

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、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、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

78K0S/Kx1+

サンプル・プログラム（ソフトウェアUART半二重通信）

高速内蔵発振クロック&キャリブレーション編

この資料は、サンプル・プログラムの動作概要や使用方法、およびソフトウェア制御によるUART半二重通信の設定方法や活用方法を説明したものです。サンプル・プログラムでは、初期設定完了後にキャリブレーションを行い、ボー・レートを決定します。その後、受信テストとして8データを受信し、送信テストとしてそれらの8データを送信します。

対象デバイス

78K0S/KA1+マイクロコントローラ
 78K0S/KB1+マイクロコントローラ
 78K0S/KU1+マイクロコントローラ
 78K0S/KY1+マイクロコントローラ

目次

- 第1章 概要 ... 3
 - 1.1 初期設定の主な内容 ... 3
 - 1.2 メイン・ループ以降の内容 ... 4
- 第2章 回路図 ... 6
 - 2.1 回路図 ... 6
 - 2.2 周辺ハードウェア ... 6
- 第3章 ソフトウェアについて ... 7
 - 3.1 ファイル構成 ... 7
 - 3.2 使用する内蔵周辺機能 ... 8
 - 3.3 初期設定と動作概要 ... 8
 - 3.4 フロー・チャート ... 10
- 第4章 設定方法について ... 12
 - 4.1 ソフトウェアUARTの初期設定 ... 12
 - 4.1.1 ポートの設定 ... 12
 - 4.1.2 通信プロトコルの設定 ... 13
 - 4.2 キャリブレーション ... 14
 - 4.2.1 キャリブレーションの使用方法 ... 14
 - 4.2.2 キャリブレーションの動作概要 ... 15
 - 4.3 UART受信 ... 18
 - 4.3.1 UART受信の使用方法 ... 18
 - 4.3.2 UART受信の動作概要 ... 19
 - 4.4 UART送信 ... 22
 - 4.4.1 UART送信の使用方法 ... 22
 - 4.4.2 UART送信の動作概要 ... 23
- 第5章 システム・シミュレータ SM+での動作確認 ... 25
 - 5.1 サンプル・プログラムのビルド ... 25
 - 5.2 SM+での動作 ... 27
- 第6章 関連資料 ... 32
- 付録A プログラム・リスト ... 33
- 付録B 改版履歴 ... 46

資料番号 U18916JJ2V0AN00（第2版）
 発行年月 July 2008 NS

- 本資料に記載されている内容は2008年7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。当社製品の不具合により生じた生命、身体および財産に対する損害の危険を最小限度にするために、冗長設計、延焼対策設計、誤動作防止設計等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。

M8E 02.11

第1章 概 要

このサンプル・プログラムでは、ソフトウェアで通信タイミングの調整やポートの制御を行うことによって、マイクロコントローラのUART機能の有無に関わらず、またその機能を使用することなく、UART半二重通信を実現する例を示しています（以降、この通信を「ソフトウェアUART」と記載します）。ソフトウェアUARTは、シリアル・インタフェースUART6を搭載していない製品（78K0S/KU1+, 78K0S/KY1+）でシリアル通信を行いたい場合や、シリアル・インタフェースUART6を搭載している製品（78K0S/KA1+, 78K0S/KB1+）でシリアル通信のチャンネルを増やしたい場合に使用します。

初期設定完了後にキャリブレーションを行い、8ビット分のロウ・レベル（80H）受信により、ボー・レートを決定します。その後、受信テストとして8データを受信し、送信テストとしてそれらの8データを送信します。また、送受信中はLEDが点灯します。

1.1 初期設定の主な内容

初期設定の主な内容は、次のとおりです。

システム・クロック・ソースとして、高速内蔵発振器を選択[※]

ウォッチドッグ・タイマの動作停止

V_{LVI}（低電圧検出電圧）を4.3 V ± 0.2 Vに設定

V_{DD}（電源電圧） V_{LVI}になったあとに、V_{DD} < V_{LVI}を検出した場合、内部リセット（LVIリセット）信号を発生

CPUクロック周波数を8 MHzに設定

入出力ポートの設定

注 オプション・バイトで設定します。



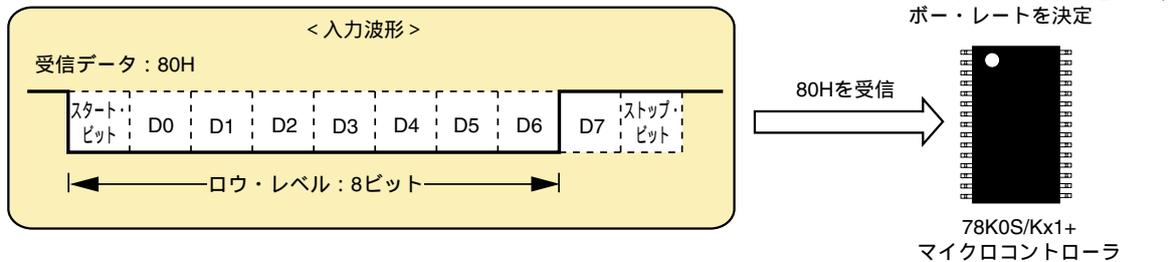
【コラム】半二重通信とは

半二重通信とは、受信動作と送信動作を交互に行う通信のことです。このサンプル・プログラムは、半二重通信方式であり、ソフトウェアにより受信動作または送信動作を行うことができます。

1.2 メイン・ループ以降の内容

初期設定完了後にキャリブレーションを行い、8ビット分のロウ・レベル（80H）受信により、ボー・レートを決定します。

キャリブレーション



キャリブレーション終了後、受信テストとして8データを受信し、送信テストとしてそれらの8データを送信します。また、送受信中はLEDが点灯します。

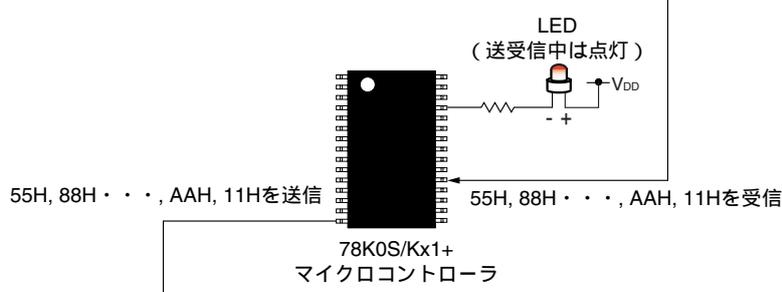
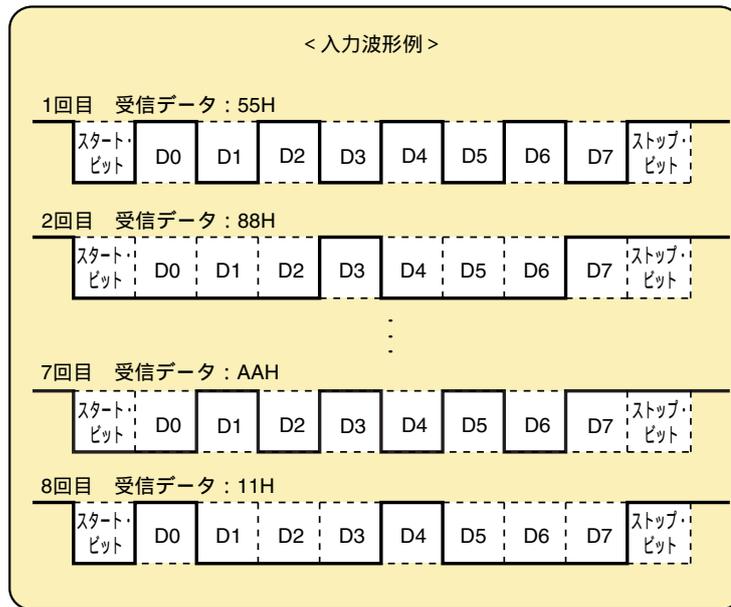
通信プロトコルは次のように設定されます。

- ・ボー・レート：4800～19200 bps^{注1}
- ・データのキャラクタ長：8ビット
- ・パリティ指定：パリティなし
- ・ストップ・ビット数：1ビットまたは2ビット^{注2}
- ・先頭ビットの指定：LSBファースト

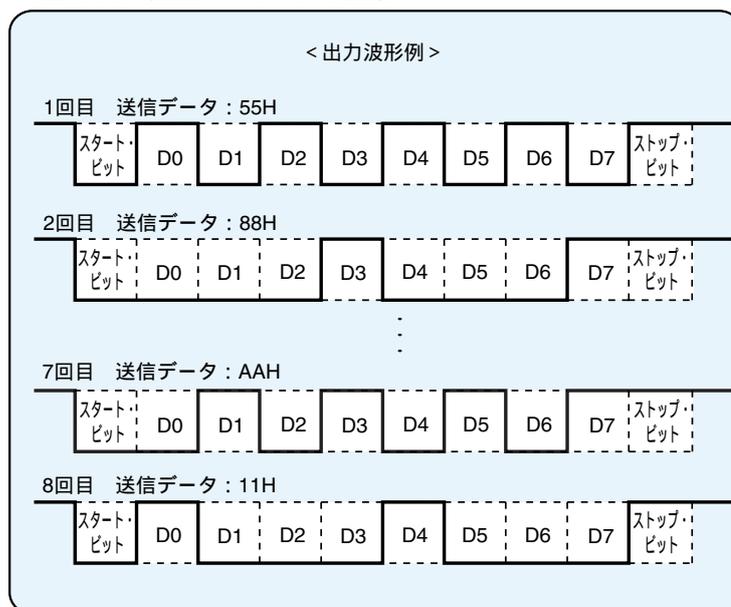
- 注1. キャリブレーションにより、ボー・レート値を決定します。キャリブレーションを行わない場合は、ソフトウェアにより設定可能です。デフォルトは9600 bpsです。
2. ソフトウェアにより設定可能です。デフォルトは1ビットです。

受信テスト 送信テスト

受信テスト (8データを受信)



送信テスト (受信した8データを送信)



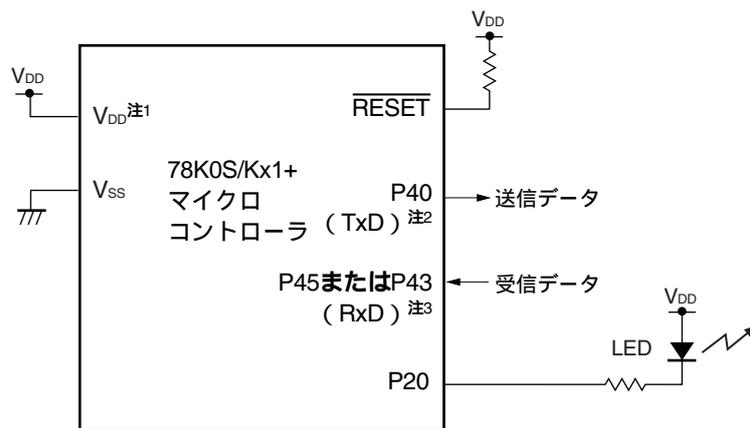
注意 デバイス使用上の注意事項については、各製品のユーザズ・マニュアル ([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)) を参照してください。

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. 4.5 V V_{DD} 5.5 Vの電圧範囲で使用してください。

2. P40端子をUART送信端子として使用します。

3. P45端子(78K0S/KA1+, 78K0S/KB1+マイクロコントローラ)またはP43端子(78K0S/KY1+, 78K0S/KU1+マイクロコントローラ)をUART受信端子として使用します。

注意1. AV_{REF} 端子は V_{DD} に直接接続してください(78K0S/KA1+, 78K0S/KB1+マイクロコントローラのみ)。

2. AV_{SS} 端子はGNDに直接接続してください(78K0S/KB1+マイクロコントローラのみ)。

3. 回路図中の端子および AV_{REF} , AV_{SS} 端子以外の未使用端子はすべて出力ポートのため、オープン(未接続)にしてください。

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。

・LED

データ受信中および送信中に点灯します。

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 ^{注1}	マイコンのハードウェア初期化処理とメイン処理のソース・ファイル			
op.asm	オプション・バイト設定用アセンブラ・ソース・ファイル (システム・クロック・ソースなどを設定)			
softuart.prw	統合開発環境 PM+用ワーク・スペース・ファイル			
softuart.prj	統合開発環境 PM+用プロジェクト・ファイル			
softuart.pri softuart.prs softuart.prm	システム・シミュレータ SM+ for 78K0S/Kx1+用プロジェクト・ファイル		注2	
softuart0.pnl	システム・シミュレータ SM+ for 78K0S/Kx1+用入出力パネル・ファイル (周辺ハードウェア動作を確認するために使用)		注2	
softuart0.wvi	システム・シミュレータ SM+ for 78K0S/Kx1+用信号データ・エディタ・ファイル (外部信号波形を入力するために使用)		注2	
softuart0.wvo	システム・シミュレータ SM+ for 78K0S/Kx1+用タイミング・チャート・ファイル (波形を確認するために使用)			

注1. ソフトウェアUARTのサンプル・プログラムは、アセンブリ言語版のみです。

2. 78K0S/KU1+マイクロコントローラには、同封されていません。

備考



: ソース・ファイルのみ同封



: 統合開発環境 PM+とシステム・シミュレータ SM+ for 78K0S/Kx1+で使用するファイルを同封



: システム・シミュレータ SM+ for 78K0S/Kx1+で使用するマイコン動作シミュレーション・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。

- ・ $V_{DD} < V_{LVI}$ 検出 : 低電圧検出 (LVI) 回路
- ・ UART受信 (RxD) : P45またはP43^注
- ・ UART送信 (TxD) : P40
- ・ LED出力 : P20

注 P45 : 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ

P43 : 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

3.3 初期設定と動作概要

このサンプル・プログラムでは、初期設定にて、低電圧検出機能の設定、クロック周波数の選択、入出力ポートの設定、ボー・レートのデフォルト値の設定などを行います。

初期設定完了後は、キャリブレーションを行い、8ビット分のロウ・レベル (80H) 受信により、ボー・レートを決定します。その後、受信テストとして8データを受信し、送信テストとしてそれらの8データを送信します。また、送受信中はLEDが点灯します。

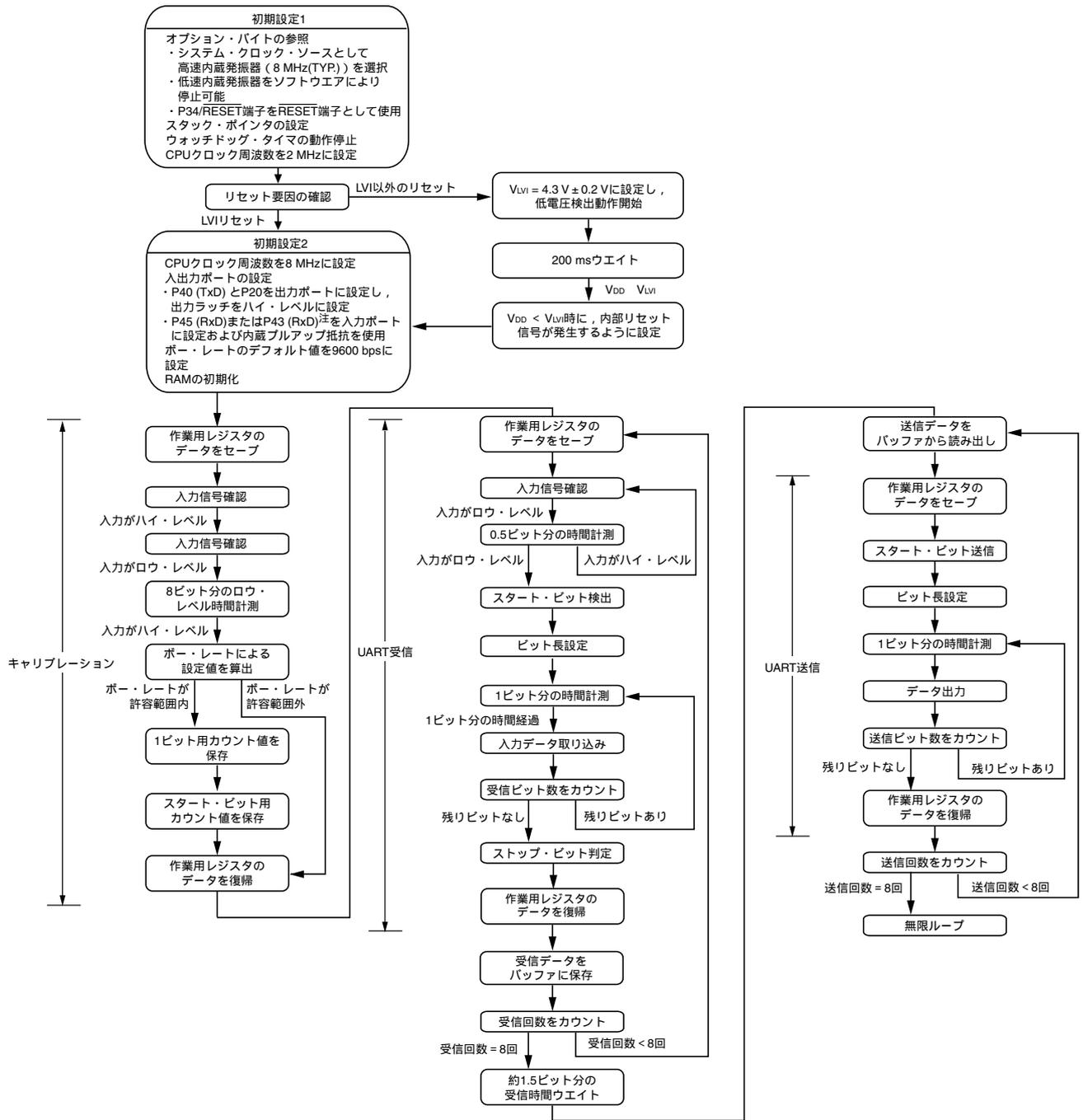
通信プロトコルは次のように設定されます。

- ・ ボー・レート : 4800 ~ 19200 bps^{注1}
- ・ データのキャラクタ長 : 8ビット
- ・ パリティ指定 : パリティなし
- ・ ストップ・ビット数 : 1ビットまたは2ビット^{注2}
- ・ 先頭ビットの指定 : LSBファースト

注1. キャリブレーションにより、ボー・レート値を決定します。キャリブレーションを行わない場合は、ソフトウェアにより設定可能です。デフォルトは9600 bpsです。

2. ソフトウェアにより設定可能です。デフォルトは1ビットです。

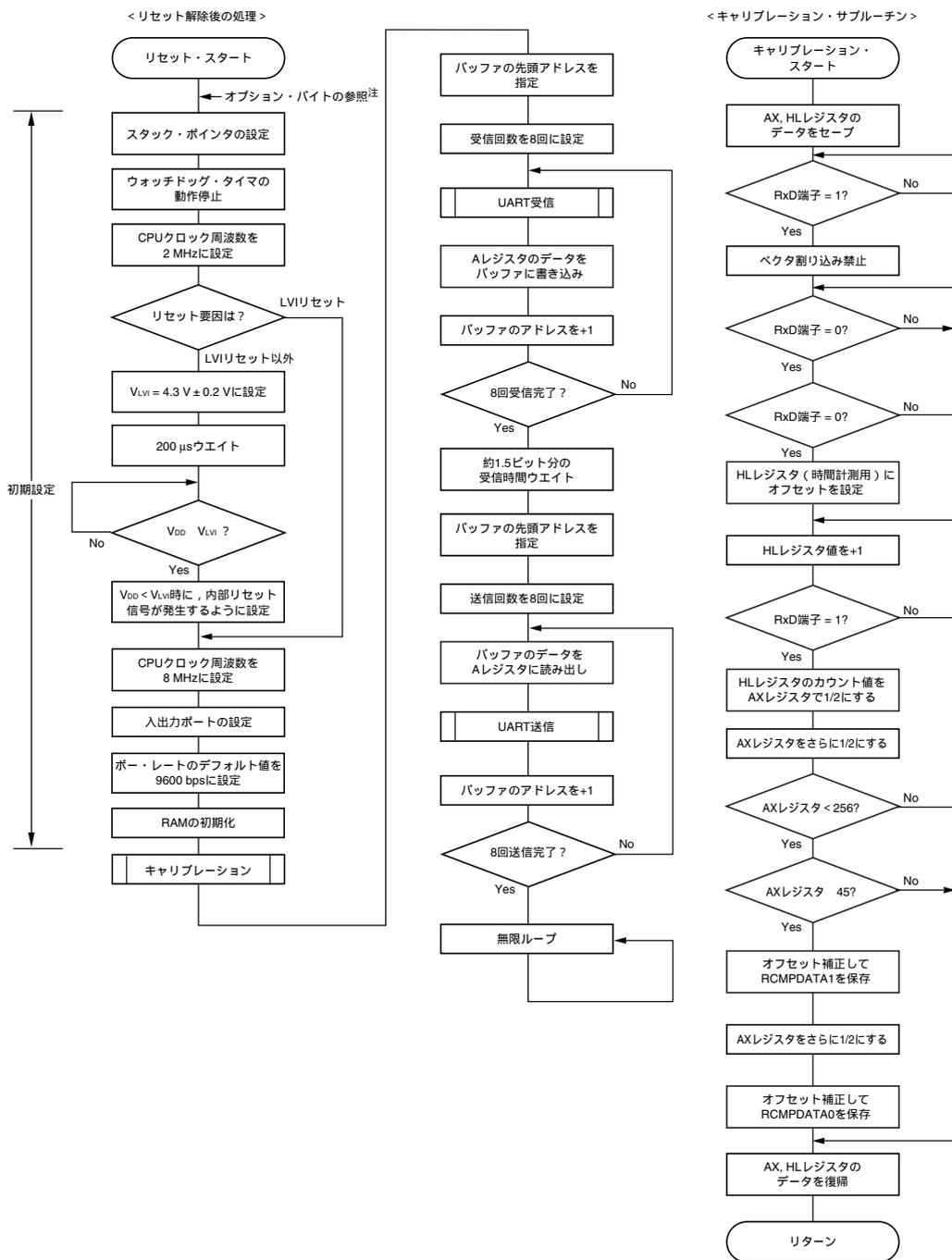
詳細については、次の状態遷移図（ステート・チャート）に示します。



注 P45 : 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ
 P43 : 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

3.4 フロー・チャート

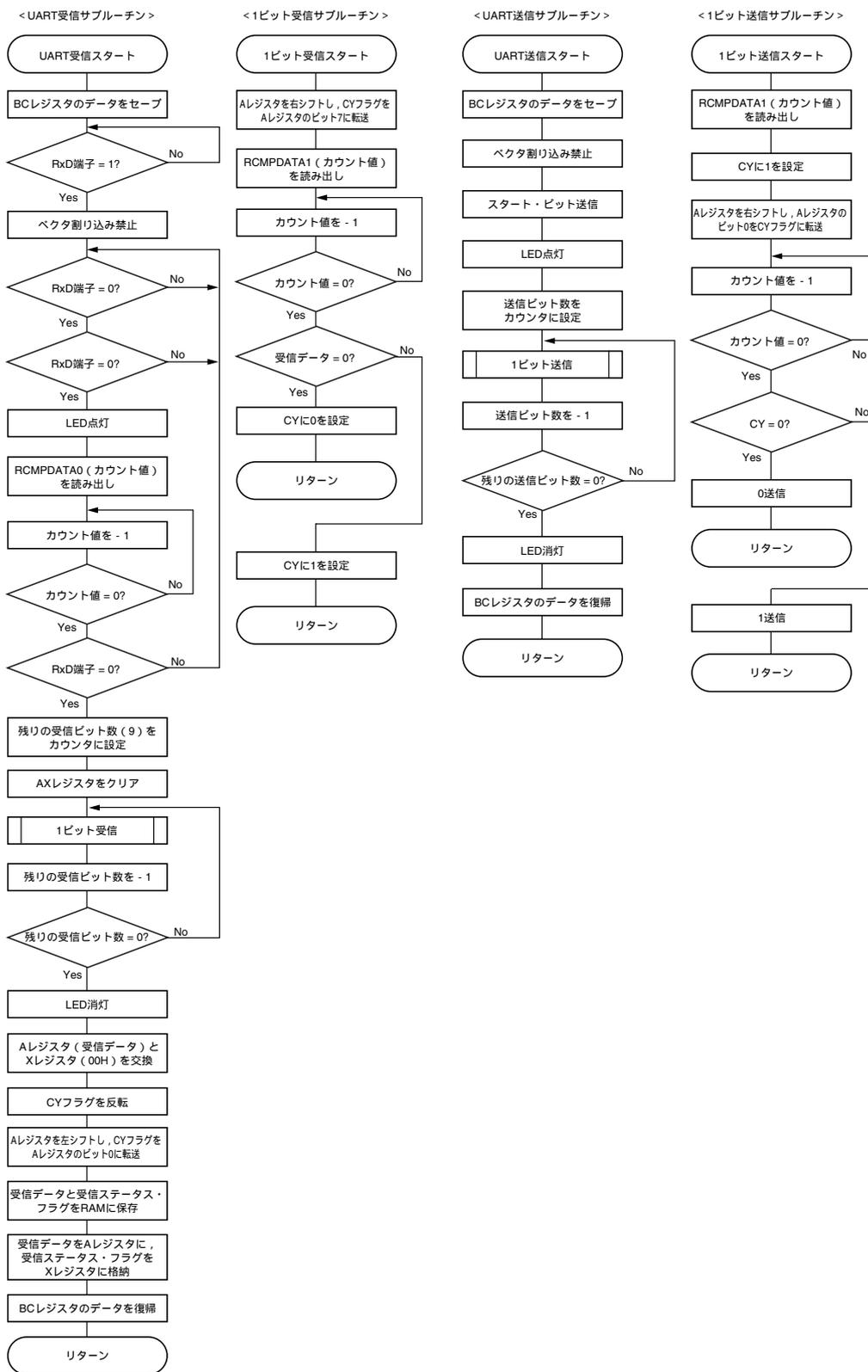
このサンプル・プログラムのフロー・チャートを次に示します。



注 オプション・バイトの参照は、リセット解除後に、マイコンが自動的に行います。このサンプル・プログラムでは、オプション・バイトの参照により、次の内容が設定されます。

- ・システム・クロック・ソースとして、高速内蔵発振クロック (8 MHz (TYP.)) を使用
- ・低速内蔵発振器をソフトウェアで停止可
- ・P34/RESET端子をRESET端子として使用

備考 <UART受信サブルーチン> , <1ビット受信サブルーチン> , <UART送信サブルーチン> , <1ビット送信サブルーチン> のフロー・チャートは、次ページにあります。



第4章 設定方法について

この章では、ソフトウェアUARTの初期設定、および、"キャリブレーション"、"UART受信"、"UART送信" の3つのサブルーチンの使用方法と動作概要について説明します。

その他の初期設定については、[78K0S/Kx1+ サンプル・プログラム\(初期設定\) LED点灯のスイッチ制御編 アプリケーション・ノート](#)を、低電圧検出(LVI)については、[78K0S/Kx1+ サンプル・プログラム\(低電圧検出\) 2.7V未満検出時リセット発生編 アプリケーション・ノート](#)を参照してください。

レジスタ設定方法の詳細については、各製品のユーザーズ・マニュアル ([78K0S/KU1+](#), [78K0S/KY1+](#), [78K0S/KA1+](#), [78K0S/KB1+](#)) を参照してください。

アセンブラ命令については、[78K0Sシリーズ 命令編 ユーザーズ・マニュアル](#)を参照してください。

4.1 ソフトウェアUARTの初期設定

ソフトウェアUART通信を使用する際の初期設定として、次の3つの内容を設定しています。

- ・ソフトウェアUART通信で使用するポート
- ・ボー・レートのデフォルト値
- ・ストップ・ビット数

4.1.1 ポートの設定

このサンプル・プログラムでは、ソフトウェアUART通信で使用する端子を次のように設定しています。

- ・UART受信(RxD) : P45 (78K0S/KA1+, 78K0S/KB1+マイクロコントローラ)
P43 (78K0S/KY1+, 78K0S/KU1+マイクロコントローラ)
- ・UART送信(TxD) : P40

リセット解除後の初期設定では、次の3つのレジスタを表のように設定しています。

- ・ポート・レジスタ : P4
- ・ポート・モード・レジスタ : PM4
- ・プルアップ抵抗オプション・レジスタ : PU4

	P4レジスタ	PM4レジスタ	PU4レジスタ
78K0S/KA1+, 78K0S/KB1+マイクロコントローラ	P40 = 1	PM40 = 0, PM45 = 1	PU45 = 1
78K0S/KY1+, 78K0S/KU1+マイクロコントローラ	P40 = 1	PM40 = 0, PM43 = 1	PU43 = 1

ソース・ファイル中では、頻繁に使用するポート・レジスタについて、次のようにシンボル定義をしています。

【本サンプル・プログラム・ソースからの抜粋(78K0S/KA1+, 78K0S/KB1+マイクロコントローラの場合)】

PTXD	EQU	P4.0	; UART送信用端子 (TxD端子)
PRXD	EQU	P4.5	; UART受信用端子 (RxD端子)

ソフトウェアUART通信では、汎用入出力ポートをUART送受信端子として使用するため、これらのポート設定を変更することにより、任意のポートで通信が可能となります。

4.1.2 通信プロトコルの設定

通信プロトコルは、次のとおりです。

- ・ボー・レート：4800～19200 bps (デフォルトは9600 bps)
- ・データのキャラクタ長：8ビット
- ・パリティ指定：パリティなし
- ・ストップ・ビット数：1ビットまたは2ビット (デフォルトは1ビット)
- ・先頭ビットの指定：LSBファースト

ボー・レートとストップ・ビット数は、ソフトウェアにより設定することができます。

ボー・レートに関しては、キャリブレーションにより、決定することもできます。

(1) ボー・レートの設定

次の2つのRAMデータで、ボー・レートを設定します。

- ・RCMPDATA1：1ビット・カウント用
- ・RCMPDATA0：スタート・ビット・カウント用

このサンプル・プログラムでは、初期設定で、このRAMデータのデフォルト値を次のように設定しています。

【本サンプル・プログラム・ソースからの抜粋】

MOV	RCMPDATA1, #CB9600	; 1ビット・カウント用タイマのデフォルト値(9600bps)
MOV	RCMPDATA0, #CHB9600	; スタート・ビット・カウント用タイマのデフォルト値(9600bps)

ボー・レート設定用の定数は、シンボル定義により次の3種類が用意されています。

ボー・レート	シンボル	
	RCMPDATA1用	RCMPDATA0用
4800 bps	CB4800	CHB4800
9600 bps (デフォルト)	CB9600 (デフォルト)	CHB9600 (デフォルト)
19200 bps	CB19200	CHB19200

キャリブレーションを行わない場合、またはキャリブレーション時にボー・レートの推奨範囲外と判定された場合は、初期設定で設定したこのRAMデータが適用されます。

(2) ストップ・ビット数の設定

このサンプル・プログラムでは、シンボル定義により、デフォルトで、ストップ・ビット数を1に設定しています。

【本サンプル・プログラム・ソースからの抜粋】

```
CSTOPBIT EQU 1 ; ストップ・ビット数の指定
```

この "1" を "2" に変更することで、送信時のストップ・ビット数を2に設定することができます。受信時は常に "ストップ・ビット数 = 1" として動作します。

4.2 キャリブレーション

4.2.1 キャリブレーションの使用方法

このサンプル・プログラムでは、キャリブレーション処理をサブルーチン化しています。次のようにコールすることにより、キャリブレーションを実行することができます。

【キャリブレーション・サブルーチンの呼び出し例】

```
CALL !SCALIB
```

このサブルーチンにより、ボー・レートを修正するために、何度でもキャリブレーションを行うことができます。

キャリブレーション処理により、次の2つの設定値がRAMに格納されます。このRAMデータは、ボー・レートにより異なります。

- ・RCMPDATA1： 1ビット・カウント用
- ・RCMPDATA0： スタート・ビット・カウント用

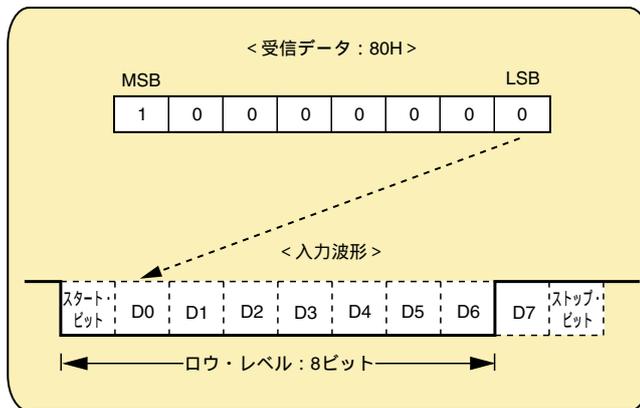
キャリブレーション時に、ボー・レートが推奨範囲から大きく外れている場合では、このRAMデータは更新されず、直前のボー・レートでのデータを保持します（このサンプル・プログラムでは、初期設定の値（ボー・レートが9600 bpsの場合の値）となります）。詳細は、4.2.2 **キャリブレーションの動作概要**を参照してください。

4.2.2 キャリブレーションの動作概要

シリアル通信の入力として使用する端子 (RxD) から8ビット分のロウ・レベル (受信データ: 80H) を受信することにより, キャリブレーションを行い, ポー・レートを決めます。

このサンプル・プログラムでは, 初期設定完了後, すぐにキャリブレーションを行います。

キャリブレーション検出用の80Hの受信データの波形は, 次のようになります。8ビット分のロウ・レベル幅は, スタート・ビットからキャラクタ・ビットのビット6までの長さになります。



ロウ・レベル幅測定開始前に, 割り込み受け付けを禁止し, 測定時に割り込み処理が発生しないように設定します。

ロウ・レベル幅測定開始時は, 約1.5 μs以下のノイズ除去を行うために, RxD端子がロウ・レベルであることを2回確認します。確認後, 通信速度に適した処理時間になるように, 命令の実行クロック数から割り出した補正値をHLレジスタに設定してから, 計測用のHLレジスタをカウント開始します。

時間計測用のHLレジスタは, 命令の実行クロック数より, 16クロックごとに1インクリメントします。

RxD端子がハイ・レベルであることを確認したら, HLレジスタをカウント終了します。

【本サンプル・プログラム・ソースからの抜粋 (キャリブレーション)】

```

JCAL000:
    BF    PRXD, $JCAL000    ; RxD端子 = 0なら1になるのを待つ
    DI                                ; ベクタ割り込み禁止

JCAL100:
    BT    PRXD, $JCAL100    ;10: キャリブレーション開始待ち
    BT    PRXD, $JCAL100    ;10: ノイズならキャリブレーション開始待ちへ戻る

    NOP                                ; 2: 時間調整用
    NOP                                ; 2: 時間調整用
    MOVW  HL, #CCALOFFSET    ; 6: 時間補正

JCAL200:
    NOP                                ; 2:
    INCW  HL                    ; 4: 時間計測
    BF    PRXD, $JCAL200    ;10: RxD端子 = 1待ち
    
```

測定開始前に, 割り込み禁止に設定 → **DI**
 ロウ・レベルを2回検出 → **BT PRXD, \$JCAL100**
 補正値を設定 → **MOVW HL, #CCALOFFSET**
 HLレジスタを+1 → **INCW HL**
 ハイ・レベル検出待ち → **BF PRXD, \$JCAL200**

2+4+10 = 16クロック
 ハイ・レベル検出するまで, NOP~BF命令を繰り返すので, HLレジスタは16クロックごとに1インクリメントする

HLレジスタのカウンタ値とボー・レート値の関係は次のようになります。

使用するクロック周波数が
8 MHzなので、8 MHz = 8×10^6 Hz

$$\text{HLレジスタのカウンタ値} = \frac{1}{\underbrace{\text{ボー・レート値}}_{(a)} \times \underbrace{8 \times 10^6}_{(b)} \times 8 \times \underbrace{\frac{1}{16}}_{(c)}}$$

(a) : 1ビット分の時間 (s)
 (b) : 1ビット分のクロック数
 (c) : 8ビット分のクロック数

HLレジスタは、命令の実行クロック
数より、16クロックごとに1インク
リメントするので、
「8ビット分のクロック数 $\times 1/16$ 」
がカウンタ値となる

$$\text{ボー・レート値} = \frac{4 \times 10^6}{\text{HLレジスタのカウンタ値}}$$

【例】HLレジスタのカウンタ値が416の場合

$$\text{ボー・レート値} = \frac{4 \times 10^6}{416} \quad 9615 \text{ [bps]}$$

HLレジスタのカウンタ値を1/4にした値から、ボー・レートが推奨範囲内であることを判定します。このとき、カウンタ値の判定値から、ボー・レートの許容範囲は、次のようになります。

$$45 \quad \frac{\text{HLレジスタのカウンタ値}}{4} < 256$$

→ ボー・レートの許容範囲 : 3907 ~ 22222 [bps]
 (ボー・レートの推奨範囲 : 4800 ~ 19200 [bps])

【本サンプル・プログラム・ソースからの抜粋（キャリブレーション結果判定～カウント値保存）】

```

MOVW  AX,  HL
CLR1  CY
RORC  A,  1           ; 結果を1/2する
XCH   A,  X
RORC  A,  1
XCH   A,  X
CLR1  CY
RORC  A,  1           ; さらに1/2する(結果を1/4に)
XCH   A,  X
RORC  A,  1
XCH   A,  X
COUNT値/4 < 256?
(3907 bps以上?)
CMP   A,  #0         ; 上位8ビットが0?
BNZ   $JCAL300      ; 遅すぎるなら抜ける
XCH   A,  X
COUNT値/4 45?
(22222 bps以下?)
CMP   A,  #45       ; 下限のチェック
BC    $JCAL300      ; 速すぎるなら抜ける
; CY=0
MOV   RCMPDATA1, A  ; RCMPDATA1設定値を保存領域に
SUB   RCMPDATA1, #CTROFFSET ; 送受信時の48クロック分を補正
RORC  A,  1         ; さらに1/2に
SUB   A,  #CSTOFFSET ; スタート・ビット処理補正
MOV   RCMPDATA0, A  ; スタート・ビット処理用に
COUNT値をさらに1/2
JCAL300:
POP   HL             ; HLレジスタのデータを復帰
POP   AX             ; AXレジスタのデータを復帰
RET
    
```

カウント値を1/2
 カウント値をさらに1/2
 カウント値/4 < 256? (3907 bps以上?)
 カウント値/4 45? (22222 bps以下?)
 「カウント値/4 - 補正值」を RCMPDATA1に保存
 「カウント値/8 - 補正值」を RCMPDATA0に保存
 ポー・レートが許容範囲外ならば、キャリブレーション終了

4.3 UART受信

4.3.1 UART受信の使用方法

このサンプル・プログラムでは、UART受信処理をサブルーチン化しています。次のように使用することにより、データ受信処理および受信データと受信ステータス・フラグの保存をすることができます。

【UART受信サブルーチンの呼び出しとデータ保存例】

CALLT [ZRXDATA]	; UART受信サブルーチン・コール
MOV RDATA, A	; 受信データをRDATAに保存
XCH A, X	; 受信ステータス・フラグをAレジスタに転送
MOV SDATA, A	; 受信ステータス・フラグをSDATAに保存

このUART受信サブルーチン・コールでは、受信データと受信ステータス・フラグを、汎用レジスタとRAM領域の両方に格納し、サブルーチンから復帰します。それぞれの対応は次のとおりです。

	保存先の汎用レジスタ	保存先のRAMアドレス
受信データ	Aレジスタ	RRXDATA (RRXDATAの下部1バイト)
受信ステータス・フラグ	Xレジスタ	RRXFLAG (RRXDATAの上部1バイト)

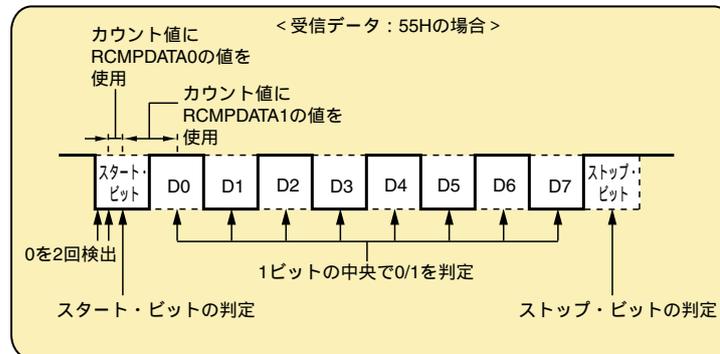
受信ステータス・フラグは、フレーミング・エラー（ストップ・ビット未検出）のみ判定することができます。

- ・受信ステータス・フラグ = 00H : 正常受信
- ・受信ステータス・フラグ = 01H : フレーミング・エラー

4.3.2 UART受信の動作概要

データ受信開始前に、割り込み受け付けを禁止し、受信時に割り込み処理が発生しないように設定します。

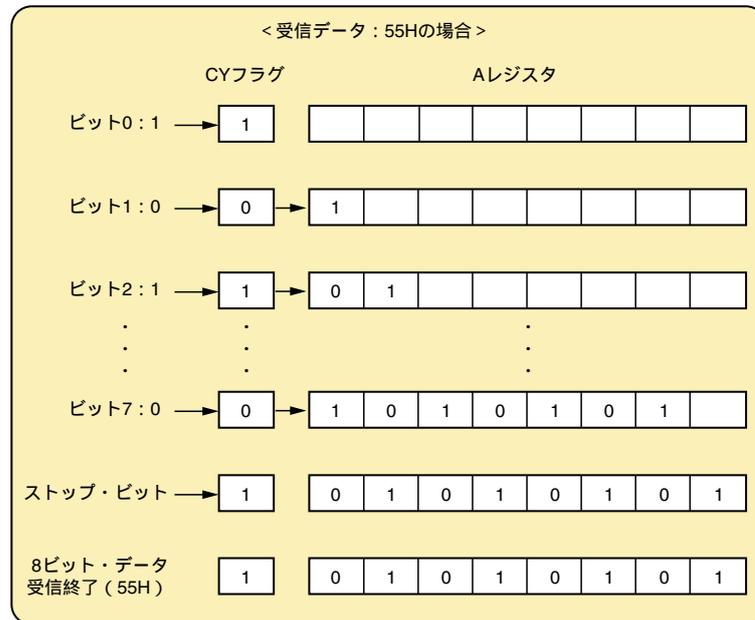
データ受信開始時は、約 $1.5\ \mu\text{s}$ 以下のノイズ除去を行うために、UART受信端子 (RxD) がロウ・レベルであることを2回確認し、確認後、ビットの中央でスタート・ビットの判定を行います。



上図のように、ビットの判定タイミングをビットの中央で行うために、スタート・ビット判定時にはキャリブレーションで決定したRCMPDATA0の値(0.5ビット分のカウント値)を、ビット0以降の判定はRCMPDATA1の値(1ビット分のカウント値)を使用し、判定のタイミングを決定します。

備考 RCMPDATA1とRCMPDATA0には、デフォルトでボー・レートが9600 bpsの場合の値が設定されています。キャリブレーションが行われない場合、またはキャリブレーションの結果、ボー・レートが許容範囲外である場合は、デフォルト値がUART送受信に使用されます。

データ受信時は、AレジスタとCYフラグを受信用のバッファ・レジスタとして使用します。1ビット受信ごとに受信データと同じ値をCYフラグに設定し、Aレジスタを右にシフトしていくことで、CYフラグのデータ（受信ビット）をAレジスタのビット7にシフトします。



8ビット・データ受信終了後、受信データはAレジスタとRRXDATA（RAM領域）に保存します。また、CYフラグの値を反転し、受信ステータス・フラグ（0：正常受信，1：フレーミング・エラー）として、XレジスタとRRXFLAG（RAM領域）に保存します。

【本サンプル・プログラム・ソースからの抜粋 (UART受信)】

```

XCALT CSEG CALLT0
ZRXDATA: DW SRXDATA ; UART受信サブルーチン
.
.
.
MMAINLOOP:
.
.
.
MOVW HL, #RRXDATABUFF ; バッファの先頭アドレスを指定
MOV B, #8 ; 受信回数を指定
LMLP100:
UART受信サブルーチン呼び出しで、データ受信
CALLT [ZRXDATA] ; データ受信
MOV [HL], A ; データをバッファへ書き込み
INC L ; バッファのアドレスを+1
DBNZ B, $LMLP100 ; 8回繰り返す
.
.
.
SRXDATA:
PUSH BC ; BCレジスタのデータをスタックへ退避
JRXD000:
受信開始前に、割り込み禁止に設定
DI ; RxD端子 = 0なら1になるのを待つ
; ベクタ割り込み禁止
JRXD100:
ロウ・レベルを2回検出
BT PRXD, $JRXD100 ; 10: スタート・ビット検出待ち
BT PRXD, $JRXD100 ; 10: ノイズならスタート・ビット検出待ちへ戻る
.
.
.
CLR1 PLED ; 6: LED点灯 (データ受信中)
NOP ; 2: 時間調整用
MOV A, RCMPDATA0 ; 4: 設定値の読み出し
MOV RTIMECNT, A ; 4: ビット中央までをセット
JRXD200:
0.5ビット分カウンタ(8クロックごとに1デクリメント)
DBNZ RTIMECNT, $JRXD200 ; 8: スタート・ビット中央待ち
BT PRXD, $JRXD100 ; 10: スタート・ビット未検出なら検出待ちへ戻る
NOP ; 2: 時間調整用
MOV B, #8+1 ; 6: 残りの受信ビット数を設定
MOVW AX, #0000H ; 6: 初期データをセット
JRXD300:
受信ビット数を設定
スタート・ビットの判定
CALL !SRXBIT ; 6: ビット受信
DBNZ B, $JRXD300 ; 6: 受信ビット数をカウント
SET1 PLED ; LED消灯 (データ受信終了)
XCH A, X ; 受信データをXレジスタに退避
NOT1 CY
ROL A, 1 ; ストップ・ビット未検出ならbit0 = 1に
MOVW RRXDATA, AX ; 受信データとエラー・ステータスを保存
XCH A, X ; 受信データをAレジスタに格納
; エラー・ステータスをXレジスタに格納
POP BC ; BCレジスタのデータを復帰
RET
.
.
.

```

UART受信サブルーチン

8クロック × RCMPDATA0値

受信データとエラー・ステータスをRAM領域に保存

備考 1ビット受信サブルーチン (SRXBIT) は次ページに続きます。

4.4.2 UART送信の動作概要

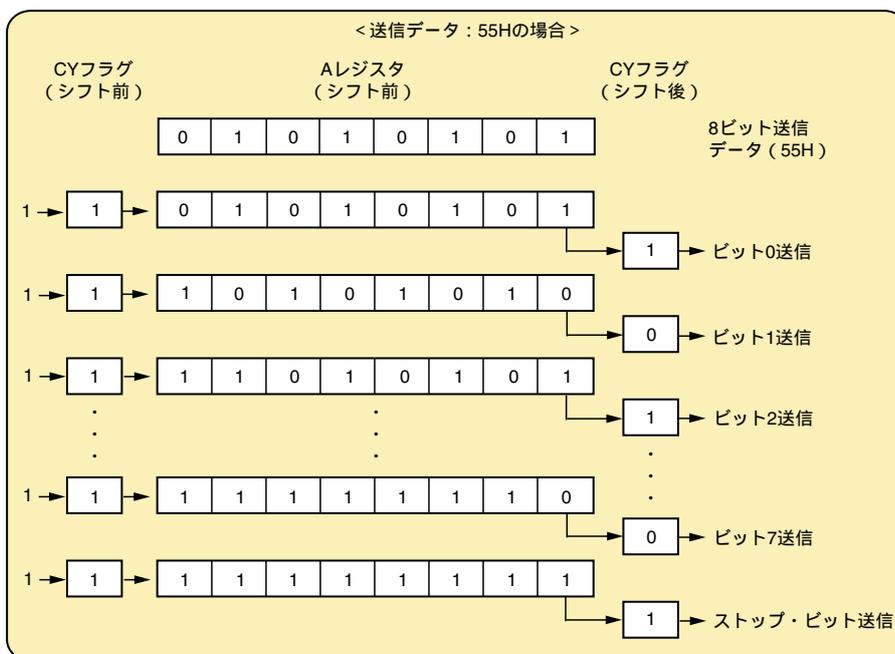
データ送信開始前に、割り込み受け付けを禁止し、送信時に割り込み処理が発生しないように設定します。

送信データはAレジスタから読み出すため、サブルーチン・コール前に送信データをAレジスタに格納しておく必要があります。

送信をする際に、キャリブレーションで決定したRCMPDATA1の値(1ビット分のカウント値)を使用し、送信する1ビットの長さを決定します。

備考 RCMPDATA1には、デフォルトでボー・レートが9600 bpsの場合の値が設定されています。キャリブレーションが行われない場合、またはキャリブレーションの結果、ボー・レートが許容範囲外である場合は、デフォルト値がUART送受信に使用されます。

データ送信時は、AレジスタとCYフラグを送信用のバッファ・レジスタとして使用します。CYフラグに1を設定し、データを右にシフトすることで、CYフラグのデータがAレジスタのビット7に、Aレジスタのビット0のデータ(送信ビット)がCYフラグにシフトし、シフト後のCYフラグと同じ値を1ビットごとに送信します。



【本サンプル・プログラム・ソースからの抜粋 (UART送信)】

```

XCALT CSEG CALLT0
ZTXDATA: DW STXDATA ; UART送信サブルーチン・
.
.
.
MMAINLOOP:
.
.
.
MOVW HL, #RRXDATA_BUFF ; バッファの先頭アドレスを指定
MOV B, #8 ; 送信回数を指定
LMLP200:
MOV A, [HL] ; バッファのデータを読み出し
CALLT [ZTXDATA] ; データ送信
INC L ; バッファのアドレスを+1
DBNZ B, $LMLP200 ; 8回繰り返す
.
.
.
STXDATA:
PUSH BC ; BCレジスタのデータをスタックへ退避
DI ; ベクタ割り込み禁止
CLR1 PTXD ; 6:スタート・ビット送信
CLR1 PLED ; 6:LED点灯(データ送信中)
MOV B, #1+8+CSTOPBIT ; 6:送信ビット数を設定
JTXD100:
CALL !STXBIT ; 6:ビット送信
DBNZ B, $JTXD100 ; 6:送信ビット数をカウント
SET1 PLED ; LED消灯(データ送信終了)
POP BC ; BCレジスタのデータを復帰
RET
.
.
.
1ビット送信用サブルーチン
STXBIT:
MOV RTIMECNT, A ; 4:送信データをセーブ
MOV A, RCMPDATA1 ; 4:送信時間カウント値をゲット
XCH A, RTIMECNT ; 6:カウント値セット&送信データ復帰
SET1 CY ; 2:出力後にデータを1にする
RORC A, 1 ; 2:送信データをCYフラグに右シフト
JTXB100:
DBNZ RTIMECNT, $JTXB100 ; 8*n:時間待ち
BC $JTXB200 ; 6:CY=1なら分岐
CLR1 PTXD ; 6:0送信
RET ; 6:
JTXB200:
SET1 PTXD ; 6:1送信
RET ; 6:
    
```

送信データをバッファから読み出し

UART送信用サブルーチン呼び出して、データ送信

送信開始前に、割り込み禁止に設定

スタート・ビットを送信

1ビット送信用サブルーチン呼び出して、1ビットずつ送信

送信ビット数を設定

1ビット送信用に設定値を読み出し

CYフラグに1を設定

1ビット分カウント(8クロックごとに1デクリメント)

送信ビット数を設定

Aレジスタのビット0をCYフラグにシフト

8クロック×RCMPDATA1

CY = 0ならば0を送信, CY = 1ならば1を送信

UART送信用サブルーチン

1ビット送信用サブルーチン

第5章 システム・シミュレータ SM+での動作確認

この章では、のアイコンを選択してダウンロードしたアセンブリ言語用のファイル（ソース・ファイル+プロジェクト・ファイル）を用い、サンプル・プログラムが、システム・シミュレータ SM+ for 78K0S/Kx1+でどのように動作するかを説明します。

注意 SM+ for 78K0S/Kx1+は、78K0S/KU1+マイクロコントローラには対応していません（2008年7月現在）。
したがって、78K0S/KU1+マイクロコントローラはSM+ for 78K0S/Kx1+で動作確認することはできません。

5.1 サンプル・プログラムのビルド

サンプル・プログラムをSM+ for 78K0S/Kx1+（以降、「SM+」と表記します）で動作確認をするために、サンプル・プログラムをビルドしてから、SM+を起動する必要があります。ここでは、サンプル・プログラムのビルド方法について、のアイコンからダウンロードしたアセンブリ言語用サンプル・プログラム（ソース・プログラム+プロジェクト・ファイル）を使用し、説明します。その他のダウンロードしたプログラムのビルド方法については、[78K0S/Kx1+ サンプル・プログラム スタートアップ・ガイド アプリケーション・ノート](#)を参照してください。

PM+操作方法の詳細については、[PM+ プロジェクト・マネージャ ユーザーズ・マニュアル](#)を参照してください。

【コラム】ビルドのエラー

PM+でビルドしているときに「A006 File not found 'C:¥NECTOOLS32¥LIB78K0S¥s0sl.rel'」または、「*** ERROR F206 Segment '@@DATA' can't allocate to memory - ignored.」というエラー・メッセージが出た場合、次の手順にてコンパイラオプションの設定を変更してください。

[ツール] [コンパイラオプションの設定]を選択してください。

[コンパイラオプションの設定]ダイアログが開いたら、「スタートアップ・ルーチン」タグを選択してください。

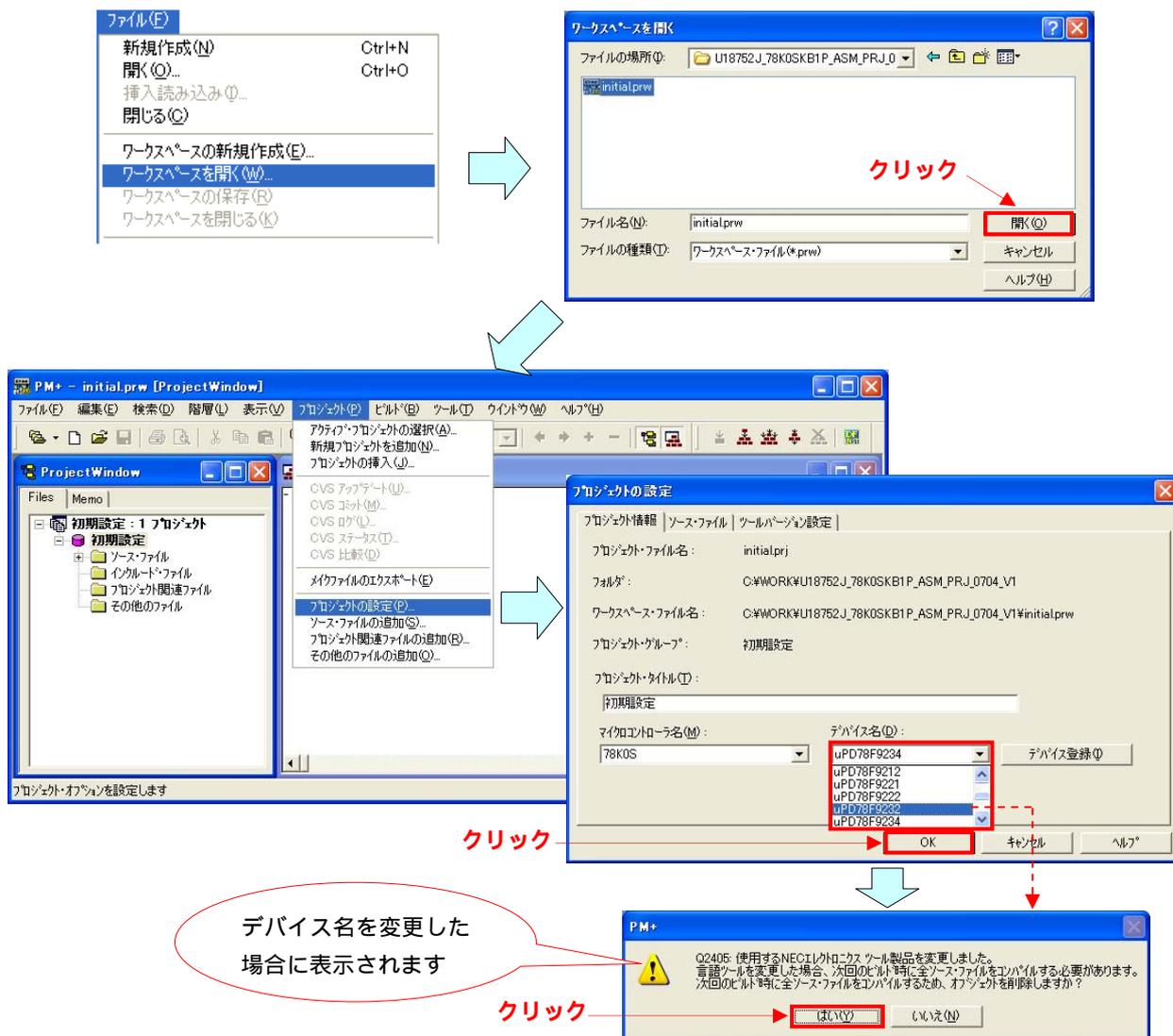
「標準ライブラリ固定領域を使用する」のチェックを外してください（それ以外のチェックは、そのまま）。

「標準ライブラリ固定領域を使用する」のチェックを外すと、標準ライブラリ固定領域として確保されていた118バイトのRAM領域が使用可能になりますが、標準ライブラリ（getchar関数やmalloc関数など）を使用できなくなります。

このサンプル・プログラムでは、のアイコンを選択してダウンロードしたファイルを使用する場合、デフォルトで「標準ライブラリ固定領域を使用する」のチェックが外されています。

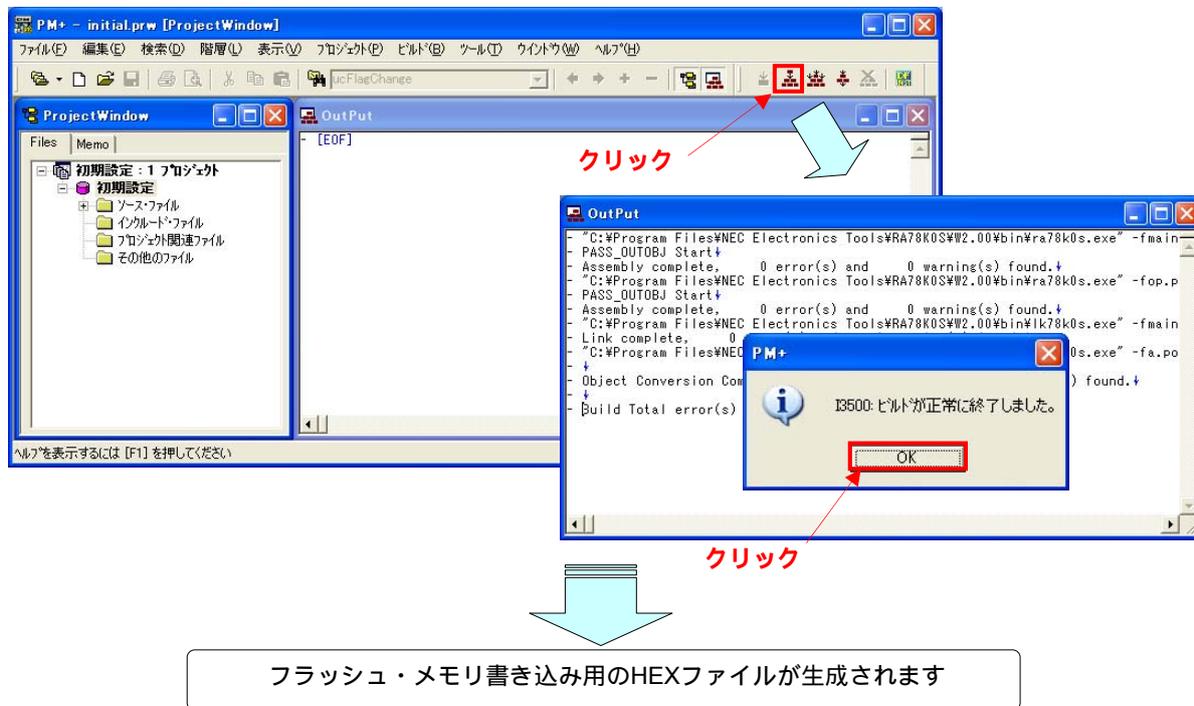
- (1) PM+を起動してください。
- (2) [ファイル] [ワークスペースを開く] から、「softuart.prw」を選択し、[開く] ボタンをクリックしてください。ワークスペースが作成され、その中にソース・ファイルが自動的に読み込まれます。
- (3) [プロジェクト] [プロジェクトの設定] を選択してください。[プロジェクトの設定] 画面が立ち上がったら、使用するデバイス名を選択(デフォルトでは、ROM/RAMサイズの最も大きいデバイスが選択)し、[OK] ボタンをクリックしてください。

備考 下の図は、「サンプル・プログラム(初期設定) LED点灯のスイッチ制御」の画面例です。



- (4)  (「ビルド」ボタン)をクリックしてください。ソース・ファイルが正常にビルドされると、「I3500: ビルドが正常に終了しました」というメッセージ画面が立ち上がります。
- (5) メッセージ画面にある [OK] ボタンをクリックしてください。フラッシュ・メモリ書き込み用のHEXファイルが作成されます。

備考 下の図は、「サンプル・プログラム(初期設定) LED点灯のスイッチ制御」の画面例です。



5.2 SM+での動作

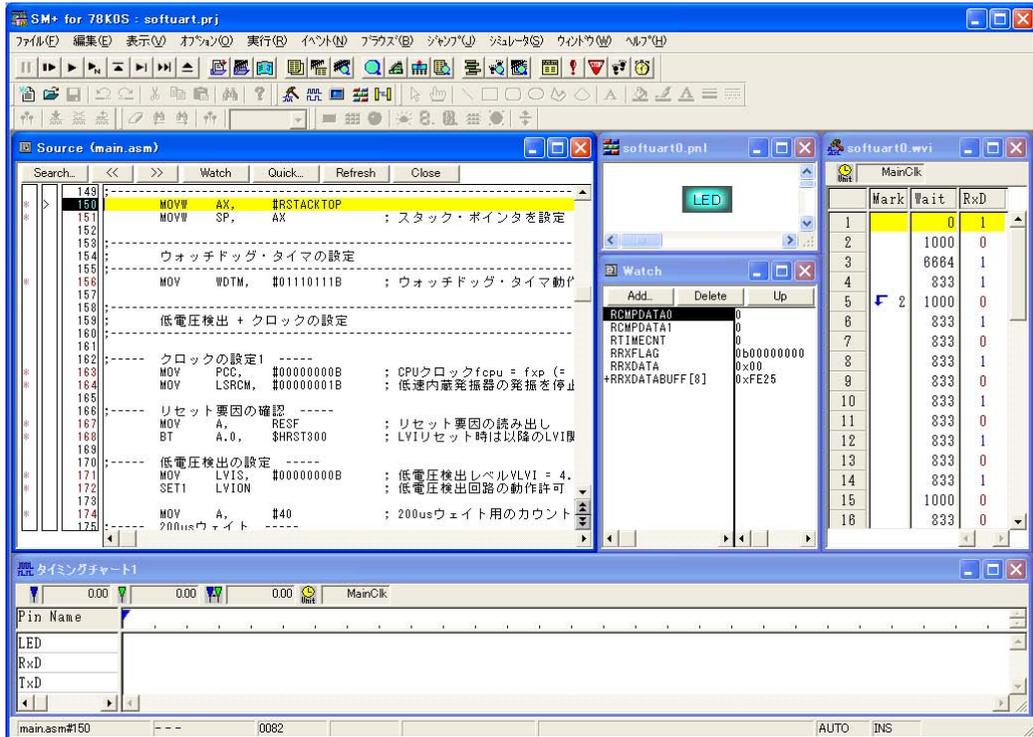
ここでは、SM+の入出力パネル・ウィンドウやタイミング・チャート・ウィンドウ上での動作確認の例を説明します。

SM+操作方法の詳細については、[SM+ システム・シミュレータ 操作編 ユーザーズ・マニュアル](#)を参照してください。

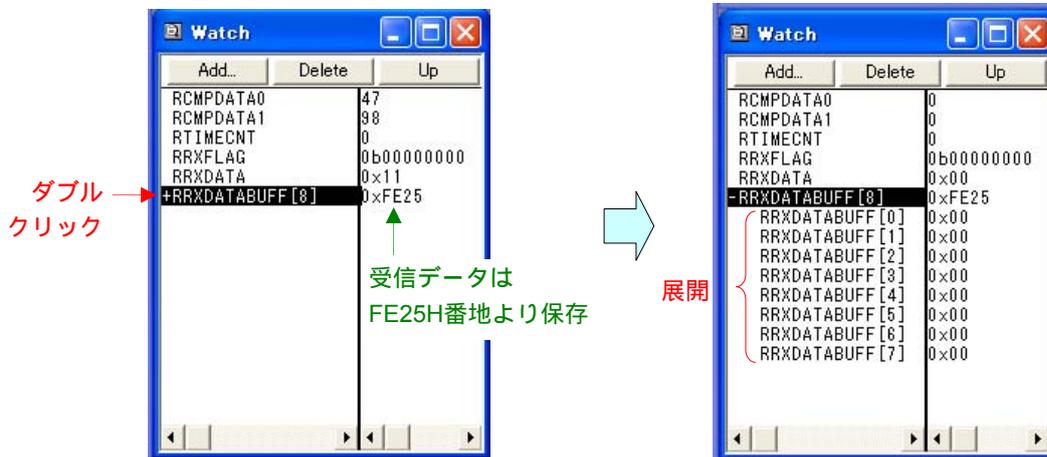
- (1) SM+ for 78K0S/Kx1+ W1.02をPM+ Ver6.30の環境で使用するために、次のサイトにあるPDFファイルを参照して、「外部ツールの登録」を行い、SM+を起動してください。

・ <http://www.necel.com/micro/ja/freesoft/pdf/ZUD-CD-07-0189.pdf>

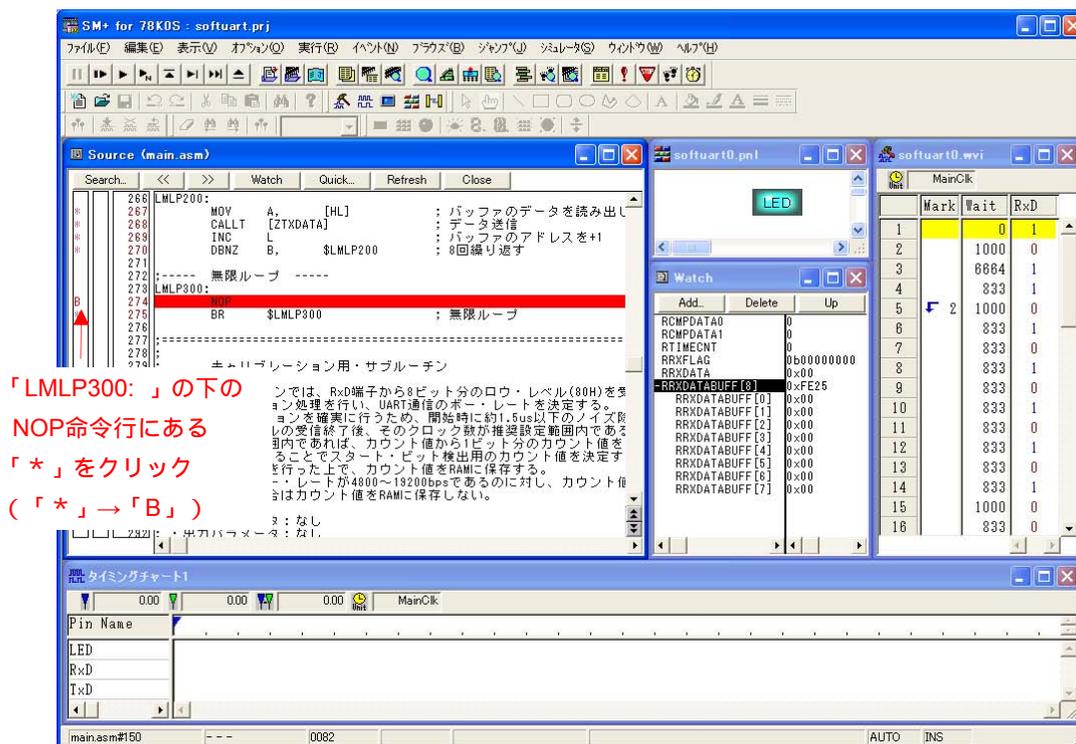
(2) SM+を起動すると、次のような画面になります。



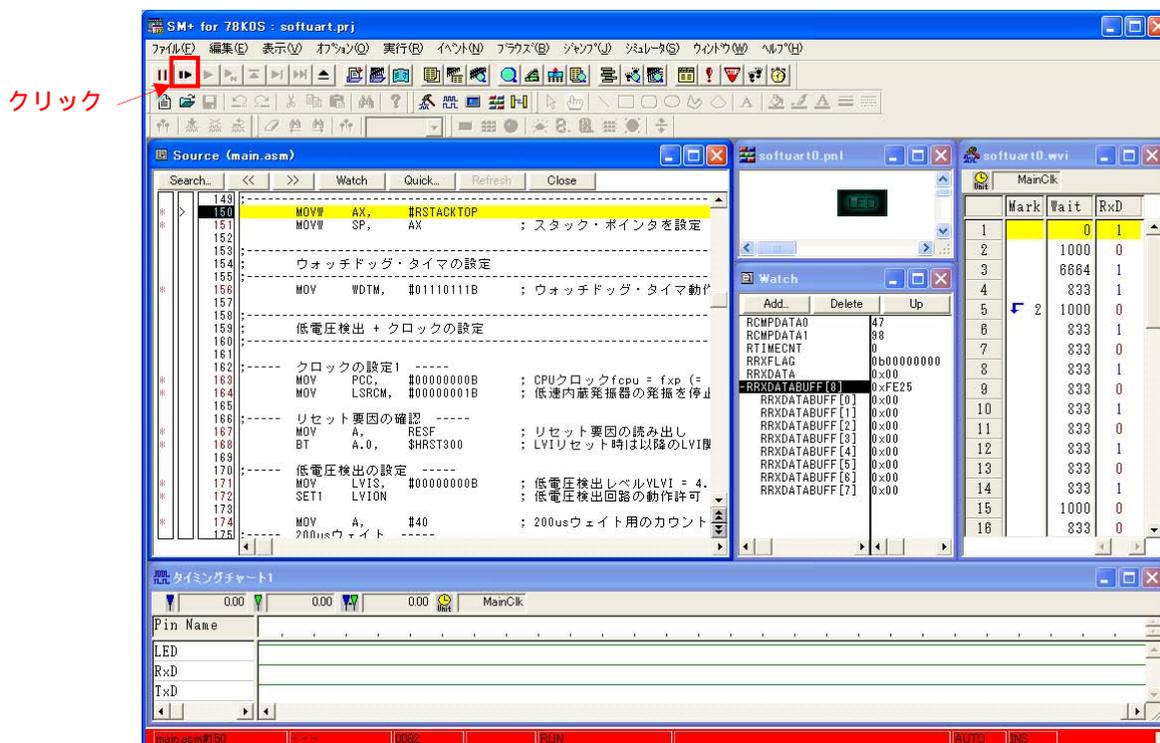
(3) ウォッチ・ウィンドウ (Watch) を選択し、「+RRXDATABUFF [8]」をダブルクリックすると、先頭文字が「+」から「-」に切り替わり、「-RRXDATABUFF [8]」の下に、保存される受信データが展開表示されます。



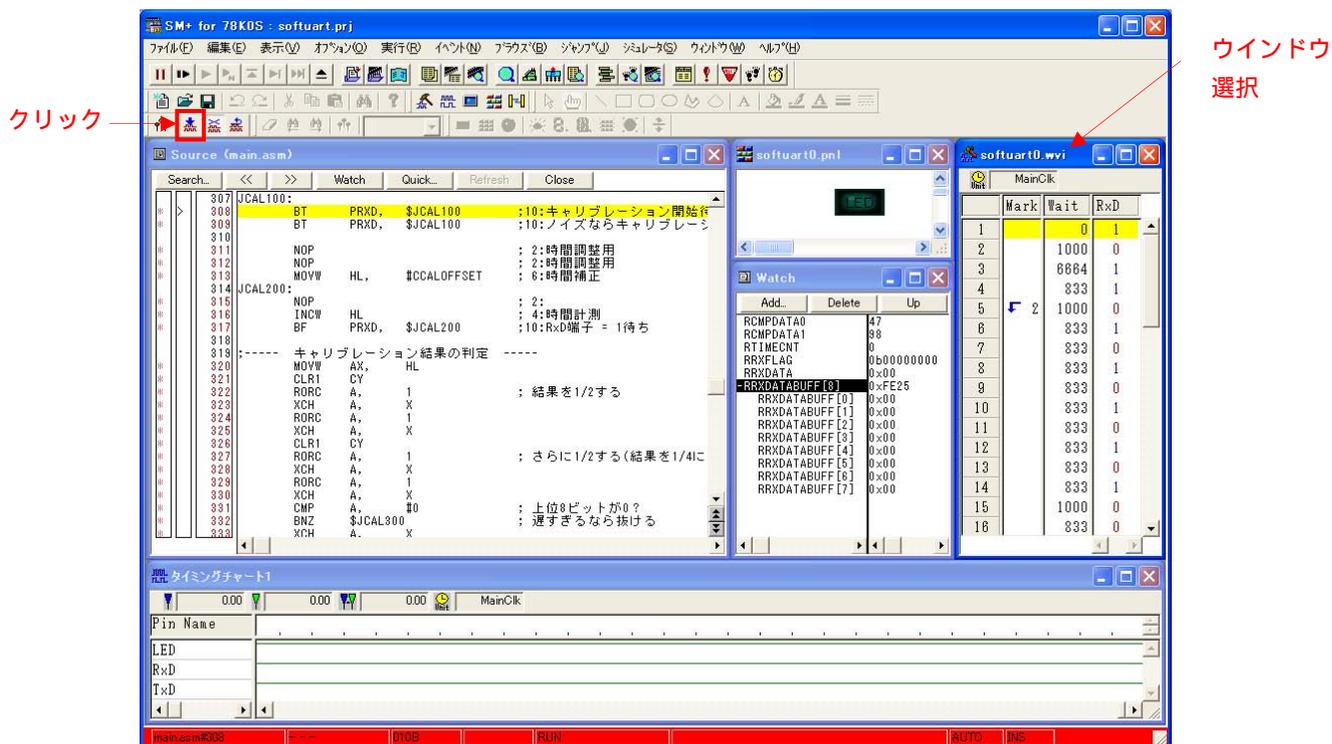
- (4) 全処理終了後にシミュレーションを停止するために、ソース・テキスト・ウィンドウ (Source (main.asm)) を選択し、ラベル" LMLP300: " の下のNOP命令行に、ブレーク・ポイントを設定してください。



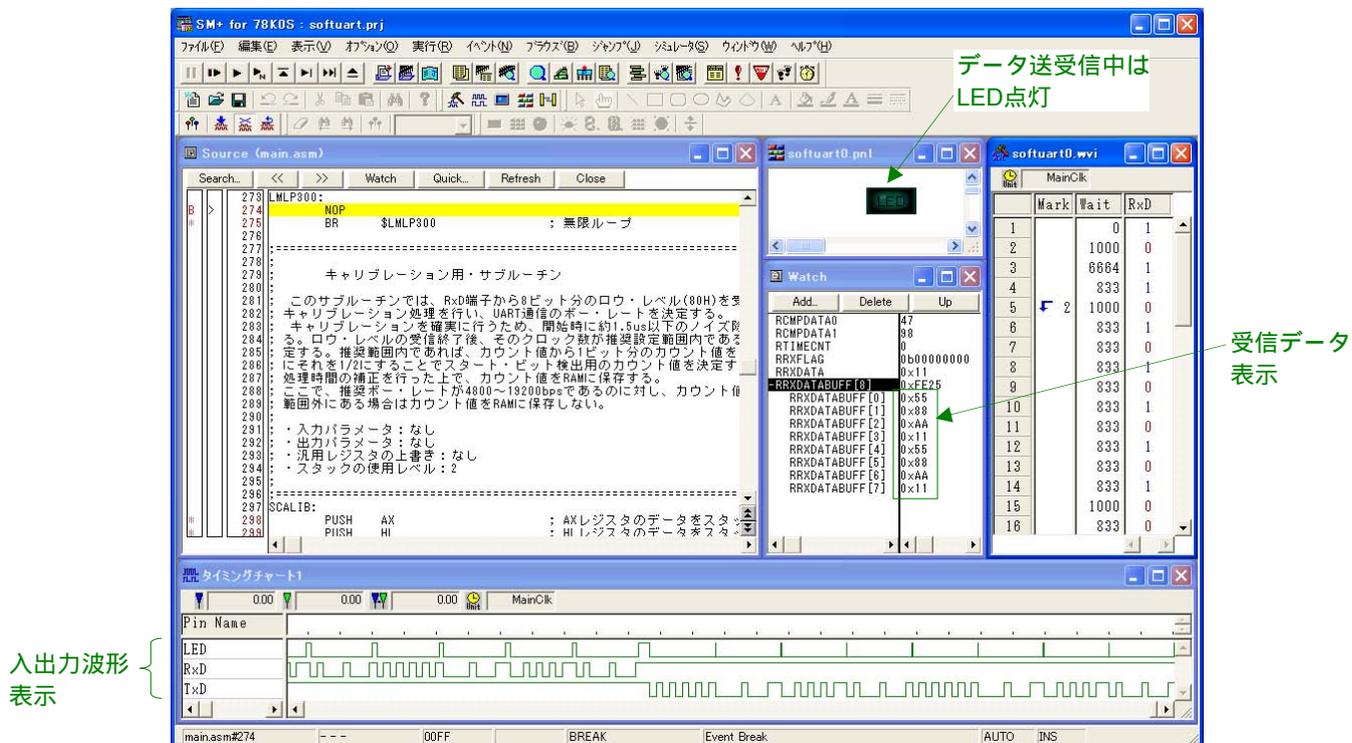
- (5)  (「リスタート」ボタン) をクリックしてください。CPUリセット後、プログラムが実行され、次のような画面になります。



(6) 信号データ・エディタ・ウィンドウ (softuart0.wvi) を選択し,  (信号入力の開始ボタン) をクリックしてください。

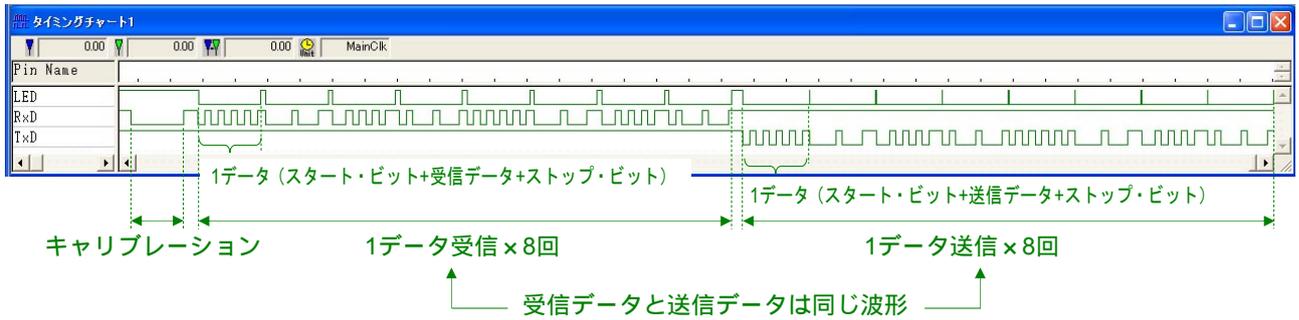


(7) ソフトウェアUARTによるデータ送受信のシミュレーションが実行され, 全処理終了後に停止します。

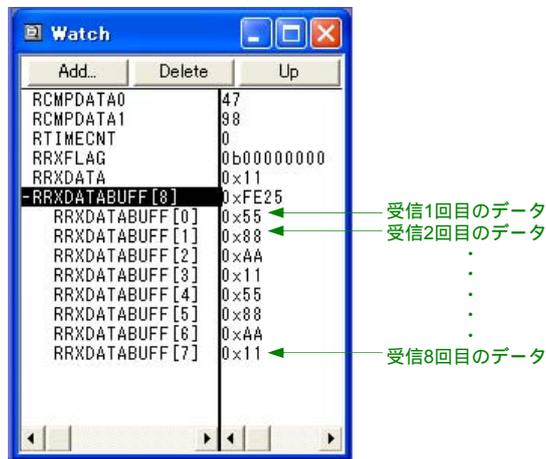


(8) タイミング・チャート・ウインドウには入出力波形が，ウォッチ・ウインドウには受信したデータが，次のように表示されます。

・タイミング・チャート・ウインドウ



・ウォッチ・ウインドウ



第6章 関連資料

資料名		和文 / 英文
78K0S/KU1+ ユーザーズ・マニュアル		PDF
78K0S/KY1+ ユーザーズ・マニュアル		PDF
78K0S/KA1+ ユーザーズ・マニュアル		PDF
78K0S/KB1+ ユーザーズ・マニュアル		PDF
78K0Sシリーズ 命令編 ユーザーズ・マニュアル		PDF
RA78K0S 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
CC78K0S 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
SM+ システム・シミュレータ 操作編 ユーザーズ・マニュアル		PDF
フラッシュ書き込み簡単マニュアル (MINICUBE2編) インフォメーション	78K0S/KU1+	PDF
	78K0S/KY1+	PDF
	78K0S/KA1+	PDF
	78K0S/KB1+	PDF
78K0S/Kx1+	サンプル・プログラム スタートアップ・ガイド	PDF
アプリケーション・ ノート	サンプル・プログラム (初期設定) LED点灯のスイッチ制御編	PDF
	サンプル・プログラム (低電圧検出) 2.7V未満検出時リセット発生編	PDF

付録A プログラム・リスト

プログラム・リスト例として、78K0S/KB1+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
;*****
;
; NEC Electronics      78K0S/KB1+シリーズ
;
;*****
; 78K0S/KB1+シリーズ      サンプル・プログラム
;*****
; ソフトウェアUART
;*****
; 【履歴】
; 2007.9.--      新規作成
;*****
;
; 【概要】
;
;本サンプルプログラムは、ソフトウェア制御によるUART通信の例を示すものである。
;初期設定完了後、8ビット分のロウ・レベル(=80H)を受信することでキャリブレーション
;を行い、ボー・レートを決定する。キャリブレーション完了後、受信テストとして8キャラクタのデータを受信し、その後、送信テストとしてその8キャラクタを送信する。受信
;を行う際は、受信端子の入力レベルを監視し、ロウ・レベルの検出をトリガとしてキャ
;リブレーションやデータ受信を開始する。ボー・レートの推奨設定範囲は4800～19200
;bpsであり、キャリブレーションを行わない場合はデフォルト9600bpsに設定している。
;また、送受信中はLEDが点灯する。
;
;
; <主な設定内容>
;
; ・ウォッチドッグ・タイマの動作停止
; ・低電圧検出電圧VLVIを4.3V±0.2Vに設定
; ・VDD VLVIとなった後にVDD < VLVIとなった場合、内部リセット信号発生(低電圧検出回路)
; ・CPUクロックを8MHzに設定
;
;
; <シリアル通信プロトコル>
;
```

```

;   ・ボー・レート           : 4800~19200bps (デフォルト9600bps)
;   ・データのキャラクタ長   : 8bit
;   ・パリティ指定           : パリティなし
;   ・ストップ・ビット数    : 1bit or 2bit (デフォルト1bit)
;   ・先頭ビットの指定      : LSBファースト
;
;
;   <受信エラーについて>
;
;   ・フレーミング・エラーのみを検出する。
;   ・パリティ・エラーやオーバラン・エラーは検出しない。
;
;
; 【ポート入出力の設定】
;
;   入力ポート : P45
;   出力ポート : P00-P03, P20-P23, P30-P33, P40-P44, P46, P47, P120-P123, P130
;   未使用のポートは全て出力ポートに設定しておく
;
;
; *****

```

```

;=====

```

シンボル定義

```

;=====
PTXD      EQU    P4.0      ; UART送信用端子(TxD端子)
PRXD      EQU    P4.5      ; UART受信用端子(RxD端子)
PLED      EQU    P2.0      ; 送受信状態表示LED用端子

CSTOPBIT  EQU    1        ; ストップ・ビット数の指定

CCALOFFSET EQU    (10+20-13)/16 ; キャリブレーション開始時の17クロック補正用
CTROFFSET  EQU    (6+18+18+6)/8  ; 送受信時の48クロック分補正用
CSTOFFSET  EQU    (10+26+5)/8    ; スタート・ビット検出時の41クロック補正用

CB4800    EQU    202        ; 4800ボーでの1ビット・カウント値
CHB4800    EQU    (CB4800+CTROFFSET)/2-CSTOFFSET ; 4800ボーでのスタート・ビット・カウント値
CB9600    EQU    98        ; 9600ボーでの1ビット・カウント値
CHB9600    EQU    (CB9600+CTROFFSET)/2-CSTOFFSET ; 9600ボーでのスタート・ビット・カウント値
CB19200   EQU    46        ; 19200ボーでの1ビット・カウント値
CHB19200   EQU    (CB19200+CTROFFSET)/2-CSTOFFSET ; 19200ボーでのスタート・ビット・カウント値

```

```

;=====
;
;   ベクタ・テーブルの設定
;
;=====

```

```

XVCT  CSEG   AT      0000H
          DW     IRESET      ;(00)  RESET
          DW     IRESET      ;(02)  --
          DW     IRESET      ;(04)  --
          DW     IRESET      ;(06)  INTLVI
          DW     IRESET      ;(08)  INTP0
          DW     IRESET      ;(0A)  INTP1
          DW     IRESET      ;(0C)  INTTMH1
          DW     IRESET      ;(0E)  INTTM000
          DW     IRESET      ;(10)  INTTM010
          DW     IRESET      ;(12)  INTAD
          DW     IRESET      ;(14)  --
          DW     IRESET      ;(16)  INTP2
          DW     IRESET      ;(18)  INTP3
          DW     IRESET      ;(1A)  INTTM80
          DW     IRESET      ;(1C)  INTSRE6
          DW     IRESET      ;(1E)  INTSR6
          DW     IRESET      ;(20)  INTST6

```

```

;=====
;
;   CALLT・テーブルの設定
;
;   頻繁に呼び出すサブルーチンは、1バイトのコール命令であるCALLT命令を
;   使用すると命令コードを短くすることが可能である。
;
;=====

```

```

XCALT CSEG   CALLTO
ZRADATA:  DW     SRADATA      ; UART受信サブルーチン
ZTXDATA:  DW     STXDATA      ; UART送信サブルーチン

```

```

;=====
;
;   RAMの定義
;
;=====

```

```

DRAM  DSEG   SADDRP

```

```

RRXDATA:    DS    1    ; 受信データ(受信ステータスとペア)
RRXFLAG:    DS    1    ; 受信ステータス・フラグ
              ; (bit0 = 1ならフレーミング・エラー)

      DSEG    SADDR
RCMPDATA0:  DS    1    ; スタート・ビット・カウント用
RCMPDATA1:  DS    1    ; 1ビット間隔カウント用
RTIMECNT:   DS    1    ; 実際のカウント用
RRXDATABUFF: DS    8    ; 送受信テスト用のデータ・バッファ

;=====
;
;   スタック領域の確保
;
;=====
DSTK  DSEG    AT      OFEE0H
RSTACKEND: DS    20H    ; スタック領域を32バイト確保
RSTACKTOP:          ; スタック領域の先頭アドレス = FF00H

;*****
;
;   リセット解除後の初期化処理
;
;*****
XMAIN CSEG    UNIT
IRESET:
;-----
;   スタック・ポインタの設定
;-----
      MOVW    AX,    #RSTACKTOP
      MOVW    SP,    AX    ; スタック・ポインタを設定

;-----
;   ウォッチドッグ・タイマの設定
;-----
      MOV     WDTM,  #01110111B    ; ウォッチドッグ・タイマ動作停止

;-----
;   低電圧検出 + クロックの設定
;-----

;----- クロックの設定1 -----
      MOV     PCC,    #00000000B    ; CPUクロックfcpu = fxp (= fx/4 = 2MHz)
      MOV     LSRM,   #00000001B    ; 低速内蔵発振器の発振を停止

```

;----- リセット要因の確認 -----

```
MOV  A,    RESF          ; リセット要因の読み出し
BT   A.0,  $HRST300     ; LVIリセット時は以降のLVI関連処理を省略し、SET_CLOCKへ
```

;----- 低電圧検出の設定 -----

```
MOV  LVIS, #00000000B   ; 低電圧検出レベルVLVI = 4.3V ± 0.2Vに設定
SET1 LVION          ; 低電圧検出回路の動作許可
```

```
MOV  A,    #40          ; 200usウェイト用のカウント値を代入
```

;----- 200usウェイト -----

HRST100:

```
DEC  A
BNZ  $HRST100          ; 0.5[us/cclk]×10[cclk]×40[count] = 200[us]
```

;----- VDD VLVI待ち処理 -----

HRST200:

```
NOP
BT   LVIF,  $HRST200   ; VDD < VLVIなら分岐

SET1 LVIMD          ; VDD < VLVI時に内部リセット信号が発生するように設定
```

;----- クロックの設定2 -----

HRST300:

```
MOV  PPCC, #00000000B   ; 周辺ハードウェアへの供給クロックfxp = fx (= 8MHz)
; -> CPUクロックfcpu = fxp = 8MHz
```

;----- ポート0の設定 -----

```
MOV  P0,    #00000000B   ; P00-P03の出力ラッチLow
MOV  PM0,   #11110000B   ; P00-P03を出力ポートに設定
```

;----- ポート2の設定 -----

```
MOV  P2,    #00000001B   ; P21-P23の出力ラッチLow、P20の出力ラッチHigh(LED消灯)
MOV  PM2,   #11110000B   ; P20-P23を出力ポートに設定
```

;----- ポート3の設定 -----

```
MOV  P3,    #00000000B   ; P30-P33の出力ラッチLow
```

```

MOV    PM3,    #11110000B    ; P30-P33を出力ポートに設定

;-----
;    ポート4の設定
;-----
MOV    P4,     #00000001B    ; P41-P47の出力ラッチLow、P40の出力ラッチHigh(シリアル送信用の
設定)
MOV    PU4,    #00100000B    ; P45に内蔵プルアップ抵抗を使用
MOV    PM4,    #00100000B    ; P40-P44,P46,P47を出力ポートに、P45(シリアル受信用)を入力ポ
トに設定

;-----
;    ポート12の設定
;-----
MOV    P12,    #00000000B    ; P120-P123の出力ラッチLow
MOV    PM12,   #11110000B    ; P120-P123を出力ポートに設定

;-----
;    ポート13の設定
;-----
MOV    P13,    #00000001B    ; P130の出力High

;-----
;    RAMの初期化
;-----
MOV    RCMPDATA1, #CB9600    ; 1ビット・カウント用タイマのデフォルト値(9600bps)
MOV    RCMPDATA0, #CHB9600   ; スタート・ビット・カウント用タイマのデフォルト値(9600bps)
MOVW   AX,     #0000H
MOVW   RRXDATA, AX           ; 受信データ&受信ステータスの初期化

;*****
;
;
;    メイン・ループ
;
;*****
MMAINLOOP:

;----- キャリブレーション -----
CALL   !SCALIB              ; キャリブレーション処理(80H受信待ち)

;----- 受信テスト -----
MOVW   HL,     #RRXDATABUFF  ; バッファの先頭アドレスを指定
MOV    B,     #8              ; 受信回数を指定

```

```

LMLP100:
    CALLT  [ZRXDATA]          ; データ受信
    MOV    [HL],  A           ; データをバッファへ書き込み
    INC    L                  ; バッファのアドレスを+1
    DBNZ   B,      $LMLP100   ; 8回繰り返す

;----- 通信相手の処理待ち -----
    MOV    A,      RCMPDATA1  ; 1ビット間隔カウント用データを読み出し
LMLP150:
    NOP                                ; このウェイト時間は通信相手の送信完了および
    NOP                                ; 受信許可までの処理時間を考慮して設定
    DEC    A
    BNZ    $LMLP150

;----- 送信テスト -----
    MOVW   HL,      #RRXDATABUFF ; バッファの先頭アドレスを指定
    MOV    B,      #8           ; 送信回数を指定
LMLP200:
    MOV    A,      [HL]        ; バッファのデータを読み出し
    CALLT  [ZTXDATA]          ; データ送信
    INC    L                  ; バッファのアドレスを+1
    DBNZ   B,      $LMLP200   ; 8回繰り返す

;----- 無限ループ -----
LMLP300:
    NOP
    BR     $LMLP300           ; 無限ループ

;=====
;
;   キャリブレーション用・サブルーチン
;
;   このサブルーチンでは、RxD端子から8ビット分のロウ・レベル(80H)を受信することで
;   キャリブレーション処理を行い、UART通信のボー・レートを決定する。
;   キャリブレーションを確実にを行うため、開始時に約1.5us以下のノイズ除去を行って
;   り。ロウ・レベルの受信終了後、そのクロック数が推奨設定範囲内であるかどうかを判
;   定する。推奨範囲内であれば、カウント値から1ビット分のカウント値を決定し、さら
;   にそれを1/2にすることでスタート・ビット検出用のカウント値を決定する。それぞれ
;   処理時間の補正を行った上で、カウント値をRAMに保存する。
;   ここで、推奨ボー・レートが4800~19200bpsであるのに対し、カウント値が大幅にその
;   範囲外にある場合はカウント値をRAMに保存しない。
;
;   ・入力パラメータ：なし

```

```

; ・出力パラメータ：なし
; ・汎用レジスタの上書き：なし
; ・スタックの使用レベル：2
;
;=====
SCALIB:
    PUSH    AX                ; AXレジスタのデータをスタックへ退避
    PUSH    HL                ; HLレジスタのデータをスタックへ退避

;----- キャリブレーション開始前処理 -----
JCAL000:
    BF      PRXD,    $JCAL000    ; RxD端子 = 0なら1になるのを待つ
    DI                          ; ベクタ割り込み禁止

;----- キャリブレーション処理 -----
JCAL100:
    BT      PRXD,    $JCAL100    ;10:キャリブレーション開始待ち
    BT      PRXD,    $JCAL100    ;10:ノイズならキャリブレーション開始待ちへ戻る

    NOP                                ; 2:時間調整用
    NOP                                ; 2:時間調整用
    MOVW   HL,      #CCALOFFSET    ; 6:時間補正
JCAL200:
    NOP                                ; 2:
    INCW   HL                          ; 4:時間計測
    BF      PRXD,    $JCAL200    ;10:RxD端子 = 1待ち

;----- キャリブレーション結果の判定 -----
    MOVW   AX,      HL
    CLR1   CY
    RORC   A,       1                ; 結果を1/2する
    XCH    A,       X
    RORC   A,       1
    XCH    A,       X
    CLR1   CY
    RORC   A,       1                ; さらに1/2する(結果を1/4に)
    XCH    A,       X
    RORC   A,       1
    XCH    A,       X
    CMP    A,       #0                ; 上位8ビットが0?
    BNZ    $JCAL300                ; 遅すぎるなら抜ける
    XCH    A,       X
    CMP    A,       #45                ; 下限のチェック

```

```

BC      $JCAL300          ; 速すぎるなら抜ける
                        ; CY=0

;----- タイマ・カウント用レジスタへの保存 -----
MOV     RCMPDATA1, A      ; RCMPDATA1設定値を保存領域に
SUB     RCMPDATA1, #CTROFFSET ; 送受信時の48クロック分を補正
RORC   A, 1              ; さらに1/2に
SUB     A, #CSTOFFSET    ; スタート・ビット処理補正
MOV     RCMPDATA0, A     ; スタート・ビット処理用に

JCAL300:
POP     HL                ; HLレジスタのデータを復帰
POP     AX                ; AXレジスタのデータを復帰
RET

;=====
;
;   UART受信用サブルーチン
;
;   このサブルーチンでは、1キャラクタ分のデータ受信処理を行う。
;   データ受信を確実にを行うため、開始時に約1.5us以下のノイズ除去を行っている。スタ
;   ート・ビット検出後、1ビット・データの中央で0/1を判定し、保存していく。1ビット
;   ・データの判定・保存は1ビット受信用サブルーチンで行い、ストップ・ビットの検出
;   を終了した時点でデータをRRXDATA(2バイト)に保存する。このとき、受信データは下位
;   1バイト(RRXDATA)に、受信ステータス・フラグは上位1バイト(RRXFLAG)に保存される。
;   また、受信データはAレジスタに、受信ステータス・フラグはXレジスタに格納してサブ
;   ルーチンから復帰するので、受信データの受け渡しをRAMとAXレジスタの両方で行うこ
;   とができる。
;
;   ・入力パラメータ：なし
;   ・出力パラメータ：Aレジスタ(受信データ)、Xレジスタ(受信ステータス・フラグ)
;   ・汎用レジスタの上書き：AXレジスタ
;   ・スタックの使用レベル：2
;
;=====
SRXDATA:
PUSH   BC                ; BCレジスタのデータをスタックへ退避

;----- 受信開始前処理 -----
JRXD000:
BF     PRXD, $JRXD000    ; RxD端子 = 0なら1になるのを待つ
DI                      ; ベクタ割り込み禁止

```

;----- スタート・ビット検出処理 -----

JRXD100:

```

BT    PRXD,  $JRXD100    ;10:スタート・ビット検出待ち
BT    PRXD,  $JRXD100    ;10:ノイズならスタート・ビット検出待ちへ戻る

CLR1  PLED                    ; 6:LED点灯(データ受信中)
NOP                                ; 2:時間調整用
MOV   A,    RCMPDATA0      ; 4:設定値の読み出し
MOV   RTIMECNT, A          ; 4:ビット中央までをセット

```

JRXD200:

```

DBNZ  RTIMECNT, $JRXD200    ; 8:スタート・ビット中央待ち
BT    PRXD,  $JRXD100    ;10:スタート・ビット未検出なら検出待ちへ戻る

NOP                                ; 2:時間調整用
MOV   B,    #8+1          ; 6:残りの受信ビット数を設定
MOVW  AX,   #0000H        ; 6:初期データをセット

```

;----- データ受信処理 -----

JRXD300:

```

CALL  !SRXBIT                ; 6:ビット受信
DBNZ  B,    $JRXD300        ; 6:受信ビット数をカウント

SET1  PLED                    ; LED消灯(データ受信終了)

```

;----- 受信データ保存処理 -----

```

XCH   A,    X                ; 受信データをXレジスタに退避
NOT1  CY
ROLC  A,    1                ; ストップ・ビット未検出ならbit0 = 1に
MOVW  RRXDATA, AX           ; 受信データとエラー・ステータスを保存
XCH   A,    X                ; 受信データをAレジスタに格納
                                ; エラー・ステータスをXレジスタに格納
POP   BC                    ; BCレジスタのデータを復帰
RET

```

; 1ビット受信用サブルーチン

- ; ;
- ; ・入力パラメータ：Aレジスタ(受信データ)、CYフラグ(受信ビット)
 - ; ・出力パラメータ：Aレジスタ(受信データ)、CYフラグ(受信ビット)
 - ; ・汎用レジスタの上書き：Aレジスタ
 - ; ・スタックの使用レベル：0
- ; ;
-

SRXBIT:

```

NOP                ; 2:送信処理との時間調整用
RORC   A,          1          ; 2:受信データ(CYフラグ)を右シフトで取り込む
MOV    RTIMECNT, A          ; 4:受信データをセーブ
MOV    A,          RCMPDATA1 ; 4:受信時間カウント値をゲット
XCH   A,          RTIMECNT   ; 6:カウント値セット&受信データ復帰

```

JRXB100:

```

DBNZ   RTIMECNT, $JRXB100    ; 8*n:時間待ち

BT     PRXD,  $JRXB200       ;10:受信データを確認
CLR1   CY                               ; 2:0受信ならCY=0
RET                                         ; 6:

```

JRXB200:

```

SET1   CY                               ; 2:1受信ならCY=1
RET                                         ; 6:

```

=====

```

;
;   UART送信用サブルーチン
;
;   このサブルーチンでは、1キャラクタ分のデータ送信を行う。
;   下記の例のように、送信したいデータをAレジスタに格納してこのサブルーチン呼び
;   出す。データの送信は1ビット送信用サブルーチンで行い、ストップ・ビットの送信を
;   完了した時点でこのサブルーチンから復帰する。

```

; プログラム例 :

```

;   MOV    A,          #54H    ; 54HをAレジスタに格納
;   CALLT  [ZTXDATA]         ; UART送信用サブルーチン呼び出し

```

- ; ・入力パラメータ : Aレジスタ(送信データ)
- ; ・出力パラメータ : なし
- ; ・汎用レジスタの上書き : Aレジスタ
- ; ・スタックの使用レベル : 2

=====

STXDATA:

```

PUSH   BC                ; BCレジスタのデータをスタックへ退避

```

;----- 送信開始前処理 -----

```

DI                ; ベクタ割り込み禁止

```

;----- スタート・ビット送信処理 -----

```

CLR1   PTXD           ; 6:スタート・ビット送信

```

```

CLR1  PLED                ; 6:LED点灯(データ送信中)
MOV   B, #1+8+CSTOPBIT   ; 6:送信ビット数を設定

;----- データ送信処理 -----
JTXD100:
CALL  !STXBIT            ; 6:ビット送信
DBNZ  B, $JTXD100       ; 6:送信ビット数をカウント

SET1  PLED              ; LED消灯(データ送信終了)

POP   BC                ; BCレジスタのデータを復帰
RET

;-----
;      1ビット送信用サブルーチン
;
; ・入力パラメータ：Aレジスタ(送信データ)
; ・出力パラメータ：Aレジスタ(送信データ)
; ・汎用レジスタの上書き：Aレジスタ
; ・スタックの使用レベル：0
;
;-----
STXBIT:
MOV   RTIMECNT, A       ; 4:送信データをセーブ
MOV   A, RCMPDATA1     ; 4:送信時間カウント値をゲット
XCH  A, RTIMECNT       ; 6:カウント値セット&送信データ復帰
SET1  CY                ; 2:出力後にデータを1にする
RORC  A, 1              ; 2:送信データをCYフラグに右シフト
JTXB100:
DBNZ  RTIMECNT, $JTXB100 ; 8*n:時間待ち

BC    $JTXB200         ; 6:CY=1なら分岐
CLR1  PTXD             ; 6:0送信
RET   ; 6:
JTXB200:
SET1  PTXD             ; 6:1送信
RET   ; 6:

end

```

```

op.asm
;=====
;
; オプション・バイトの設定
;
;=====
OPBT CSEG AT 0080H
      DB 10011100B ; オプション・バイトの設定
;
;          ||||
;          |||+----- 低速内蔵発振器はソフトウェアで停止可能
;          |++----- 高速内蔵発振クロック(8MHz)を使用
;          +----- P34/RESET端子をリセット端子として使用

      DB 11111111B ; プロテクト・バイトの設定(セルフプログラミング用)
;
;          |||||
;          +----- 全てのブロックへの書き込み許可

end

```

付録B 改版履歴

本文欄外の 印は、本版で改訂された主な箇所を示しています。この" "をPDF上でコピーして「検索する文字列」に指定することによって、改版箇所を容易に検索できます。

版 数	発行年月	改版箇所	改版内容
第1版	October 2007	-	-
第2版	July 2008	p.25	第5章 システム・シミュレータ SM+での動作確認 ・注意文中の（2007年10月現在）を（2008年7月現在）に変更
		pp.25-27	5.1 サンプル・プログラムのビルドを変更
		p.27	5.2 SM+での動作 ・（1）を追加
		p.32	第6章 関連資料 ・フラッシュ書き込み簡単マニュアル（MINICUBE2編） インフォメーションを追加

【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。
