

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

## 78K0/Lx3

### サンプル・プログラム (リアルタイム・カウンタ)

### 低電圧時のリアルタイム・カウンタ動作継続編

この資料は、サンプル・プログラムの動作概要や使用方法について説明したものです。本サンプル・プログラムは、アプリケーション上の電源がOFF (バッテリー取り外し時などによる) した場合にも、電源-グランド間のスーパー・キャパシタに保持している電荷により、低電圧状態においても、リアルタイム・カウンタの動作を継続するものです。また、低電圧状態から通常電圧状態に復帰後 (バッテリー交換後など)、リアルタイム・カウンタが動作継続した結果であるカウンタ値をUART通信にて送信します。低電圧状態から通常電圧状態に復帰時に、時計データを再設定することなくカウンタ動作を継続動作することを目的としています。

#### 対象デバイス

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

#### 目次

- 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 初期設定と動作概要 ... 10
  - 3.4 UART送信データのフォーマット ... 12
  - 3.5 フロー・チャート ... 13
- 第4章 設定方法について ... 16
  - 4.1 使用する周辺の初期設定 ... 16
  - 4.2 メイン処理 ... 28
  - 4.3 低電圧検出割り込み処理 (INTLVI使用) ... 31
  - 4.4 UART送信処理 ... 33
- 第5章 関連資料 ... 38
- 付録A プログラム・リスト ... 39
- 付録B 改版履歴 ... 70

資料番号 U19541JJ1V0AN00 (第1版)  
 発行年月 November 2008 NS

- 本資料に記載されている内容は2008年11月現在のもので、今後、予告なく変更することがあります。量産設計の際には最新の個別データ・シート等をご参照ください。
  - 文書による当社の事前の承諾なしに本資料の転載複製を禁じます。当社は、本資料の誤りに関し、一切その責を負いません。
  - 当社は、本資料に記載された当社製品の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、一切その責を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
  - 本資料に記載された回路、ソフトウ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いません。当社製品をお客様の機器にご使用の際には、当社製品の不具合の結果として、生命、身体および財産に対する損害や社会的損害を生じさせないよう、お客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計を行ってください。
  - 当社は、当社製品の品質水準を「標準水準」、「特別水準」およびお客様に品質保証プログラムを指定していただく「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。
    - 標準水準：コンピュータ、OA機器、通信機器、計測機器、AV機器、家電、工作機械、パーソナル機器、産業用ロボット
    - 特別水準：輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器
    - 特定水準：航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器、生命維持のための装置またはシステム等
- 当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。意図されていない用途で当社製品の使用をお客様が希望する場合には、事前に当社販売窓口までお問い合わせください。

(注)

- (1) 本事項において使用されている「当社」とは、NECエレクトロニクス株式会社およびNECエレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいう。
- (2) 本事項において使用されている「当社製品」とは、(1)において定義された当社の開発、製造製品をいう。

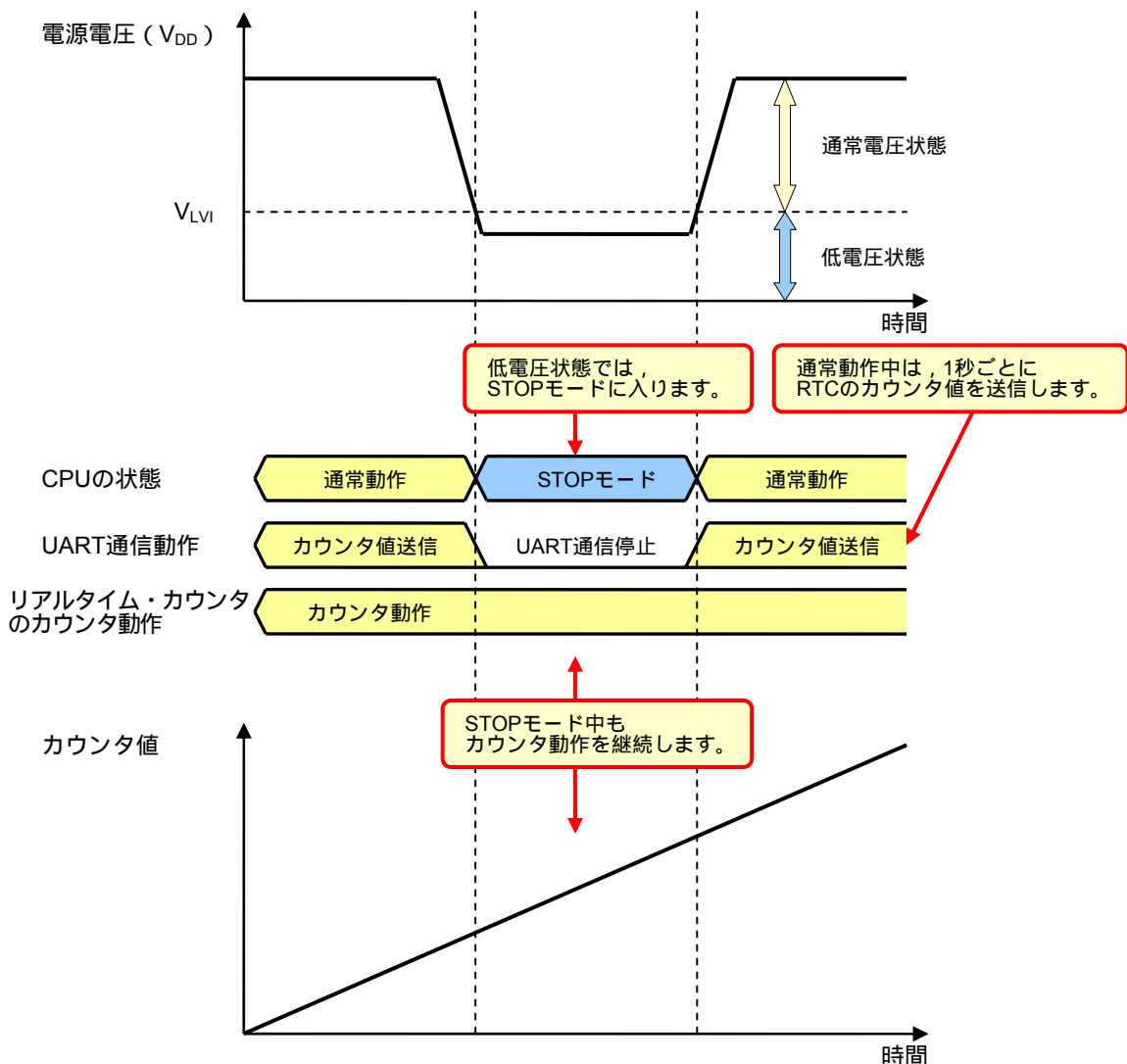
# 第1章 概 要

本サンプル・プログラムは、アプリケーション上の電源がOFF（バッテリー取り外し時などによる）した場合にも、電源-グランド間のスーパー・キャパシタに保持している電荷により、低電圧状態においても、リアルタイム・カウンタの動作を継続するものです。また、低電圧状態から通常電圧状態に復帰後（バッテリー交換後など）、リアルタイム・カウンタが動作継続した結果であるカウンタ値をUART通信にて送信します。低電圧状態から通常電圧状態に復帰時に、時計データを再設定することなくカウンタ動作を継続動作することを目的としています。

メイン処理では、リアルタイム・カウンタのカウンタ値（年、月、日、曜日、時、分、秒）を読み出し、UART送信処理を呼び出します。

低電圧状態は、低電圧検出回路により検出します。電源電圧（ $V_{DD}$ ）の電圧が検出電圧（ $V_{LVI}$ ）より低くなった場合、または $V_{LVI}$ 以上になった場合に割り込み信号（INTLVI）を発生します。低電圧状態を検出した場合は、割り込み処理にてSTOPモードに入ります。低電圧状態から復帰後に割り込み処理を終了します。

UART送信処理ではシリアル・インタフェースUART6にて、リアルタイム・カウンタのカウンタ値（年、月、日、曜日、時、分、秒）を送信します。



## (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

レジスタ・バンクの設定

スタック・ポインタの設定

ROM/RAMサイズの設定

ポートの設定

CPUクロックを高速内蔵発振動作（8 MHz）に設定

周辺ハードウェア・クロックを高速内蔵発振動作（8 MHz）に設定

低電圧検出回路を電源電圧（V<sub>DD</sub>）のレベル検出により割り込み信号を発生するように設定

シリアル・インタフェースUART6をデータ送信用に設定

リアルタイム・カウンタを定周期割り込みに設定し、カウンタ値を初期化

INTLVI割り込みマスク解除

割り込みの許可

## (2) メイン処理の内容

リアルタイム・カウンタからのカウンタ値（年，月，日，曜日，時，分，秒）読み出しと，UART送信処理の呼び出しを行います。

処理を行うタイミングは，リアルタイム・カウンタの1秒に1度の定周期割り込みにより作成します。

リアルタイム・カウンタからのカウンタ値の読み出し処理では，リアルタイム・カウンタのウェイト制御をカウンタ値読み出し，書き込みモードに設定し，カウンタ動作を停止させ，カウンタ値（年，月，日，曜日，時，分，秒）を読み出します。カウンタ値（年，月，日，曜日，時，分，秒）の読み出し完了後にリアルタイム・カウンタのウェイト制御をカウンタ動作に設定し，動作を再開させます。

カウンタ値読み出し，書き込みモード中は，カウント・レジスタのカウンタ動作が停止します。そのため，カウンタ値読み出し，書き込みモード中にSTOPモードに入ってしまうと，STOPモード中にカウンタ動作ができなくなります。カウンタ動作の停止中にSTOPモードに入らないように，リアルタイム・カウンタからのカウンタ値読み出し，書き込みモード中は割り込みを禁止します。

### (3) UART送信処理の内容

リアルタイム・カウンタから読み出したカウンタ値(年,月,日,曜日,時,分,秒)をASCIIコードに変換し,シリアル・インタフェースUART6にて送信する処理です。

送信動作を行う前に低電圧状態を検出した場合は,シリアル・インタフェースUART6を再スタートさせます。

送信動作中に低電圧状態を検出した場合は,送信動作を中断します。

UART通信の設定内容詳細,および送信データの詳細については,3.4 UART送信データのフォーマットを参照してください。

### (4) INTLVI割り込み処理(低電圧検出割り込み)の内容

低電圧状態を検出した場合に行う割り込み処理です。

割り込み信号(INTLVI)は,電圧降下時に電源電圧( $V_{DD}$ ) < 検出電圧( $V_{LVI}$ )になったとき,または,電圧上昇時に $V_{DD} < V_{LVI}$ となった場合に発生します。 $V_{LVI}$ は $1.93V \pm 0.1V$ に設定します。

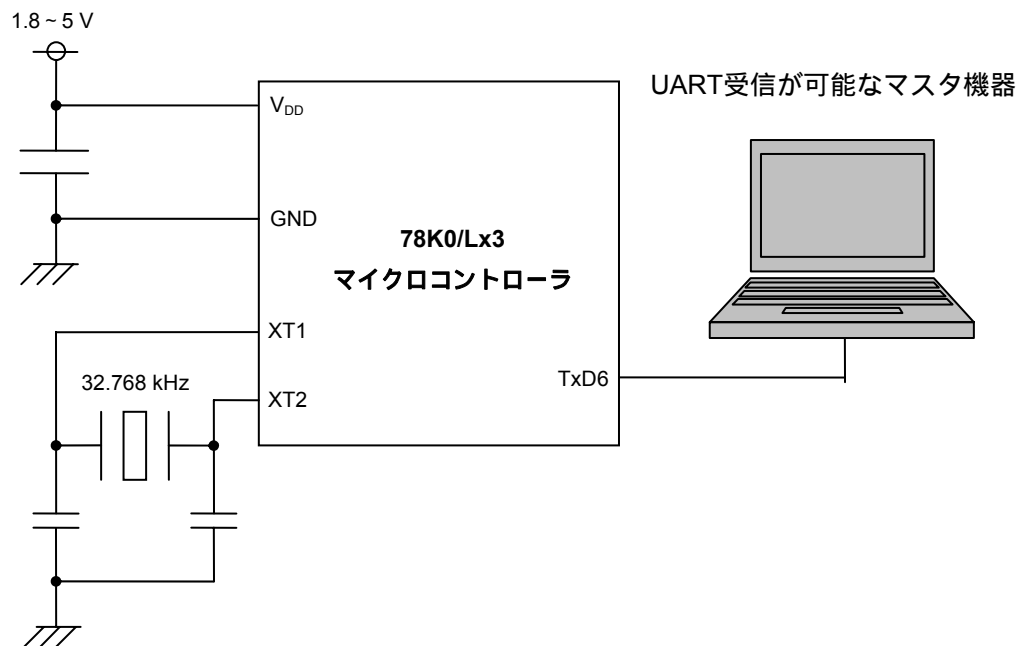
$V_{DD} < V_{LVI}$ となってINTLVI信号が発生した場合は,STOPモードに入ります。 $V_{DD} < V_{LVI}$ となってINTLVI信号が発生した場合は,割り込み処理を終了します。STOPモードに入る際は,STOPモード中にリアルタイム・カウンタが動作できるように,カウンタ値読み出し,書き込みモード中でないかを確認します。

## 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。



- 注意1. AV<sub>REF</sub>端子はV<sub>DD</sub> (1.8 ~ 5 V供給) に直接接続してください。
2. AV<sub>SS</sub>端子はGNDに直接接続してください。
  3. 回路図中の端子およびAV<sub>REF</sub>, AV<sub>SS</sub>端子以外の未使用端子はすべて出力ポートのため、オープン（未接続）にしてください。
  4. TxD6端子は、UART受信が可能な機器に接続します。



## 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) サブシステム・クロック

XT1, XT2端子に, 32.768 kHzの水晶発振子を接続します。

### (2) UART通信用機器 (TxD6)

TxD6端子にUART受信用の機器を接続します。

### (3) スーパー・キャパシタ

V<sub>DD</sub>-GND端子間にスーパー・キャパシタを接続します。

**注意1.** AV<sub>REF</sub>端子はV<sub>DD</sub> (1.8~5 V供給) に直接接続してください。



**2.** AV<sub>SS</sub>端子はGNDに直接接続してください。

## 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、UART送信データのフォーマット、およびフロー・チャートを説明します。

### 3.1 ファイル構成

ダウンロードする圧縮ファイルのファイル構成は、次のようになっています。

ファイル名	説明	同封圧縮 (*.zip) ファイル	
			
main.asm (アセンブリ言語版) ----- main.c (C言語版)	マイコンのハードウェア初期化処理、メイン処理、UART送信処理と低電圧検出割り込み処理のソース・ファイル	注	注
op.asm	オプション・バイト設定用アセンブラ・ソース・ファイル (ウォッチドッグ・タイマの動作設定、低速内蔵発振器の設定などを行います)	●	●
78K0Lx3_LVI_RTC.prw	統合開発環境 PM plus用ワーク・スペース・ファイル		
78K0Lx3_LVI_RTC.prj	統合開発環境 PM plus用プロジェクト・ファイル		

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

備考



: ソース・ファイルのみ同封



: 統合開発環境 PM plusで使用するファイルを同封

## 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。

- ・ 高速内蔵発振回路  
CPUクロック，周辺ハードウェア・クロックに使用します。
- ・ リアルタイム・カウンタ  
年，月，曜日，日，時，分，秒のカウント，および1秒ごとの定周期割り込みに使用します。
- ・ 低電圧検出回路  
低電圧状態検出に使用します。
- ・ シリアル・インタフェースUART6  
リアルタイム・カウンタのカウント値の送信に使用します。

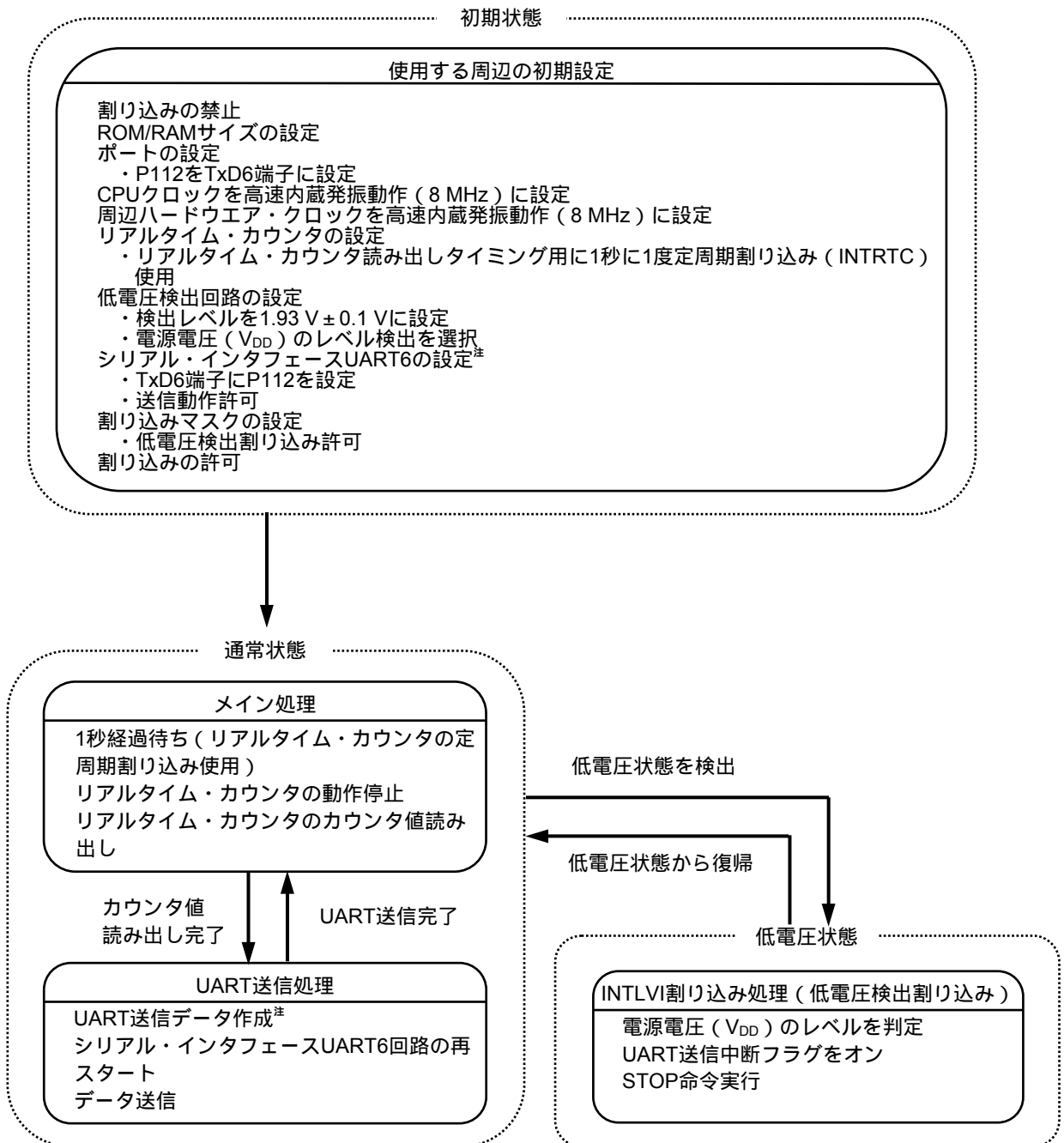
### 3.3 初期設定と動作概要

このサンプル・プログラムでは、使用する周辺の初期設定にて、ポートの設定や、クロック周波数の選択、リアルタイム・カウンタの設定、低電圧検出回路の設定、シリアル・インタフェースUART6の設定などを行います。使用する周辺の初期設定完了後、1秒経過待ちとなります。1秒経過すると、リアルタイム・カウンタのウェイト制御をカウンタ値読み出し、書き込みモードに設定し、カウンタ動作を停止させ、カウンタ値を読み出します。カウンタ値の読み出し完了後、リアルタイム・カウンタのウェイト制御をカウンタ動作に設定し、カウンタ動作を開始させ、読み出したカウンタ値をUART通信にて送信し、再び1秒経過待ちとなります。UART通信にて送信するデータは、年、月、曜日、日、時、分、秒のカウンタ値です<sup>※</sup>。

電源電圧 ( $V_{DD}$ ) が1.93 Vより低くなると、割り込み処理にてSTOPモードに入ります。電源電圧 ( $V_{DD}$ ) が1.93 V以上になると、STOPモードから復帰し、割り込み処理を終了します。UART通信の際、送信動作を行う以前に低電圧状態を検出した場合は、シリアル・インタフェースUART6の回路を再スタートします。

詳細については、次の状態遷移図（ステート・チャート）に示します。

**注** UART通信の設定内容詳細、および送信データの詳細については、3.4 UART送信データのフォーマットを参照してください。



注 UART通信の設定内容詳細、および送信データの詳細については、3.4 UART送信データのフォーマットを参照してください。

### 3.4 UART送信データのフォーマット

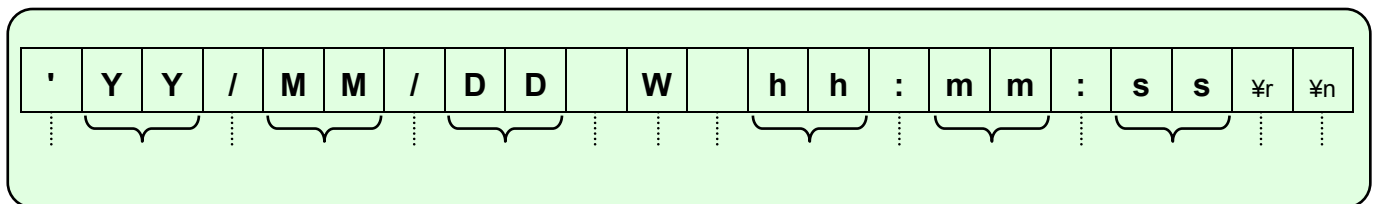
シリアル・インタフェースUART6により送信するデータの内容を説明します。

シリアル・インタフェースUART6の設定は下記のとおりです。

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

データの送信は、1秒に1回行います。1回の送信データの長さは22バイトです。リアルタイム・カウンタから読み出した年、月、曜日、日、時、分、秒のカウンタ値をそれぞれASCIIコードに変換して送信します。送信するデータのフォーマットを図3-1に示します。

図3-1 UART送信データのフォーマット



西暦表示の略を示すクォーターションです。

年のカウンタ値です。西暦の下2桁を2桁のBCDで表します。

年月日の区切りを示します。

月のカウンタ値です。2桁のBCDで表します。

日のカウンタ値です。2桁のBCDで表します。

年月日、曜日、時間を区切るスペースです。

曜日のカウンタ値です。カウンタ値と曜日の対応は以下のとおりです。

曜日	日	月	火	水	木	金	土
カウンタ値	0	1	2	3	4	5	6

時のカウンタ値です。値は24時間制で、2桁のBCDで表します。

時間の区切りを示します。

分のカウンタ値です。2桁のBCDで表します。

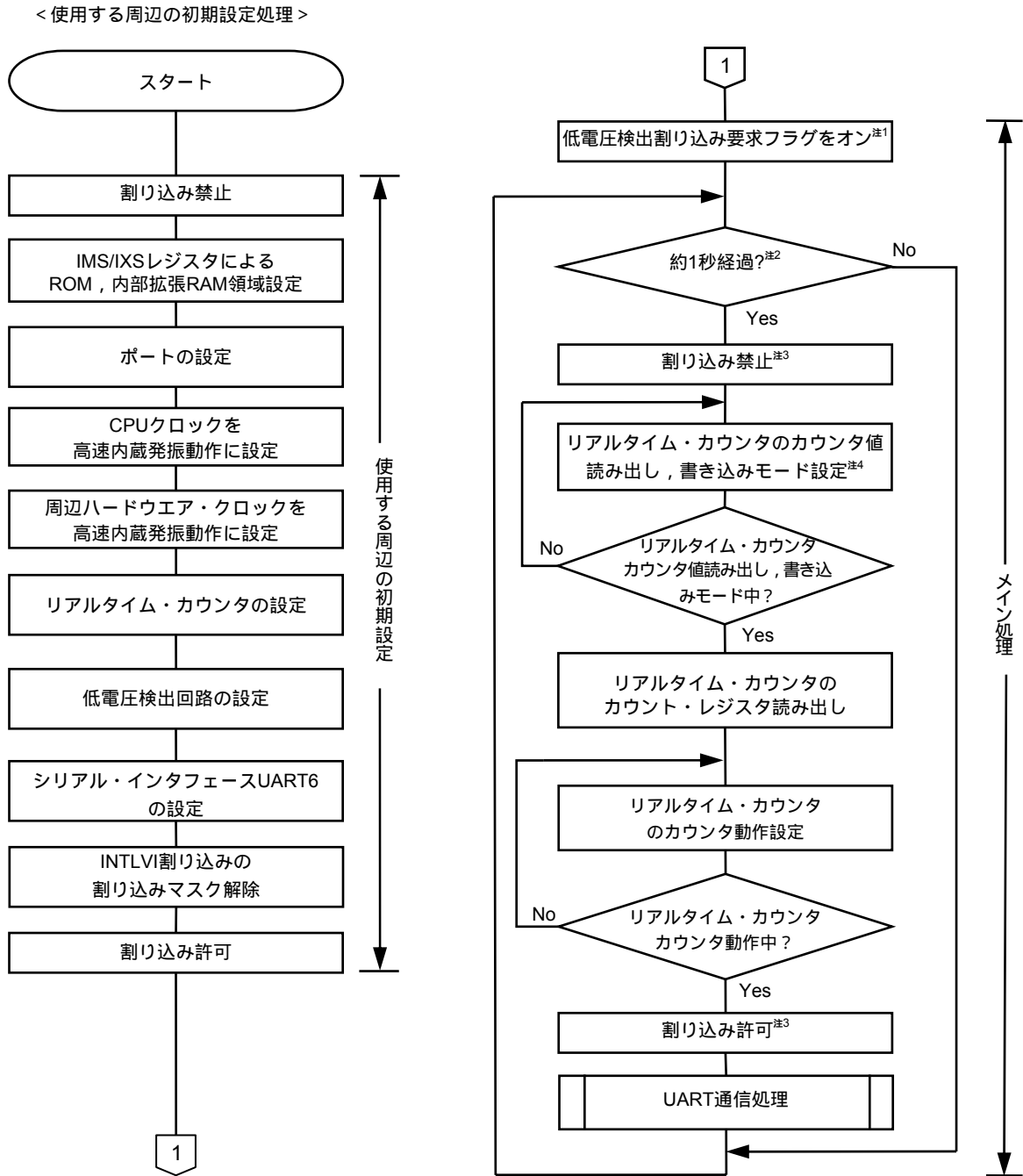
秒のカウンタ値です。2桁のBCDで表します。

復帰コードです。

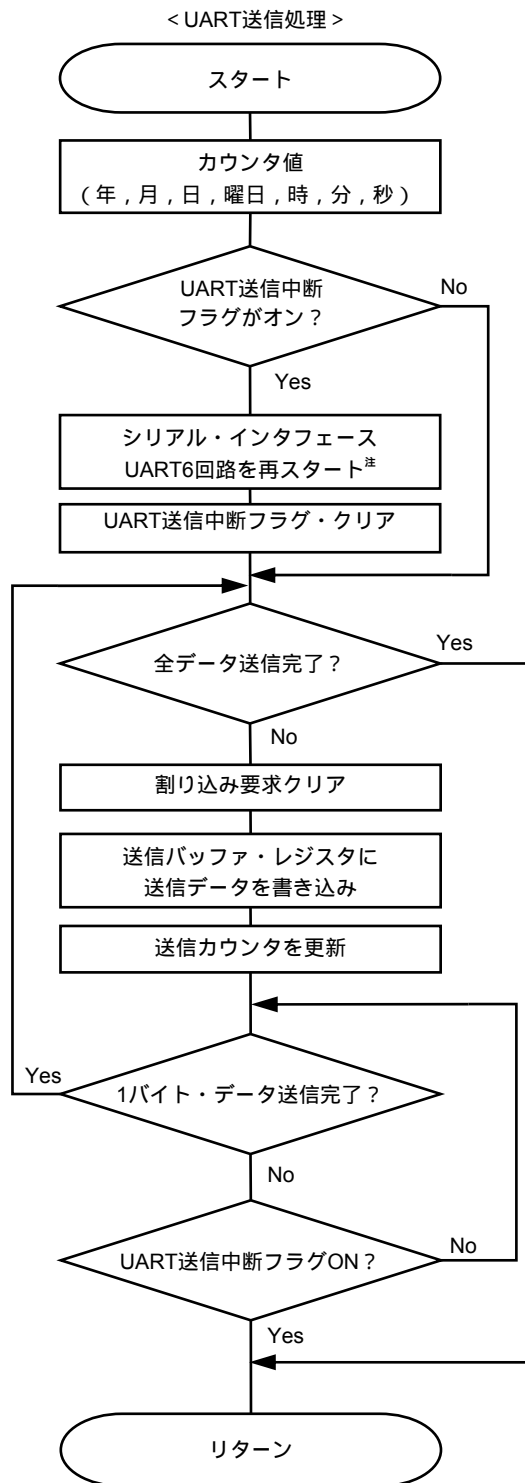
改行コードです。

### 3.5 フロー・チャート

このサンプル・プログラムのフロー・チャートを次に示します。



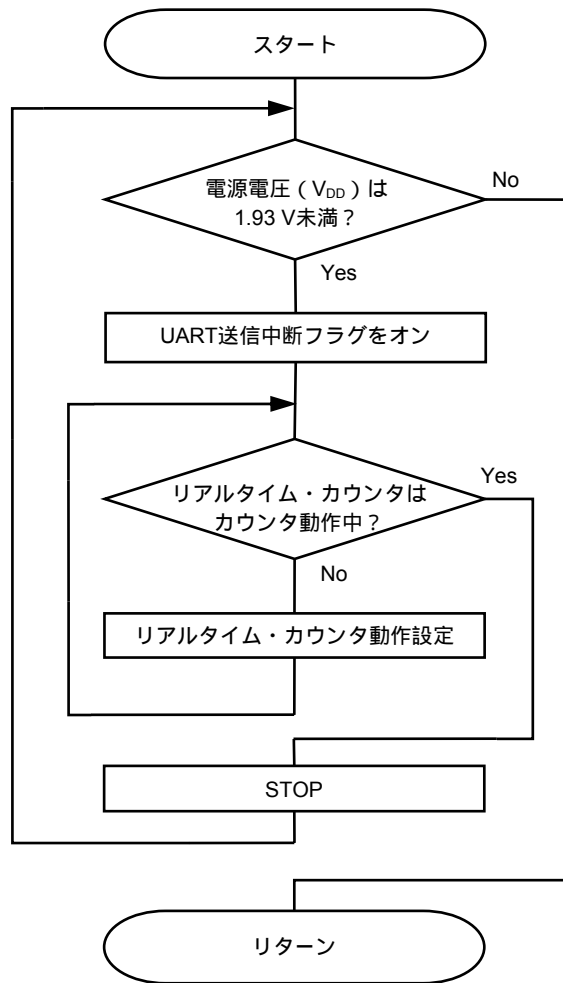
- 注1. リセット・スタート後、低電圧検出割り込み処理にて電源電圧 ( $V_{DD}$ ) のレベルを検出します。
2. リアルタイム・カウンタの定周期割り込み (INTRTC) により、1秒経過を待ちます。
3. カウンタ停止中にSTOPモードに入り、リアルタイム・カウンタのカウンタ値が更新されなくなるのを防止します。
4. リアルタイム・カウンタ停止後、リアルタイム・カウンタが停止するまで最大1クロック (32.768 kHz) の時間がかかります。



注 シリアル・インタフェースUART6への供給クロックが停止した場合，クロック供給再開後の動作は保証していないので，再開後はPOWER6 = 0, RXE6 = 0, TXE6 = 0として，回路を再スタートしてください。



< 低電圧検出割り込み処理 (INTLVI使用) >





C言語で定義する変数は次のとおりです。

- ucSec : リアルタイム・カウンタから読み出した秒カウンタ値を格納する領域です。
- ucMin : リアルタイム・カウンタから読み出した分カウンタ値を格納する領域です。
- ucHour : リアルタイム・カウンタから読み出した時カウンタ値を格納する領域です。
- ucWeek : リアルタイム・カウンタから読み出した曜日カウンタ値を格納する領域です。
- ucYear : リアルタイム・カウンタから読み出した年カウンタ値を格納する領域です。
- ucMonth : リアルタイム・カウンタから読み出した月カウンタ値を格納する領域です。
- ucDay : リアルタイム・カウンタから読み出した日カウンタ値を格納する領域です。
- ucTxBuffer[22] : UART通信で送信するデータを格納する配列です。リアルタイム・カウンタから読み出したカウンタ値(年, 月, 日, 曜日, 時, 分, 秒)をASCIIコードに変換して格納します。
- ucTxBufferCounter : UART送信処理で, 送信したデータ数を数えます。
- bUARTStop : UART送信の中断とシリアル・インタフェースUART6の回路の再スタートを通知するためのフラグです。低電圧状態を検出しSTOPモードに入る際にオンにします。

```

/*=====
RAMの定義
=====*/
/* リアルタイム・カウンタからのデータ取得用変数 */
----- unsigned char ucSec /* 秒 */
----- ,ucMin /* 分 */
----- ,ucHour /* 時 */
----- ,ucWeek /* 曜日 */
----- ,ucYear /* 年 */
----- ,ucMonth /* 月 */
----- ,ucDay; /* 日 */
----- unsigned char ucTxBuffer[22]; /* 送信データ・バッファ */
----- unsigned char ucTxBufferCounter; /* 送信カウンタ */
----- boolean bUARTStop; /* UART送信中断フラグ */

```

**(2) 使用する周辺の初期設定処理**

アセンブリ言語での使用する周辺の初期設定処理では、次の動作を行います。

割り込みを禁止します。

レジスタ・バンクを設定します。

スタック・ポインタの設定を行います。

メモリ・サイズと内部拡張RAMサイズの設定を行います。

使用するマイコンに適したIMS, IXS<sup>註</sup>を設定してください。

ポートの設定を行います。

P112をTxD6端子として使用するため、ハイ・レベル出力に設定します。その他のポートはロウ・レベル出力に設定します。

クロック周波数の設定を行います。

CPUクロック、および周辺ハードウェア・クロックを高速内蔵発振動作に設定します。また、サブシステム・クロック端子の動作モードをXT1発振モードに設定します。

リアルタイム・カウンタの設定を行います。

入クロックにサブシステム・クロック(32.768 kHz)を設定し、1秒に1度定周期割り込み(INTRTC)を発生するよう設定します。

- (a) サブシステム・クロックの発振安定待ち処理です。1秒ウエイトすると、サブシステム・クロックの発振は十分安定するので、リアルタイム・カウンタの定周期割り込みを利用して1秒ウエイトします。1秒ウエイト後、リアルタイム・カウンタを一度停止させ、各カウント・レジスタの初期値を設定し、再びカウンタ動作を開始させます。

低電圧検出回路を設定します。

電源電圧( $V_{DD}$ )のレベル検出をし、低電圧検出割り込み(INTLVI)を発生させるように設定します。検出レベルを $1.93\text{ V} \pm 0.1\text{ V}$ に設定します。

- (a) LVIONをセット(1)すると、LVI回路内のコンパレータの動作を開始します。LVIONをセット(1)してからLVIFで電圧を確認するまでの時間(10  $\mu\text{s}$  (MAX.))をソフトウェアでウエイトします。シリアル・インタフェースUART6を下記のとおり設定します。

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

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

割り込みマスクを設定します。

低電圧検出割り込みのマスクを解除します。低電圧検出割り込み(INTLVI)が発生した際、低電圧検出割り込み処理を行いますので、あらかじめベクタ・テーブルのアドレス04Hに低電圧検出割り込み処理を登録しておきます。

割り込みを許可します。

注 78K0/LF3, 78K0/LE3のみ

```

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

-----
:   割り込み禁止
:
DI
-----
:   レジスタ・バンク設定
:
SEL    RB0
-----
:   スタック・ポインタの設定
:
MOVW   SP,    #STACKTOP
-----
:   ROM/RAMサイズの設定
:
:   モデルにより設定値が異なるので注意してください。
:   使用モデルの設定を有効にしてください。(デフォルトではuPD78F0485)
:
:   ;uPD78F0471,uPD78F0481,uPD78F0491使用時の設定
:   ;MOV   IMS,    #04H           ;ROMサイズの設定
:   ;MOV   IXS,    #0CH           ;内部拡張RAMサイズの設定
:
:   ;uPD78F0472,uPD78F0482,uPD78F0492使用時の設定
:   ;MOV   IMS,    #0C6H          ;ROMサイズの設定
:   ;MOV   IXS,    #0CH           ;内部拡張RAMサイズの設定
:
:   ;uPD78F0473,uPD78F0483,uPD78F0493使用時の設定
:   ;MOV   IMS,    #0C8H          ;ROMサイズの設定
:   ;MOV   IXS,    #0CH           ;内部拡張RAMサイズの設定
:
:   ;uPD78F0474,uPD78F0484,uPD78F0494使用時の設定
:   ;MOV   IMS,    #0CCH          ;ROMサイズの設定
:   ;MOV   IXS,    #0AH           ;内部拡張RAMサイズの設定
:
:   ;uPD78F0475,uPD78F0485,uPD78F0495使用時の設定
:   MOV   IMS,    #0CFH           ;ROMサイズの設定
:   MOV   IXS,    #0AH           ;内部拡張RAMサイズの設定
:
-----
:   ポート1の設定
:
MOV     P1,    #00000000B         ;P1初期値設定
:   ;+++++----- P17/P16/P15/P14/P13/P12/P11/P10:未使用(0)
MOV     PM1,   #00000000B         ;P1入出力設定
:   ;+++++----- PM17/PM16/PM15/PM14/PM13/PM12/PM11/PM10:未使用(0)
:
-----
:   ポート2の設定
:
MOV     P2,    #00000000B         ;P2初期値設定
:   ;+++++----- P27/P26/P25/P24/P23/P22/P21/P20:未使用(0)
MOV     PM2,   #00000000B         ;P2入出力設定
:   ;+++++----- PM27/PM26/PM25/PM24/PM23/PM22/PM21/PM20:未使用(0)
:
-----
:   ポート3の設定
:
MOV     P3,    #00000000B         ;P3初期値設定
:   ;|||++++----- P34/P33/P32/P31/P30:未使用(0)
:   ;+++----- <000固定>
MOV     PM3,   #11100000B         ;P3入出力設定
:   ;|||++++----- PM34/PM33/PM32/PM31/PM30:未使用(0)
:   ;+++----- <111固定>
:
-----
:   ポート4の設定
:
MOV     P4,    #00000000B         ;P4初期値設定
:   ;+++++----- P47/P46/P45/P44/P43/P42/P41/P40:未使用(0)
MOV     PM4,   #00000000B         ;P4入出力設定
:   ;+++++----- PM47/PM46/PM45/PM44/PM43/PM42/PM41/PM40:未使用(0)
:
-----
:   ポート8の設定
:
MOV     P8,    #00000000B         ;P8初期値設定
:   ;|||++++----- P83/P82/P81/P80:未使用(0)
:   ;++++----- <0000固定>
MOV     PM8,   #11110000B         ;P8入出力設定
:   ;|||++++----- PM83/PM82/PM81/PM80:未使用(0)
:   ;++++----- <1111固定>

```

-----			
ポート9の設定			
MOV	P9,	#0000000B	;P9初期値設定
		;        + + + +-----	P93/P92/P91/P90:未使用(0)
		;+ + + +-----	<0000固定>
MOV	PM9,	#1111000B	;P9入出力設定
		;        + + + +-----	PM93/PM92/PM91/PM90:未使用(0)
		;+ + + +-----	<1111固定>
-----			
ポート10の設定			
MOV	P10,	#0000000B	;P10初期値設定
		;        + + + +-----	P103/P102/P101/P100:未使用(0)
		;+ + + +-----	<0000固定>
MOV	PM10,	#1111000B	;P10入出力設定
		;        + + + +-----	PM103/PM102/PM101/PM100:未使用(0)
		;+ + + +-----	<1111固定>
-----			
ポート11の設定			
MOV	P11,	#00000100B	;P11初期値設定
		;        +   + +-----	P113/P111/P110:未使用(0)
		;        +-----	P112:Hi (1)
		;+ + + +-----	<0000固定>
MOV	PM11,	#1111000B	;P11入出力設定
		;        +   + +-----	PM113/PM111/PM110:未使用(0)
		;        +-----	PM112:出力(0) Tx06として使用
		;+ + + +-----	<1111固定>
-----			
ポート12の設定			
MOV	P12,	#00000000B	;P12初期値設定
		;            +-----	P120:未使用(0)
		;      + + + +-----	P124/P123/P122/P121:Read Only
		;+ + + +-----	<000固定>
MOV	PM12,	#11111110B	;P12入出力設定
		;            +-----	PM120:未使用(0)
		;+ + + + + + +-----	<1111111固定>
-----			
ポート13の設定			
MOV	P13,	#0000000B	;P13初期値設定
		;        + + + +-----	P133/P132/P131/P130:未使用(0)
		;+ + + +-----	<0000固定>
MOV	PM13,	#1111000B	;P13入出力設定
		;        + + + +-----	PM133/PM132/PM131/PM130:未使用(0)
		;+ + + +-----	<1111固定>
-----			
ポート14の設定			
MOV	P14,	#0000000B	;P14初期値設定
		;        + + + +-----	P143/P142/P141/P140:未使用(0)
		;+ + + +-----	<0000固定>
MOV	PM14,	#1111000B	;P14入出力設定
		;        + + + +-----	PM143/PM142/PM141/PM140:未使用(0)
		;+ + + +-----	<1111固定>
-----			
ポート15の設定			
MOV	P15,	#0000000B	;P15初期値設定
		;        + + + +-----	P153/P152/P151/P150:未使用(0)
		;+ + + +-----	<0000固定>
MOV	PM15,	#1111000B	;P15入出力設定
		;        + + + +-----	PM153/PM152/PM151/PM150:未使用(0)
		;+ + + +-----	<1111固定>
-----			



```

サブシステム・クロックの発振安定待ち
サブシステム・クロックの発振安定待ちに1秒の定周期割り込みを使用します。
;最初の定周期割り込み要求が発生した後、リアルタイム・カウンタを再スタートさせます。
HRST400:
BF      RTCIF, $HRST400      ;最初の定周期割り込み発生待ち
CLR1    RTCE                  ;リアルタイム・カウンタ動作停止

;カウント・レジスタ初期設定 : '08/06/01 (日) 00:00:00
MOV     SEC, #00H             ;秒 : 00秒
MOV     MIN, #00H            ;分 : 00分
MOV     HOUR, #00H           ;時 : 00時
MOV     WEEK, #00H           ;曜日 : 日曜日
MOV     DAY, #01H            ;日 : 1日
MOV     MONTH, #06H          ;月 : 6月
MOV     YEAR, #08H           ;年 : 08年

CLR1    RTCIF                ;定周期割り込み要求フラグをクリア
SET1    RTCE                  ;リアルタイム・カウンタ動作開始

-----
低電圧検出回路の設定
-----
MOV     LVIS, #00001111B     ;低電圧検出レベル選択レジスタ
;|||||++++----- LVIS3/LVIS2/LVIS1/LVIS0: 検出レベルはVLVI15 (1.93 V±0.1 V)
;++++----- <0000固定>

MOV     LVIM, #10000000B     ;低電圧検出レジスタ
;|||||+----- LVIF: Read Only
;|||||+----- LVIMD: 電圧降下時に電源電圧 (VDD) < 検出電圧 (VLVI) になったとき、
;|||||+----- または、電圧上昇時にVDD > VLVIになったとき内部割り込み信号を発生
;|||||+----- LVISEL: 電源電圧 (VDD) のレベルを検出
;++++----- <0000固定>
;+----- LVION: 低電圧検出動作許可

; LVIONをセット(1)してからLVIFで電圧を確認するまでの時間(10 μs(MAX.))をウエイト
; 10 μs == 80CLK(8MHz)
MOV     B, #13                ;4CLK
HRST300:
DBNZ    B, $HRST300          ;6CLK
;4 + 6 * 13 = 82 CLK

-----
シリアル・インタフェースUART6の設定
-----
リアルタイム・カウンタでのカウント値を送信します。
-----
MOV     CKSR6, #00000000B    ;UART6基本クロック選択
;|||||++++----- TPS63-60: 基本クロック (fXCLK6) = fPRS
;++++----- <0固定>

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

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

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

MOV     ISC, #00001000B      ;入力切り替え制御
;|||||+----- ISC0: 未使用
;|||||+----- ISC1: TI000入力をそのまま使用する(通常動作)
;|||||+----- ISC2: 未使用
;|||++----- ISC3: RxD6/P113入力許可
;|||++----- ISC5-4: TxD6=P112, RxD6=P113
;|||++----- <0固定>

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

```



```

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

CLR1  LV1IF             ;低電圧検出回路割り込み要求クリア
CLR1  LV1MK            ;低電圧検出回路割り込みマスク解除
;
-----
;
; 割り込み許可
;
EI
;
-----

```

C言語の初期化処理も、アセンブリ言語と同様な動作を行います。

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

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

```

/*****
リセット解除後の初期化処理
*****/
void hdwinit(void)
{
    unsigned short temp;          /* ワーク領域 */

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

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

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

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

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

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

    /*-----
    ポートの設定 (未使用ポートはLow出力に設定)
    -----*/
    /* ポート1 */
    P1 = 0b00000000;          /* P1初期値設定 */
    /*+++++----- P17/P16/P15/P14/P13/P12/P11/P10:未使用(0) */
    PM1 = 0b00000000;          /* P1入出力設定 */
    /*+++++----- PM17/PM16/PM15/PM14/PM13/PM12/PM11/PM10:未使用(0) */

    /* ポート2 */
    P2 = 0b00000000;          /* P2初期値設定 */
    /*+++++----- P27/P26/P25/P24/P23/P22/P21/P20:未使用(0) */
    PM2 = 0b00000000;          /* P2入出力設定 */
    /*+++++----- PM27/PM26/PM25/PM24/PM23/PM22/PM21/PM20:未使用(0) */

    /* ポート3 */
    P3 = 0b00000000;          /* P3初期値設定 */
    /*|||++++----- P34/P33/P32/P31/P30:未使用(0) */
    /*++++----- <000固定> */
    PM3 = 0b11100000;          /* P3入出力設定 */
    /*|||++++----- PM34/PM33/PM32/PM31/PM30:未使用(0) */
    /*++++----- <111固定> */

    /* ポート4 */
    P4 = 0b00000000;          /* P4初期値設定 */
    /*+++++----- P47/P46/P45/P44/P43/P42/P41/P40:未使用(0) */
    PM4 = 0b00000000;          /* P4入出力設定 */
    /*+++++----- PM47/PM46/PM45/PM44/PM43/PM42/PM41/PM40:未使用(0) */

    /* ポート8 */
    P8 = 0b00000000;          /* P8初期値設定 */
    /*|||++++----- P83/P82/P81/P80:未使用(0) */
    /*++++----- <0000固定> */
    PM8 = 0b11110000;          /* P8入出力設定 */
    /*|||++++----- PM83/PM82/PM81/PM80:未使用(0) */
    /*++++----- <1111固定> */

    /* ポート9 */
    P9 = 0b00000000;          /* P9初期値設定 */
    /*|||++++----- P93/P92/P91/P90:未使用(0) */
    /*++++----- <0000固定> */
    PM9 = 0b11110000;          /* P9入出力設定 */
    /*|||++++----- PM93/PM92/PM91/PM90:未使用(0) */
    /*++++----- <1111固定> */

```

```

/* ポート10 */
P10 = 0b00000000; /* P10初期値設定 */
/* |||+---- P103/P102/P101/P100:未使用(0) */
/* +---- <0000固定> */
PM10 = 0b11110000; /* P10入出力設定 */
/* |||+---- PM103/PM102/PM101/PM100:未使用(0) */
/* +---- <1111固定> */

/* ポート11 */
P11 = 0b00000100; /* P11初期値設定 */
/* |||+|+---- P113/P111/P110:未使用(0) */
/* |||+---- P112:Hi(1) */
/* +---- <0000固定> */
PM11 = 0b11110000; /* P11入出力設定 */
/* |||+|+---- PM113/PM111/PM110:未使用(0) */
/* |||+---- PM112:出力(0) TxD6として使用 */
/* +---- <1111固定> */

/* ポート12 */
P12 = 0b00000000; /* P12初期値設定 */
/* |||+---- P120:未使用(0) */
/* |||+---- P124/P123/P122/P121:Read Only */
/* +---- <000固定> */
PM12 = 0b11111110; /* P12入出力設定 */
/* |||+---- PM120:未使用(0) */
/* +---- <1111111固定> */

/* ポート13 */
P13 = 0b00000000; /* P13初期値設定 */
/* |||+---- P133/P132/P131/P130:未使用(0) */
/* +---- <0000固定> */
PM13 = 0b11110000; /* P13入出力設定 */
/* |||+---- PM133/PM132/PM131/PM130:未使用(0) */
/* +---- <1111固定> */

/* ポート14 */
P14 = 0b00000000; /* P14初期値設定 */
/* |||+---- P143/P142/P141/P140:未使用(0) */
/* +---- <0000固定> */
PM14 = 0b11110000; /* P14入出力設定 */
/* |||+---- PM143/PM142/PM141/PM140:未使用(0) */
/* +---- <1111固定> */

/* ポート15 */
P15 = 0b00000000; /* P15初期値設定 */
/* |||+---- P153/P152/P151/P150:未使用(0) */
/* +---- <0000固定> */
PM15 = 0b11110000; /* P15入出力設定 */
/* |||+---- PM153/PM152/PM151/PM150:未使用(0) */
/* +---- <1111固定> */

/* -----
クロック周波数の設定
-----
高速内蔵発振8MHz(TYP.)で動作を行うように設定します。
----- */
OSCCTL = 0b00010000; /* クロック動作モード */
/* |||+---- <0固定> */
/* ||+---- OSCSELS: サブシステム・クロック端子の設定 */
/* ||+---- サブシステム・クロック端子の動作モード:XT1発振モード */
/* ||+---- P123/XT1,P124/XT2:水晶/セラミック発振子接続 */
/* ||+---- <0固定> */
/* ++---- EXCLK/OSCSEL: */
/* ++---- 高速システム・クロック端子の動作モード:入力ポート・モード */
/* ++---- P121/X1,P122/X2/EXCLK:入力ポート */

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

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

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

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

```





## 4.2 メイン処理

アセンブリ言語のメイン処理では、次の動作を行います。

リセット・スタート後、低電圧検出割り込み処理にて電源電圧 ( $V_{DD}$ ) のレベルを検出するため、低電圧検出割り込み要求フラグをオンに設定します。

1秒に1度の定周期割り込みをチェックします。定周期割り込みの発生があった場合は、～を行います。

の処理中はリアルタイム・カウンタのウェイト制御をカウンタ値読み出し、書き込みモードに設定にします。カウンタ値読み出し、書き込みモード中はカウンタ動作が停止し、カウント・レジスタが更新されません。カウンタ動作停止中にSTOPモードに入るのを防止するため、割り込み禁止とします。

リアルタイム・カウンタのウェイト制御をカウンタ値読み出し、書き込みモードに設定にし、カウンタ値読み出し、書き込みモード中になるまでウェイトします。ウェイト時間はサブシステム・クロック (32.768 kHz) で最大1クロックです。

リアルタイム・カウンタのカウント・レジスタからカウンタ値 (年, 月, 日, 曜日, 時, 分, 秒) を読み出します。

リアルタイム・カウンタをカウンタ動作に設定し、カウンタ動作中になるまでウェイトします。

割り込みを許可します。

UART送信処理のサブルーチンを呼び出します。

割り込み処理が行われなくなる現象を防止するためのNOP命令です。割り込み機能を制御するレジスタに対する操作を行ったとき、次の命令の終了まで割り込み要求の受け付けを保留します。よって の処理がなかった場合、以下のような動作が繰り返され、割り込み処理が行われなくなる可能性があります。

- ・ から 'LMAIN200'へ分岐 (割り込み機能を制御するレジスタに対する操作なので、割り込み要求を保留)
- ・ から へ分岐 (割り込み機能を制御するレジスタに対する操作の次の命令なので、割り込み要求を保留)

へ分岐します。



C言語のメイン処理も、アセンブリ言語と同様な動作を行います。

```

/*****
メイン・ループ
*****/
void main(void)
{
    /* リセット解除後の電源電圧を低電圧割り込み処理でチェック、 */
    /* すでに検出電圧以下の場合はSTOPモードに入る */
    LVIIIF = 1; /* 低電圧検出割り込み要求オン */

    while(1){
        /*****
        /*
        /*          RTCカウント値送信処理          */
        /*
        /*****
        if(RTCIF)
        { /* 1秒経過 */
            RTCIF = 0; /* 割り込み要求をクリア */
            /*-----*/
            /* リアルタイム・カウンタからデータを取得 */
            /*-----*/
            DI(); /* リアルタイム・カウンタ読み出し中はSEC~YEARカウンタが停止するため、 */
                /* そのままSTOPモードに入ると時間が進まない。 */
                /* 読み出し中は割り込み禁止に設定する */

            do{
                RWAIT = 1; /* SEC~YEARカウンタ停止設定 */
                /* (カウンタ値読み出し/書き込みモード) */
            }while(!RWST); /* カウンタ値の読み出し/書き込みモード中になるまでウェイト */
                /* (RWAIT = 1に設定後、32.768 kHzで最大1クロック(30.5 μs)) */

            ucSec = SEC; /* 秒データ取得 */
            ucMin = MIN; /* 分データ取得 */
            ucHour = HOUR; /* 時データ取得 */
            ucWeek = WEEK; /* 曜日データ取得 */
            ucDay = DAY; /* 日データ取得 */
            ucMonth = MONTH; /* 月データ取得 */
            ucYear = YEAR; /* 年データ取得 */

            do{
                RWAIT = 0; /* SEC~YEARカウンタ動作設定 */
            }while(RWST); /* カウンタ値の読み出し/書き込みモード解除になるまでウェイト

        /*

        EI(); /* 読み出し終了。割り込み禁止解除。 */

        /*-----*/
        /*          UART6送信データ作成&データ送信          */
        /*-----*/
        fn_UART6_Tx();
    }

    NOP(); /* 割り込み機能を制御するレジスタに対する操作命令実行中は、 */
        /* 割り込み要求の受付を保留する。 */
        /* 割り込み不可になるの防止するため、NOP命令を実行。 */

    /*****
    /*
    /*          各種メイン処理          */
    /*
    /*****

    /* 各種メイン処理があればここでいきます。 */
}

```



### 4.3 低電圧検出割り込み処理（INTLVI使用）

アセンブリ言語の低電圧検出割り込み処理では、次の動作を行います。

レジスタ・バンクを切り替えます。

電源電圧 ( $V_{DD}$ ) が1.93 V未満か、1.93 V以上かを判定します。1.93 V未満の場合は の処理を実行します。

1.93 V以上の場合は、割り込み処理を終了します。

UART中断フラグ<sup>注</sup>をオンにします。

万が一、リアルタイム・カウンタがカウンタ値（年、月、日、曜日、時、分、秒）読み出し、書き込みモード中であれば、カウンタ動作設定をし、カウンタ動作中になるのを待ちます。

STOPモードに入ります。

低電圧検出割り込み（INTLVI）が発生すると、STOPモードから復帰し、 の処理を行います。

**注** 低電圧状態を検出しSTOPモードに入る際にオンにし、UART送信の中断とシリアル・インタフェースUART6の回路の再スタートを通知するためのフラグです。

```

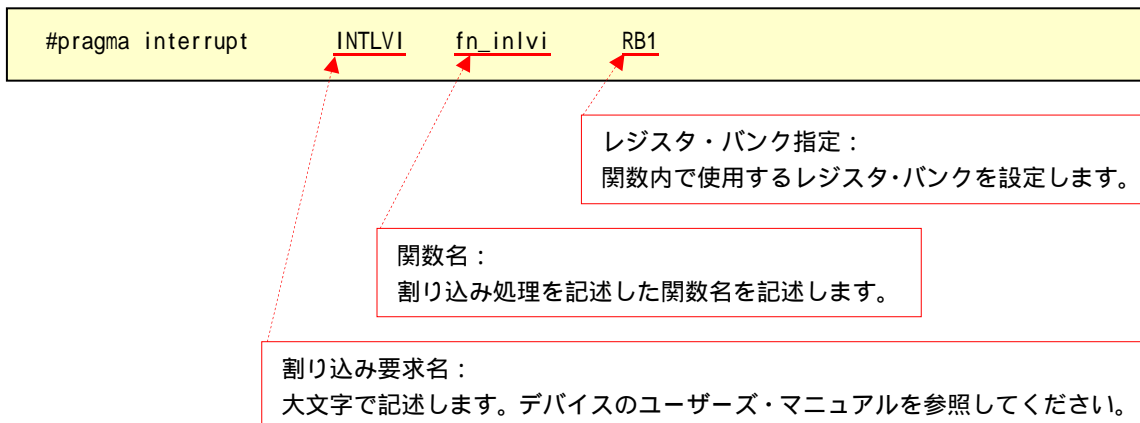
*****
;
;
; 低電圧検出完了割り込み(INTLVI使用)
;
;-----
;
; VDDが検出電圧以上、または検出電圧より低くなった場合に割り込みが発生する。
; 検出電圧は1.93V。
; レジスタバンク1使用
;-----
*****
| INTLVI:
-----|  SEL      RB1                ;割り込み処理でのレジスタバンク設定
-----|
-----|  HLV1100:
-----|  BF      LVIF,$HLV1600        ;VDD < 1.93Vとなった?  YES
-----|
-----|  SET1    FUARTSTOP            ;UART中断フラグをONにする
-----|
-----|  HLV1300:
-----|  BF      RWST,$HLV1400        ;リアルタイム・カウンタが読み出し中の場合
-----|  CLR1    RWAIT                ;読み出し中設定を解除してからSTOPモードに入る
-----|  BR      HLV1300              ;読み出し中設定が解除されるのを待つ
-----|
-----|  HLV1400:
-----|  STOP                    ;STOPモードへ移行
-----|  BR      HLV1100              ;INTLVIでSTOPモードから復帰
-----|
-----|  HLV1600:
-----|  ;VDD      1.93Vとなったので、STOPモードを解除する
-----|
-----|  HLV1900:
-----|  RETI

```

### C言語での割り込み処理の設定例

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）



・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_intlvi(void)
{
    ... (略) ...
}
    
```

詳細に関しては、[ユーザーズ・マニュアル CC78K0 コンパイラ・パッケージ 言語編](#)を参照ください。

C言語の処理も、アセンブリ言語と同様な動作を行います。

```

/*****
低電圧検出完了割り込み (INTLVI使用)
-----
VDDが検出電圧以上、または検出電圧より低くなった場合に割り込みが発生する。
検出電圧は1.93V。
*****/
__interrupt void fn_intlvi(void)
{
    while(LVIF == 1) /* VDD < 1.93V の場合はSTOPモードに入る */
    {
        bUARTStop = 1; /* UART6送信中断フラグをONにする */

        while(RWST) /* リアルタイム・カウンタが読み出し中の場合 */
        {
            RWAIT = 0; /* 読み出し中設定を解除してからSTOPモードに入る。 */
        }
        STOP(); /* STOPモードへ移行 */
    }

    /* VDD 1.93V の場合はSTOPモードを解除する */
}
    
```

## 4.4 UART送信処理

アセンブリ言語のUART送信処理では、次の動作を行います。

リアルタイム・カウンタから読み出したカウンタ値（年，月，日，曜日，時，分，秒）をASCIIコードに変換します。<sup>※</sup>

低電圧状態を検出した場合は、シリアル・インタフェースUART6の回路を再スタートさせます。

シリアル・インタフェースUART6にてASCIIコードに変換したカウンタ値（年，月，日，曜日，時，分，秒）を送信します。1バイト送信完了を待っている間に低電圧状態を検出した場合は、送信動作を中断します。

1バイトのBCDをASCIIコードに変換するためのサブルーチンです。

**注** UART通信の設定内容詳細，および送信データの詳細については，3.4 UART送信データのフォーマットを参照してください。

```

*****
;
;
;   UART6送信データ作成&データ送信
;
;-----
;   [ IN ] RSEC,RMIN,RHOUR,RWEEK,RDAY,RMONTH,RYEAR:RTCカウンタ値
;   [ OUT ] なし
;
;   リアルタイム・カウンタから読み出した値を，
;   ASCIIコードに変換して送信バッファに設定し，送信します。
;   送信開始前に低電圧を検出していた場合は，UART6の再設定を行います。
;   送信完了待ち中に低電圧を検出した場合は，UART6の送信動作を中断します。
;
;   < 送信データの内容 >
;   0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21
;
;   '  Y  Y  /  M  M  /  D  D      W      h  h  :  m  m  :  s  s  ¥r  ¥n
;
;   (Y:年,M:月,D:日,W:曜日,h:時,m:分,s:秒)
;-----
SUART6TX:
;-----
;   送信バッファにUART6送信データ作成処理
;-----
MOV   RTXBUFF,#''''           ;[0]クォーテーション(年データ略表示)保存

MOV   A,RYEAR                 ;年データを取得
CALL  !SNUM2CHR                ;年データをASCIIコードに変換
MOV   (RTXBUFF+1),A           ;[1]年データ上位保存
MOV   A,X
MOV   (RTXBUFF+2),A           ;[2]年データ下位保存

MOV   (RTXBUFF+3),#'/'       ;[3]スラッシュ(年月日データ区切り用)保存

MOV   A,RMONTH                ;月データを取得
CALL  !SNUM2CHR                ;月データをASCIIコードに変換
MOV   (RTXBUFF+4),A           ;[4]月データ上位保存
MOV   A,X
MOV   (RTXBUFF+5),A           ;[5]月データ下位保存

MOV   (RTXBUFF+6),#'/'       ;[6]スラッシュ(年月日データ区切り用)保存

MOV   A,RDAY                  ;日データを取得
CALL  !SNUM2CHR                ;日データをASCIIコードに変換
MOV   (RTXBUFF+7),A           ;[7]日データ上位保存
MOV   A,X
MOV   (RTXBUFF+8),A           ;[8]日データ下位保存

```

```

MOV    (RTXBUF+9),#' '      ;[9]スペース保存

MOV    A,RWEEK              ;曜日データを取得
CALL   !SNUM2CHR            ;曜日データをASCIIコードに変換
MOV    (RTXBUF+10),A        ;[10]曜日データ保存

MOV    (RTXBUF+11),#' '     ;[11]スペース保存

MOV    A,RHOUR              ;時データ取得
CALL   !SNUM2CHR            ;時データをASCIIコードに変換
MOV    (RTXBUF+12),A        ;[12]時データ上位保存
MOV    A,X
MOV    (RTXBUF+13),A        ;[13]時データ下位保存

MOV    (RTXBUF+14),#':'     ;[14]コロン(時間データ区切り用)保存

MOV    A,RMIN               ;分データ取得
CALL   !SNUM2CHR            ;分データをASCIIコードに変換
MOV    (RTXBUF+15),A        ;[15]分データ上位保存
MOV    A,X
MOV    (RTXBUF+16),A        ;[16]分データ下位保存

MOV    (RTXBUF+17),#':'     ;[17]コロン(時間データ区切り用)保存

MOV    A,RSEC               ;秒データ取得
CALL   !SNUM2CHR            ;秒データをASCIIコードに変換
MOV    (RTXBUF+18),A        ;[18]秒データ上位保存
MOV    A,X
MOV    (RTXBUF+19),A        ;[19]秒データ下位保存

MOV    (RTXBUF+20),#0DH     ;[20]キャリッジリターン保存
MOV    (RTXBUF+21),#0AH     ;[21]ラインフィード保存

;-----
;          UART6データ送信
;-----

BF     FUARTSTOP,$JU6TX500  ;低電圧が検出されている? NO

;===== シリアル・インタフェースUART6再設定 =====
CLR1   POWER6              ;内部動作クロックの動作禁止
CLR1   TXE6                 ;送信動作禁止

DI     ;UART6再設定中は割り込み禁止
SET1   TXE6                 ;送信動作許可
SET1   POWER6              ;内部動作クロックの動作許可
CLR1   FUARTSTOP           ;UART送信中断フラグをクリア
EI     ;割り込み許可

JU6TX500:
;===== 送信動作開始 =====
MOV    B,#22                ;送信カウンタ設定
MOVW   HL,#RTXBUF

JU6TX600:
CLR1   STIF6               ;割り込み要求クリア
MOV    A,[HL]              ;送信バッファから送信データを取得
MOV    TXB6,A              ;データ送信

JU6TX700:
BT     FUARTSTOP,$JU6TX800 ;UART6送信中断フラグがON? YES:送信完了を待たずに終了
BF     STIF6,$JU6TX700     ;UART6 1Byte送信完了? NO
CLR1   STIF6               ;割り込み要求クリア
INCW   HL                  ;送信バッファの送信データ位置更新
DBNZ   B,$JU6TX600         ;未送信データあり? YES :次のデータを送信

JU6TX800:                  ;送信終了
RET

```

```

*****
;
;
;   BCDをASCIIコードに変換
;
;-----
;   [ IN ] A:数値(BCD)
;   [ OUT ] A:上位4bitのASCIIコード
;           X:下位4bitのASCIIコード
;-----
*****
SNUM2CHR:
MOV     X,A
AND     A,#0FH           ;下位4ビットを取得
ADD     A,'#0'           ;ASCIIコードに変換
XCH     A,X              ;戻り値保存&変換対象数値取得

AND     A,#0F0H          ;上位4ビットを取得
ROL     A,1              ;上位4bitを下位4bitに落として取得
ROL     A,1
ROL     A,1
ROL     A,1
ADD     A,'#0'           ;ASCIIコードに変換
RET

```

C言語の処理も，アセンブリ言語と同様な動作を行います。

```

/*****
UART6送信データ作成 & データ送信
-----

[ IN ] なし
[ OUT ] なし

リアルタイム・カウンタから読み出した値を，
ASCIIコードに変換して送信バッファに設定し，送信します。
送信開始前に低電圧を検出していた場合は，UART6の再設定を行います。
送信完了待ち中に低電圧を検出した場合は，UART6の送信動作を中断します。

< 送信データの内容 >
 0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21
  '  Y  Y  /  M  M  /  D  D      W      h  h  :  m  m  :  s  s  ¥r  ¥n

          (Y:年, M:月, D:日, W:曜日, h:時, m:分, s:秒)
*****/
static void fn_UART6_Tx(void)
{
    /*****/
    /*
    /*          UART6送信データ作成処理          */
    /*
    /*****/
    ucTxBuffer[0] = '¥';
    ucTxBuffer[1] = (( ucYear & ( 0xF0 )) >> 4 ) + '0';
    ucTxBuffer[2] = ( ucYear & ( 0x0F )) + '0';
    ucTxBuffer[3] = '/';
    ucTxBuffer[4] = (( ucMonth & ( 0xF0 )) >> 4 ) + '0';
    ucTxBuffer[5] = ( ucMonth & ( 0x0F )) + '0';
    ucTxBuffer[6] = '/';
    ucTxBuffer[7] = (( ucDay & ( 0xF0 )) >> 4 ) + '0';
    ucTxBuffer[8] = ( ucDay & ( 0x0F )) + '0';
    ucTxBuffer[9] = ' ';
    ucTxBuffer[10] = ( ucWeek & ( 0x0F )) + '0';
    ucTxBuffer[11] = ' ';
    ucTxBuffer[12] = (( ucHour & ( 0xF0 )) >> 4 ) + '0';
    ucTxBuffer[13] = ( ucHour & ( 0x0F )) + '0';
    ucTxBuffer[14] = ':';
    ucTxBuffer[15] = (( ucMin & ( 0xF0 )) >> 4 ) + '0';
    ucTxBuffer[16] = ( ucMin & ( 0x0F )) + '0';
    ucTxBuffer[17] = ':';
    ucTxBuffer[18] = (( ucSec & ( 0xF0 )) >> 4 ) + '0';
    ucTxBuffer[19] = ( ucSec & ( 0x0F )) + '0';
    ucTxBuffer[20] = '¥r';
    ucTxBuffer[21] = '¥n';

    /* クォーテーション(年データ略表示) */
    /* 年データ上位(ASCIIコードに変換) */
    /* 年データ下位(ASCIIコードに変換) */
    /* スラッシュ(年月日データ区切り用) */
    /* 月データ上位(ASCIIコードに変換) */
    /* 月データ下位(ASCIIコードに変換) */
    /* スラッシュ(年月日データ区切り用) */
    /* 日データ上位(ASCIIコードに変換) */
    /* 日データ下位(ASCIIコードに変換) */
    /* スペース */
    /* 曜日データ */
    /* スペース */
    /* 時データ上位(ASCIIコードに変換) */
    /* 時データ下位(ASCIIコードに変換) */
    /* コロン(時間データ区切り用) */
    /* 分データ上位(ASCIIコードに変換) */
    /* 分データ下位(ASCIIコードに変換) */
    /* コロン(時間データ区切り用) */
    /* 秒データ上位(ASCIIコードに変換) */
    /* 秒データ下位(ASCIIコードに変換) */
    /* キャリッジリターン */
    /* ラインフィード */
}

```

```
/*-----*/
/*          UART6データ送信          */
/*-----*/
/*-----*/
/* シリアル・インタフェースUART6再設定 */
/*-----*/
if(bUARTStop) /* すでに低電圧が検出されている場合は，UART6の再設定が必要 */
{
    POWER6 = 0;          /* 内部動作クロックの動作禁止 */
    TXE6 = 0;            /* 送信動作禁止 */

    DI();                /* UART6再設定中は割り込み禁止 */
    TXE6 = 1;            /* 送信動作許可 */
    POWER6 = 1;          /* 内部動作クロックの動作許可 */
    bUARTStop = 0;       /* UART送信中断フラグをクリア */
    EI();                /* 割り込み許可 */
}

/*-----*/
/*          送信動作開始          */
/*-----*/
for(ucTxBufferCounter = 0; ucTxBufferCounter < sizeof(ucTxBuffer); ucTxBufferCounter++)
{ /* 全データ送信完了するまで，送信動作継続 */
    STIF6 = 0;          /* 割り込み要求クリア */
    TXB6 = ucTxBuffer[ucTxBufferCounter]; /* データ送信 */
    while(!STIF6)
    { /* UART6で1Byte送信完了待ち */
        if(bUARTStop)
        { /* 送信途中で送信中断フラグがONになった場合，送信動作を中断 */
            ucTxBufferCounter = sizeof(ucTxBuffer);
            break;
        }
    }
}
}
```

## 第5章 関連資料

資料名		和文 / 英文
78K0/LC3 ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0/LD3 ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0/LE3 ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0/LF3 ユーザーズ・マニュアル		<a href="#">PDF</a>
78K0シリーズ 命令編 ユーザーズ・マニュアル		<a href="#">PDF</a>
RA78K0 アセンブラ・パッケージ ユーザーズ・マニュアル	言語編	<a href="#">PDF</a>
	操作編	<a href="#">PDF</a>
CC78K0 Cコンパイラ ユーザーズ・マニュアル	言語編	<a href="#">PDF</a>
	操作編	<a href="#">PDF</a>
PM+ プロジェクト・マネージャ ユーザーズ・マニュアル		<a href="#">PDF</a>



## 付録A プログラム・リスト

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

main.asm (アセンブリ言語版)

```
*****
;
;
;   NEC Electronics   78K0/Lx3シリーズ
;
;
;*****
;   78K0/LF3シリーズ   サンプル・プログラム
;*****
;   低電圧時のリアルタイム・カウンタ動作継続編
;*****
; 【履歴】
;   2008.05.--   新規作成
;*****
;
;
; 【概要】
;
; 本サンプルプログラムは、低電圧状態でリアルタイム・カウンタのカウント動作を継続す
; るプログラムです。
; 1秒毎にリアルタイム・カウンタのカウント・レジスタの値を取得し、シリアル・インタ
; フェースUART6にて送信します。電源電圧が1.93V以下になるとSTOPモードに入ります。電
; 源電圧が1.93Vを超えるとSTOPモードから復帰します。
; 本サンプルプログラムのUARTは、送信のみを行います。
;
;
;
; <初期設定の主な設定内容>
;
; ・ベクタ・テーブルの設定
; ・レジスタ・バンクの設定
; ・スタック・ポインタの設定
; ・ROM/RAMサイズの設定
; ・ポートの設定
; ・CPUクロック周波数を高速内蔵発振8MHz(TYP.)に設定
; ・リアルタイム・カウンタの設定
; ・低電圧検出回路の設定
; ・シリアル・インタフェースUART6の設定
```

```

; ・割り込みマスクの設定
;
;
; <メイン処理の主な内容>
;
; ・リアルタイム・カウンタのカウンタ動作停止
; ・リアルタイム・カウンタのカウント・レジスタの値を取得
; ・リアルタイム・カウンタのカウンタ動作設定
; ・UART6の送信データ作成 & データ送信処理
;
;
; <UART6の送信データ作成 & データ送信処理の主な内容>
;
; ・送信データ作成
; ・通信動作の開始
; ・送信データのカウンタ
; ・送信データの設定
;
;
; <低電圧検出割り込み(INTLVI使用)>
;
; ・リアルタイム・カウンタのカウンタ動作設定
; ・STOP命令実行
;
;
;*****
;
;=====
;
;      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     IINTLVI              ;(04)  INTLVI
        DW     RESET_START          ;(06)  INTP0
        DW     RESET_START          ;(08)  INTP1
        DW     RESET_START          ;(0A)  INTP2
        DW     RESET_START          ;(0C)  INTP3
        DW     RESET_START          ;(0E)  INTP4
        DW     RESET_START          ;(10)  INTP5
        DW     RESET_START          ;(12)  INTSRE6

```

```

DW    RESET_START                ;(14)  INTSR6
DW    RESET_START                ;(16)  INTST6
DW    RESET_START                ;(18)  INTCSI10/INTST0
DW    RESET_START                ;(1A)  INTTMH1
DW    RESET_START                ;(1C)  INTTMH0
DW    RESET_START                ;(1E)  INTTM50
DW    RESET_START                ;(20)  INTTM000
DW    RESET_START                ;(22)  INTTM010
DW    RESET_START                ;(24)  INTAD
DW    RESET_START                ;(26)  INTSR0
DW    RESET_START                ;(28)  INTRTC
DW    RESET_START                ;(2A)  INTTM51
DW    RESET_START                ;(2C)  INTKR
DW    RESET_START                ;(2E)  INTRTCI
DW    RESET_START                ;(30)  INTDSAD
DW    RESET_START                ;(32)  INTTM52
DW    RESET_START                ;(34)  INTTMH2
DW    RESET_START                ;(36)  INTMCG
DW    RESET_START                ;(38)  INTRIN
DW    RESET_START                ;(3A)  INTRERR/INTGP/INTREND/INTDFULL
DW    RESET_START                ;(3C)  INTACSI
DW    RESET_START                ;(3E)  BRK

```

=====

```

;
;   スタック領域の確保
;

```

=====

```

DSTK      DSEG   AT      0FB00H      ;RAM先頭アドレス
STACKEND:
          DS      20H              ;スタック領域を32バイト確保
STACKTOP:
          ;スタック領域の先頭アドレス = FB20H

```

=====

```

;
;   RAMの定義
;

```

=====

```

DRTC      DSEG   SADDR   ;RTC関連RAM
RSEC:     DS      1          ;秒データ取得変数
RMIN:     DS      1          ;分データ取得変数
RHOURL:   DS      1          ;時データ取得変数
RWEEK:    DS      1          ;曜日データ取得変数

```

```

RDAY:      DS      1          ;日データ取得変数
RMOTH:     DS      1          ;月データ取得変数
RYEAR:     DS      1          ;年データ取得変数

RTXBUF:    DS      22        ;送信データ・バッファ

DLVI   BSEG          ;INTLVI処理関連RAM
FUARTSTOP  DBIT      ;UART送信中断フラグ

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

;-----
;      割り込み禁止
;-----
;
;      DI
;-----
;      レジスタ・バンク設定
;-----
;
;      SEL      RB0
;-----
;
;      スタック・ポインタの設定
;-----
;
;      MOVW    SP,    #STACKTOP
;-----
;
;      ROM/RAMサイズの設定
;-----
;
;      モデルにより設定値が異なるので注意してください。
;      使用モデルの設定を有効にしてください。（デフォルトではuPD78F0485）
;-----
;
;      ;uPD78F0471,uPD78F0481,uPD78F0491使用時の設定
;      ;MOV    IMS,    #04H          ;ROMサイズの設定
;      ;MOV    IXS,    #0CH          ;内部拡張RAMサイズの設定

;
;      ;uPD78F0472,uPD78F0482,uPD78F0492使用時の設定

```

```
;MOV IMS, #0C6H ;ROMサイズの設定
;MOV IXS, #0CH ;内部拡張RAMサイズの設定
```

```
;uPD78F0473,uPD78F0483,uPD78F0493使用時の設定
```

```
;MOV IMS, #0C8H ;ROMサイズの設定
;MOV IXS, #0CH ;内部拡張RAMサイズの設定
```

```
;uPD78F0474,uPD78F0484,uPD78F0494使用時の設定
```

```
;MOV IMS, #0CCH ;ROMサイズの設定
;MOV IXS, #0AH ;内部拡張RAMサイズの設定
```

```
;uPD78F0475,uPD78F0485,uPD78F0495使用時の設定
```

```
MOV IMS, #0CFH ;ROMサイズの設定
MOV IXS, #0AH ;内部拡張RAMサイズの設定
```

```
-----
```

```
;
; ポート1の設定
```

```
-----
```

```
MOV P1, #0000000B ;P1初期値設定
;+++++----- P17/P16/P15/P14/P13/P12/P11/P10:未使用(0)
MOV PM1, #0000000B ;P1入出力設定
;+++++----- PM17/PM16/PM15/PM14/PM13/PM12/PM11/PM10:未使用(0)
```

```
-----
```

```
;
; ポート2の設定
```

```
-----
```

```
MOV P2, #0000000B ;P2初期値設定
;+++++----- P27/P26/P25/P24/P23/P22/P21/P20:未使用(0)
MOV PM2, #0000000B ;P2入出力設定
;+++++----- PM27/PM26/PM25/PM24/PM23/PM22/PM21/PM20:未使用(0)
```

```
-----
```

```
;
; ポート3の設定
```

```
-----
```

```
MOV P3, #0000000B ;P3初期値設定
;||+++++----- P34/P33/P32/P31/P30:未使用(0)
;+++----- <000固定>
MOV PM3, #1110000B ;P3入出力設定
;||+++++----- PM34/PM33/PM32/PM31/PM30:未使用(0)
;+++----- <111固定>
```

```
-----
```

```
;
; ポート4の設定
```

```
-----
```

```
MOV P4, #0000000B ;P4初期値設定
;+++++----- P47/P46/P45/P44/P43/P42/P41/P40:未使用(0)
```

```

MOV    PM4,    #0000000B          ;P4入出力設定
;+++++----- PM47/PM46/PM45/PM44/PM43/PM42/PM41/PM40:未使用(0)
;-----
;
;   ポート8の設定
;-----
MOV    P8,     #0000000B          ;P8初期値設定
;|||++++----- P83/P82/P81/P80:未使用(0)
;++++----- <0000固定>
MOV    PM8,    #1111000B          ;P8入出力設定
;|||++++----- PM83/PM82/PM81/PM80:未使用(0)
;++++----- <1111固定>
;-----
;
;   ポート9の設定
;-----
MOV    P9,     #0000000B          ;P9初期値設定
;|||++++----- P93/P92/P91/P90:未使用(0)
;++++----- <0000固定>
MOV    PM9,    #1111000B          ;P9入出力設定
;|||++++----- PM93/PM92/PM91/PM90:未使用(0)
;++++----- <1111固定>
;-----
;
;   ポート10の設定
;-----
MOV    P10,    #0000000B          ;P10初期値設定
;|||++++----- P103/P102/P101/P100:未使用(0)
;++++----- <0000固定>
MOV    PM10,   #1111000B          ;P10入出力設定
;|||++++----- PM103/PM102/PM101/PM100:未使用(0)
;++++----- <1111固定>
;-----
;
;   ポート11の設定
;-----
MOV    P11,    #00000100B          ;P11初期値設定
;|||+|++----- P113/P111/P110:未使用(0)
;||| +----- P112:Hi(1)
;++++----- <0000固定>
MOV    PM11,   #1111000B          ;P11入出力設定
;|||+|++----- PM113/PM111/PM110:未使用(0)
;||| +----- PM112:出力(0) TxD6として使用
;++++----- <1111固定>
;-----
;
;   ポート12の設定
;-----

```

```

MOV    P12,    #0000000B          ;P12初期値設定
        ;|||||+----- P120:未使用(0)
        ;|||++++----- P124/P123/P122/P121:Read Only
        ;+++----- <000固定>

MOV    PM12,   #11111110B         ;P12入出力設定
        ;|||||+----- PM120:未使用(0)
        ;+++++----- <1111111固定>

;-----
;
;   ポート13の設定
;-----
MOV    P13,    #0000000B          ;P13初期値設定
        ;|||++++----- P133/P132/P131/P130:未使用(0)
        ;++++----- <0000固定>

MOV    PM13,   #11110000B         ;P13入出力設定
        ;|||++++----- PM133/PM132/PM131/PM130:未使用(0)
        ;++++----- <1111固定>

;-----
;
;   ポート14の設定
;-----
MOV    P14,    #0000000B          ;P14初期値設定
        ;|||++++----- P143/P142/P141/P140:未使用(0)
        ;++++----- <0000固定>

MOV    PM14,   #11110000B         ;P14入出力設定
        ;|||++++----- PM143/PM142/PM141/PM140:未使用(0)
        ;++++----- <1111固定>

;-----
;
;   ポート15の設定
;-----
MOV    P15,    #0000000B          ;P15初期値設定
        ;|||++++----- P153/P152/P151/P150:未使用(0)
        ;++++----- <0000固定>

MOV    PM15,   #11110000B         ;P15入出力設定
        ;|||++++----- PM153/PM152/PM151/PM150:未使用(0)
        ;++++----- <1111固定>

;-----
;
;   クロック周波数の設定
;-----
;
;   高速内蔵発振8MHz(TYP.)で動作を行うように設定します。
;-----
MOV    OSCCTL, #00010000B         ;クロック動作モード
        ;|||++++----- <0000固定>
        ;||+----- OSCSELS: サブシステム・クロック端子の設定

```

```

;|||                サブシステム・クロック端子の動作モード:XT1発振モード
;|||                P123/XT1,P124/XT2:水晶 / セラミック発振子接続
;||+----- <0固定>
;++++----- EXCLK/OSCSEL:
;                高速システム・クロック端子の動作モード：入力ポート・モード
;                P121/X1,P122/X2/EXCLK：入力ポート

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

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

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

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

;-----
;   リアルタイム・カウンタの設定
;-----
;   1秒毎にUARTでリアルタイム・カウンタの値を送信するため、
;   年月日と時間のカウンタと、UART送信間隔の測定を行います。
;-----

MOV    RTCCL, #00000000B    ;リアルタイム・カウンタ・クロック選択レジスタ
;|||||+----- RTCCL1/RTCCLO:入力クロック (fRTC) 選択
;|||||        fRTC = fSUB(32.768kHz)
;+++++----- <000000固定>

MOV    RTCC0, #00001010B    ;リアルタイム・カウンタ・コントロール・レジスタ0

```



```

;||||+++----- CT2/CT1/CT0:1秒に1度定周期割り込み (INTRTC) を発生
;||||+----- AMPM:24時間制
;|||+----- RCLOE0:RTCCL端子の出力 (32.768 kHz) 禁止
;||+----- RCLOE1:RTC1HZ端子の出力 (1 Hz) 禁止
;|+----- <0固定>
;+----- RTCE:リアルタイム・カウンタ動作停止

MOV    RTCC1, #00001000B    ;リアルタイム・カウンタ・コントロール・レジスタ1
;|||||+----- RWAIT:カウンタ動作設定
;|||||+----- RWST:Read Only
;||||+----- <0固定>
;|||+----- RIFG:定周期割り込み発生クリア
;||+----- WAFG:無効
;|+----- <0固定>
;+----- WALIE:アラームの一致による割り込み (INTRTC) を発生しない
;+----- WALE:一致動作無効

MOV    RTCC2, #00000000B    ;リアルタイム・カウンタ・コントロール・レジスタ2
;+||||+++----- RINTE/CT2/CT1/CT0:インターバル割り込みを発生しない。
;||+----- <00固定>
;|+----- RCKDIV:無効
;+----- RCLOE2:RTCDIV端子の出力禁止

CLR1   RTCIF    ;定周期割り込み要求をクリア
SET1   RTCE     ;リアルタイム・カウンタ動作開始

;-----;
;   サブシステム・クロックの発振安定待ち   ;
;-----;
;サブシステム・クロックの発振安定待ちに1秒の定周期割り込みを使用します。
;最初の定周期割り込み要求が発生した後，リアルタイム・カウンタを再スタートさせます。

HRST400:
BF     RTCIF,$HRST400      ;最初の定周期割り込み発生待ち
CLR1   RTCE                ;リアルタイム・カウンタ動作停止

;カウント・レジスタ初期設定： '08/06/01 (日) 00:00:00
MOV    SEC,   #00H         ;秒：00秒
MOV    MIN,   #00H         ;分：00分
MOV    HOUR,  #00H         ;時：00時
MOV    WEEK,  #00H         ;曜日：日曜日
MOV    DAY,   #01H         ;日：1日
MOV    MONTH, #06H         ;月：6月
MOV    YEAR,  #08H         ;年：08年

```

```
CLR1   RTCIF   ;定周期割り込み要求フラグをクリア
SET1   RTCE    ;リアルタイム・カウンタ動作開始
```

```
;-----
; 低電圧検出回路の設定
```

```
MOV    LVIS,   #00001111B    ;低電圧検出レベル選択レジスタ
;||||++++----- LVIS3/LVIS2/LVIS1/LVIS0:検出レベルはVLVI15 ( 1.93 V ± 0.1 V )
;++++----- <0000固定>

MOV    LVIM,   #10000000B    ;低電圧検出レジスタ
;|||||+----- LVIF:Read Only
;|||||+----- LVIMD:電圧降下時に電源電圧 ( VDD ) < 検出電圧 ( VLVI ) になったとき,
;|||||         または, 電圧上昇時にVDD  VLVIになったとき内部割り込み信号を発生
;|||||+----- LVISEL:電源電圧 ( VDD ) のレベルを検出
;++++----- <0000固定>
;+----- LVION:低電圧検出動作許可
```

```
; LVIONをセット(1)してからLVIFで電圧を確認するまでの時間(10 μs(MAX.))をウェイト
```

```
; 10μs == 80CLK(8MHz)
```

```
MOV    B,     #13            ;4CLK
```

```
HRST300:
```

```
DBNZ   B,     $HRST300      ;6CLK
```

```
;4 + 6 * 13 = 82 CLK
```

```
;-----
; シリアル・インタフェースUART6の設定
```

```
;-----
; リアルタイム・カウンタでのカウント値を送信します。
```

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

```
;ボー・レート用クロックの分周値設定
```

```
MOV    BRGC6, #35          ;ボーレート =  $8 \cdot 10^6 [\text{Hz}] / (2 \cdot 115200 [\text{bps}]) = 34.72$ 
```

```
; 誤差を小さくするため, 小数点以下を切り上げ
```

```
;ボーレート: 115200bps 114285bps(ERR:-0.79%)
```

```
MOV    ASIM6, #01000101B    ;UART6動作モード選択
```

```
;|||||+----- ISRM6: 受信エラー発生時にINTSR6を割り込み
```

```
;|||||+----- SL6: ストップ・ビット数=1
```

```

;||||+----- CL6: データ長=8
;|||++----- PS61-60: パリティなし
;||+----- RXE6: 受信動作禁止
;|+----- TXE6: 送信動作許可
;+----- POWER6: 内部動作クロックの動作禁止

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

MOV    ISC,    #00001000B    ;入力切り替え制御
;|||||+----- ISC0: 未使用
;|||||+----- ISC1: TI000入力をそのまま使用する(通常動作)
;||||+----- ISC2: 未使用
;|||+----- ISC3: RxD6/P113入力許可
;||+----- ISC5-4: TxD6=P112,RxD6=P113
;++----- <0固定>

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

;-----
;   割り込みマスクの設定
;-----

MOVW   MK0,#0FFFFH
MOVW   MK1,#0FFFFH          ;全ての割り込みをマスク

CLR1   LVIF                  ;低電圧検出回路割り込み要求クリア
CLR1   LVIMK                 ;低電圧検出回路割り込みマスク解除

;-----
;   割り込み許可
;-----

EI

;*****
;
;
;
;   メイン処理
;
;

```

```

;
;*****
MAIN_LOOP:

;リセット解除後の電源電圧を低電圧割り込み処理でチェックし、
;すでに検出電圧以下の場合はSTOPモードに入る
SET1  LVIIF          ;低電圧検出割り込み要求オン

;*****
;
;          RTCカウント値送信処理          ;
;
;*****
BF      RTCIF,$LMAIN500      ;1秒経過?  NO
CLR1    RTCIF              ;割り込み要求クリア

;-----
; リアルタイム・カウンタからデータを取得
;-----
DI          ;リアルタイム・カウンタ読み出し中はSEC~YEARカウンタが停止するため、
;そのままSTOPモードに入ると時間が進まない。読み出し中は割り込み禁止に設
定する

LMAIN200:
SET1  RWAIT          ;SEC~YEARカウンタ停止設定(カウンタ値読み出し/書き込みモー
ド)
BF      RWST,$LMAIN200      ;カウンタ値の読み出し/書き込みモード中になるまでウエイト
;(RWAIT = 1に設定後, 32.768 kHzで最大1クロック(30.5 μs))

MOV    A,SEC
MOV    RSEC,A          ;秒データ取得
MOV    A,MIN
MOV    RMIN,A          ;分データ取得
MOV    A,HOUR
MOV    RHOUR,A         ;時データ取得
MOV    A,WEEK
MOV    RWEEK,A         ;曜日データ取得
MOV    A,DAY
MOV    RDAY,A          ;日データ取得
MOV    A,MONTH
MOV    RMOTH,A         ;月データ取得
MOV    A,YEAR
MOV    RYEAR,A         ;年データ取得

```

LMAIN300:

```
CLR1   RWAIT           ;SEC ~ YEARカウンタ動作設定
BT     RWST,$LMAIN300  ;カウンタ値の読み出し/書き込みモード解除になるまでウエイト
```

```
EI           ;読み出し終了。割り込み禁止解除。
```

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

LMAIN500:

```
NOP           ;割り込み機能を制御するレジスタに対する操作命令実行中は、
              ;割り込み要求の受付を保留する。
              ;割り込み不可になるの防止するため、NOP命令を実行。
```

```
.*****
;
;
;   各種メイン処理
;
;
;*****
```

```
; 各種メイン処理があればここでいきます。
```

```
BR     MAIN_LOOP
```

```
.*****
;
;
;   UART6送信データ作成 & データ送信
;
;-----
;   [ IN ] RSEC,RMIN,RHOUR,RWEEK,RDAY,RMONTH,RYEAR:RTCカウンタ値
;   [ OUT ] なし
;
;   リアルタイム・カウンタから読み出した値を、
;   ASCIIコードに変換して送信バッファに設定し、送信します。
;   送信開始前に低電圧を検出していた場合は、UART6の再設定を行います。
;   送信完了待ち中に低電圧を検出した場合は、UART6の送信動作を中断します。
;
;   < 送信データの内容 >
;
;   0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 18 19 20 21
;
;   '  Y  Y  /  M  M  /  D  D      W      h  h  :  m  m  :  s  s  ¥r  ¥n
;
;*****
```

```

;          (Y:年, M:月, D:日, W:曜日, h:時, m:分, s:秒)
;*****
SUART6TX:
;-----
; 送信バッファにUART6送信データ作成処理
;-----
MOV     RTXBUF,#""           ;[0]クォーテーション(年データ略表示)保存

MOV     A,RYEAR              ;年データを取得
CALL    !SNUM2CHR            ;年データをASCIIコードに変換
MOV     (RTXBUF+1),A         ;[1]年データ上位保存
MOV     A,X
MOV     (RTXBUF+2),A         ;[2]年データ下位保存

MOV     (RTXBUF+3),#'/'      ;[3]スラッシュ(年月日データ区切り用)保存

MOV     A,RMOTH              ;月データを取得
CALL    !SNUM2CHR            ;月データをASCIIコードに変換
MOV     (RTXBUF+4),A         ;[4]月データ上位保存
MOV     A,X
MOV     (RTXBUF+5),A         ;[5]月データ下位保存

MOV     (RTXBUF+6),#'/'      ;[6]スラッシュ(年月日データ区切り用)保存

MOV     A,RDAY               ;日データを取得
CALL    !SNUM2CHR            ;日データをASCIIコードに変換
MOV     (RTXBUF+7),A         ;[7]日データ上位保存
MOV     A,X
MOV     (RTXBUF+8),A         ;[8]日データ下位保存

MOV     (RTXBUF+9),#' '      ;[9]スペース保存

MOV     A,RWEEK              ;曜日データを取得
CALL    !SNUM2CHR            ;曜日データをASCIIコードに変換
MOV     (RTXBUF+10),A        ;[10]曜日データ保存

MOV     (RTXBUF+11),#' '     ;[11]スペース保存

MOV     A,RHOUR              ;時データ取得
CALL    !SNUM2CHR            ;時データをASCIIコードに変換
MOV     (RTXBUF+12),A        ;[12]時データ上位保存
MOV     A,X
MOV     (RTXBUF+13),A        ;[13]時データ下位保存

```

```

MOV    (RTXBUF+14),#:'      ;[14]コロン(時間データ区切り用)保存

MOV    A,RMIN                ;分データ取得
CALL   !SNUM2CHR            ;分データをASCIIコードに変換
MOV    (RTXBUF+15),A        ;[15]分データ上位保存
MOV    A,X
MOV    (RTXBUF+16),A        ;[16]分データ下位保存

MOV    (RTXBUF+17),#:'      ;[17]コロン(時間データ区切り用)保存

MOV    A,RSEC                ;秒データ取得
CALL   !SNUM2CHR            ;秒データをASCIIコードに変換
MOV    (RTXBUF+18),A        ;[18]秒データ上位保存
MOV    A,X
MOV    (RTXBUF+19),A        ;[19]秒データ下位保存

MOV    (RTXBUF+20),#0DH      ;[20]キャリッジリターン保存
MOV    (RTXBUF+21),#0AH      ;[21]ラインフィード保存

;-----
;          UART6データ送信
;-----

BF     FUARTSTOP,$JU6TX500   ;低電圧が検出されている?   NO

;==== シリアル・インタフェースUART6再設定 ====
CLR1   POWER6                ;内部動作クロックの動作禁止
CLR1   TXE6                  ;送信動作禁止

DI                      ;UART6再設定中は割り込み禁止
SET1   TXE6                  ;送信動作許可
SET1   POWER6                ;内部動作クロックの動作許可
CLR1   FUARTSTOP              ;UART送信中断フラグをクリア
EI                      ;割り込み許可

JU6TX500:
;===== 送信動作開始 =====
MOV    B,#22                 ;送信カウンタ設定
MOVW   HL,#RTXBUF

JU6TX600:
CLR1   STIF6                 ;割り込み要求クリア
MOV    A,[HL]                ;送信バッファから送信データを取得

```

```

MOV    TXB6,A           ;データ送信
JU6TX700:
BT     FUARTSTOP,$JU6TX800 ;UART6送信中断フラグがON? YES:送信完了を待たずに終了
BF     STIF6,$JU6TX700    ;UART6 1Byte送信完了? NO
CLR1   STIF6            ;割り込み要求クリア
INCW   HL               ;送信バッファの送信データ位置更新
DBNZ   B,$JU6TX600      ;未送信データあり? YES :次のデータを送信

```

```

JU6TX800:                ;送信終了
RET

```

```

;*****
;
;
;   BCDをASCIIコードに変換
;
;-----
;   [ IN ] A:数値(BCD)
;   [ OUT ] A:上位4bitのASCIIコード
;           X:下位4bitのASCIIコード
;*****

```

```

SNUM2CHR:
MOV    X,A
AND    A,#0FH           ;下位4ビットを取得
ADD    A,#'0'          ;ASCIIコードに変換
XCH    A,X              ;戻り値保存 & 変換対象数値取得

AND    A,#0F0H          ;上位4ビットを取得
ROL    A,1              ;上位4bitを下位4bitに落として取得
ROL    A,1
ROL    A,1
ROL    A,1
ADD    A,#'0'          ;ASCIIコードに変換
RET

```

```

;*****
;
;
;   低電圧検出完了割り込み(INTLVI使用)
;
;-----
;   VDDが検出電圧以上，または検出電圧より低くなった場合に割り込みが発生する。
;   検出電圧は1.93V。
;   レジスタ・バンク1使用
;*****

```



```

IINTLVI:
    SEL    RB1                ;割り込み処理でのレジスタ・バンク設定

HLVI100:
    BF     LVIF,$HLVI600     ;VDD < 1.93Vとなった?  YES

    SET1   FUARTSTOP        ;UART中断フラグをONにする

HLVI300:
    BF     RWST,$HLVI400     ;リアルタイム・カウンタが読み出し中の場合
    CLR1   RWAIT            ;読み出し中設定を解除してからSTOPモードに入る
    BR     HLVI300          ;読み出し中設定が解除されるのを待つ

HLVI400:
    STOP                   ;STOPモードへ移行
    BR     HLVI100         ;INTLVIでSTOPモードから復帰

HLVI600:
    ;VDD    1.93Vとなったので , STOPモードを解除する

HLVI900:
    RETI

end

```

main.c (C言語版)

/\*\*\*\*\*

NEC Electronics 78K0/Lx3シリーズ

\*\*\*\*\*

78K0/LF3シリーズ サンプル・プログラム

\*\*\*\*\*

低電圧時のリアルタイム・カウンタ動作継続編

\*\*\*\*\*

## 【履歴】

2008.5.-- 新規作成

\*\*\*\*\*

## 【概要】

本サンプルプログラムは、低電圧状態でリアルタイム・カウンタのカウント動作を継続するプログラムです。

1秒毎にリアルタイム・カウンタのカウント・レジスタの値を取得し、シリアル・インタフェースUART6にて送信します。電源電圧が1.93V以下になるとSTOPモードに入ります。電源電圧が1.93Vを超えるとSTOPモードから復帰します。

本サンプルプログラムのUARTは、送信のみを行います。

## &lt; 初期設定の主な設定内容 &gt;

- ・ベクタ・テーブルの設定
- ・レジスタ・バンクの設定
- ・スタック・ポインタの設定
- ・ROM/RAMサイズの設定
- ・ポートの設定
- ・CPUクロック周波数を高速内蔵発振8MHz(TYP.)に設定
- ・リアルタイム・カウンタの設定
- ・低電圧検出回路の設定
- ・シリアル・インタフェースUART6の設定
- ・割り込みマスクの設定

## &lt; メイン処理の主な内容 &gt;

- ・リアルタイム・カウンタのカウント動作停止
- ・リアルタイム・カウンタのカウント・レジスタの値を取得

- ・リアルタイム・カウンタのカウンタ動作設定
- ・UART6の送信データ作成&データ送信処理

< UART6の送信データ作成 & データ送信処理の主な内容 >

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

< 低電圧検出割り込み(INTLVI使用) >

- ・リアルタイム・カウンタのカウンタ動作設定
- ・STOP命令実行

```

*****/
#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 STOP              /* STOP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 INTLVI  fn_intlvi RB1 /* 割り込み関数宣言:INTLVI */

```

/\*=====

関数プロトタイプ宣言

```

=====*/
static void fn_UART6_Tx(void);          /* UART6の送信データ作成 & データ送信処理 */

```

/\*=====

R A Mの定義

```

=====*/
/* リアルタイム・カウンタからのデータ取得用変数 */
unsigned char   ucSec           /* 秒 */
                ,ucMin         /* 分 */
                ,ucHour        /* 時 */
                ,ucWeek        /* 曜日 */
                ,ucYear         /* 年 */

```

```

        ,ucMonth                /* 月 */
        ,ucDay;                 /* 日 */

unsigned char ucTxBuffer[22];   /* 送信データ・バッファ */
unsigned char ucTxBufferCounter; /* 送信カウンタ */

boolean bUARTStop;            /* UART送信中断フラグ */

/*****

                リセット解除後の初期化処理

*****/

void hdwinit(void)
{
    unsigned short temp;       /* ワーク領域 */

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

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

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

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

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

    /* uPD78F0475,uPD78F0485,uPD78F0495使用時の設定 */
    IMS = 0xCF;                 /* ROMサイズの設定 */

```

IXS = 0x0A; /\* 内部拡張RAMサイズの設定 \*/

/\*-----\*/

ポートの設定 (未使用ポートはLow出力に設定)

-----\*/

/\* ポート1 \*/

P1 = 0b00000000; /\* P1初期値設定 \*/

/\*+++++----- P17/P16/P15/P14/P13/P12/P11/P10:未使用(0) \*/

PM1 = 0b00000000; /\* P1入出力設定 \*/

/\*+++++----- PM17/PM16/PM15/PM14/PM13/PM12/PM11/PM10:未使用(0) \*/

/\* ポート2 \*/

P2 = 0b00000000; /\* P2初期値設定 \*/

/\*+++++----- P27/P26/P25/P24/P23/P22/P21/P20:未使用(0) \*/

PM2 = 0b00000000; /\* P1入出力設定 \*/

/\*+++++----- PM27/PM26/PM25/PM24/PM23/PM22/PM21/PM20:未使用(0) \*/

/\* ポート3 \*/

P3 = 0b00000000; /\* P3初期値設定 \*/

/\*|||++++----- P34/P33/P32/P31/P30:未使用(0) \*/

/\*+++----- <000固定> \*/

PM3 = 0b11100000; /\* P3入出力設定 \*/

/\*|||++++----- PM34/PM33/PM32/PM31/PM30:未使用(0) \*/

/\*+++----- <111固定> \*/

/\* ポート4 \*/

P4 = 0b00000000; /\* P4初期値設定 \*/

/\*+++++----- P47/P46/P45/P44/P43/P42/P41/P40:未使用(0) \*/

PM4 = 0b00000000; /\* P4入出力設定 \*/

/\*+++++----- PM47/PM46/PM45/PM44/PM43/PM42/PM41/PM40:未使用(0) \*/

/\* ポート8 \*/

P8 = 0b00000000; /\* P8初期値設定 \*/

/\*||||++++----- P83/P82/P81/P80:未使用(0) \*/

/\*++++----- <0000固定> \*/

PM8 = 0b11110000; /\* P8入出力設定 \*/

/\*||||++++----- PM83/PM82/PM81/PM80:未使用(0) \*/

/\*++++----- <1111固定> \*/

/\* ポート9 \*/

P9 = 0b00000000; /\* P9初期値設定 \*/

/\*||||++++----- P93/P92/P91/P90:未使用(0) \*/

/\*++++----- <0000固定> \*/

PM9 = 0b11110000; /\* P9入出力設定 \*/

/\*||||++++----- PM93/PM92/PM91/PM90:未使用(0) \*/

/\*++++----- <1111固定> \*/

/\* ポート10 \*/

P10 = 0b00000000; /\* P10初期値設定 \*/

```

/*|||++++----- P103/P102/P101/P100:未使用(0) */
/*++++----- <0000固定> */
PM10 = 0b11110000;          /* P10入出力設定 */
/*|||++++----- PM103/PM102/PM101/PM100:未使用(0) */
/*++++----- <1111固定> */
/* ポート11 */
P11 = 0b00000100;          /* P11初期値設定 */
/*|||+|++----- P113/P111/P110:未使用(0) */
/*||| +----- P112:Hi(1)*/
/*++++----- <0000固定> */
PM11 = 0b11110000;          /* P11入出力設定 */
/*|||+|++----- PM113/PM111/PM110:未使用(0) */
/*||| +----- PM112:出力(0) TxD6として使用*/
/*++++----- <1111固定> */
/* ポート12 */
P12 = 0b00000000;          /* P12初期値設定 */
/*|||||+----- P120:未使用(0) */
/*|||++++----- P124/P123/P122/P121:Read Only */
/*++++----- <000固定> */
PM12 = 0b11111110;          /* P12入出力設定 */
/*|||||+----- PM120:未使用(0) */
/*+++++----- <1111111固定> */
/* ポート13 */
P13 = 0b00000000;          /* P13初期値設定 */
/*|||++++----- P133/P132/P131/P130:未使用(0) */
/*++++----- <0000固定> */
PM13 = 0b11110000;          /* P13入出力設定 */
/*|||++++----- PM133/PM132/PM131/PM130:未使用(0) */
/*++++----- <1111固定> */
/* ポート14 */
P14 = 0b00000000;          /* P14初期値設定 */
/*|||++++----- P143/P142/P141/P140:未使用(0) */
/*++++----- <0000固定> */
PM14 = 0b11110000;          /* P14入出力設定 */
/*|||++++----- PM143/PM142/PM141/PM140:未使用(0) */
/*++++----- <1111固定> */
/* ポート15 */
P15 = 0b00000000;          /* P15初期値設定 */
/*|||++++----- P153/P152/P151/P150:未使用(0) */
/*++++----- <0000固定> */
PM15 = 0b11110000;          /* P15入出力設定 */
/*|||++++----- PM153/PM152/PM151/PM150:未使用(0) */
/*++++----- <1111固定> */

```

```
/*-----
```

### クロック周波数の設定

```
-----
```

高速内蔵発振8MHz(TYP.)で動作を行うように設定します。

```
-----*/
```

```
OSCCTL = 0b00010000;          /* クロック動作モード */
/*|||++++----- <0固定> */
/*||+----- OSCSELS: サブシステム・クロック端子の設定 */
/*||              サブシステム・クロック端子の動作モード:XT1発振モード */
/*||              P123/XT1,P124/XT2:水晶 / セラミック発振子接続 */
/*||+----- <0固定> */
/*++----- EXCLK/OSCSEL: */
/*              高速システム・クロック端子の動作モード:入力ポート・モード */
/*              P121/X1,P122/X2/EXCLK: 入力ポート */
```

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

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

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

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

```
/*-----
```

### リアルタイム・カウンタの設定

```
-----
```

1秒毎にUARTでリアルタイム・カウンタの値を送信するため、  
年月日と時間のカウントと、UART送信間隔の測定を行います。

```

-----*/

RTCCL = 0b00000000;          /* リアルタイム・カウンタ・クロック選択レジスタ */
/*|||||++----- RTCCL1/RTCCL0:入力クロック (fRTC) 選択 */
/*|||||          fRTC = fSUB(32.768kHz) */
/*+++++----- <000000固定> */

RTCC0 = 0b00001010;         /* リアルタイム・カウンタ・コントロール・レジスタ0 */
/*|||||+++----- CT2/CT1/CT0:1秒に1度定周期割り込み (INTRTC) を発生 */
/*|||+----- AMPM:24時間制 */
/*||+----- RCLOE0:RTCCL端子の出力 ( 32.768 kHz) 禁止 */
/*|+----- RCLOE1:RTC1HZ端子の出力 ( 1 Hz) 禁止 */
/*|+----- <0固定> */
/*+----- RTCE:リアルタイム・カウンタ動作停止 */

RTCC1 = 0b00001000;         /* リアルタイム・カウンタ・コントロール・レジスタ1 */
/*|||||+----- RWAIT:カウンタ動作設定 */
/*|||||+----- RWST:Read Only */
/*|||||+----- <0固定> */
/*|||+----- RIFG:定周期割り込み発生クリア */
/*||+----- WAFG:無効 */
/*|+----- <0固定> */
/*|+----- WALIE:アラームの一致による割り込み (INTRTC) を発生しない */
/*+----- WALE:一致動作無効 */

RTCC2 = 0b00000000;         /* リアルタイム・カウンタ・コントロール・レジスタ2 */
/*+||||+++----- RINTE/CT2/CT1/CT0:インターパル割り込みを発生しない。 */
/* ||++----- <00固定> */
/* |+----- RCKDIV:無効 */
/* +----- RCLOE2:RTCDIV端子の出力禁止 */

RTCIF = 0;          /* 定周期割り込み要求をクリア */
RTCE = 1;          /* リアルタイム・カウンタ動作開始 */

/*-----*/
/*   サブシステム・クロックの発振安定待ち   */
/*-----*/
/*   サブシステム・クロックの発振安定待ちに1秒の定周期割り込みを使用します。 */
/*   最初の定周期割り込み要求が発生した後, リアルタイム・カウンタを再スタートさせます。 */
while(!RTCIF)
{
    NOP();
}

```



```
RTCIF = 0;      /* 定周期割り込み要求をクリア */
RTCE = 0;      /* リアルタイム・カウンタ動作停止 */
```

```
/* カウント・レジスタ初期設定：'08/06/01 (日) 00:00:00 */
```

```
SEC = 0;        /* 秒：00秒 */
MIN = 0;        /* 分：00分 */
HOUR = 0;       /* 時：00時 */
WEEK = 0;       /* 曜日：日曜日 */
DAY = 1;        /* 日：1日 */
MONTH = 6;      /* 月：6月 */
YEAR = 8;       /* 年：08年 */
```

```
RTCE = 1;      /* リアルタイム・カウンタ動作開始 */
```

```
/*-----
```

```
低電圧検出回路の設定
```

```
-----*/
```

```
LVIS = 0b00001111;      /* 低電圧検出レベル選択レジスタ */
/*|||+----- LVIS3/LVIS2/LVIS1/LVIS0:検出レベルはVLVI15 (1.93 V ± 0.1 V) */
/*++++----- <0000固定> */
```

```
LVIM = 0b10000000;      /* 低電圧検出レジスタ */
/*|||||+----- LVIF:Read Only */
/*|||||+----- LVIMD:電圧降下時に電源電圧 (VDD) < 検出電圧 (VLVI) になったとき, */
/*|||||          または, 電圧上昇時にVDD > VLVIになったとき内部割り込み信号を発生
```

```
*/
```

```
/*|||||+----- LVISEL:電源電圧 (VDD) のレベルを検出 */
/*|++++----- <0000固定> */
/*+----- LVION:低電圧検出動作許可 */
```

```
/* LVIONをセット(1)してからLVIFで電圧を確認するまでの時間(10 μs(MAX.))をウエイト */
```

```
/* 10μs == 80CLK(8MHz) */
```

```
for(temp = 2; temp > 0; temp--)
{
    NOP();
}
```

```
/*-----
```

```
シリアル・インタフェースUART6の設定
```

```
-----
```

```
リアルタイム・カウンタでのカウント値を送信します。
```

```
-----*/
```

```
CKSR6 = 0b00000000;      /* UART6基本クロック選択 */
```

```

/*|||++++----- TPS63-60: 基本クロック(fXCLK6) = fPRS) */
/*++++----- <0固定> */

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

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

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

ISC = 0b00001000; /* 入力切り替え制御 */
/*|||++++----- ISC0: 未使用 */
/*|||++++----- ISC1: TI000入力をそのまま使用する(通常動作) */
/*|||++++----- ISC2: 未使用 */
/*|||++++----- ISC3: RxD6/P113入力許可 */
/*||+----- ISC5-4: TxD6=P112,RxD6=P113 */
/*++----- <0固定> */

POWER6 = 1; /* 内部動作クロックの動作許可 */

/*-----
割り込みマスクの設定
-----*/

MK0 = 0x0FFFF;
MK1 = 0x0FFFF; /* 全ての割り込みをマスク */

LVIIIF = 0; /* 低電圧検出回路割り込み要求クリア */
LVIMK = 0; /* 低電圧検出回路割り込みマスク解除 */

```

```

    EI();          /* 割り込み許可 */
}

/*****

メイン・ループ

*****/

void main(void)
{
    /* リセット解除後の電源電圧を低電圧割り込み処理でチェック、 */
    /* すでに検出電圧以下の場合はSTOPモードに入る */
    LVIIIF = 1;      /* 低電圧検出割り込み要求オン */

    while(1){
        /*****
        /*
        /*          RTCカウント値送信処理          */
        /*
        /*****
        if(RTCIF)
        { /* 1秒経過 */
            RTCIF = 0;      /* 割り込み要求をクリア */
            /*-----*/
            /* リアルタイム・カウンタからデータを取得 */
            /*-----*/
            DI(); /* リアルタイム・カウンタ読み出し中はSEC～YEARカウンタが停止するため、 */
                /* そのままSTOPモードに入ると時間が進まない。読み出し中は割り込み禁止に
設定する */

            do{
                RWAIT = 1;      /* SEC～YEARカウンタ停止設定(カウンタ値読み出し/書き込
みモード) */

            }while(!RWST); /* カウンタ値の読み出し/書き込みモード中になるまでウエイト */
                /* (RWAIT = 1に設定後、32.768 kHzで最大1クロック(30.5 μs) */

            ucSec = SEC;      /* 秒データ取得 */
            ucMin = MIN;      /* 分データ取得 */
            ucHour = HOUR;    /* 時データ取得 */
            ucWeek = WEEK;    /* 曜日データ取得 */

```

```

ucDay = DAY;          /* 日データ取得 */
ucMonth = MONTH;     /* 月データ取得 */
ucYear = YEAR;       /* 年データ取得 */

do{
    RWAIT = 0;       /* SEC~YEARカウンタ動作設定 */
}while(RWST);       /* カウンタ値の読み出し/書き込みモード解除になるまでウエイト */

EI(); /* 読み出し終了。割り込み禁止解除。 */

/*-----*/
/*   UART6送信データ作成&データ送信   */
/*-----*/
fn_UART6_Tx();
}

NOP(); /* 割り込み機能を制御するレジスタに対する操作命令実行中は、 */
/* 割り込み要求の受付を保留する。 */
/* 割り込み不可になるの防止するため、NOP命令を実行。 */

/*****/
/*                               */
/*   各種メイン処理               */
/*                               */
/*                               */
/*****/

/* 各種メイン処理があればここでいきます。 */
}
}

```

```

/*****

```

#### UART6送信データ作成 & データ送信

-----

[ IN ] なし  
[ OUT ] なし

リアルタイム・カウンタから読み出した値を、  
ASCIIコードに変換して送信バッファに設定し、送信します。  
送信開始前に低電圧を検出していた場合は、UART6の再設定を行います。  
送信完了待ち中に低電圧を検出した場合は、UART6の送信動作を中断します。

< 送信データの内容 >

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21  
 ' Y Y / M M / D D W h h : m m : s s ¥r ¥n

(Y:年, M:月, D:日, W:曜日, h:時, m:分, s:秒)

\*\*\*\*\*/

static void fn\_UART6\_Tx(void)

{

/\*  
 \*\*\*\*\*/

/\*  
 /\* UART6送信データ作成処理 \*/  
 /\*  
 /\*  
 \*\*\*\*\*/

ucTxBuffer[0] = '¥'; /\* クォーターション(年データ略表示) \*/  
 ucTxBuffer[1] = (( ucYear & ( 0xF0 )) >> 4 ) + '0'; /\* 年データ上位(ASCIIコードに変換) \*/  
 ucTxBuffer[2] = ( ucYear & ( 0x0F )) + '0'; /\* 年データ下位(ASCIIコードに変換) \*/  
 ucTxBuffer[3] = '/'; /\* スラッシュ(年月日データ区切り用) \*/  
 ucTxBuffer[4] = (( ucMonth & ( 0xF0 )) >> 4 ) + '0'; /\* 月データ上位(ASCIIコードに変換) \*/  
 ucTxBuffer[5] = ( ucMonth & ( 0x0F )) + '0'; /\* 月データ下位(ASCIIコードに変換) \*/  
 ucTxBuffer[6] = '/'; /\* スラッシュ(年月日データ区切り用) \*/  
 ucTxBuffer[7] = (( ucDay & ( 0xF0 )) >> 4 ) + '0'; /\* 日データ上位(ASCIIコードに変換) \*/  
 ucTxBuffer[8] = ( ucDay & ( 0x0F )) + '0'; /\* 日データ下位(ASCIIコードに変換) \*/  
 ucTxBuffer[9] = ' '; /\* スペース \*/  
 ucTxBuffer[10] = ( ucWeek & ( 0x0F )) + '0'; /\* 曜日データ \*/  
 ucTxBuffer[11] = ' '; /\* スペース \*/  
 ucTxBuffer[12] = (( ucHour & ( 0xF0 )) >> 4 ) + '0'; /\* 時データ上位(ASCIIコードに変換) \*/  
 ucTxBuffer[13] = ( ucHour & ( 0x0F )) + '0'; /\* 時データ下位(ASCIIコードに変換) \*/  
 ucTxBuffer[14] = ':'; /\* コロン(時間データ区切り用) \*/  
 ucTxBuffer[15] = (( ucMin & ( 0xF0 )) >> 4 ) + '0'; /\* 分データ上位(ASCIIコードに変換) \*/  
 ucTxBuffer[16] = ( ucMin & ( 0x0F )) + '0'; /\* 分データ下位(ASCIIコードに変換) \*/  
 ucTxBuffer[17] = ':'; /\* コロン(時間データ区切り用) \*/  
 ucTxBuffer[18] = (( ucSec & ( 0xF0 )) >> 4 ) + '0'; /\* 秒データ上位(ASCIIコードに変換) \*/  
 ucTxBuffer[19] = ( ucSec & ( 0x0F )) + '0'; /\* 秒データ下位(ASCIIコードに変換) \*/  
 ucTxBuffer[20] = '¥r'; /\* キャリッジリターン \*/  
 ucTxBuffer[21] = '¥n'; /\* ラインフィード \*/

/\*  
 \*\*\*\*\*/

/\*  
 /\* UART6データ送信 \*/  
 /\*  
 /\*  
 \*\*\*\*\*/

/\*-----\*/  
 /\* シリアル・インタフェースUART6再設定 \*/

```

/*-----*/
if(bUARTStop) /* すでに低電圧が検出されている場合は，UART6の再設定が必要 */
{
    POWER6 = 0;          /* 内部動作クロックの動作禁止 */
    TXE6 = 0;           /* 送信動作禁止 */

    DI();               /* UART6再設定中は割り込み禁止 */
    TXE6 = 1;           /* 送信動作許可 */
    POWER6 = 1;        /* 内部動作クロックの動作許可 */
    bUARTStop = 0;     /* UART送信中断フラグをクリア */
    EI();               /* 割り込み許可 */
}

/*-----*/
/*          送信動作開始          */
/*-----*/
for(ucTxBufferCounter = 0; ucTxBufferCounter < sizeof(ucTxBuffer); ucTxBufferCounter++)
/* 全データ送信完了するまで，送信動作継続 */
    STIF6 = 0;          /* 割り込み要求クリア */
    TXB6 = ucTxBuffer[ucTxBufferCounter]; /* データ送信 */

    while(!STIF6)
    /* UART6で1Byte送信完了待ち */
        if(bUARTStop)
        /* 送信途中で送信中断フラグがONになった場合，送信動作を中断 */
            ucTxBufferCounter = sizeof(ucTxBuffer);
            break;
    }
}
}

```

```

/*****

```

低電圧検出完了割り込み(INTLVI使用)

-----

VDDが検出電圧以上，または検出電圧より低くなった場合に割り込みが発生する。  
検出電圧は1.93V。

```

*****/
__interrupt void fn_intlvi(void)
{
    while(LVIF == 1) /* VDD < 1.93V の場合はSTOPモードに入る */
    {

```

```
bUARTStop = 1; /* UART6送信中断フラグをONにする */

while(RWST)      /* リアルタイム・カウンタが読み出し中の場合 */
{
    RWAIT = 0;   /* 読み出し中設定を解除してからSTOPモードに入る。 */
}
STOP();         /* STOPモードへ移行 */
}

/* VDD 1.93V の場合はSTOPモードを解除する */
}
```

## 付録B 改版履歴

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



## 【発 行】

NECエレクトロニクス株式会社

〒211-8668 神奈川県川崎市中原区下沼部1753

電話（代表）：044(435)5111

—— お問い合わせ先 ——

---

## 【ホームページ】

NECエレクトロニクスの情報がインターネットでご覧になれます。

URL(アドレス) <http://www.necel.co.jp/>

---

## 【営業関係，技術関係お問い合わせ先】

半導体ホットライン

（電話：午前 9:00～12:00，午後 1:00～5:00）

電 話 : 044-435-9494

E-mail : [info@necel.com](mailto: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。

---