お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願い申し上げます。

ルネサスエレクトロニクス ホームページ（http://www.renesas.com）

2010年4月1日
ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社（http://www.renesas.com）

【問い合わせ先】http://japan.renesas.com/inquiry

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したものですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。

   標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット

   高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）

   特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

RENESAS

アプリケーション・ノート

78K0S/Kx1+

サンプル・プログラム（低電圧検出）

2.7 V未満検出時リセット発生編

この資料は，サンプル・プログラムの動作概要や使用方法，および低電圧検出機能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，V_{DD}＜V_{LVI}（V_{LVI} = 2.85 V±0.15 V）を検出して，内部リセット（LVIリセット）信号を発生します。LVIリセットでは，リセット直前のLED表示データが保持され，リセット解除後にLED出力が復元されます。

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

資料番号　U18821JJ2V0AN00（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，ソフトウエ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，ソフトウエ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。

M8E　02.11

第1章　概　　要

このサンプル・プログラムでは，低電圧検出（LVI）機能の使用例を示しています。

LVI回路を使用し，V_{DD}＜V_{LVI}（V_{LVI} = 2.85 V±0.15 V）を検出して，内部リセット（LVIリセット）信号を発生するように設定します。

初期設定完了後は，スイッチ入力の立ち下がりエッジを検出して割り込み処理を行い，スイッチの入力回数に応じたLED点灯パターンを表示します（この処理は，「サンプル・プログラム（割り込み）」と同様です）。

LVI以外によるリセットが発生した場合では，プログラムにてスイッチの入力回数を初期化します。LVIリセットが発生した場合では，POC電圧を下回らないかぎり，RAMはリセット直前のデータを保持するため，LVIリセット解除後に，リセット発生直前のスイッチの入力回数を復元し，それに応じたLED点灯パターンを表示します。

1. 1　初期設定の主な内容

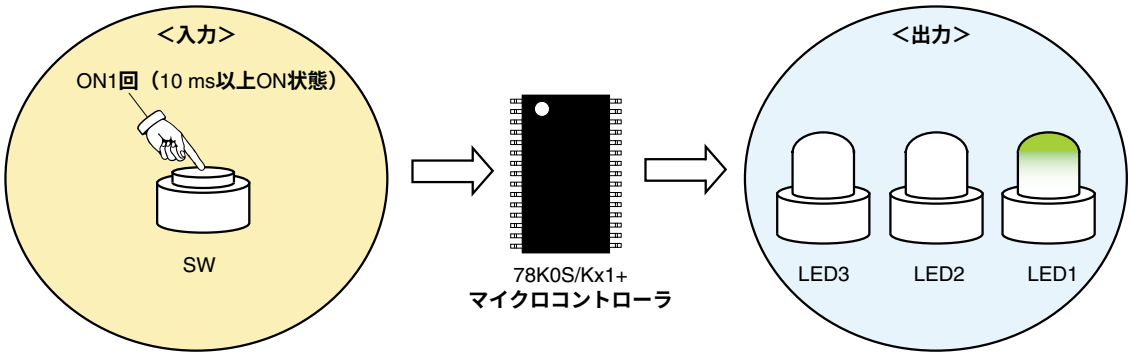
初期設定の主な内容は，次のとおりです。

- システム・クロック・ソースとして，高速内蔵発振器を選択[注]
- 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2.85 V±0.15 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4 MHzに設定
- 入出力ポートの設定
- INTP1（外部割り込み）の有効エッジを立ち下がりエッジに設定
- 割り込み許可

注　オプション・バイトで設定します。

1.2　メイン・ループ以降の内容

スイッチ入力によるINTP1端子の立ち下がりエッジを検出し，割り込み処理を行います。割り込み処理では，INTP1端子の立ち下がりエッジを検出してから約10 ms経過後に，スイッチがONであることを確認し，LEDの点灯パターンを変化させます。約10 ms経過後に，スイッチがOFFである場合は，チャタリングであると判定し，LEDの点灯パターンを変化させません。



スイッチの入力回数[注]	LED出力		
	LED3	LED2	LED1
0	OFF	OFF	OFF
1	OFF	OFF	ON
2	OFF	ON	OFF
3	OFF	ON	ON
4	ON	OFF	OFF
5	ON	OFF	ON
6	ON	ON	OFF
7	ON	ON	ON

注　8回目以降は，0回目からの点灯パターンの繰り返しになります。

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

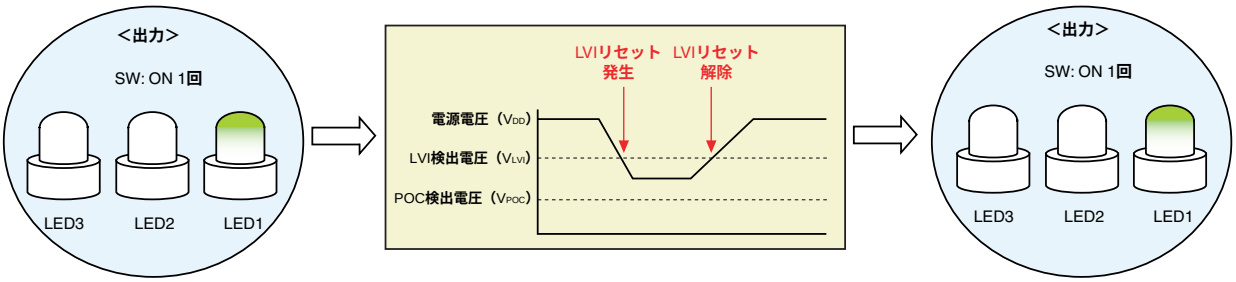
【コラム】チャタリングとは

スイッチが切り替わった直後に，接点が機械的にばたつくことにより，電気信号がONとOFFを繰り返す現象のことです。

1. 3　低電圧検出（LVI）機能の内容

このサンプル・プログラムでは，V_{DD}＜V_{LVI}になった場合，低電圧検出（LVI）機能による内部リセット（LVIリセット）を発生します。このとき，レジスタ値は初期化されますが，RAMはPOC電圧を下回らないかぎり，リセット直前のデータを保持します。したがって，LVIリセット時は，リセット直前のスイッチの入力回数を保持し，リセット解除後にスイッチの入力回数を復元し，それに応じたLED点灯パターンを表示することが可能となります[注]。LVI以外のリセット時は，プログラムにてスイッチの入力回数を初期化し，全消灯となります。

注　このページの【コラム】にあるように，C言語のプログラムでは，標準のスタートアップ・ルーチンを使用する場合，main関数の前でRAMデータが初期化されます。これを防ぐために，このC言語版のサンプル・プログラムでは，標準のスタートアップ・ルーチンの一部をコメント・アウトし，初期値のないRAMデータが初期化（0にクリア）されないようにしています。



【コラム】スタートアップ・ルーチンの処理

標準のスタートアップ・ルーチンは，主に次の処理を行います。

- スタック・ポインタの設定
- ハードウエアの初期化（早期に行う必要のあるもの）
- ライブラリで使用する変数の初期化
- 初期値あり外部変数，sreg変数は，初期値をROMからRAMに転送
- 初期値なし外部変数，sreg変数は，RAMに0を代入[注]

注　このC言語版のサンプル・プログラムに同封されている，スタートアップ・ルーチンのソース・ファイル（cstart.asm）では，この処理のみコメント・アウトしています。

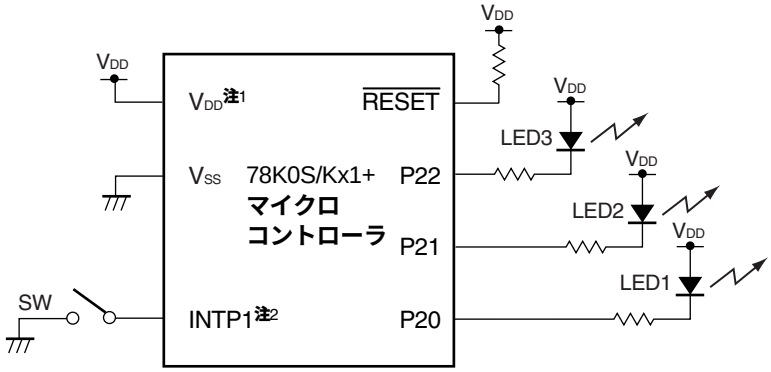
詳細については，CC78K0S　Cコンパイラ　操作編　ユーザーズ・マニュアルのスタートアップ・ルーチンの章を参照してください。

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場合の回路図および周辺ハードウエアを説明します。

2. 1　回路図

回路図を次に示します。



注1.　3.0 V≦V_{DD}≦5.5 Vの電圧範囲で使用してください。

2.　INTP1/P43: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ
INTP1/P32: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

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　周辺ハードウエア

使用する周辺ハードウエアを次に示します。

（1）スイッチ（SW）

LED点灯制御用の入力として，スイッチを使用します。

（2）LED（LED1, LED2, LED3）

スイッチ入力に対応した出力として，LEDを使用します。

第3章　ソフトウエアについて

この章では，ダウンロードする圧縮ファイルのファイル構成，使用するマイコンの内蔵周辺機能，サンプル・プログラムの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。

(1) アセンブリ言語版

ファイル名	説 明	同封圧縮（*.zip）ファイル	
			
main.asm	マイコンのハードウエ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）	●	●
lvi.prw	統合開発環境 PM+用ワーク･スペース・ファイル		●
lvi.prj	統合開発環境 PM+用プロジェクト・ファイル		●

(2) C言語版

ファイル名	説 明	同封圧縮（*.zip）ファイル	
			
main.c	マイコンのハードウエ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）	●	●
cstart.asm	スタートアップ・ルーチンのソース・ファイル（ROM化処理の一部をコメント・アウト）	●	●
def.inc	ライブラリ種別設定用ファイル（「cstart.asm」のインクルード・ファイル）	●	●
macro.inc	各種定型パターンについてのマクロ定義ファイル（「cstart.asm」のインクルード・ファイル）	●	●
lvi.prw	統合開発環境 PM+用ワーク･スペース・ファイル		●
lvi.prj	統合開発環境 PM+用プロジェクト・ファイル		●

備考　  　：ソース・ファイルのみ同封

 　：統合開発環境 PM+で使用するファイルを同封

3. 2　使用する内蔵周辺機能

このサンプル・プログラムでは，マイコンに内蔵する次の周辺機能を使用します。

- V_{DD}＜V_{LVI}検出　　：低電圧検出（LVI）回路
- スイッチ入力　　：INTP1[注]（外部割り込み）
- LED出力　　：P20, P21, P22（出力ポート）

注　INTP1/P43: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ
　　INTP1/P32: 78K0S/KY1+, 78K0S/KU1+マイクロコントローラ

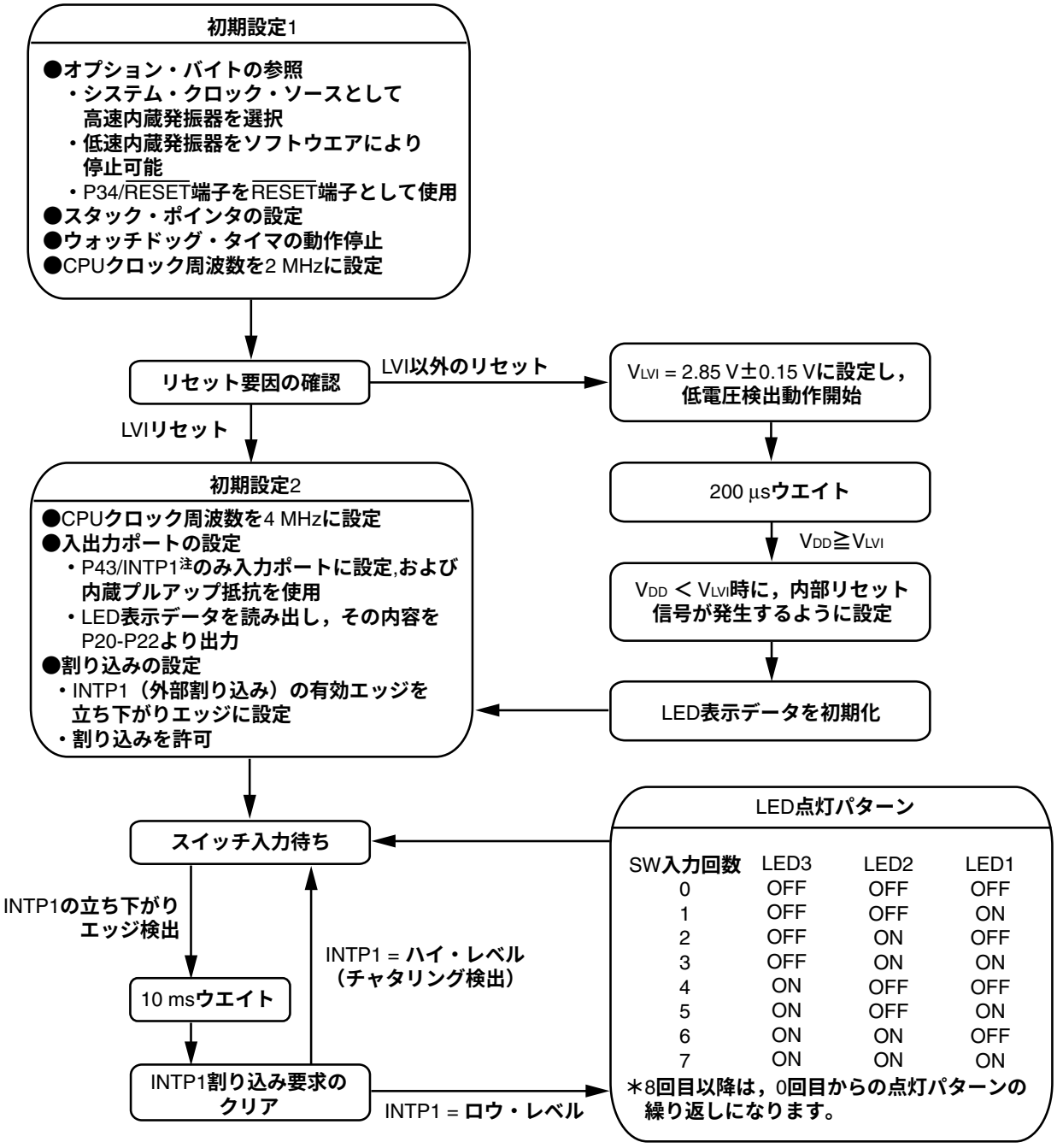
3. 3　初期設定と動作概要

このサンプル・プログラムでは，初期設定にて，クロック周波数の選択や，入出力ポートの設定，割り込みの設定，LVIの設定などを行います。

初期設定完了後は，スイッチ入力（SW）の立ち下がりエッジを検出して割り込み処理を行い，スイッチ入力回数に応じて，3つのLED（LED1, LED2, LED3）の点灯を制御します（この処理は，「サンプル・プログラム（割り込み）」と同様です）。

LVI以外によるリセットが発生した場合では，プログラムにてスイッチの入力回数を初期化します。LVIリセットが発生した場合では，POC電圧を下回らないかぎり，RAMはリセット直前のデータを保持するため，LVIリセット解除後に，リセット発生直前のスイッチの入力回数を復元し，それに応じたLED点灯パターンを表示します。

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。

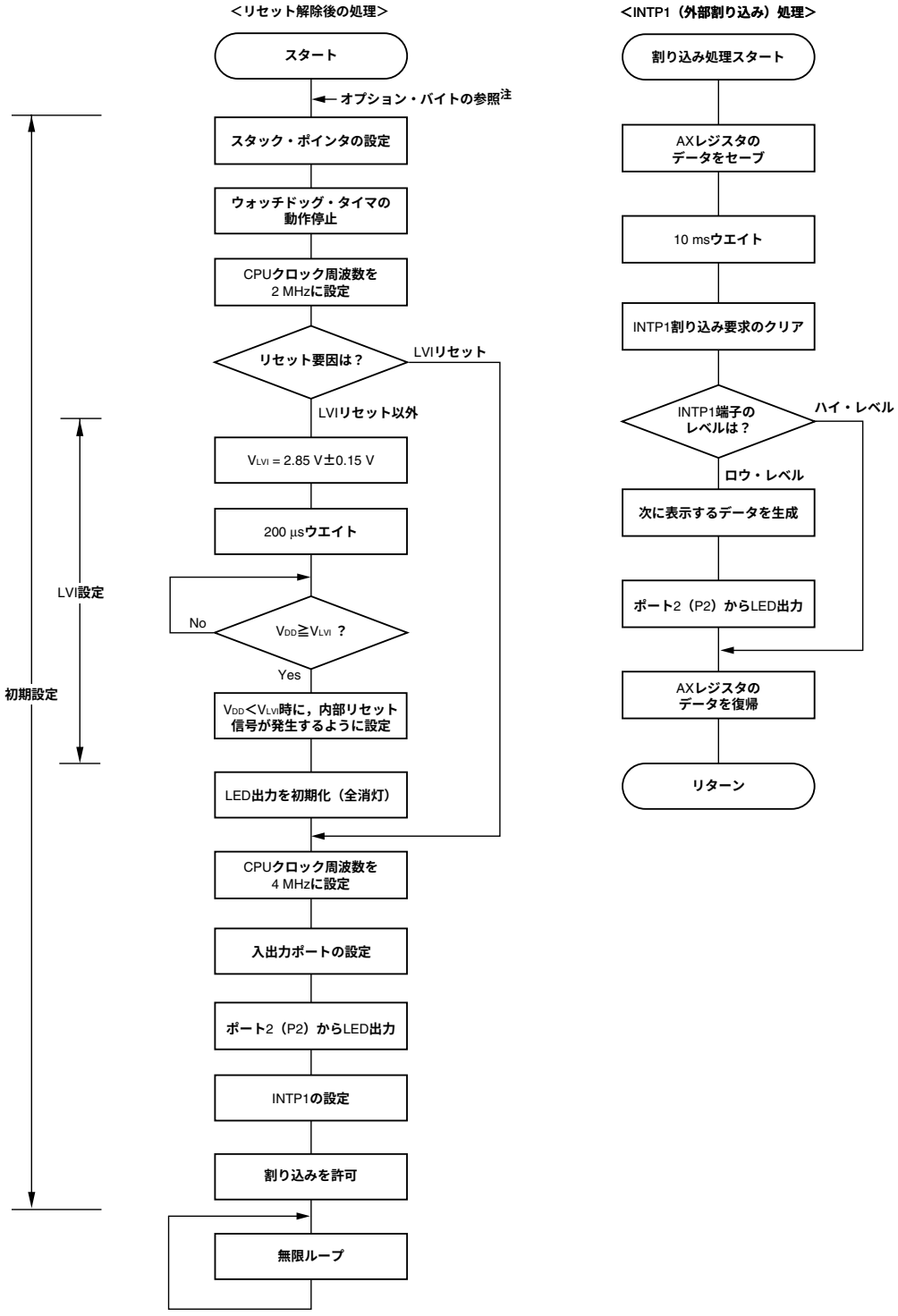


注　INTP1/P43: 78K0S/KA1+, 78K0S/KB1+マイクロコントローラ

INTP1/P32: 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エアで停止可**
- **P34/$\overline{\text{RESET}}$端子を$\overline{\text{RESET}}$端子として使用**

第4章　設定方法について

この章では，低電圧検出機能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，割り込みについては，78K0S/Kx1+　サンプル・プログラム（割り込み）　スイッチ入力による外部割り込み編　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，78K/0S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　低電圧検出（LVI）機能の設定

低電圧検出機能には，次の2種類の動作モードがあります。

・リセットとして使用（→【例　1】）
・割り込みとして使用（→【例　2】）

低電圧検出機能は，主に次の2種類のレジスタで制御します。

・低電圧検出レジスタ（LVIM）
・低電圧検出レベル選択レジスタ（LVIS）

(1) 低電圧検出動作についての設定

低電圧検出レジスタ（LVIM）で，低電圧検出の動作モードの設定，および動作の制御を行います。

図4－1　低電圧検出レジスタ（LVIM）のフォーマット

LVIM

LVION	0	0	0	0	0	LVIMD	LVIF

低電圧検出フラグ（Read Only）

0	$V_{DD} \geqq V_{LVI}$または動作禁止時
1	$V_{DD} < V_{LVI}$

低電圧検出の動作モード選択

0	$V_{DD} < V_{LVI}$時に割り込み信号発生
1	$V_{DD} < V_{LVI}$時に内部リセット信号発生

低電圧検出動作の許可／禁止

0	動作禁止
1	動作許可

(2) 低電圧検出レベルについての設定

低電圧検出レベル選択レジスタ（LVIS）で，低電圧検出レベルを設定します。

図4－2　低電圧検出レベル選択レジスタ（LVIS）のフォーマット

LVIS

0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

低電圧検出レベル

0	0	0	0	V_{LVI0}（4.3 V±0.2 V）
0	0	0	1	V_{LVI1}（4.1 V±0.2 V）
0	0	1	0	V_{LVI2}（3.9 V±0.2 V）
0	0	1	1	V_{LVI3}（3.7 V±0.2 V）
0	1	0	0	V_{LVI4}（3.5 V±0.2 V）
0	1	0	1	V_{LVI5}（3.3 V±0.15 V）
0	1	1	0	V_{LVI6}（3.1 V±0.15 V）
0	1	1	1	V_{LVI7}（2.85 V±0.15 V）
1	0	0	0	V_{LVI8}（2.6 V±0.1 V）
1	0	0	1	V_{LVI9}（2.35 V±0.1 V）
上記以外				設定禁止

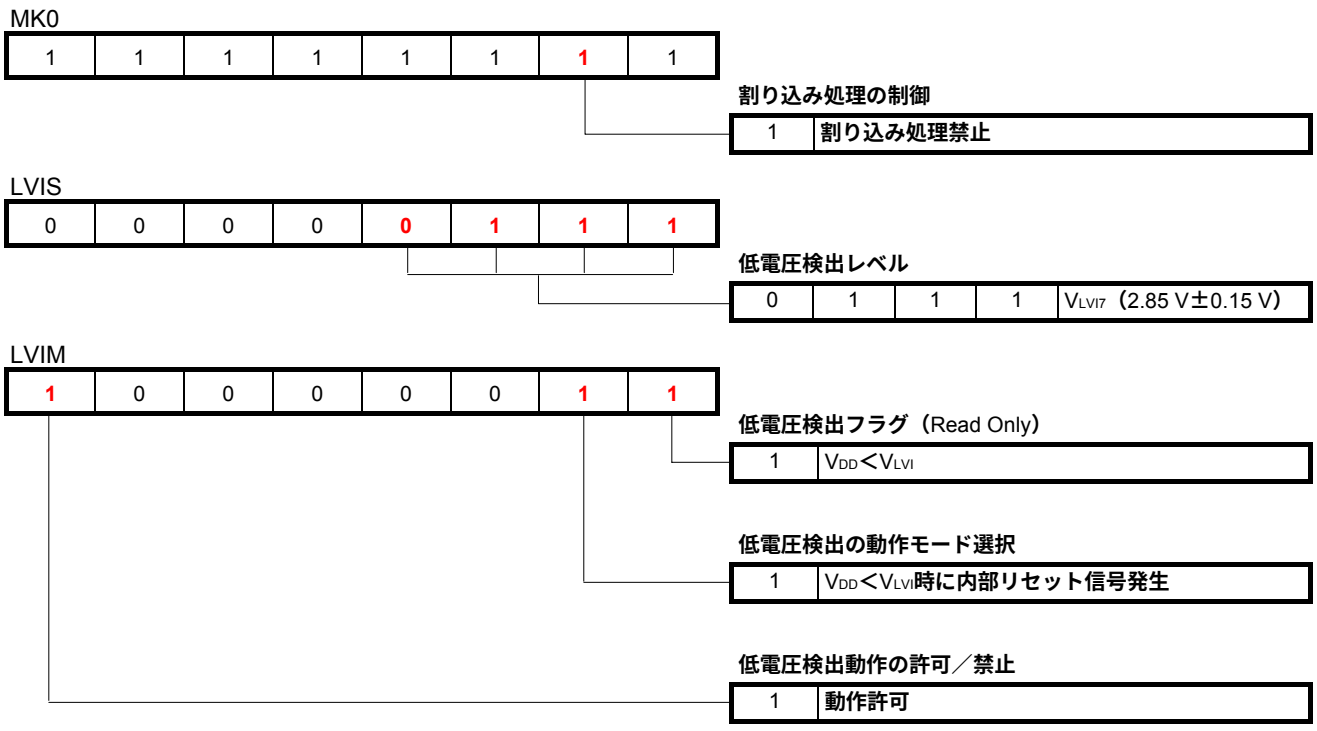
【例　1】　低電圧検出レベルをV_{LVI} = 2.85 V±0.15 Vに設定し，低電圧検出機能をリセットとして使用する（サンプル・プログラムと同内容）

●設定手順

① LVIの割り込みをマスクする（LVIMK = 1）[注]
② LVISレジスタのビット3-0（LVIS3-LVIS0）で検出レベルを設定する
③ LVIMレジスタのビット7（LVION）を1にセットし，LVI動作許可にする
④ ソフトウエアで200 μs以上ウエイトする
⑤ LVIMレジスタのビット0（LVIF）で，「$V_{DD} \geqq V_{LVI}$（LVIF = 0）」であることを確認するまで待つ
⑥ LVIMレジスタのビット1（LVIMD）を1に設定し，LVI検出時に内部リセット信号発生するように設定する

注　リセット後にLVIの割り込みはマスクされるため，サンプル・プログラム，次ページのプログラム例では設定を省略しています。

備考　上述の②～⑥と，次ページの②～⑥は対応しています。



●アセンブリ言語のプログラム例（サンプル・プログラムと同内容）

```
XMAIN  CSEG   UNIT
RESET_START:
②       MOV    LVIS,  #00000111B    ; 低電圧検出レベルVLVI = 2.85V±0.15Vに設定
③       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
WAIT_200US:
④       DEC    A
        BNZ    $WAIT_200US          ; 0.5[us/clk]×10[clk]×40[count] = 200[us]
WAIT_LVI:
        NOP
⑤       BT     LVIF,  $WAIT_LVI     ; VDD<VLVIなら分岐

⑥       SET1   LVIMD                ; VDD<VLVI時に内部リセット信号が発生するように設定
```

●C言語のプログラム例（サンプル・プログラムと同内容）

```
void hdwinit(void){
        unsigned char ucCnt200us;  /* 200usウェイト用8ビット変数 */
②              LVIS  = 0b00000111; /* 低電圧検出レベルVLVI = 2.85V±0.15Vに設定 */
③              LVION = 1;          /* 低電圧検出回路の動作許可 */

④              for (ucCnt200us = 0; ucCnt200us < 9; ucCnt200us++){   /*約200usウェイト */
                       NOP();
               }

⑤              while (LVIF){       /* VDD≧VLVI待ち */
                       NOP();
               }

⑥              LVIMD = 1;          /* VDD<VLVI時に内部リセット信号が発生するように設定 */
}
```

備考1. 上述のウエイト時間（200 μs）は，サンプル・プログラムと同様に，f_{CPU}（CPUクロック周波数） = 2 MHzで計算しています。

2. 上述の②～⑥と，前ページの②～⑥は対応しています

【例　2】　低電圧検出レベルをV_{LVI} = 4.3 V±0.2 Vに設定し，低電圧検出機能を割り込みとして使用する

●設定手順

① LVIの割り込みをマスクする（LVIMK = 1）[注]
② LVISレジスタのビット3-0（LVIS3-LVIS0）で検出レベルを設定する
③ LVIMレジスタのビット7（LVION）を1にセットし，LVI動作許可にする
④ ソフトウエアで200 μs以上ウエイトする
⑤ LVIMレジスタのビット0（LVIF）で，「$V_{DD} \geqq V_{LVI}$（LVIF = 0）」であることを確認するまで待つ
⑥ LVIの割り込み要求フラグをクリアする（LVIIF = 0）
⑦ LVIの割り込みマスク・フラグを解除する（LVIMK = 0）
⑧ （ベクタ割り込みを使用する場合は）EI命令を実行する

注　リセット後にLVIの割り込みはマスクされるため，下記のプログラム例では設定を省略しています。

備考　上述の②～⑧と，次ページ，次々ページの②～⑧は対応しています。

MK0

1	1	1	1	1	1	0/1	1

割り込み処理の制御

0	割り込み処理許可
1	割り込み処理禁止

IF0

0	0	0	0	0	0	0	0

割り込み要求フラグ

0	割り込み要求信号が発生していない

PSW

1	0	0	0	0	0	1	0

割り込み受け付けの許可／禁止

1	許可

LVIS

0	0	0	0	0	0	0	0

低電圧検出レベル

0	0	0	0	V_{LVI0}（4.3 V±0.2 V）

LVIM

1	0	0	0	0	0	0	1

低電圧検出フラグ（Read Only）

1	$V_{DD} < V_{LVI}$

低電圧検出の動作モード選択

0	$V_{DD} < V_{LVI}$時に割り込み信号発生

＊デフォルト値が0なので，設定不要

低電圧検出動作の許可／禁止

1	動作許可

●アセンブリ言語のプログラム例（低電圧検出の割り込みで，CPUクロック周波数を低速にする）

```
HIGH_SPEED    EQU    00H
LOW_SPEED     EQU    02H

XVCT   CSEG   AT     0000H
       DW     RESET_START   ;(00)  RESET
       DW     RESET_START   ;(02)  --
       DW     RESET_START   ;(04)  --
       DW     INTERRUPT_LVI ;(06)  INTLVI

XMAIN  CSEG   UNIT
RESET_START:
       MOV    PCC,   #00000000B    ; CPUクロックfcpu = fxp (= fx/4 = 2MHz)

②      MOV    LVIS,  #00000000B    ; 低電圧検出レベルVLVI = 4.3V±0.2Vに設定
③      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
WAIT_200US:
④      DEC    A
       BNZ    $WAIT_200US          ; 0.5[us/clk]×10[clk]×40[count] = 200[us]

WAIT_LVI1:
       NOP
⑤      BT     LVIF,  $WAIT_LVI1    ; VDD<VLVIなら分岐
       MOV    PPCC,  #HIGH_SPEED   ; 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

⑥      CLR1   LVIIF                ; LVIの割り込み要求フラグをクリア
⑦      CLR1   LVIMK                ; LVIの割り込み要求マスク・フラグを解除
⑧      EI                          ; ベクタ割り込み許可

MAIN_LOOP:
       MOV    A,     PPCC
       CMP    A,     #HIGH_SPEED
       BZ     $MAIN_LOOP           ; CPUクロックfcpu = 8MHzならMAIN_LOOPへ

WAIT_LVI2:
       NOP
       BT     LVIF,  $WAIT_LVI2    ; VDD<VLVIなら分岐
       MOV    PPCC,  #HIGH_SPEED   ; 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
       BR     $MAIN_LOOP           ; MAIN_LOOPへ

INTERRUPT_LVI:
       MOV    PPCC,  #LOW_SPEED    ; 周辺ハードウェアへの供給クロックfxp = fx/4 (= 2MHz)
                                   ;  -> CPUクロックfcpu = fxp = 2MHz
       RETI                        ; 割り込み処理から復帰
```

DW疑似命令で，初期値（「INTERRUPT_LVI」）を記述することにより，LVI検出による割り込み発生時には，「INTERRUPT_LVI」のシンボルから割り込み処理開始

CPUクロック周波数を設定

電源立ち上げ後，$V_{DD} \geqq V_{LVI}$になったら，CPUクロック周波数を高速にする

LVI検出後，$V_{DD} \geqq V_{LVI}$になったら，CPUクロック周波数を高速にする

LVI検出による割り込み発生により，「INTERRUPT_LVI」から割り込み処理開始

$V_{DD} < V_{LVI}$時に，CPUクロック周波数を低速にする

備考1.　上述のウエイト時間（200 μs）とCPUクロック周波数は，fx（システム・クロック発振周波数） = 8 MHz で計算しています。

2.　上述の②～⑧と，前ページの②～⑧は対応しています

●C言語のプログラム例（低電圧検出の割り込みで，CPUクロック周波数を低速にする）

```
#pragma       SFR                          /* 特殊機能レジスタ(SFR)名を記述可能にする */
#pragma       EI                           /* EI命令を記述可能にする */
#pragma       NOP                          /* NOP命令を記述可能にする */
#pragma interrupt INTLVI fn_intlvi         /* 割り込み関数宣言:INTLVI */

#define       HighSpeed 0x00
#define       LowSpeed 0x02

void hdwinit(void){
       PCC   = 0b00000000;  /* CPUクロックfcpu = fxp (= fx/4 = 2MHz) */

②      LVIS  = 0b00000000;  /* 低電圧検出レベルVLVI = 4.3V±0.2Vに設定 */
③      LVION = 1;           /* 低電圧検出回路の動作許可 */

④      for (ucCnt200us = 0; ucCnt200us < 9; ucCnt200us++){   /* 約200usウェイト */
              NOP();
       }

⑤      while (LVIF){        /* VDD≧VLVI待ち */
              NOP();
       }

       PPCC  = HighSpeed;   /* 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 */

⑥      LVIIF = 0;           /* LVIの割り込み要求フラグをクリア */
⑦      LVIMK = 0;           /* LVIの割り込みマスク・フラグを解除 */
⑧      EI();                /* ベクタ割り込み許可 */

       return;
}

void main(void){
       while (1){
              while (PPCC == HighSpeed){ /* LVI割り込み待ち */
                     NOP();
              }

              while (LVIF){        /* VDD≧VLVI待ち */
                     NOP();
              }

              PPCC  = HighSpeed;   /* 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 */
       }
}

__interrupt void fn_intlvi(){
       PPCC  = LowSpeed;           /* 周辺ハードウェアへの供給クロックfxp = fx/4 (= 2MHz)
                                      -> CPUクロックfcpu = fxp = 2MHz */

       return;
}
```

前処理指令（#pragma指令）にて，INTLVIの割り込み関数（この例では「fn_intlvi」）を宣言し，その割り込み関数を__interrupt修飾子で宣言することにより，割り込み発生時には，__interrupt修飾子で宣言した割り込み関数から割り込み処理開始

CPUクロック周波数を設定

電源立ち上げ後，$V_{DD} \geqq V_{LVI}$になったら，CPUクロック周波数を高速にする

LVI検出後，$V_{DD} \geqq V_{LVI}$になったら，CPUクロック周波数を高速にする

$V_{DD} < V_{LVI}$時に，CPUクロック周波数を低速にする

LVI検出による割り込み発生により，「fn_intlvi」から割り込み処理開始

備考1.　**上述のウエイト時間（200 μs）とCPUクロック周波数は，fx（システム・クロック発振周波数）** = 8 MHz **で計算しています。**

2.　**上述の②～⑧と，前々ページの②～⑧は対応しています**

第5章　デバイスでの動作確認

この章では，ダウンロードしたサンプル・プログラムを使用し，ビルドからデバイスでの動作確認までの流れを説明します。

★　5. 1　サンプル・プログラムのビルド

サンプル・プログラムのビルド方法について，アセンブリ言語版の  のアイコンからダウンロードしたサンプル・プログラム（ソース・ファイル＋プロジェクト・ファイル）とC言語版の  のアイコンからダウンロードしたサンプル・プログラム（ソース・ファイルのみ）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。

また，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。

①［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。
②［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。
③「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。

5. 1. 1　アセンブリ言語版（ソース・ファイル＋プロジェクト・ファイル）の場合

78K0S/KB1+マイクロコントローラのサンプル・プログラム（低電圧検出）のアセンブリ言語版の， 

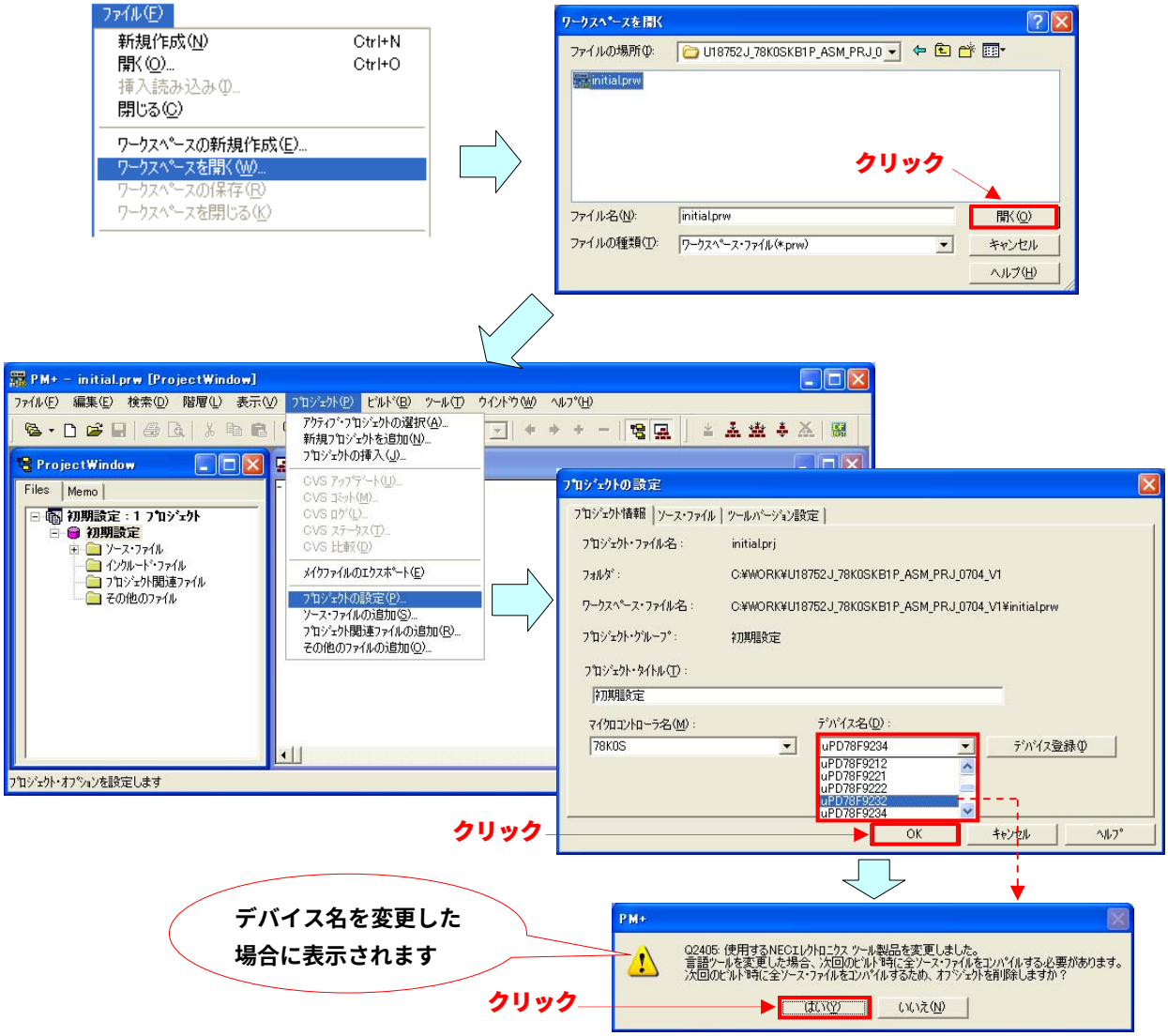
のアイコンからダウンロードしたファイルを使用して，説明します。

(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，「lvi.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。

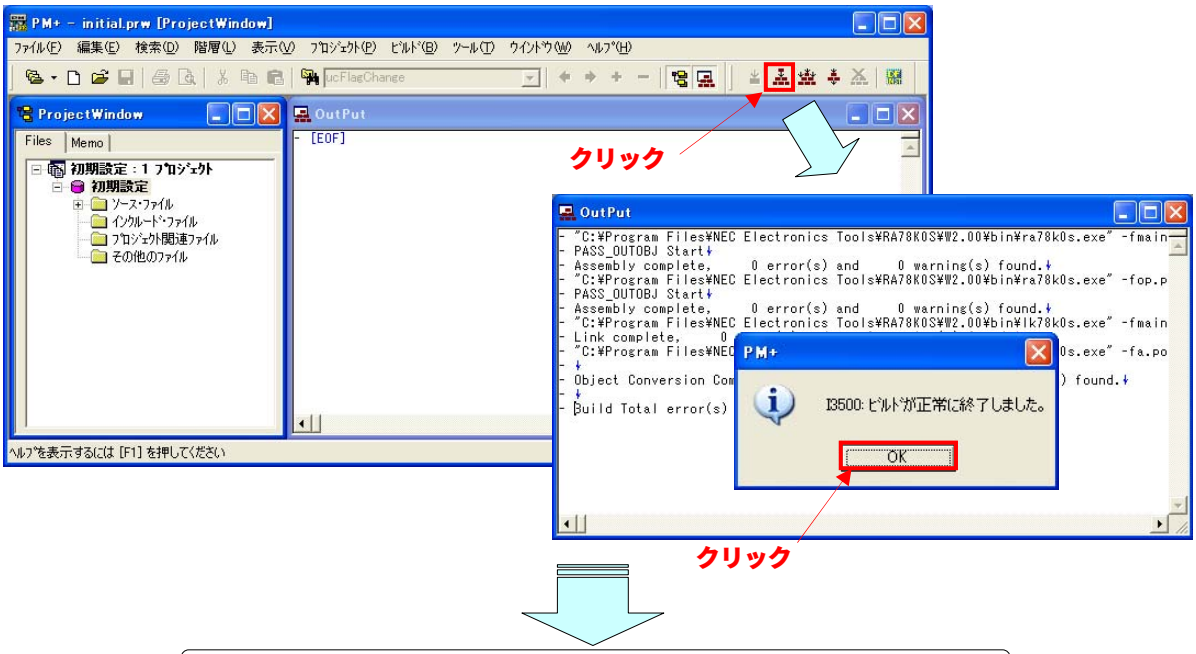
備考　下の図は，「サンプル・プログラム（初期設定）　LED点灯のスイッチ制御」の画面例です。



(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。



フラッシュ・メモリ書き込み用のHEXファイルが生成されます → 5. 2へ

5. 1. 2　C言語版（ソース・ファイルのみ）の場合

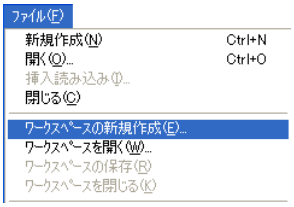
78K0S/KB1+マイクロコントローラのサンプル・プログラム（低電圧検出）のC言語版の，  のアイコンからダウンロードしたファイルを使用して，説明します。

このサンプル・プログラムのC言語版には，「main.c」，「op.asm」のほかに，スタートアップ・ルーチンのソース・ファイルの「cstart.asm」，ライブラリ種別設定ファイルの「def.inc」，マクロ定義ファイルの「macro.inc」が含まれています。これらのファイルは，標準のスタートアップ・ルーチンの一部をコメント・アウトしたものであり，これらのファイルを適用してPM+でプロジェクトを作成する際には，他のC言語版のサンプル・プログラムとは異なる操作が必要となります。次にその手順を説明します。

●プロジェクトの登録

(1) PM+を起動してください。

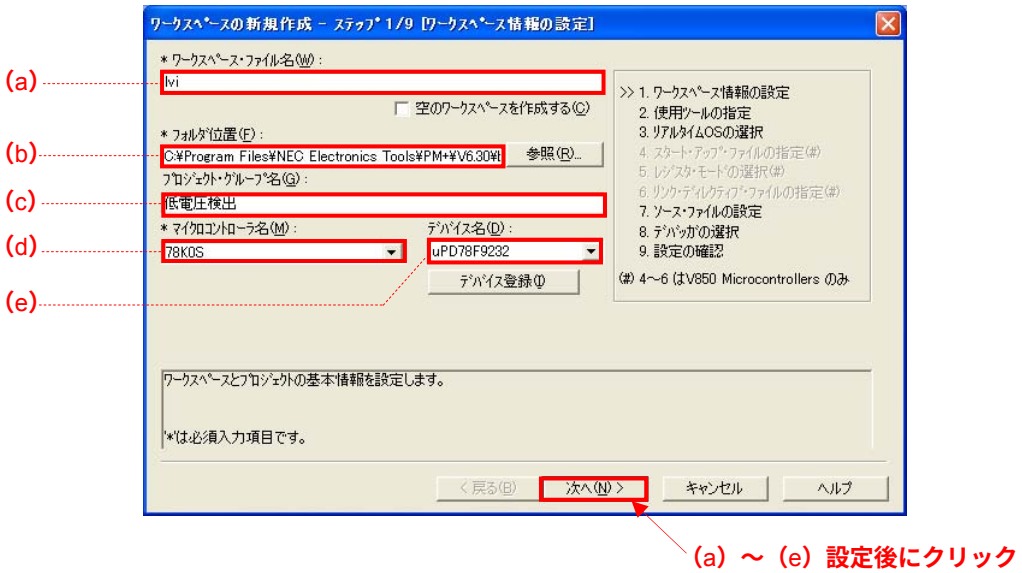
(2) ［ファイル］→［ワークスペースの新規作成］を選択してください。



(3) ［ワークスペースの新規作成－ステップ1/9 [ワークスペース情報の設定] ］画面が立ち上がります。次の項目を設定してください。

(a) ワークスペース・ファイル名（この画面例では，ファイル名を「lvi」と入力）。
(b) フォルダ位置（この画面例では，デフォルトのフォルダ（PM+が存在する「bin」フォルダ）の下の，任意で作成した「work」フォルダを指定）
(c) プロジェクト・グループ名（この画面例では，グループ名を「低電圧検出」と入力）。
(d) シリーズ名（この画面例では，シリーズ名を「78K0S」を選択）
(e) デバイス名（この画面例では，78K0S/KB1+マイクロコントローラ製品の「uPD78F9232」を設定）。

(a) ～ (e) を設定したら，［次へ］ボタンをクリックしてください。



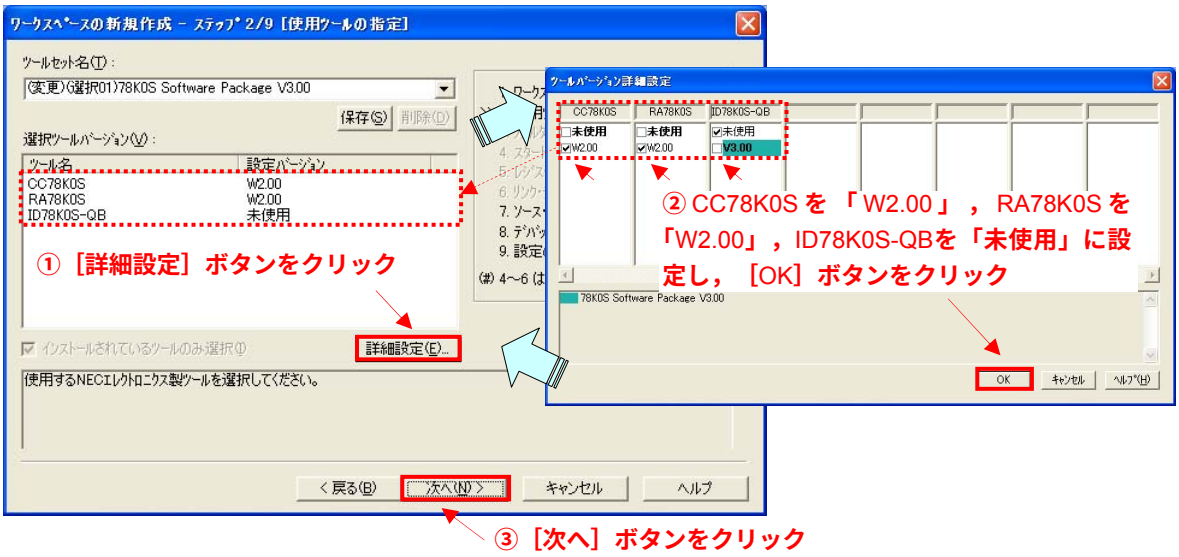
(4) ［ワークスペースの新規作成－ステップ2/9 [使用ツールの指定] ］画面が立ち上がります。次の手順で使用するツールとそのバージョンを設定してください。

① ［詳細設定］ボタンをクリックしてください。

② ［ツールバージョン詳細設定］画面が立ち上がります。使用するツールとそのバージョンを次のように設定し，［OK］ボタンをクリックしてください。

- CC78K0S：　W2.00以上のバージョン
- RA78K0S：　W2.00以上のバージョン
- ID78K0S-QB：未使用

③ ②で選択したツールとそのバージョンが設定されます。［次へ］ボタンをクリックしてください。



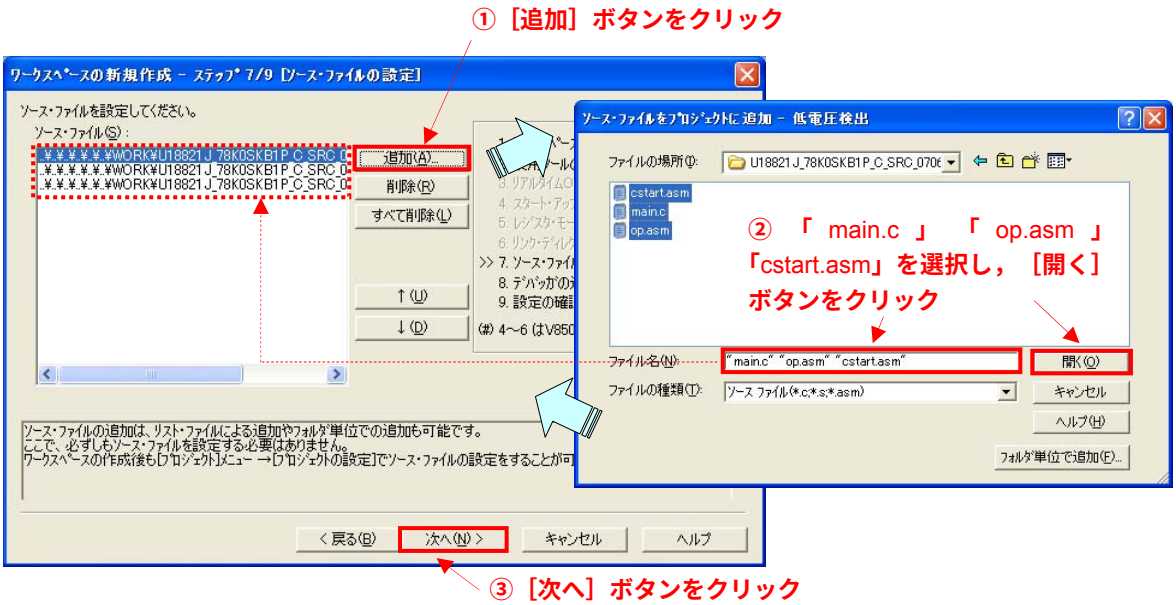
(5) ［ワークスペースの新規作成－ステップ7/9 [ソース・ファイルの設定] ］画面が立ち上がります。次の手順でソース・ファイルを設定してください。

① ［追加］ボタンをクリックしてください。

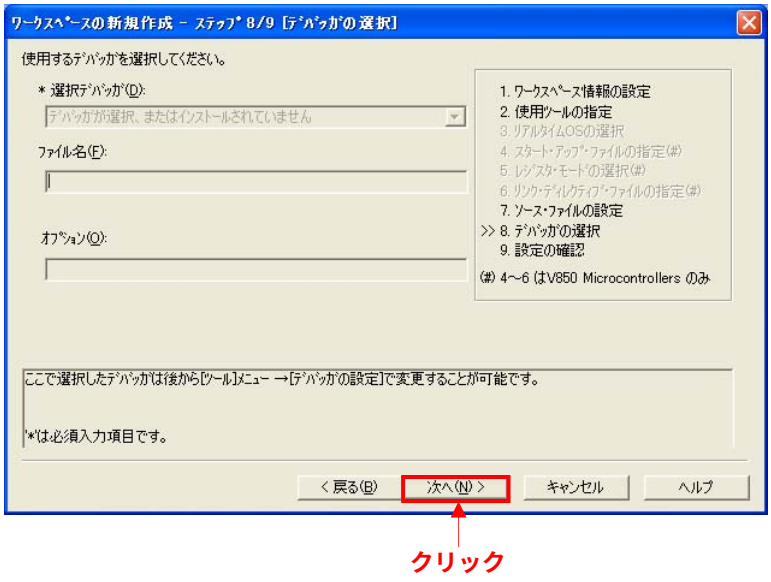
② ［ソースファイルをプロジェクトに追加］画面が立ち上がります。次のソース・ファイルを選択し，［開く］ボタンをクリックしてください。

- main.c
- op.asm
- cstart.asm

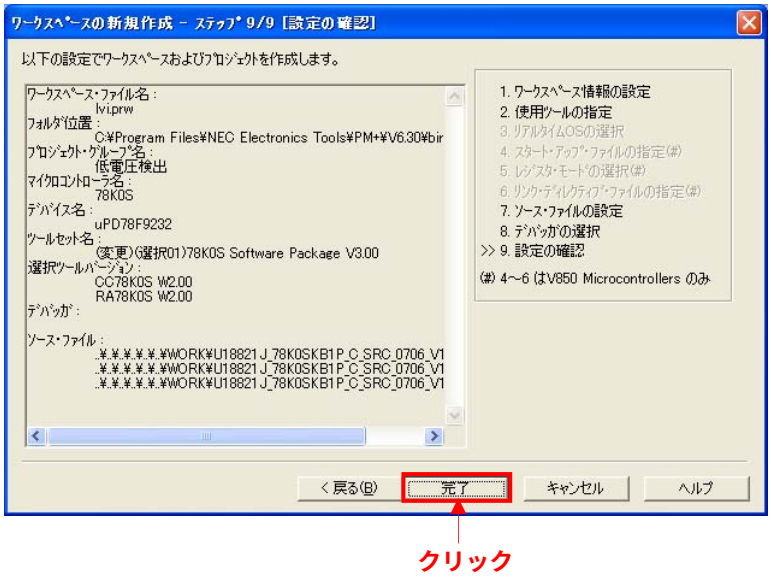
③ ②で選択したソース・ファイルが設定されます。［次へ］ボタンをクリックしてください。



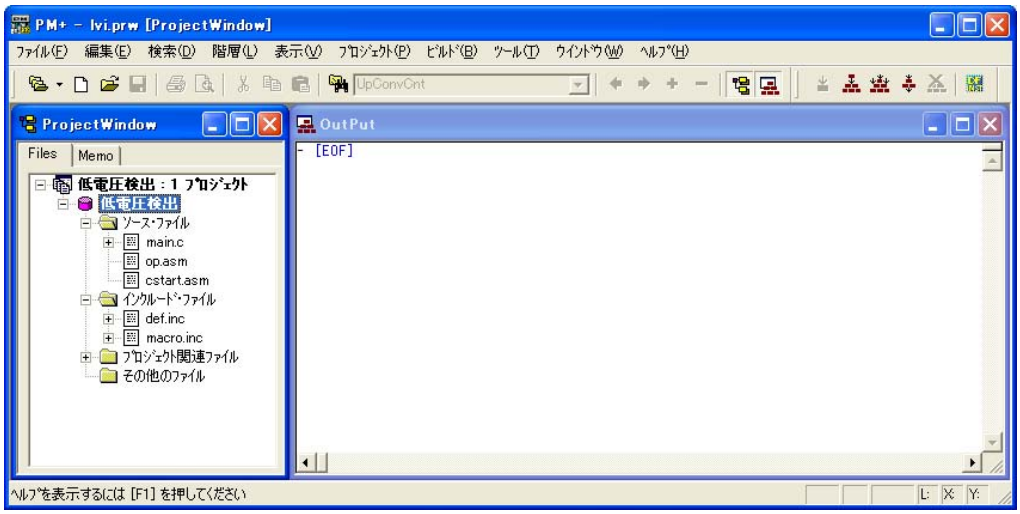
(6) ［ワークスペースの新規作成－ステップ8/9 [デバッガの選択] ］画面が立ち上がります。［次へ］ボタンをクリックしてください。



(7) ［ワークスペースの新規作成－ステップ9/9 [設定の確認] ］画面が立ち上がります。内容を確認後，［完了］ボタンをクリックしてください。



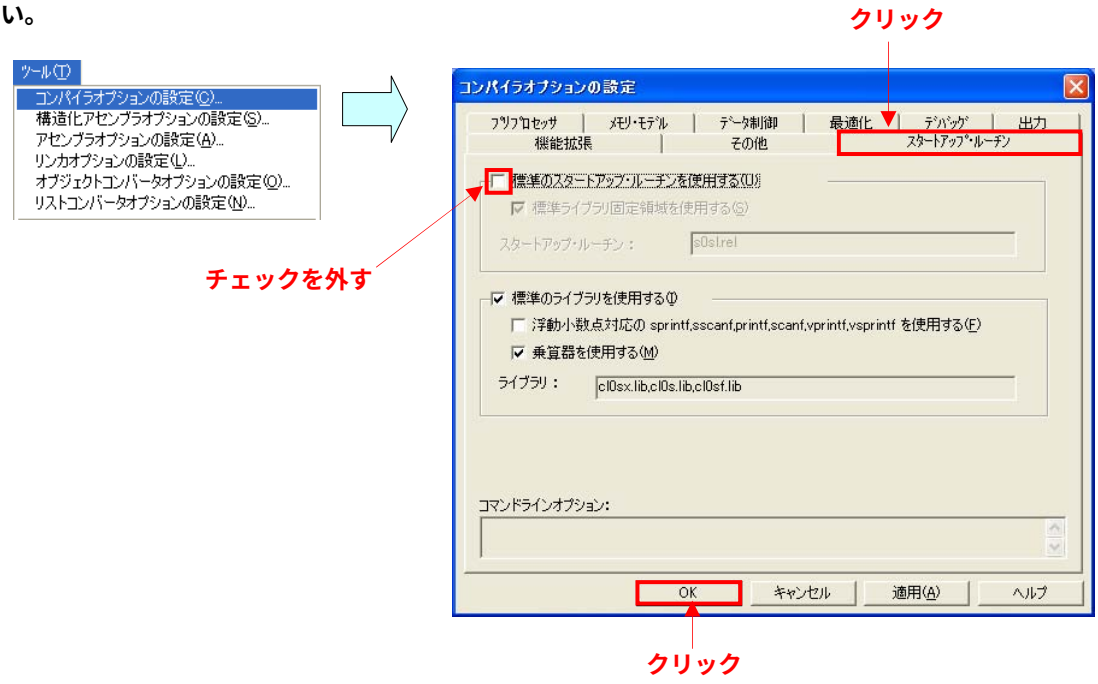
(8) ワークスペースが作成され，プロジェクトが登録されます。



●コンパイラオプションの設定

(9)　［ツール］→［コンパイラオプションの設定］を設定ください。

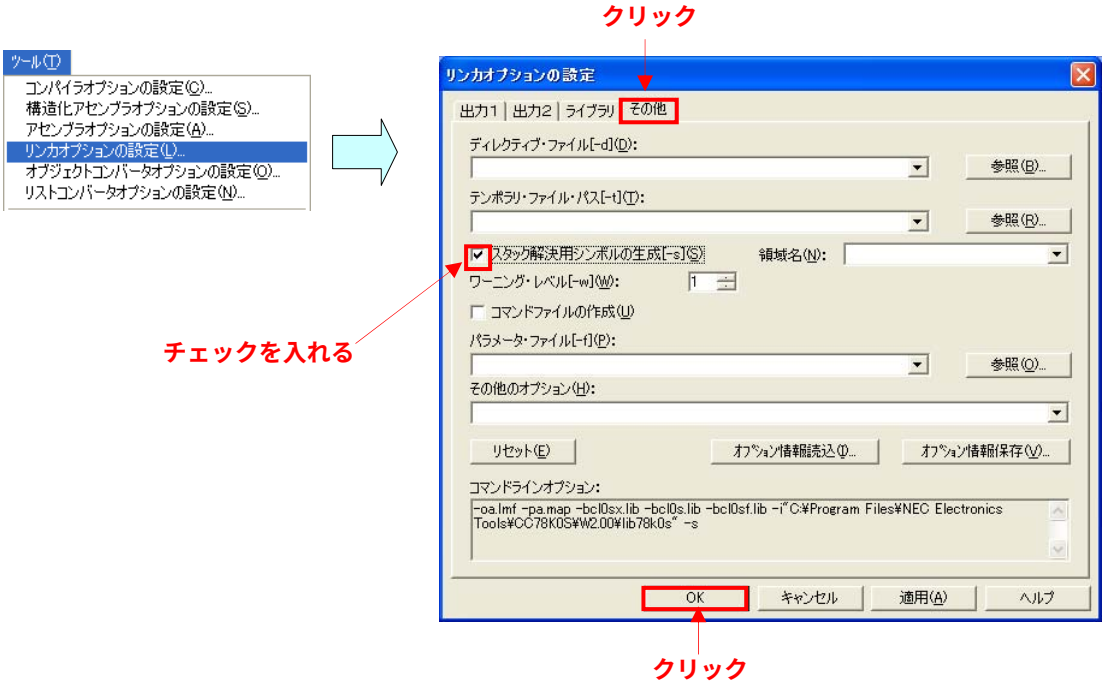
(10)　［コンパイラオプションの設定］画面が立ち上がります。「スタートアップ・ルーチン」タブをクリックし，「標準のスタートアップ・ルーチンを使用する」のチェックを外して，［OK］ボタンをクリックしてください。



●リンカオプションの設定

(11)　［ツール］→［リンカオプションの設定］を設定ください。

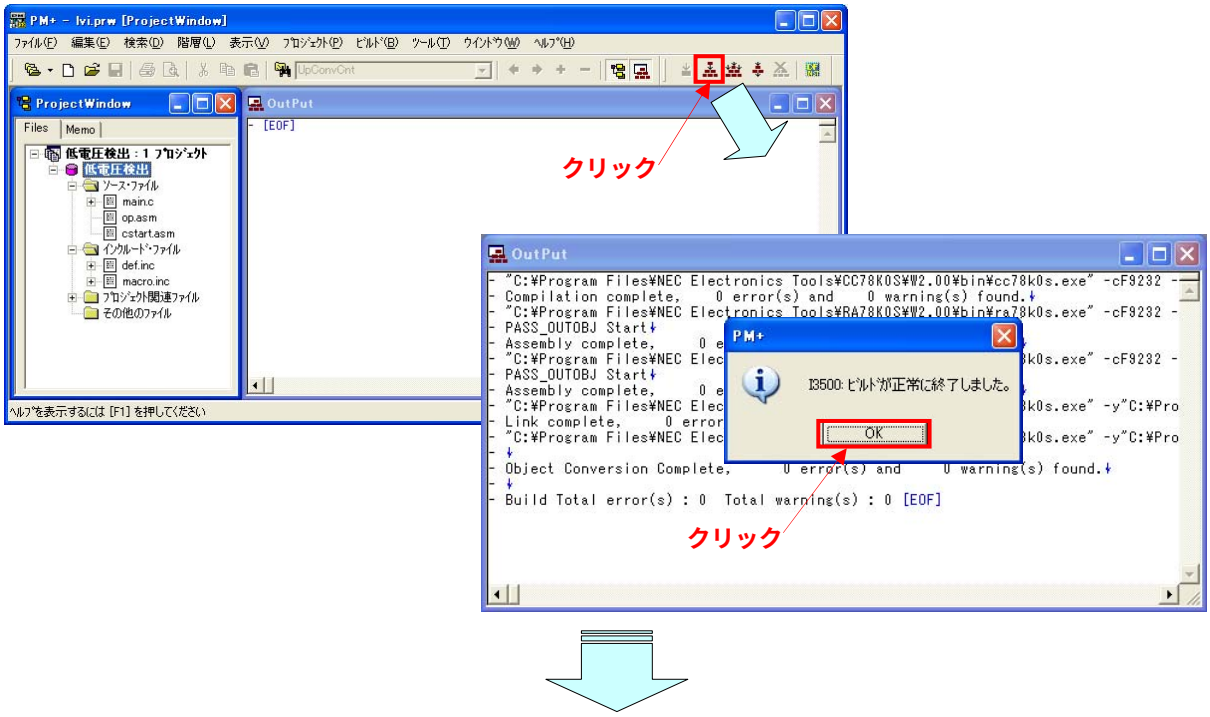
(12)　［リンカオプションの設定］画面が立ち上がります。「その他」タブをクリックし，「スタック解決用のシンボルの生成［-s］」のチェックを入れて，［OK］ボタンをクリックしてください。



●ビルドの実行

(13)  （「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。

(14) 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。



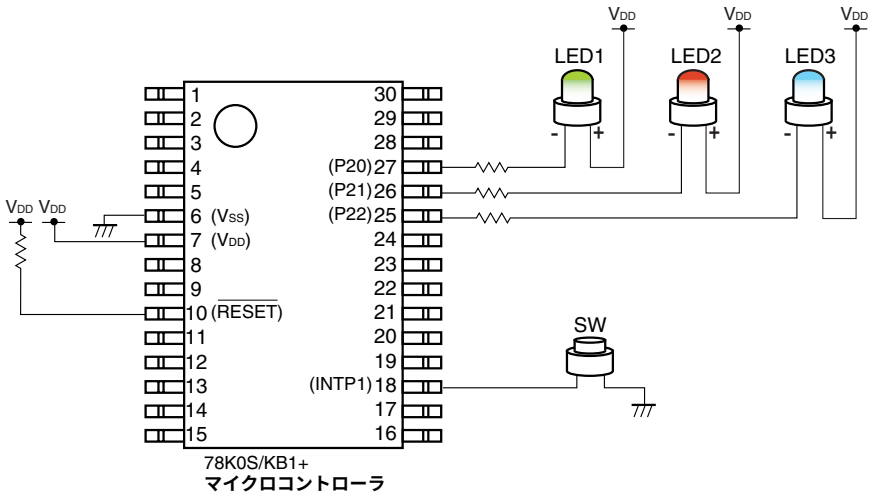
フラッシュ・メモリ書き込み用のHEXファイルが生成されます → 5. 2へ

5. 2　デバイスでの動作

ここでは，デバイスでの動作確認の例を説明します。

ビルド実行により生成されたHEXファイルは，デバイスのフラッシュ・メモリに書き込みすることができます。デバイスへのフラッシュ・メモリ書き込み方法例については，各製品のフラッシュ書き込み簡単マニュアル　インフォメーション（78K0S/KU1+, 78K0S/KY1+, 78K0S/KA1+, 78K0S/KB1+）を参照してください。

デバイスと使用する周辺ハードウエア（スイッチ，LED）の接続例を，次に示します。



このサンプル・プログラムを書き込んだデバイスの動作例は，次のようになります。

(1) V_{DD}≧3.0 V（通常動作時）[注]の場合

スイッチの入力回数により，LEDの点灯パターンが変化します。

注　低電源検出電圧をV_{LVI} = 2.85 V±0.15 Vに設定しているため，スイッチ入力による割り込み処理動作は，V_{DD}≧3.0 Vで行ってください。

スイッチの入力回数[注]	LED出力		
	LED3	LED2	LED1
0	OFF	OFF	OFF
1	OFF	OFF	ON
2	OFF	ON	OFF
3	OFF	ON	ON
4	ON	OFF	OFF
5	ON	OFF	ON
6	ON	ON	OFF
7	ON	ON	ON

注　8回目以降は，0回目からの点灯パターンの繰り返しになります。

また，スイッチを押す時間が10 ms未満の場合は，そのスイッチ入力がチャタリングであると判定され，LED表示パターンは，スイッチを押す前の表示パターンのまま，変化しません。

(2) V_{DD}≧3.0 V → V_{DD} = 2.5 V **（低電圧検出）** → V_{DD}≧3.0 V

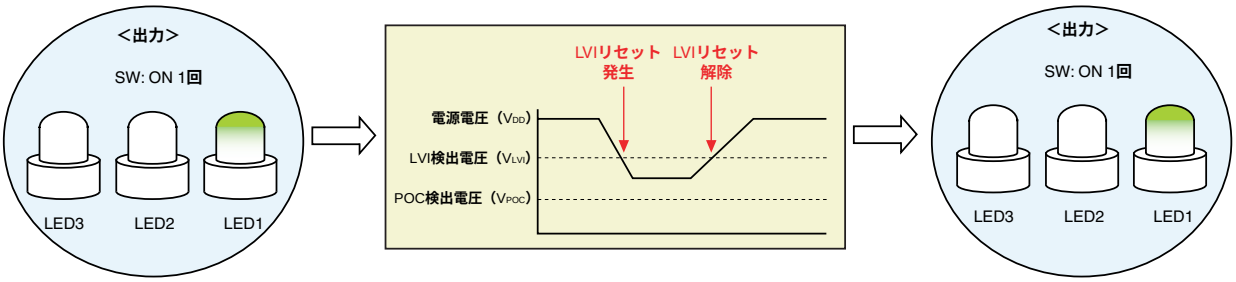
電源電圧の変化により，LEDの点灯は次のようになります。

① V_{DD}≧3.0 V → V_{DD} = 2.5 V

低電圧検出レベルをV_{LVI} = 2.85 V±0.15 Vに，低電圧検出機能をリセットとして使用するように設定しているため，LVIリセットが発生します。このとき，LEDは全消灯となりますが，RAMはリセット直前のLED表示データを保持しています。

② V_{DD} = 2.5 V → V_{DD}≧3.0 V

通常モードに戻ります。このとき，リセット要因がLVIリセットであることを確認し，RAMに保持されているLED表示データを読み出すので，リセット直前の点灯パターンが復元されます。



(3) V_{DD}≧3.0 V → 2.0 V>V_{DD} **（データ保持電源電圧未満）** → V_{DD}≧3.0 V

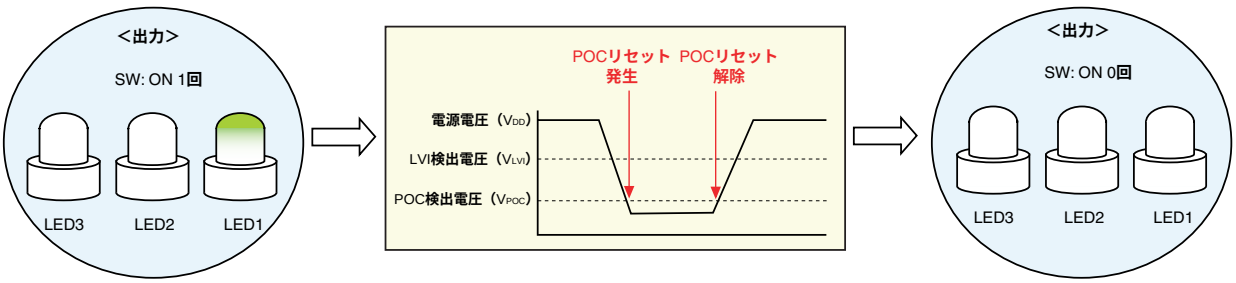
電源電圧の変化により，LEDの点灯は次のようになります。

① V_{DD}≧3.0 V → 2.0 V>V_{DD}

パワーオン・クリア（POC）によるリセットが発生します。このとき，LED点灯は全消灯となり，RAMデータは不定となります。

② 2.0 V>V_{DD} → V_{DD}≧3.0 V

通常モードに戻ります。このとき，リセット要因がLVIリセット以外であることを確認し，RAMのLED点灯パターンを初期化するので，LEDは全消灯（スイッチの入力回数 = 0回）となります。



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
78K/0S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	言語編	PDF
ユーザーズ・マニュアル	操作編	PDF
CC78K0S　Cコンパイラ	言語編	PDF
ユーザーズ・マニュアル	操作編	PDF
PM+　プロジェクト・マネージャ　ユーザーズ・マニュアル		PDF
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）	78K0S/KU1+	PDF
インフォメーション	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・	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
ノート	サンプル・プログラム（割り込み）　スイッチ入力による外部割り込み編	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 （アセンブリ言語版）

```
;*******************************************************************************
;
;     NEC Electronics     78K0S/KB1+シリーズ
;
;*******************************************************************************
;     78K0S/KB1+シリーズ       サンプル･プログラム
;*******************************************************************************
;     低電圧検出
;*******************************************************************************
;【履歴】
;     2007.6.--        新規作成
;*******************************************************************************
;
;【概要】
;
;本サンプルプログラムは、低電圧検出(LVI)機能の使用例を示すものである。
;低電圧検出(LVI)回路を使用することでVDD＜VLVI(2.85V±0.15V)において内部リセット
;信号が発生するように設定を行う。
;初期設定完了後は、スイッチ入力回数に応じてLEDの点灯パターンを変化させる。
;(この処理は「サンプル・プログラム　割り込み」と同様。)
;ここで、LVI以外によるリセット時はスイッチ入力回数が初期化されるが、LVIによる
;リセット時はRAMデータが保持されている為、リセット前のスイッチ入力回数を復元し、
;それに応じたLED点灯パターンを表示する。
;
;
;  ＜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2.85V±0.15V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クロック周波数を4MHzに設定
;  ・外部割り込みINTP1の有効エッジ：立ち下りエッジに設定
;  ・スイッチ入力時のチャタリング除去時間 = 10ms
;
;
```

```
;  ＜低電圧検出とリセット解除後のLED点灯パターン＞
;
;  ・低電圧検出回路以外によるリセット・・・全LED消灯
;  ・低電圧検出回路によるリセット・・・・・リセット前のLED点灯パターンを保持
;
;
;  ＜スイッチ入力回数とLED点灯パターン＞
;
;    +----------------------------------+
;    | SW入力回数 |  LED3 |  LED2 |  LED1 |
;    |   (P43)    | (P22) | (P21) | (P20) |
;    |------------|----------------------|
;    |     0回    |  OFF  |  OFF  |  OFF  |
;    |     1回    |  OFF  |  OFF  |  ON   |
;    |     2回    |  OFF  |  ON   |  OFF  |
;    |     3回    |  OFF  |  ON   |  ON   |
;    |     4回    |  ON   |  OFF  |  OFF  |
;    |     5回    |  ON   |  OFF  |  ON   |
;    |     6回    |  ON   |  ON   |  OFF  |
;    |     7回    |  ON   |  ON   |  ON   |
;    +----------------------------------+
;    # 8回目以降は0回からの繰り返し
;
;
;【ポート入出力の設定】
;
;  入力ポート：P43
;  出力ポート：P00-P03，P20-P23，P30-P33，P40-P42，P44-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
;
;*****************************************************************************


;=============================================================================
;
;     ベクタ・テーブルの設定
;
;=============================================================================
XVCT  CSEG    AT      0000H
      DW      RESET_START             ;(00)   RESET
      DW      RESET_START             ;(02)   --
      DW      RESET_START             ;(04)   --
      DW      RESET_START             ;(06)   INTLVI
```

```
        DW      RESET_START             ;(08)   INTP0
        DW      INTERRUPT_P1            ;(0A)   INTP1
        DW      RESET_START             ;(0C)   INTTMH1
        DW      RESET_START             ;(0E)   INTTM000
        DW      RESET_START             ;(10)   INTTM010
        DW      RESET_START             ;(12)   INTAD
        DW      RESET_START             ;(14)   --
        DW      RESET_START             ;(16)   INTP2
        DW      RESET_START             ;(18)   INTP3
        DW      RESET_START             ;(1A)   INTTM80
        DW      RESET_START             ;(1C)   INTSRE6
        DW      RESET_START             ;(1E)   INTSR6
        DW      RESET_START             ;(20)   INTST6

;=============================================================================
;
;       RAMの定義
;
;=============================================================================
XRAM    DSEG    SADDR
CNT_1:          DS      1               ; メジャー・ループ用
CNT_2:          DS      1               ; マイナー・ループ用
LEDDATA:        DS      1               ; LED表示パターン変数

;=============================================================================
;
;       スタック領域の確保
;
;=============================================================================
XSTK    DSEG    AT      0FEE0H
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 = FF00H

;*****************************************************************************
;
;       リセット解除後の初期化処理
;
;*****************************************************************************
XMAIN   CSEG    UNIT
RESET_START:
```

```
;-------------------------------------------------------------------------------
;     スタック・ポインタの設定
;-------------------------------------------------------------------------------
      MOVW    AX,     #STACKTOP
      MOVW    SP,     AX              ; スタック・ポインタを設定

;-------------------------------------------------------------------------------
;     低電圧検出 + ウォッチドッグ・タイマの設定 + クロックの設定
;-------------------------------------------------------------------------------
;-----  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

;-----  クロックの設定1  -----
      MOV     PCC,    #00000000B      ; CPUクロックfcpu = fxp (= fx/4 = 2MHz)
      MOV     LSRCM,  #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,    $SET_CLOCK      ; 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,   #00000111B      ; 低電圧検出レベルVLVI = 2.85V±0.15V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  -----
WAIT_200US:
      DEC     A
      BNZ     $WAIT_200US             ; 0.5[us/clk]×10[clk]×40[count] = 200[us]

;-----  VDD≧VLVI待ち処理  -----
WAIT_LVI:
      NOP
      BT      LVIF,   $WAIT_LVI       ; 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

      MOV     LEDDATA,#00000111B      ; LED表示データを初期化

;-----  クロックの設定2  -----
SET_CLOCK:
      MOV     PPCC,   #00000001B      ; 周辺ハードウェアへの供給クロックfxp = fx/2 (= 4MHz)
                                      ; -> CPUクロックfcpu = fxp = 4MHz
```

```
;-------------------------------------------------------------------------------
;     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     A,      LEDDATA         ; LED表示データ読み出し
      MOV     P2,     A               ; LED出力(P20-P22)、P23の出力ラッチLow
      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,     #00000000B      ; P40-P47の出力ラッチLow
      MOV     PU4,    #00001000B      ; P43に内蔵プルアップ抵抗を使用
      MOV     PM4,    #00001000B      ; P43を入力ポートに、P40-P42,P44-P47を出力ポートに設定

;-------------------------------------------------------------------------------
;     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

;-------------------------------------------------------------------------------
;     割り込みの設定
;-------------------------------------------------------------------------------
      MOV     INTM0,  #00000000B      ; INTP1の有効エッジ = 立下りエッジ
      CLR1    PIF1                    ; 無効割り込み要求をクリアしておく
```

```
        CLR1    PMK1                    ; INTP1割り込みマスク解除

        EI                              ; ベクタ割り込み許可

;*******************************************************************************
;
;       メイン・ループ
;
;*******************************************************************************
MAIN_LOOP:
        NOP
        BR      $MAIN_LOOP              ; MAIN_LOOPへ

;*******************************************************************************
;
;       外部割り込みINTP1
;
;*******************************************************************************
INTERRUPT_P1:
        PUSH    AX                      ; AXレジスタのデータをスタックへ退避

;-----  チャタリング対策の10msウェイト  -----
        MOV     CNT_1,  #215            ; メジャー・ループ用のカウント値を代入
        NOP
        NOP
LOOP_1:
        MOV     CNT_2,  #17             ; マイナー・ループ用のカウント値を代入
        NOP
LOOP_2:
        NOP
        DBNZ    CNT_2,  $LOOP_2         ; マイナー・ループ
        DBNZ    CNT_1,  $LOOP_1         ; メジャー・ループ

        CLR1    PIF1                    ; INTP1割り込み要求をクリア

;-----  チャタリング検出の判定  -----
        BT      P4.3,   $END_INTP1      ; スイッチ入力がなければ分岐する

;-----  LED点灯処理  -----
        DEC     LEDDATA                 ; 現在のLED表示データを-1
        AND     LEDDATA, #00000111B     ; ビット0-2以外をマスク
        MOV     A,      LEDDATA         ; LED表示データを読み出し
        MOV     P2,     A               ; LED出力
```

```
END_INTP1:
        POP     AX                      ; AXレジスタのデータを復帰
        RETI                            ; 割り込み処理から復帰

end
```

● main.c （C言語版）

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

        NEC Electronics      78K0S/KB1+シリーズ

*****************************************************************************
        78K0S/KB1+シリーズ        サンプル･プログラム
*****************************************************************************
        低電圧検出
*****************************************************************************
【履歴】
        2007.6.--         新規作成
*****************************************************************************

【概要】

本サンプルプログラムは、低電圧検出(LVI)機能の使用例を示すものである。
低電圧検出(LVI)回路を使用することでVDD＜VLVI(2.85V±0.15V)において内部リセット
信号が発生するように設定を行う。
初期設定完了後は、スイッチ入力回数に応じてLEDの点灯パターンを変化させる。
(この処理は「サンプル・プログラム　割り込み」と同様。)
ここで、LVI以外によるリセット時はスイッチ入力回数が初期化されるが、LVIによる
リセット時はRAMデータが保持されている為、リセット前のスイッチ入力回数を復元し、
それに応じたLED点灯パターンを表示する。


  ＜主な設定内容＞

  ・割り込みで起動される関数の宣言: INTP1 -> fn_intp1()
  ・ウォッチドッグ・タイマの動作停止
  ・低電圧検出電圧VLVIを2.85V±0.15V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クロック周波数を4MHzに設定
  ・外部割り込みINTP1の有効エッジ：立ち下りエッジに設定
  ・スイッチ入力時のチャタリング検出時間 = 10ms
```

＜低電圧検出とリセット解除後のLED点灯パターン＞

・低電圧検出回路以外によるリセット・・・全LED消灯
・低電圧検出回路によるリセット・・・・・リセット前のLED点灯パターンを保持

＜スイッチ入力回数とLED点灯パターン＞

SW入力回数 (P43)	LED3 (P22)	LED2 (P21)	LED1 (P20)
0回	OFF	OFF	OFF
1回	OFF	OFF	ON
2回	OFF	ON	OFF
3回	OFF	ON	ON
4回	ON	OFF	OFF
5回	ON	OFF	ON
6回	ON	ON	OFF
7回	ON	ON	ON

8回目以降は0回からの繰り返し

【ポート入出力の設定】

入力ポート：P43
出力ポート：P00-P03，P20-P23，P30-P33，P40-P42，P44-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

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


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

      前処理指令（#pragma指令）

=========================================================================*/
#pragma       SFR                           /* 特殊機能レジスタ(SFR)名を記述可能にする */
#pragma       EI                            /* EI命令を記述可能にする */
#pragma       NOP                           /* NOP命令を記述可能にする */
#pragma interrupt INTP1 fn_intp1      /* 割り込み関数宣言:INTP1 */
```

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

        リセット解除後の初期化処理

*****************************************************************************/
        sreg unsigned char g_ucLED;     /* LED表示データ用8ビット変数(内部高速RAM領域) */

void hdwinit(void){
        unsigned char ucCnt200us;       /* 200usウェイト用8ビット変数 */

/*---------------------------------------------------------------------------
        ウォッチドッグ・タイマの設定 + 低電圧検出 + クロックの設定
---------------------------------------------------------------------------*/
        /* ウォッチドッグ・タイマの設定 */
        WDTM  = 0b01110111;             /* ウォッチドッグ・タイマ動作停止 */

        /* クロックの設定1 */
        PCC   = 0b00000000;             /* CPUクロックfcpu = fxp (= fx/4 = 2MHz) */
        LSRCM = 0b00000001;             /* 低速内蔵発振器の発振を停止 */

        /* リセット要因の確認 */
        if (!(RESF & 0b00000001)){      /* LVIリセット時は以降のLVI関連処理を省略 */

                /* 低電圧検出の設定 */
                LVIS  = 0b00000111;     /* 低電圧検出レベルVLVI = 2.85V±0.15Vに設定 */
                LVION = 1;              /* 低電圧検出回路の動作許可 */

                for (ucCnt200us = 0; ucCnt200us < 9; ucCnt200us++){     /* 約200usウェイト */
                        NOP();
                }

                while (LVIF){           /* VDD≧VLVI待ち */
                        NOP();
                }

                LVIMD = 1;              /* VDD＜VLVI時に内部リセット信号が発生するように設定 */

                g_ucLED = 0b00000111;   /* LED表示データを初期化 */
        }

        /* クロックの設定2 */
        PPCC  = 0b00000001;             /* 周辺ハードウェアへの供給クロックfxp = fx/2 (= 4MHz)
                                           -> CPUクロックfcpu = fxp = 4MHz */
```

```
/*-----------------------------------------------------------------------------
      ポート0の設定
-----------------------------------------------------------------------------*/
      P0    = 0b00000000;             /* P00-P03の出力ラッチLow */
      PM0   = 0b11110000;             /* P00-P03を出力ポートに設定 */

/*-----------------------------------------------------------------------------
      ポート2の設定
-----------------------------------------------------------------------------*/
      P2    = g_ucLED;                /* LED表示データを出力 */
      PM2   = 0b11110000;             /* P20-P23を出力ポートに設定 */

/*-----------------------------------------------------------------------------
      ポート3の設定
-----------------------------------------------------------------------------*/
      P3    = 0b00000000;             /* P30-P33の出力ラッチLow */
      PM3   = 0b11110000;             /* P30-P33を出力ポートに設定 */

/*-----------------------------------------------------------------------------
      ポート4の設定
-----------------------------------------------------------------------------*/
      P4    = 0b00000000;             /* P40-P47の出力ラッチLow */
      PU4   = 0b00001000;             /* P43に内蔵プルアップ抵抗を使用 */
      PM4   = 0b00001000;             /* P43を入力ポートに、P40-42,P44-P47を出力ポートに設定 */

/*-----------------------------------------------------------------------------
      ポート12の設定
-----------------------------------------------------------------------------*/
      P12   = 0b00000000;             /* P120-P123の出力ラッチLow */
      PM12  = 0b11110000;             /* P120-P123を出力ポートに設定 */

/*-----------------------------------------------------------------------------
      ポート13の設定
-----------------------------------------------------------------------------*/
      P13   = 0b00000001;             /* P130の出力High */

/*-----------------------------------------------------------------------------
      割り込みの設定
-----------------------------------------------------------------------------*/
      INTM1 = 0b00000000;             /* INTP1の有効エッジ = 立下りエッジ */
      PIF1  = 0;                      /* 無効割り込み要求をクリアしておく */
      PMK1  = 0;                      /* INTP1割り込みマスク解除 */
```

```
        return;
}

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

        メイン・ループ

*****************************************************************************/
void main(void){

        EI();                           /* ベクタ割り込み許可 */

        while (1){
                NOP();
                NOP();
        }
}

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

        外部割込みINTP1

*****************************************************************************/
__interrupt void fn_intp1(){
        unsigned int unChat;    /* チャタリング除去タイマ用16ビット変数 */

        for (unChat = 0; unChat < 555; unChat++){       /* 約10msウェイト(チャタリング除去用) */
                NOP();
        }

        PIF1  = 0;                      /* INTP1割り込み要求をクリア */

        if (!P4.3){             /* 10ms以上SWオンの場合の処理 */
                g_ucLED -= 1;           /* 現在のLED表示データを-1 */
                g_ucLED &= 0b00000111;  /* ビット0-2以外をマスク */
                P2 = g_ucLED;           /* LED出力 */
        }

        return;
}
```

● op.asm （アセンブリ言語版とC言語版共通）

```
;================================================================================
;
;   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	June 2007	－	－
第2版	July 2008	p.10	3. 4　フロー・チャートを変更
		pp.18-26	5. 1　サンプル・プログラムのビルドを変更
		p.29	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変更

【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。