をシリアル・データ・レジスタ12 (SDR12) およびシリアル・データ・レジスタ13 (SDR13) のフォーマットに合わせ、設定します。

シリアル・アレイ・ユニット1 (SAU1) のチャンネル1のUART3受信動作を開始します。

キャリブレーション状態をキャリブレーション完了に設定します。

```

;*****
;
;      INTTM07割り込み処理
;      (INTTM07使用, TI07端子での両エッジ検出)
;
;-----
;
;      キャリブレーション開始処理の呼び出し後、マスタ機器から入力したパルスの
;      ロウ・レベル間の計測完了時に発生する割り込み処理です。立ち下がりエッジ
;      から立ち上がりエッジまでをTAU0のチャンネル7のキャプチャ・モードで計測した
;      ロウ・レベル間の高速内蔵発振クロックのタイマ・カウント値からUART3の転送
;      ボー・レートの設定を行います。
;*****
INT_DATAIN:
-----
SEL      RB1          ;レジスタバンク切り替え
-----
SET1     TTOL.7      ;TS07:TAU0のチャンネル7のキャプチャ動作停止(トリガ動作)
MOVW    AX,          TCR07 ;ロウ・レベル間の高速内蔵発振クロックでのタイマ・カウント値を取
得
-----
SET1     TMMK07      ;割り込み禁止
-----
CMP      TSR07L,    #0 ;オーバーフロー発生?
BNZ     $HIDIN050   ; Yes, (キャリブレーション失敗)
CMPW    AX,         #0FFH*9+1 ;ボー・レート用分周値の範囲内?
BC      $HIDIN100   ; Yes,
HIDIN050:
MOV     RCALIBST, #CCALIB_ERROR ;キャリブレーション失敗
BR      HIDINRET
HIDIN100:
;-----
;      ボー・レートの算出式
;-----
;      取得したタイマ・カウント値から転送ボー・レートが9600bpsとなるように
;      動作クロックの分周値用に変換
;-----
;
;      【入力したパルスの波形】
;
;      ・データ長=8ビット
;      ・転送レート=9600bps
;      ・データ位相=正転出力
;      ・パリティ・ビット=なし
;      ・ストップ・ビット=1ビット付加
;      ・8ビット・データ=00H
;
;
;      スタート | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | ストップ
;
;      ビット数   1   2   3   4   5   6   7   8   9
;
;      |                               |
;
;      【ボー・レートの算出式】

```



```

;
; (ボー・レート) =
; {対象チャンネルの動作クロック (MCK) 周波数} / (SDRmn[15:9]+1) / 2 [bps] ...
;
; SDRmn[15:9]は、SDRmnレジスタのビット15-9の値を示します。
;
; で表されます。(詳細はユーザーズ・マニュアルを参照してください。)
;
; ・CPU/周辺ハードウェア・クロック・・・fCLK [Hz]
; ・ボー・レート・・・9600 [bps]
;
; として、式から、
;
; (SDRmn[15:9]+1) = {fCLK/2^2}/9600/2 ...
;
; と表せます。
; また、高速内蔵発振クロックの2^2分周で動作するTAU0のチャンネル7の
; キャプチャで計測した時間と、転送レートが9600bpsのデータ9ビット分の
; 時間は等しいことから、タイマ・カウント値 (AX) は、
;
; 1/9600*9 = {2^2/fCLK}*AX [s]
;
; と表せますので、fCLKは、
;
; fCLK/2^2=AX*9600/9 ...
;
; となり、式から、
;
; SDRmn[15:9] = {AX*9600/9}/9600/2-1 = AX/18-1
;
; 上記の式より、シリアル・データ・レジスタ (SDR1x) のUART3転送ボー・レート用
; 動作クロックの分周値の算出が行えます。
;
; -----
; AX/18-1の計算
MOV B, #0 ;B=AX/18の計算結果
HIDIN300:
SUBW AX, #18
BC $HIDIN500
INC B
BR $HIDIN300
HIDIN500:
DEC B ;B=(AX/18の計算結果)-1

;シリアル・データ・レジスタ (SDR1x) のビット15-9に合わせる
MOV A, B ;B=AX/600-1の計算結果 (SDRmn[15:9])
ROL A, 1 ;ビット15-9に合わせる
AND A, #11111110B ;ビット8は0固定
MOV X, #0 ;SDRmnの下位7-0ビット用の設定

MOVW SDR12, AX ;送信側転送ボー・レートの設定
MOVW SDR13, AX ;受信側転送ボー・レートの設定

; -----
; UART3受信動作開始
; -----
CLR1 SRIF3 ;割り込み要求クリア
CLR1 SRMK3 ;割り込み許可
SET1 !SS1L.3 ;SS13:UART3受信動作許可

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

```

C言語のINTTM07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。

```

/*****
INTTM07割り込み処理
(INTTM07使用、TI07端子での両エッジ検出)

-----
キャリブレーション開始処理の呼び出し後、マスタ機器から入力したパルスの
ロウ・レベル間の計測完了時に発生する割り込み処理です。立ち下がりエッジ
から立ち上がりエッジまでをTAU0のチャンネル7のキャプチャ・モードで計測した
ロウ・レベル間的高速内蔵発振クロックのタイマ・カウント値からUART3の転送
ボー・レートの設定を行います。
*****/

```

```

__interrupt void fn_inttm07(void)
{
    register unsigned short ushnTimeCnt;      /* タイマ・カウンタ値 */

    TT0L.7 = 1;                               /* TS07:TAU0のチャンネル7のキャプチャ動作停止(トリガ動作) */
    ushnTimeCnt = TCR07;                       /* ロウ・レベル間の高速内蔵発振クロックでのタイマ・カウント値を
取得 */

    TMMK07 = 1;                               /* 割り込み禁止 */

    if((TSR07L!=0) || (ushnTimeCnt>0x0ff*9)){
        /* オーバーフロー発生 もしくは ボー・レート用分周値の範囲外 */
        ucCalibrationStatus = CALIBRATION_STATUS_ERROR;      /* キャリブレーション失敗 */
    }
    else{
        /*-----
        ボー・レートの算出式
        -----
        取得したタイマ・カウント値から転送ボー・レートが9600bpsとなるように
        動作クロックの分周値用に変換
        -----

        【入力したパルスの波形】

        ・データ長=8ビット
        ・転送レート=9600bps
        ・データ位相=正転出力
        ・パリティ・ビット=なし
        ・ストップ・ビット=1ビット付加
        ・8ビット・データ=00H

        ┌───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┐
        │   │   │   │   │   │   │   │   │   │   │   │   │
        │   │   │   │   │   │   │   │   │   │   │   │   │
        │   │   │   │   │   │   │   │   │   │   │   │   │
        └───┴───┴───┴───┴───┴───┴───┴───┴───┴───┴───┘
        1 2 3 4 5 6 7 8 9
        |               |
        測定

        【ボー・レートの算出式】

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

        SDRmn[15:9]は、SDRmnレジスタのビット15-9の値を示します。
        で表されます。(詳細はユーザーズ・マニュアルを参照してください。)

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

        として、式から、

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

        と表せます。
        また、高速内蔵発振クロックの2^2分周で動作するTAU0のチャンネル7の
        キャプチャで計測した時間と、転送レートが9600bpsのデータ9ビット分の
        時間は等しいことから、タイマ・カウント値(ushnTimeCnt)は、

        1/9600*9 = {2^2/fCLK}*ushnTimeCnt [s]

        と表せますので、fCLKは、

        fCLK/2^2=ushnTimeCnt*9600/9 ...

        となり、式から、

        SDRmn[15:9] = {ushnTimeCnt*9600/9}/9600/2-1 = ushnTimeCnt/18-1

        上記の式より、シリアル・データ・レジスタ(SDR1x)のUART3転送ボー・レート用
        動作クロックの分周値の算出が行えます。

        ----- */
        /* 受信転送ボー・レートの設定 */
        SDR12 = SDR13 = ((ushnTimeCnt/18)-1)<<9) & 0b1111111100000000;

        /*-----
        受信動作開始
        ----- */
        SRIF3 = 0;                               /* 割り込み要求クリア */
        SRMK3 = 0;                               /* 割り込み許可 */
        SSL.3 = 1;                               /* SS13:UART3受信動作許可 */

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

```

第5章 関連資料

資料名		和文 / 英文
78K0R/KE3 ユーザーズ・マニュアル		PDF
78K0R/KF3 ユーザーズ・マニュアル		PDF
78K0R/KG3 ユーザーズ・マニュアル		PDF
78K0R/KH3 ユーザーズ・マニュアル		PDF
78K0R/KJ3 ユーザーズ・マニュアル		PDF
78K0Rマイクロコントローラ 命令編 ユーザーズ・マニュアル		PDF
RA78K0R アセンブラ・パッケージ ユーザーズ・マニュアル	言語編	PDF
	操作編	PDF
CC78K0R Cコンパイラ ユーザーズ・マニュアル	言語編	PDF
	操作編	PDF
PM+ プロジェクト・マネージャ ユーザーズ・マニュアル		PDF

付録A プログラム・リスト

プログラム・リスト例として、78K0R/KG3マイクロコントローラのソース・プログラムを次に示します。

```
main.asm (アセンブリ言語版)
;*****
;
;
;   NEC Electronics   78K0R/KG3シリーズ
;
;*****
;
;   78K0R/KG3シリーズ   サンプル・プログラム
;*****
;
;   ロウ・レベルの入力パルスを使用したポー・レート補正付きUART
;*****
;
; 【履歴】
;
;   2007.11.--   新規作成
;*****
;
;
; 【概要】
;
;
; このサンプル・プログラムは、シリアル・アレイ・ユニットのUARTの機能を使用したUART
; 送受信を行うプログラムです。まず、UART送受信を行う前にキャリブレーション開始処理
; の呼び出しにより、キャリブレーションを行います。キャリブレーションは、マスタ機器
; から送信されるデータ長が8ビット、転送レートが9600bps、データ位相が正転出力、パリ
; ティ・ビットなし、ストップ・ビットが1ビット付加、8ビットのデータが00Hのデータを
; 受信します。受信したデータをタイマ入力端子へのロウ・レベルの入力パルスとして、立
; ち上がりエッジから立ち上がりエッジまでのロウ・レベル間的高速内蔵発振クロックでの
; カウント数を計測します。計測したタイマ・カウント値を高速内蔵発振クロックを動作ク
; ロックとしているシリアル・アレイ・ユニットの転送ポー・レート用に計算し、設定しま
; す。キャリブレーション後、マスタ機器からA/Dコンバータのチャンネルと分解能の指定値
; を受信し、指定されたA/Dコンバータのチャンネルで変換したA/D値を指定された分解能でマ
; スタ機器へ送信します。このサンプルプログラムのUARTは、データ長が8ビット、転送レー
; トが9600bps、データ位相が正転出力、パリティ・ビットなし、ストップ・ビットが1ビッ
; ト付加、データ方向がLSBファースト転送となるように設定し、半2重での送受信としてい
; ます。
;
;
;
; <使用する周辺の初期設定の主な内容>
;
;
;   ・割り込みの禁止
```

```
; ・CPU / 周辺ハードウェア・クロックを高速内蔵発振クロックの動作に設定
;
; ・ポートの設定
;
; ・A/Dコンバータの設定
;
; ・SAU1のチャンネル2をUART3送信用，チャンネル3をUART3受信用に設定
;
; ・割り込みの許可
;
;
;
; <メイン処理の主な内容>
;
;
; ・UART3のエラー・チェック
;
; ・UART3受信データの解析
;
; ・A/D値の取り込み
;
; ・UART3送信データの作成
;
; ・UART3送受信動作の開始
;
;
;
; <キャリブレーション開始処理の主な内容>
;
;
; ・ISCレジスタにより，RxD3端子の入力信号をタイマ入力に設定
;
; ・TAU0のチャンネル7をキャプチャ・モードに設定
;
; ・TAU0のチャンネル7をT107端子での両エッジ検出に設定
;
; ・TAU0のチャンネル7のキャプチャ動作開始
;
;
;
; <INTST3割り込み処理（UART3の送信完了割り込み）>
;
;
; ・UART3送信データのカウンタ
;
; ・UART3送信データの設定
;
; ・UART3送信動作の停止
;
;
;
; <INTSR3割り込み処理（UART3受信完了割り込み）>
;
;
; ・UART3受信データのカウンタ
;
; ・UART3受信データの保存
;
; ・UART3受信動作の停止
;
;
;
; <INTTM07割り込み処理（INTTM07使用，T107端子での両エッジ検出）>
;
;
; ・TAU0のチャンネル7のキャプチャ動作停止
;
; ・キャプチャ測定値から高速内蔵発振で動作するUART用転送ボー・レートの設定
;
; ・TAU0のチャンネル7のキャプチャ・モードで計測したロウ・レベル間の高速内蔵発振
```

```
; クロックのタイマ・カウンタ値からUART3の転送ボー・レートの設定
;   ・ UART3の受信動作開始
;
;
;
;
;*****
```

```
=====
```

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

```
TVCT1 CSEG   AT      000000H
      DW      RESET_START      ;(00)  RESET入力,POC,LVI,WDT,TRAP
TVCT2 CSEG   AT      000004H
      DW      RESET_START      ;(04)  INTWDTI
      DW      RESET_START      ;(06)  INTLVI
      DW      RESET_START      ;(08)  INTP0
      DW      RESET_START      ;(0A)  INTP1
      DW      RESET_START      ;(0C)  INTP2
      DW      RESET_START      ;(0E)  INTP3
      DW      RESET_START      ;(10)  INTP4
      DW      RESET_START      ;(12)  INTP5
      DW      INT_UARTST       ;(14)  INTST3
      DW      INT_UARTSR       ;(16)  INTSR3
      DW      RESET_START      ;(18)  INTSRE3
      DW      RESET_START      ;(1A)  INTDMA0
      DW      RESET_START      ;(1C)  INTDMA1
      DW      RESET_START      ;(1E)  INTST0/INTCS100
      DW      RESET_START      ;(20)  INTSR0/INTCS101
      DW      RESET_START      ;(22)  INTSRE0
      DW      RESET_START      ;(24)  INTST1/INTCS110/INTIIC10
      DW      RESET_START      ;(26)  INTSR1
      DW      RESET_START      ;(28)  INTSRE1
      DW      RESET_START      ;(2A)  INTIIC0
      DW      RESET_START      ;(2C)  INTTM00
      DW      RESET_START      ;(2E)  INTTM01
      DW      RESET_START      ;(30)  INTTM02
      DW      RESET_START      ;(32)  INTTM03
      DW      RESET_START      ;(34)  INTAD
      DW      RESET_START      ;(36)  INTRTC
      DW      RESET_START      ;(38)  INTRTCI
      DW      RESET_START      ;(3A)  INTKR
```

```

DW      RESET_START          ;(3C)  INTST2/INTCS120/INTI1C20
DW      RESET_START          ;(3E)  INTSR2
DW      RESET_START          ;(40)  INTSRE2
DW      RESET_START          ;(42)  INTTM04
DW      RESET_START          ;(44)  INTTM05
DW      RESET_START          ;(46)  INTTM06
DW      INT_DATAIN           ;(48)  INTTM07
DW      RESET_START          ;(4A)  INTP6
DW      RESET_START          ;(4C)  INTP7
DW      RESET_START          ;(4E)  INTP8
DW      RESET_START          ;(50)  INTP9
DW      RESET_START          ;(52)  INTP10
DW      RESET_START          ;(54)  INTP11

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

;=====
;
;   スタック領域の確保
;
;=====
DSTK   DSEG   AT      0FEFC0H
STACKEND:
      DS      20H                ;スタック領域を32バイト確保
STACKTOP:                          ;スタック領域の先頭アドレス = FEFE0H

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

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

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

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

```

```
RCALIBST:    DS      1                ; キャリブレーション状態
CCALIB_RESET EQU    0                ; キャリブレーション未終了状態
CCALIB_END   EQU    1                ; キャリブレーション完了状態
CCALIB_ERROR EQU    2                ; キャリブレーション失敗
```

```
;*****
;
;   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    RBO
```

```
;-----
;   スタック・ポインタの設定
;-----
```

```
    MOVW   SP,    #LOWW STACKTOP ; スタック・ポインタを設定
```

```
;-----
;   クロック周波数の設定
;-----
```

```
;   高速内蔵発振クロックで動作が行えるように設定します。
;-----
```

```
    MOV    CMC,    #00000000B      ; クロック動作モード
                                ; |||||+----- AMPH: 2MHz fMX 10MHz
                                ; |||+++----- <000>
                                ; ||+----- OSCSELS: P123/P124端子を入力ポート・モード
                                ; |+----- <0>
                                ; ++----- EXCLK/OSCSEL: 入力ポート・モード
```

```
    MOV    CSC,    #11000000B      ; 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発振回路停止
```



```

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

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

;-----
;   ポート0の設定
;-----

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

;-----
;   ポート1の設定
;-----

MOV   P1, #00000000B      ;P10-P17の出力ラッチLow
      ;[後でP13(TxD3)の設定を行う]
MOV   PM1, #00000000B     ;P10-P17を出力ポートに設定
      ;[後でP14(RxD3)の設定を行う]

;-----
;   ポート2の設定
;-----

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

;-----
;   ポート3の設定
;-----

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

;-----
;   ポート4の設定
;-----

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

```

```

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

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

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

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

;-----
;   ポート11の設定
;-----
MOV   P11,   #00000000B           ;P110-P111の出力ラッチLow
MOV   PM11,  #111111100B          ;P110-P111を出力ポートに設定

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

;-----
;   ポート13の設定
;-----
MOV   P13,   #00000000B           ;P130-P131の出力ラッチLow
MOV   PM13,  #111111100B          ;P130-P131を出力ポートに設定

```

```

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

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

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

MOV   ADPC,  #00000000B           ;A/Dポート・コンフィギュレーション
;|||++++----- ADPC4-0: 使用するP20-P27をすべてアナログ入力に切り替え
;+++----- <00>

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

;-----
;   UART3の設定
;-----
SET1  !SAU1EN                ;UART3のあるSAU1使用許可

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

MOV   !SPS1L, #00000010B          ;プリスケアラ(動作クロック)の設定
;|||++++----- PRS103-100(CK10): fCLK/2^2
;++++----- PRS113-110(CK11): 未使用

```

```

;-----
; UART3送信設定(SAU1のチャンネル2使用)
;-----
MOV     RTXBUF, #0                ;送信データ・カウンタ初期化

MOVW   AX,    #000000000100010B   ;動作モードなどの設定
;|||||||||||||+----- MD120: 転送完了割り込み
;|||||||||||||++----- MD122-121: UARTモード
;|||||||||+----- <100>
;|||||||+----- SIS120: 立ち下がりエッジをスタートビットとして検出
;|||||||+----- <0>
;|||||||+----- STS12: UART送信
;||+++++----- <00000>
;|+----- CCS12: CKS00ビットで指定した動作クロックMCK
;+----- CKS12: PRSで選択した動作クロックCK10(MCK)

MOVW   !SMR12, AX

MOVW   AX,    #1000000010010111B   ;通信フォーマットの設定
;|||||||||||||+++----- DLS122-120: 8ビット・データ長
;|||||||||||||+----- <0>
;|||||||||++----- SLC121-120: ストップ・ビット長 = 1ビット
;|||||||||+----- <00>
;|||||||+----- DIR12: LSBファーストで出力
;|||||++----- PTC121-120: パリティ・ビットを出力しない
;||||+----- EOC12: UART送信時はエラー割り込み未使用
;|||+----- <0>
;||++----- DAP12/CKP12: UARTモードでは未使用
;++----- TXE12/RXE12: 送信のみを行う

MOVW   !SCR12, AX

; SDR12 転送ボー・レートの設定はINTTM07割り込み処理にて設定

MOV     SOL1L, #00000000B          ;出力データのレベルの設定(反転なし)

MOVW   AX,    #0000110000001100B   ;初期出力レベルの設定
;|||||||||||||+----- S010: チャンネル0のシリアル・クロック出力
;|||||||||||||+----- <1>
;|||||||||||||+----- S012: チャンネル2のシリアル・クロック出力
;|||||||+++++----- <00001>
;|||||||+----- CK010: チャンネル0のシリアル・データ出力
;+++++----- <0000111>

MOVW   !S01,  AX
SET1   !SOE1L.2                  ;SOE12:シリアル出力許可

```

```

;済 CLR1 PM1.3 ;ポート1の設定にて設定済み
SET1 P1.3 ;P13=TxD3

;-----
; UART3受信設定(SAU1のチャンネル3使用)
;-----

MOV RRXBUFP,#0 ;受信データ・カウンタ初期化

SET1 PM1.4 ;P14=RxD3

MOVW AX, #0000000100100010B ;動作モードなどの設定
;|||||||||||||+----- MD130: 転送完了割り込み
;|||||||||||||++----- MD132-131: UARTモード
;|||||||||+++----- <100>
;|||||||+----- SIS130: 立ち下がりエッジをスタートビットとして検出
;|||||||+----- <0>
;||||||+----- STS13: RxD端子の有効エッジ(UART受信設定)
;||+++++----- <00000>
;|+----- CCS13: CKS13ビットで指定した動作クロックMCK
;+----- CKS13: PRSで選択した動作クロックCK10(MCK)

MOVW !SMR13, AX

MOVW AX, #0100000010010111B ;通信フォーマットの設定
;|||||||||||||+++----- DLS132-130: 8ビット・データ長
;|||||||||||||+----- <0>
;|||||||||++----- SLC131-130: ストップ・ビット長 = 1ビット
;|||||||+----- <0>
;|||||||+----- DIR13: LSBファーストで入力を行う
;|||||++----- PTC131-130: パリティなしで受信
;||||+----- EOC13: UART受信のエラー割り込み許可
;|||+----- <0>
;||++----- DAP13/CKP13: UARTモードでは未使用
;+----- TXE13/RXE13: 受信のみを行う

MOVW !SCR13, AX

; SDR13 転送ボー・レートの設定はINTTM07割り込み処理にて設定

EI ;割り込み許可

```

```

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

;*****
;
;   メイン処理
;
;*****
MAIN_LOOP:
;-----
;   動作状態による分岐
;-----
      CMP     RCALIBST,#CCALIB_END      ;キャリブレーション完了?
      BZ      $LMAIN100                ; Yes,
      BR      LMAINRET                 ; No,
LMAIN100:
      MOV     A,      SE1L              ;動作停止状態/停止状態を確認
      BT     A.2,    $LMAINRET         ;SE12:UART3送信動作停止中? No,
      BT     A.3,    $LMAINRET         ;SE13:UART3受信動作停止中? No,

;*****
;   UART3受信完了
;*****
      CMP     RRXBUFP,#RRXBUFE-RRXBUF ;最終データの受信完了?
      BC     $LMAINTX                  ; No,
      MOV     RRXBUFP,#0                ;受信データ・カウンタ初期化

      MOV     A,      SSR13L            ;UART3のステータスを確認
      AND     A,      #00000111B       ;(フレーミング,パリティ,オーバラン)エラーあり?
      BNZ    $LMAIN800                 ; Yes,

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

```

```

; 【受信データ】
; byte |          +0          |          +1          |
;      +-----+-----+
;      |A/D値分解能の指定|A/Dチャンネルの指定|
;      | 08H: 8ビット  |  0:ch0 ……  |
;      | 0AH:10ビット  |  …… 7:ch7  |
;      +-----+-----+
;
; 【送信データ】
;
; byte |          +0          |          +1          |
; bit  | 7|6|5|4|3|2|1|0| 7|6|5|4|3|2|1|0|
;      +---+-----+-----+
;      | |          | 8ビットA/D値  |
;      | |          | (8ビット指定時) |
;      +---+-----+-----+
;      +---+-----+-----+
;      | |          | 10ビットA/D値  |
;      | |          | (10ビット指定時) |
;      +---+-----+-----+
;
;      |
;      +-エラー情報：受信したデータにエラーあり(1)/エラーなし(0)
;
;=====
;-----
; 受信データのチェック
;-----
CMP    RRXBUF, #10                ;A/D値分解能の指定は10ビット?
BZ     $LMAINRX300                ; Yes,
CMP    RRXBUF, #8                 ;A/D値分解能の指定は8ビット?
BNZ    $LMAINRX400                ; No, (上記以外はエラー情報設定)
LMAINRX300:
CMP    RRXBUF+1, #7+1             ;チャンネル指定(0-7)は正常値?
BC     $LMAINRX500                ; Yes,
LMAINRX400:
;エラー
MOVW   AX, #8000H                 ;エラー情報設定
BR     LMAINRX700
LMAINRX500:
;-----
; A/D値の取得
;-----
MOV    A, RRXBUF+1                ;A/D変換するチャンネルの設定

```

```

MOV    ADS,    A

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

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

;2変換目(正規データ)
CLR1   ADIF                                ;割り込み要求クリア
SET1   ADCS                                ;A/D変換動作許可
LMAINRX550:
NOP
BF     ADIF,    $LMAINRX550

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

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

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

;UART3送信動作開始
CLR1   STIF3                                ;割り込み要求クリア
CLR1   STMK3                                ;割り込み許可
SET1   !SS1L.2                             ;SS12:UART3送信動作開始

MOV    A,     RTXBUF                       ;設定した送信データの第一バイト
MOV    TxD3,  A                            ;データを送信データ・レジスタに設定

```



```

BR      LMAINRET

LMAINTX:
;*****
; UART3送信完了
;*****
CMP     RTXBUFP,#RTXBUFE-RTXBUF ;最終データの送信完了?
BC      $LMAINRET                ; No,
MOV     RTXBUFP,#0                ;送信データ・カウンタ初期化

LMAIN800:
;UART3受信動作開始
CLR1    SRIF3                      ;割り込み要求クリア
CLR1    SRMK3                      ;割り込み許可
SET1    !SS1L.3                   ;SS13:UART3受信動作開始

LMAINRET:
BR      MAIN_LOOP                  ; MAIN_LOOPへ

;*****
;
; INTST3割り込み (UART3の送信完了割り込み)
;
;*****
INT_UARTST:

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

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

MOV     A,      RTXBUFP            ;送信データ・カウンタ
MOV     B,      A
MOV     A,      RTXBUF[B]          ;今回の送信データを取得
MOV     TxD3,  A                   ;データを送信データ・レジスタに設定
BR      HINTSTRET

HINTST800:
;データ送信終了
SET1    STMK3                      ;割り込み禁止
SET1    !ST1L.2                   ;ST12:UART3送信動作停止

HINTSTRET:
RETI

```

```

;*****
;
;
;   INTSR3割り込み (UART3受信完了割り込み)
;
;*****
INT_UARTSR:

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

    MOV    A,    RRXBUFF      ;受信データ・カウンタ
    MOV    B,    A
    MOV    A,    RxD3         ;受信データ取得
    MOV    RRXBUF[B],A       ;今回の受信データを格納

    INC    RRXBUFF           ;受信データ・カウンタ更新
    CMP    RRXBUFF,#RRXBUFE-RRXBUF ;データ受信終了?
    BC     $HINTSRRET        ; No,
;データ受信終了
    SET1   SRMK3             ;割り込み禁止
    SET1   !ST1L.3          ;ST13:UART3受信動作停止
HINTSRRET:
    RETI

;*****
;
;
;   キャリブレーション開始処理
;
;-----
;   マスタ機器から入力したパルスのロウ・レベル間的高速内蔵発振クロックでの
;   カウント数を計測するため、入力パルス間隔測定として動作させ、割り込みを
;   発生させます。
;*****
SCALIBSTT:

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

```

```

MOV    ISC,    #00000010B          ;入力切り替え制御
        ;|||||+----- ISC0: INTP0端子の入力信号を外部割り込み入力とする(通常
        ;|||||          動作)
        ;|||||+----- ISC1: RxD3端子の入力信号をタイマ入力とする(ウエイクア
        ;|||||          ップ信号検出)
        ;+++++----- <000000>
        ;
        ;                               注意 RxD3端子は使用する周辺の初期設定にて入力に設定済
        ;                               み

;-----
; キャプチャの設定
;-----
; マスタ機器から送信される
; ロウ・レベルの入力パルス幅を計測します。
;-----

SET1    !TAU0EN          ;TAU0使用許可

MOV     TPSOL, #00000010B          ;タイマ・クロック選択
        ;||||++++----- PRS003-000: fCLK/2^2(シリアル・クロックと同じにしてお
        ;||||          く)
        ;++++----- PRS013-010: 未使用

MOVW    AX,    #0000001010001100B ;動作モード設定
        ;|||||||||++++----- MD073-070: キャプチャ&ワンカウント・モード
        ;|||||||||++----- <00>
        ;|||||||||++----- CIS071-070: 両エッジ(ロウ・レベル幅測定時)
        ;|||||++++----- STS072-070: TI07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
        ;|||+----- MASTER07: 単体動作
        ;||+----- CCS07: CKS00ビットで指定した動作クロックMCK
        ;|++----- <00>
        ;+----- CKS07: PRSで選択した動作クロックCK00(MCK)

MOVW    TMR07, AX

CLR1    TMIF07          ;割り込み要求クリア
CLR1    TMMK07          ;割り込み許可

SET1    TSOL.7          ;TS07:TAU0のチャンネル7のキャプチャ動作開始(トリガ動作)

RET

```

```

;*****
;
;
;   INTTM07割り込み処理
;   (INTTM07使用, T107端子での両エッジ検出)
;
;-----
;   キャリブレーション開始処理の呼び出し後, マスタ機器から入力したパルスの
;   ロウ・レベル間の計測完了時に発生する割り込み処理です。立ち下がりエッジ
;   から立ち上がりエッジまでをTAU0のチャンネル7のキャプチャ・モードで計測した
;   ロウ・レベル間的高速内蔵発振クロックのタイマ・カウント値からUART3の転送
;   ボー・レートの設定を行います。
;*****
INT_DATAIN:

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

    SET1   TTOL.7             ;TS07:TAU0のチャンネル7のキャプチャ動作停止(トリガ動作)
    MOVW   AX,    TCR07       ;ロウ・レベル間的高速内蔵発振クロックでのタイマ・カウン
ト値を取得

    SET1   TMMK07            ;割り込み禁止

    CMP    TSR07L, #0        ;オーバーフロー発生?
    BNZ    $HIDIN050        ; Yes, (キャリブレーション失敗)
    CMPW   AX,    #0FFH*9+1  ;ボー・レート用分周値の範囲内?
    BC     $HIDIN100        ; Yes,

HIDIN050:
    MOV    RCALIBST,#CCALIB_ERROR ;キャリブレーション失敗
    BR     HIDINRET

HIDIN100:
;-----
;   ボー・レートの算出式
;-----
;   取得したタイマ・カウント値から転送ボー・レートが9600bpsとなるように
;   動作クロックの分周値用に変換
;-----
;
;   【入力したパルスの波形】
;
;   ・データ長=8ビット
;   ・転送レート=9600bps
;   ・データ位相=正転出力
;   ・パリティ・ビット=なし

```

```

;   ・ストップ・ビット=1ビット付加
;   ・8ビット・データ=00H
;
;   ───────────────────────────────────────────────────────────────────────────────────
;   |   |   |   |   |   |   |   |   |   |
;   |スタート| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 |ストップ°
;   ────┬───┬───┬───┬───┬───┬───┬───┬───┬───
;
;   ビット数    1    2    3    4    5    6    7    8    9
;
;   |                                測定                                |
;
;
;   【ボー・レートの算出式】
;
;   (ボー・レート) =
;       {対象チャンネルの動作クロック(MCK)周波数}/(SDRmn[15:9]+1)/2 [bps] ……
;
;   SDRmn[15:9]は、SDRmnレジスタのビット15-9の値を示します。
;
;   で表されます。(詳細はユーザズ・マニュアルを参照してください。)
;
;   ・CPU/周辺ハードウェア・クロック・・・fCLK [Hz]
;   ・ボー・レート・・・9600 [bps]
;
;   として、式から、
;
;   (SDRmn[15:9]+1) = {fCLK/2^2}/9600/2 ……
;
;   と表せます。
;   また、高速内蔵発振クロックの2^2分周で動作するTAU0のチャンネル7の
;   キャプチャで計測した時間と、転送レートが9600bpsのデータ9ビット分の
;   時間は等しいことから、タイマ・カウント値 (AX) は、
;
;   1/9600*9 = {2^2/fCLK}*AX [s]
;
;   と表せますので、fCLKは、
;
;   fCLK/2^2=AX*9600/9 ……
;
;   となり、式から、
;
;   SDRmn[15:9] = {AX*9600/9}/9600/2-1 = AX/18-1
;

```

```

; 上記の式より, シリアル・データ・レジスタ(SDR1x)のUART3転送ボー・レート用
; 動作クロックの分周値の算出が行えます。
;
;-----
; AX/18-1の計算
MOV    B,      #0                ;B=AX/18の計算結果
HIDIN300:
SUBW   AX,     #18
BC     $HIDIN500
INC    B
BR     $HIDIN300
HIDIN500:
DEC    B                ;B=(AX/18の計算結果)-1

;シリアル・データ・レジスタ(SDR1x)のビット15-9に合わせる
MOV    A,      B                ;B=AX/600-1の計算結果(SDRmn[15:9])
ROL    A,      1                ;ビット15-9に合わせる
AND    A,      #11111110B       ;ビット8は0固定
MOV    X,      #0                ;SDRmnの下位7-0ビット用の設定

MOVW   SDR12,  AX                ;送信側転送ボー・レートの設定
MOVW   SDR13,  AX                ;受信側転送ボー・レートの設定

;-----
; UART3受信動作開始
;-----
CLR1   SRIF3                ;割り込み要求クリア
CLR1   SRMK3                ;割り込み許可
SET1   !SS1L.3              ;SS13:UART3受信動作許可

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

ロウ・レベルの入力パルスを使用したボー・レート補正付きUART

【履歴】

2007.11.-- 新規作成

【概要】

このサンプル・プログラムは、シリアル・アレイ・ユニットのUARTの機能を使用したUART送受信を行うプログラムです。まず、UART送受信を行う前にキャリブレーション開始処理の呼び出しにより、キャリブレーションを行います。キャリブレーションは、マスタ機器から送信されるデータ長が8ビット、転送レートが9600bps、データ位相が正転出力、パリティ・ビットなし、ストップ・ビットが1ビット付加、8ビットのデータが00Hのデータを受信します。受信したデータをタイマ入力端子へのロウ・レベルの入力パルスとして、立ち上がりエッジから立ち上がりエッジまでのロウ・レベル間的高速内蔵発振クロックでのカウント数を計測します。計測したタイマ・カウント値を高速内蔵発振クロックを動作クロックとしているシリアル・アレイ・ユニットの転送ボー・レート用に計算し、設定します。キャリブレーション後、マスタ機器からA/Dコンバータのチャンネルと分解能の指定値を受信し、指定されたA/Dコンバータのチャンネルで変換したA/D値を指定された分解能でマスタ機器へ送信します。このサンプルプログラムのUARTは、データ長が8ビット、転送レートが9600bps、データ位相が正転出力、パリティ・ビットなし、ストップ・ビットが1ビット付加、データ方向がLSBファースト転送となるように設定し、半2重での送受信としています。

<使用する周辺の初期設定の主な内容>

- ・ 割り込みの禁止
- ・ CPU / 周辺ハードウェア・クロックを高速内蔵発振クロックの動作に設定
- ・ ポートの設定
- ・ A/Dコンバータの設定
- ・ SAU1のチャンネル2をUART3送信用、チャンネル3をUART3受信用に設定
- ・ 割り込みの許可

< メイン処理の主な内容 >

- ・ UART3のエラー・チェック
- ・ UART3受信データの解析
- ・ A/D値の取り込み
- ・ UART3送信データの作成
- ・ UART3送受信動作の開始

< キャリブレーション開始処理の主な内容 >

- ・ ISCレジスタにより，RxD3端子の入力信号をタイマ入力に設定
- ・ TAU0のチャンネル7をキャプチャ・モードに設定
- ・ TAU0のチャンネル7をT107端子での両エッジ検出に設定
- ・ TAU0のチャンネル7のキャプチャ動作開始

< INTST3割り込み処理（UART3の送信完了割り込み） >

- ・ UART3送信データのカウンタ
- ・ UART3送信データの設定
- ・ UART3送信動作の停止

< INTSR3割り込み処理（UART3受信完了割り込み） >

- ・ UART3受信データのカウンタ
- ・ UART3受信データの保存
- ・ UART3受信動作の停止

< INTTM07割り込み処理（INTTM07使用，T107端子での両エッジ検出） >

- ・ TAU0のチャンネル7のキャプチャ動作停止
- ・ キャプチャ測定値から高速内蔵発振で動作するUART用転送ボー・レートの設定
- ・ TAU0のチャンネル7のキャプチャ・モードで計測したロウ・レベル間的高速内蔵発振クロックのタイマ・カウンタ値からUART3の転送ボー・レートの設定
- ・ UART3の受信動作開始

```

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

```



```

#pragma      EI                                /* EI命令を記述可能にする */
#pragma      NOP                               /* NOP命令を記述可能にする */
#pragma interrupt INTST3  fn_intst3          /* 割り込み関数宣言:INTST3 */
#pragma interrupt INTSR3  fn_intsr3         /* 割り込み関数宣言:INTSR3 */
#pragma interrupt INTTM07 fn_inttm07       /* 割り込み関数宣言:INTTM07 */

/*=====
関数プロトタイプ宣言
=====*/
static void fn_CalibrationStart(void);

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

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

static unsigned char ucCalibrationStatus; /* キャリブレーション状態 */
#define CALIBRATION_STATUS_RESET 0 /* キャリブレーション未終了状態 */
#define CALIBRATION_STATUS_END 1 /* キャリブレーション完了 */
#define CALIBRATION_STATUS_ERROR 2 /* キャリブレーション失敗 */

/*****
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
-----
高速内蔵発振クロックで動作が行えるように設定します。
-----*/

```

```

CMC = 0b00000000;          /* クロック動作モード */
/* |||||+----- AMPH: 2MHz fMX 10MHz */
/* |||+++----- <000> */
/* |+----- OSCSELS: P123/P124端子を入力ポート・モード */
/* |+----- <0> */
/* ++----- EXCLK/OSCSEL: 入力ポート・モード */

CSC = 0b11000000;          /* クロック動作ステータス制御 */
/* |||||+----- HI0STOP: 高速内蔵発振回路動作 */
/* |+----- <00000> */
/* |+----- XTSTOP: XT1発振回路動作 */
/* +----- MSTOP: X1発振回路動作 */

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

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

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

/*-----
ポート1の設定
-----*/
P1 = 0b00000000;          /* P10-P17の出力ラッチLow [後でP13(TxD0)の設定を行う] */
PM1 = 0b00000000;          /* P10-P17を出力ポートに設定 [後でP14(RxD0)の設定を行う] */

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

```

```
/*-----  
    ポート3の設定  
-----*/  
P3 = 0b00000000;    /* P30-P31の出力ラッチLow */  
PM3 = 0b11111100;    /* P30-P31を出力ポートに設定 */  
  
/*-----  
    ポート4の設定  
-----*/  
P4 = 0b00000000;    /* P40-P47の出力ラッチLow */  
PM4 = 0b00000000;    /* P40-P47を出力ポートに設定 */  
  
/*-----  
    ポート5の設定  
-----*/  
P5 = 0b00000000;    /* P50-P57の出力ラッチLow */  
PM5 = 0b00000000;    /* P50-P57を出力ポートに設定 */  
  
/*-----  
    ポート6の設定  
-----*/  
P6 = 0b00000000;    /* P60-P67の出力ラッチLow */  
PM6 = 0b00000000;    /* P60-P67を出力ポートに設定 */  
  
/*-----  
    ポート7の設定  
-----*/  
P7 = 0b00000000;    /* P70-P77の出力ラッチLow */  
PM7 = 0b00000000;    /* P70-P77を出力ポートに設定 */  
  
/*-----  
    ポート8の設定  
-----*/  
P8 = 0b00000000;    /* P80-P87の出力ラッチLow */  
PM8 = 0b00000000;    /* P80-P87を出力ポートに設定 */  
  
/*-----  
    ポート11の設定  
-----*/  
P11 = 0b00000000;    /* P110-P111の出力ラッチLow */  
PM11 = 0b11111100;    /* P110-P111を出力ポートに設定 */
```

```
/*-----
   ポート12の設定
   -----*/
```

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

```
/*-----
   ポート13の設定
   -----*/
```

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

```
/*-----
   ポート14の設定
   -----*/
```

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

```
/*-----
   ポート15の設定
   -----*/
```

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

```
/*-----
   A/Dコンバータの設定
   -----*/
```

```
ADCEN = 1;                  /* ADコンバータ使用の許可 */
```

```
ADPC = 0b00000000;          /* A/Dポート・コンフィギュレーション */
/* |||+++++----- ADPC4-0: すべてアナログ入力 */
/* +++----- <000> */
```

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

```
/*-----
   UART3の設定
   -----*/
```

```
SAU1EN = 1;                 /* UART3のあるSAU1使用許可 */
```

```

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

SPS1L = 0b00000010;                   /* プリスケーラ(動作クロック)の設定 */
/*||||++++----- PRS103-100(CK10): fCLK/2^2 */
/*++++----- PRS113-110(CK11): 未使用 */

/*-----
UART3送信設定(SAU1のチャンネル2使用)
----- */
ucTxBufferCounter = 0;                /* 送信データ・カウンタ初期化 */

SMR12 = 0b0000000000100010;          /* 動作モードなどの設定 */
/*|||||||||||||+----- MD120: 転送完了割り込み */
/*|||||||||||||++----- MD122-121: UARTモード */
/*|||||||||+++----- <100> */
/*|||||||||+----- SIS120: 立ち下がりエッジをスタートビットとして検出 */
/*|||||||||+----- <0> */
/*|||||||||+----- STS12: UART送信 */
/*|++++----- <00000> */
/*|+----- CCS12: CKS00ビットで指定した動作クロックMCK */
/*+----- CKS12: PRSで選択した動作クロックCK10(MCK) */

SCR12 = 0b1000000010010111;          /* 通信フォーマットの設定 */
/*|||||||||||||+++----- DLS122-120: 8ビット・データ長 */
/*|||||||||||||+----- <0> */
/*|||||||||++----- SLC121-120: ストップ・ビット長 = 1ビット */
/*|||||||||+----- <0> */
/*|||||||+----- DIR12: LSBファーストで出力 */
/*|||||++----- PTC121-120: パリティ・ビットを出力しない */
/*|||+----- EOC12: UART送信時はエラー割り込み未使用 */
/*|||+----- <0> */
/*|++----- DAP12/CKP12: UARTモードでは未使用 */
/*++----- TXE12/RXE12: 送信のみを行う */

/* SDR12 転送ボー・レートの設定はINTTM07割り込み処理にて設定 */

SOL1L = 0b00000000;                  /* 出力データのレベルの設定(反転なし) */

```

```

S01 = 0b0000111000001110;          /* 初期出力レベルの設定 */
/*|+----- S010: チャンネル0のシリアル・クロック出力 */
/*|+----- <1> */
/*|+----- S012: チャンネル2のシリアル・クロック出力 */
/*|++++----- <00001> */
/*|+----- CK010: チャンネル0のシリアル・データ出力 */
/*++++----- <0000111> */

SOE1L.2 = 1;                        /* SOE12: シリアル出力許可 */

/*済 PM1.3 = 0;          /* ポート1の設定にて設定済み */
P1.3 = 1;                        /* P13=TxD3 */

/*-----
UART3受信設定(SAU1のチャンネル3使用)
-----*/
ucRxBufferCounter = 0;          /* 受信データ・カウンタ初期化 */

PM1.4 = 1;                        /* P14=RxD3 */

SMR13 = 0b0000000100100010;        /* 動作モードなどの設定 */
/*|+----- MD130: 転送完了割り込み */
/*|++----- MD132-131: UARTモード */
/*|+++----- <100> */
/*|+----- SIS130: 立ち下がりエッジをスタートビットとして検出 */
/*|+----- <0> */
/*|+----- STS13: RxD端子の有効エッジ(UART受信設定) */
/*|++++----- <00000> */
/*|+----- CCS13: CKS13ビットで指定した動作クロックMCK */
/*+----- CKS13: PRSで選択した動作クロックCK10(MCK) */

SCR13 = 0b0100000010010111;        /* 通信フォーマットの設定 */
/*|+++----- DLS132-130: 8ビット・データ長 */
/*|+----- <0> */
/*|++----- SLC131-130: ストップ・ビット長 = 1ビット */
/*|+----- <0> */
/*|+----- DIR13: LSBファーストで入力を行う */
/*|++----- PTC131-130: パリティなしで受信 */
/*|+----- EOC13: UART受信のエラー割り込み許可 */
/*|+----- <0> */
/*|++----- DAP13/CKP13: UARTモードでは未使用 */
/*++----- TXE13/RXE13: 受信のみを行う */

```

```

/* SDR13 転送ボー・レートの設定はINTTM07割り込み処理にて設定 */

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

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

/*****

メイン処理

*****/
void main(void)
{
    while (1)
    {
        if((ucCalibrationStatus==CALIBRATION_STATUS_END)&&!(SE1L & 0b00001100)){
            /* キャリブレーションが完了し、UART3送受信動作を行っていない状態 */
            if(ucRxBufferCounter >= sizeof(ucRxBuffer)){
                /* UART3受信完了 */

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

                if(SSR13L & 0b00000111){      /* UART3のステータスを確認 */
                    /* (フレーミング,パリティ,オーバラン)エラーあり */
                        /*-- 受信動作開始 --*/
                            SRIF3 = 0;          /* 割り込み要求クリア */
                            SRMK3 = 0;          /* 割り込み許可 */
                            SS1L.3 = 1;        /* SS13:UART3受信動作開始 */
                        }
                    }
                else{
                    /* エラーなし */
                        /*=====
                           送信データの作成
                           =====
                        */

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

```

【受信データ】

byte	+0		+1	
+-----+-----+				
A/D値分解能の指定 A/Dチャンネルの指定				
08H: 8ビット 0:ch0 …				
0AH:10ビット … 7:ch7				
+-----+-----+				

【送信データ】

byte	+0		+1	
bit	7 6 5 4 3 2 1 0		7 6 5 4 3 2 1 0	
+---+-----+-----+				
8ビットA/D値				
(8ビット指定時)				
+---+-----+-----+				
+---+-----+-----+				
10ビットA/D値				
(10ビット指定時)				
+---+-----+-----+				

+エラー情報：受信したデータにエラーあり(1)/エラーなし(0)

===== */

/* 受信データのチェック */

```
if ((ucRxBuffer[0]==10||ucRxBuffer[0]==8)&&ucRxBuffer[1]<=7){
```

/* 受信したデータが正常な場合 */

/* A/D値の取得 */

```
ADS = ucRxBuffer[1]; /* A/D変換するチャンネルの設定 */
```

```
ADCE = 1; /* コンパレータの動作許可(変換待機) */
```

/* 1変換目(不正データ) */

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

(詳細はユーザーズ・マニュアルを参照してください。) */

```
ADIF = 0; /* 割り込み要求クリア */
```

```
ADCS = 1; /* 動作許可 */
```

```
while(!ADIF){ /* 変換待ち */
```

```
NOP();
```

```
}
```

/* 2変換目(正規データ) */


```
ADIF = 0;          /* 割り込み要求クリア */
ADCS = 1;          /* 動作許可 */
while(!ADIF){     /* 変換待ち */
    NOP();
}

ADCS = 0;          /* 変換動作終了 */
ADCE = 0;

/* 送信用データの作成 */
if(ucRxBuffer[0]==8){
/* 8ビット分解能 */
    ucTxBuffer[0] = 0;
    ucTxBuffer[1] = ADCRH;
}
else{
/* 10ビット分解能 */
    ucTxBuffer[0] = (unsigned char)(ADCR >>14);
    ucTxBuffer[1] = (unsigned char)(ADCR >>6);
    /* (10ビットA/D変換結果を下位ビット詰め) */
}
}
else{
/* 受信したデータが不正な場合 */
    ucTxBuffer[0] =0x80; /* エラー情報設定 */
    ucTxBuffer[1] =0x00;
}
/* 送信動作開始 */
STIF3 = 0;        /* 割り込み要求クリア */
STMK3 = 0;        /* 割り込み許可 */
SS1L.2 = 1;      /* SS12:UART3送信動作開始 */

TxD3 = ucTxBuffer[0]; /* データを送信データ・レジスタに設定 */
}
}
else if(ucTxBufferCounter >= sizeof(ucTxBuffer)){
/* UART3送信完了 */

ucTxBufferCounter = 0; /* 送信データ・カウンタ初期化 */

/* 受信動作開始 */
SRIF3 = 0;          /* 割り込み要求クリア */
SRMK3 = 0;          /* 割り込み許可 */
```

```
        SS1L.3 = 1;                /* SS13:UART3受信動作開始 */
    }
}
}

/*****

INTST3割り込み (UART3の送信完了割り込み)

*****/
__interrupt void fn_intst3(void)
{

    ucTxBufferCounter++;           /* 送信データ・カウンタ更新 */

    if(ucTxBufferCounter < sizeof(ucTxBuffer)){
        /* 後続データあり */
        /* 今回の送信データを送信データ・レジスタに設定 */
        TxD3 = ucTxBuffer[ucTxBufferCounter];
    }
    else{
        /* 最終データ送信終了 */
        STMK3 = 1;                /* 割り込み禁止 */
        ST1L.2 = 1;              /* ST12:UART3送信動作停止 */
    }
}

/*****

INTSR3割り込み (UART3受信完了割り込み)

*****/
__interrupt void fn_intsr3(void)
{

    ucRxBuffer[ucRxBufferCounter] = (unsigned char)SDR13;
                                   /* 今回の受信データを格納 */

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

    if(ucRxBufferCounter == sizeof(ucRxBuffer)){
        /* データ受信終了 */
```

```

SRMK3 = 1;          /* 割り込み禁止 */
ST1L.3 = 1;        /* ST13:UART3受信動作停止 */
}
}

/*****

キャリブレーション開始処理

-----

マスタ機器から入力したパルスのロウ・レベル間の高速内蔵発振クロックでの
カウント数を計測するため、入力パルス間隔測定として動作させ、割り込みを
発生させます。
*****/

static void fn_CalibrationStart(void)
{

ucCalibrationStatus = CALIBRATION_STATUS_RESET; /* キャリブレーション状態の初期化 */

ISC = 0b00000010; /* 入力切り替え制御 */
/* ||| ||| +----- ISC0: INTPO端子の入力信号を外部割り込み入力とする(通常
/* ||| |||          動作) */
/* ||| ||| +----- ISC1: RxD3端子の入力信号をタイマ入力とする(ウエイクア
/* ||| |||          ップ信号検出) */
/* +++++----- <000000> */
/*
/*
/*          注意 RxD3端子は使用する周辺の初期設定にて入力に設定済み
*/

/*-----
キャプチャの設定
-----

マスタ機器から送信される
ロウ・レベルの入力パルス幅を計測します。
-----*/

TAU0EN = 1;        /* TAU0使用許可 */

TPSOL = 0b00000010; /* タイマクロック選択 */
/* ||| +++++----- PRS003-000: fCLK(シリアル・クロックと同じにしておく) */
/* +++++----- PRS013-010: 未使用 */

```

```

TMR07 = 0b0000001010001100;          /* 動作モード設定 */
/* |||||++++----- MD073-070: キャプチャ&ワンカウント・モード */
/* |||||++----- <00> */
/* |||||++----- CIS071-070: 両エッジ(ロウ・レベル幅測定時) */
/* |||||++++----- STS072-070: TI07端子入力の両エッジを、スタート・トリガ
/* |||||
/* |||||+----- MASTER07: 単体動作 */
/* ||||+----- CCS07: CKS00ビットで指定した動作クロックMCK */
/* |+----- <0> */
/* +----- CKS07: PRSで選択した動作クロックCK00(MCK) */

TMIF07 = 0;                          /* 割り込み要求クリア */
TMMK07 = 0;                          /* 割り込み許可 */

TSOL.7 = 1;                          /* TS07:TAU0のチャンネル7のキャプチャ動作開始(トリガ動作)
*/
}

/*****

```

INTTM07割り込み処理
(INTTM07使用, TI07端子での両エッジ検出)

キャリブレーション開始処理の呼び出し後, マスタ機器から入力したパルスのロウ・レベル間の計測完了時に発生する割り込み処理です。立ち下がりエッジから立ち上がりエッジまでをTAU0のチャンネル7のキャプチャ・モードで計測したロウ・レベル間的高速内蔵発振クロックのタイマ・カウント値からUART3の転送ポー・レートの設定を行います。

```

*****/
__interrupt void fn_inttm07(void)
{
    register unsigned short ushnTimeCnt;    /* タイマ・カウンタ値 */

    TT0L.7 = 1;                          /* TS07:TAU0のチャンネル7のキャプチャ動作停止(トリガ動作) */
    ushnTimeCnt = TCR07;                  /* ロウ・レベル間的高速内蔵発振クロックでのタイマ・カウント値を取得 */

    TMMK07 = 1;                          /* 割り込み禁止 */

    if((TSR07L!=0)|| (ushnTimeCnt>0x0ff*9)){
/* オーバーフロー発生 もしくは ポー・レート用分周値の範囲外 */
        ucCalibrationStatus = CALIBRATION_STATUS_ERROR;    /* キャリブレーション失敗 */
    }
}

```

```

else{
/*-----
ポー・レートの算出式
-----

取得したタイマ・カウント値から転送ポー・レートが9600bpsとなるように
動作クロックの分周値用に変換
-----

```

【入力したパルスの波形】

- ・データ長=8ビット
- ・転送レート=9600bps
- ・データ位相=正転出力
- ・パリティ・ビット=なし
- ・ストップ・ビット=1ビット付加
- ・8ビット・データ=00H



【ポー・レートの算出式】

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

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

で表されます。(詳細はユーザーズ・マニュアルを参照してください。)

- ・CPU/周辺ハードウェア・クロック・・・fCLK [Hz]
- ・ポー・レート・・・・・・・・・・9600 [bps]

として、式から、

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

と表せます。

また，高速内蔵発振クロックの 2^2 分周で動作するTAU0のチャンネル7のキャプチャで計測した時間と，転送レートが9600bpsのデータ9ビット分の時間は等しいことから，タイマ・カウント値（ushnTimeCnt）は，

$$1/9600*9 = \{2^2/fCLK\}*ushnTimeCnt [s]$$

と表せますので，fCLKは，

$$fCLK/2^2=ushnTimeCnt*9600/9 \dots$$

となり，式から，

$$SDRmn[15:9] = \{ushnTimeCnt*9600/9\}/9600/2-1 = ushnTimeCnt/18-1$$

上記の式より，シリアル・データ・レジスタ(SDR1x)のUART3転送ボー・レート用動作クロックの分周値の算出が行えます。

```

-----*/
/* 送受信転送ボー・レートの設定 */
SDR12 = SDR13 = (((ushnTimeCnt/18)-1)<<9) & 0b1111111000000000;

/*-----
受信動作開始
-----*/
SRIF3 = 0; /* 割り込み要求クリア */
SRMK3 = 0; /* 割り込み許可 */
SS1L.3 = 1; /* SS13:UART3受信動作許可 */

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

```

付録B 改版履歴

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

【発 行】

NECエレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部1753

電話（代表）：044(435)5111

—— お問い合わせ先 ——

【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.necel.co.jp/>

【営業関係，技術関係お問い合わせ先】

半導体ホットライン

（電話：午前 9:00～12:00，午後 1:00～5:00）

電 話 : 044-435-9494

E-mail : info@necel.com

【資料請求先】

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