

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日
ルネサスエレクトロニクス株式会社

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

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

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りが無いことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

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

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

アプリケーション・ノート

μ PD179F11x

サンプル・プログラム

フラッシュ・セルフ・プログラミングによる プリセット・リモコン編

この資料は、サンプル・プログラムの動作概要や使用方法、およびフラッシュ・セルフ・プログラミング・ライブラリの設定方法や使用方法について説明したものです。サンプル・プログラムは、キー・スキャンにより特殊なキー入力を検知したときに、フラッシュ・セルフ・プログラミング・ライブラリを使用してフラッシュROMの内容を書き換えます。なお、本サンプル・プログラムには、リモコン送信機能は組み込まれていません。

対象デバイス

μ PD179F11xマイクロコントローラ

目次

第1章 概要 ...	3
第2章 回路イメージ ...	5
2.1 回路イメージ ...	5
2.2 周辺ハードウェア ...	6
2.3 キー・マトリクス ...	6
2.4 機種選択スイッチ ...	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 プリセット・リモコン・データの記憶方法 ...	10
3.4 使用する周辺の初期設定と動作概要 ...	12
3.5 フロー・チャート ...	15
第4章 処理説明 ...	28
4.1 初期化処理 ...	28
4.2 メイン処理 ...	35
4.3 キー・スキャン処理 ...	36
4.4 セルフ・プログラミング処理 ...	47
4.5 タイマ関連処理 ...	71
4.6 スタンバイ処理 ...	72
第5章 関連資料 ...	76
付録A プログラム・リスト ...	77
付録B 改版履歴 ...	128

資料番号 ZUD-CB-10-0234 (第1版)

発行年月 December 2010

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情報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。

第1章 概 要

このサンプル・プログラムは、フラッシュ・セルフ・プログラミングの機能を使用したプリセット・リモコンのためのプログラムです。本サンプル・プログラムでは、プリセット・リモコン・データの保存領域としてフラッシュROM領域を1ブロックずつ、計2ブロックを使用します。この2ブロックの領域にプリセット・リモコン・データを交互に書き込みます。プリセット・リモコン・データの保存領域は書き込み状態フラグと書き込み回数カウンタを持ち、これらを比較することで最新のプリセット・リモコン・データがどちらのブロックに保存されているか判別します。そして、最新のプリセット・リモコン・データが保存されているブロックからデータをRAM上に読み込み、データを書き換えて、もう一方のブロックにRAM上のデータをフラッシュ・セルフ・プログラミングで書き込みを実行します。スイッチの切り替えにより書き込む機種（TV,ビデオ,DVD1,DVD2）を選択し、特殊なキー入力により書き込むメーカーのリモコン・データを選択することができます。

(1) 使用する周辺の初期設定の主な内容

使用する周辺の初期設定の主な内容は、次のとおりです。

- 割り込みの禁止
- CPU / 周辺ハードウェア・クロックをX1発振クロックの動作に設定
- 内蔵発振器の停止
- ポートの設定
- キー割り込みを禁止に設定
- 低電圧検出機能の設定
- 割り込み要求のマスク
- タイマ関連処理，キー・スキャン処理の設定
- 8ビット・タイマ/イベント・カウンタ50(TM50)の動作開始
- 割り込みの許可

(2) メイン処理の主な内容

メイン処理の主な内容は、次のとおりです。

- キー・スキャン処理の呼び出し
- フラッシュ・セルフ・プログラミング処理の呼び出し
- タイマ関連処理の呼び出し
- スタンバイ処理の呼び出し

(3) キー・スキャン処理の内容

キー・スキャン開始処理の主な内容は、次のとおりです。

- キー・イベント実行要求をクリア
- キー・スキャン要求発生時、以下の処理を実行する
 - キー・センスを実行する
 - キー・コードを作成する
 - キー・コードのノイズ除去を実行し、キー・コードを確定する
 - スイッチ・スキャンを実行する
 - スイッチ・コードのノイズ除去を実行し、スイッチ・コードを確定する

(4) フラッシュ・セルフ・プログラミング処理の内容

フラッシュ・セルフ・プログラミング処理の主な内容は、次のとおりです。

キー・イベント要求が発生し、かつ特殊なキー入力を実行した場合、フラッシュ・セルフ・プログラミングによりリモコン・データをフラッシュROMに書き換える

(5) タイマ関連処理の内容

タイマ関連処理の主な内容は、次のとおりです。

- 10msを生成
- 10ms経過時、キー・スキャン要求を発行

(6) スタンバイ処理の内容

スタンバイ処理の主な内容は、次のとおりです。

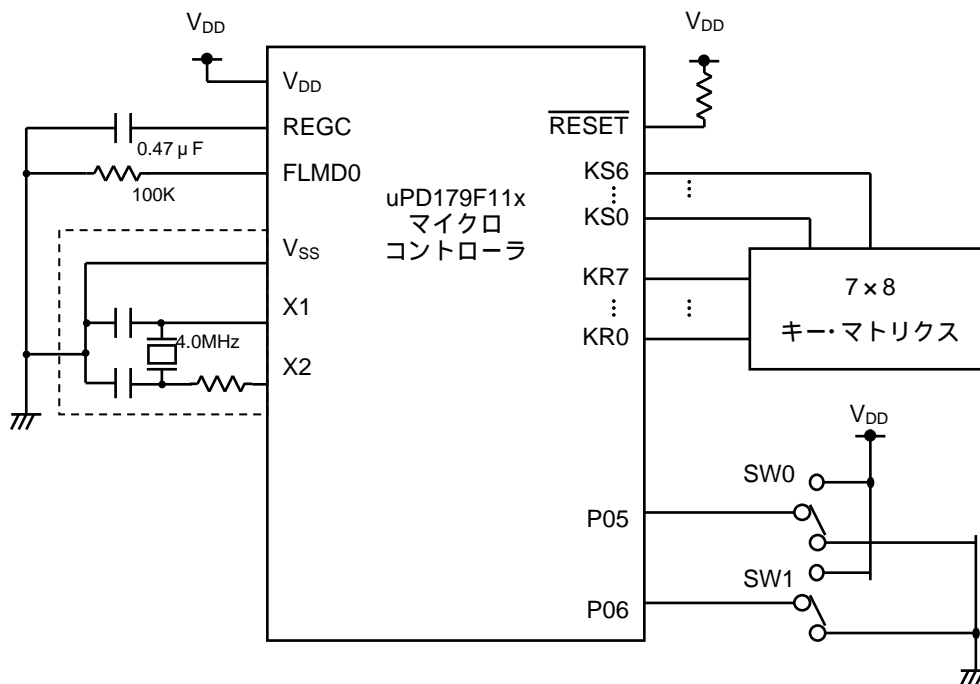
スタンバイ遷移条件を満たしているとき、キーリターン割り込みを有効にしてスタンバイに遷移
スタンバイ中のキー入力検出でスタンバイから復帰

第2章 回路イメージ

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 回路イメージ

回路イメージを次に示します。



- 注意1. 2.0 V V_{DD} 3.6 Vの電圧範囲で使用してください。
- REGCはコンデンサ (0.47 μ F) を介し、 V_{SS} に接続してください。
 - FLMD0は抵抗 (100K) を介し、 V_{SS} に接続してください。
 - 回路図中の端子以外の未使用のポート機能端子はすべて出力ポートのため、オープン (未接続) にしてください。
 - X1を使用する場合は、配線容量などの影響を避けるために、図の破線の部分を次のように配線してください。
 - 配線は極力短くする。
 - 他の信号線と交差させない。また、変化する大電流が流れる線と接近させない。
 - 発振回路のコンデンサの接地点は、常に V_{SS} と同電位となるようにする。大電流が流れるグラウンド・パターンに接地しない。
 - 発振回路から信号を取り出さない。

備考 発振子の選択および発振回路定数についてはお客様において発振評価していただくか、発振子メーカーに評価を依頼してください。

2.2 周辺ハードウェア

使用する周辺ハードウェアを次に示します。

(1) メインシステム・クロック (X1, X2)

X1, X2端子に4.0MHzの発振子を接続します。

(2) 7×8キー・マトリクス (KS0 - KS6, KR0 - KR7)

KS0 - KS6端子, KR0 - KR7端子に7×8キー・マトリクスを接続します。

詳細な構成について2.3項にて説明します。

(3) 機種選択スイッチ (P05, P06)

P05, P06端子にスイッチを接続し, 機種選択に使用します。

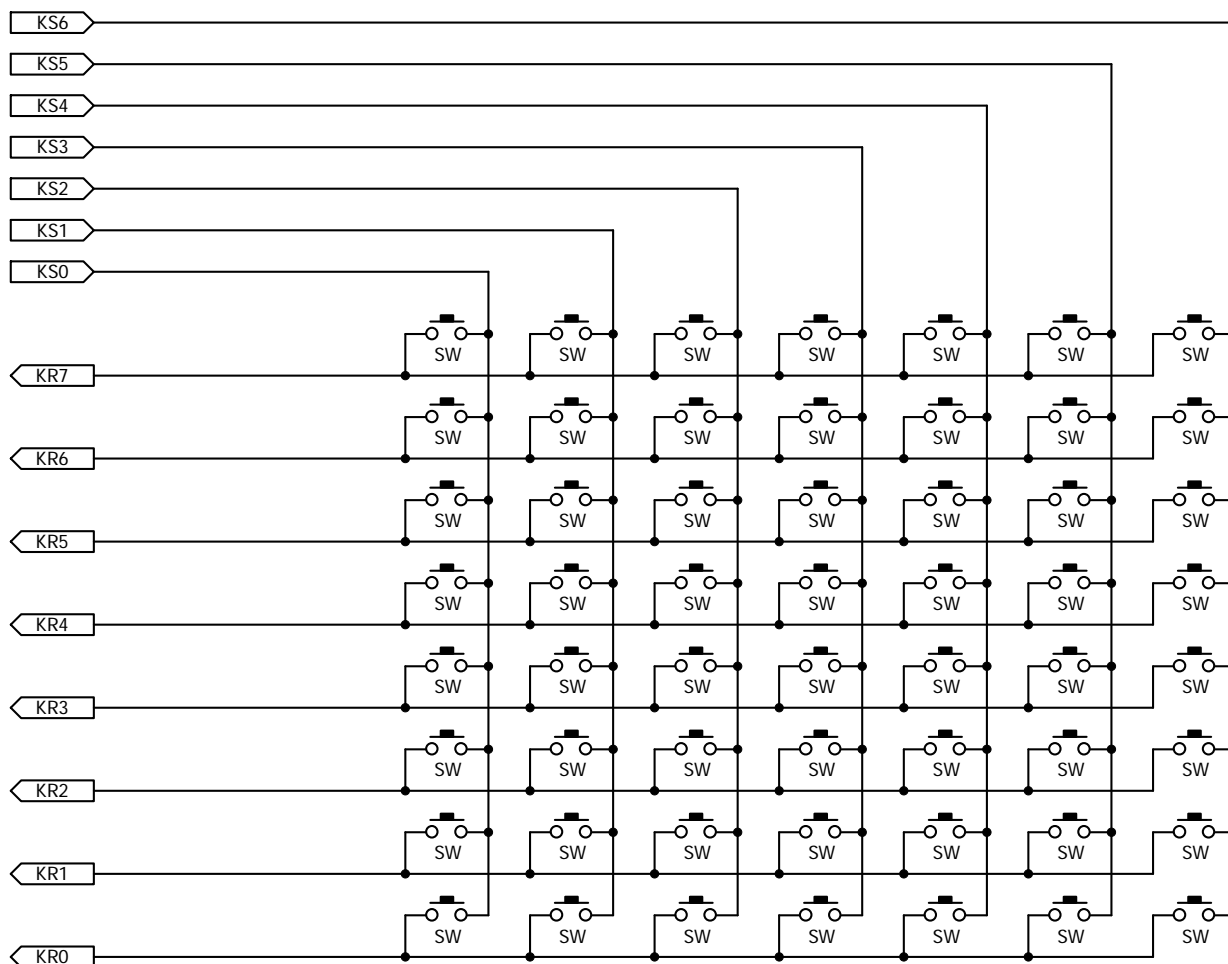
詳細な構成について2.4項にて説明します。

2.3 キー・マトリクス

使用するキー・マトリクスの構成を次に示します。また, キー・マトリクスの回路図を次ページに示します。

KR7(P17)							
KR6(P16)							
KR5(P15)							
KR4(P14)		'4'					
KR3(P13)		'3'					
KR2(P12)		'2'					
KR1(P11)		'1'					
KR0(P10)	Power						
キー・リターン	KS0	KS1	KS2	KS3	KS4	KS5	KS6
キー・センド	(P00)	(P01)	(P02)	(P03)	(P20)	(P21)	(P22)

本サンプル・プログラムでは, Powerキーと'1'~ '4'いずれかのキーを同時に押すことで, 対応するメーカーのリモコン・データをセルフ・プログラミングによってフラッシュROM領域に書き込みます。



2.4 機種選択スイッチ

本サンプル・プログラムでは、スイッチの切り替えにより使用する機種を選択することができます。機種選択スイッチの構成を次に示します。



SW1(P06)	SW0(P05)	使用機種
Low	Low	'TV'
Low	Hi	'Video'
Hi	Low	'DVD1'
Hi	Hi	'DVD2'

第3章 ソフトウェアについて

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

3.1 ファイル構成

ダウンロードする圧縮ファイルのファイル構成は、次のようになっています。

ファイル名	説明	同封圧縮 (*.zip) ファイル	
			
main.asm (アセンブリ言語版)	マイコンのハードウェア初期化処理、キー・スキャン処理、メイン処理、スタンバイ処理、フラッシュ・セルフ・プログラミング処理のソース・ファイル	注1	●
op.asm (アセンブリ言語版)	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)		●
SelfLib.inc (アセンブリ言語版)	セルフ・プログラミング・ライブラリ呼び出しのためのインクルード・ファイル (ヘッダ・ファイル)	●	●
PresetSelf.prw	統合開発環境 PM plus用ワーク・スペース・ファイル		●
PresetSelf.prj	統合開発環境 PM plus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 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。

- ・入力ポート（機種選択スイッチ用） : P05, P06
- ・入力ポート（キー・リターン用） : P10 ~ P17
- ・出力ポート（キー・センド用） : P00 ~ P03, P20 ~ P22
- ・キー・リターン用入力ポートのプルアップ抵抗：
P10 ~ P17の内蔵プルアップ抵抗を使用
- ・フラッシュ・セルフ・プログラミング開始時のFLMD0プルアップ抵抗：
FLMD0用内蔵プルアップ抵抗を使用
- ・約10ms周期でのキー・スキャン要求発行用：
8ビット・タイマ/イベント・カウンタ50をインターバル・タイマ・モードで使用

3.3 プリセット・リモコン・データの記憶方法

3.3.1 記憶領域の構造

プリセット・リモコン・データの書き込みに使用するフラッシュROM領域を、本アプリケーションではセクタと呼んでいます。1セクタの大きさは1ブロック (=1024バイト) で、2ブロックを使用します。各セクタは、先頭の4バイトをフラグ領域、次の4バイトを書き込み回数カウンタとして使用します。また、32バイトのプリセット・リモコン・データ保存領域を4個備えており、それぞれTV用リモコン・データ、VHS用リモコン・データ、DVD1用リモコン・データ、DVD2用リモコン・データを保存します。各データの31バイト目に、データの区切りとしてデリミタを配置しています。

フラグ領域 (4バイト)	書き込み回数カウンタ (4バイト)
プリセット・リモコン・データ保存領域 (TV用) (32バイト)	
プリセット・リモコン・データ保存領域 (VHS用) (32バイト)	
プリセット・リモコン・データ保存領域 (DVD1用) (32バイト)	
プリセット・リモコン・データ保存領域 (DVD2用) (32バイト)	

本サンプル・プログラムでは、2つのセクタをそれぞれブロック6と7に配置しています。これらのセクタはソース上で定義されており、セクタの設定アドレスを変更することが可能です。セクタの設定アドレスを変更するためには、asm版では『CSECT1,CSECT2, CSEC1BLK, CSEC2BLK』の設定値を変更してください。

```

;
;+-----+
;+          +
;+   セクタ1,2のアドレス定義   +
;+          +
;+-----+
CSECT1      EQU    1800H      ;ブロック6先頭アドレス
CSECT2      EQU    1C00H      ;ブロック7先頭アドレス

CSEC1BLK    EQU    6         ;ブロック6
CSEC2BLK    EQU    7         ;ブロック7

```

なお、変更の際には、以下の点について注意してください。

セクタの先頭アドレスは必ずブロックの先頭アドレスにする必要があります。

使用するセクタのアドレスと対応するブロック番号を設定してください。

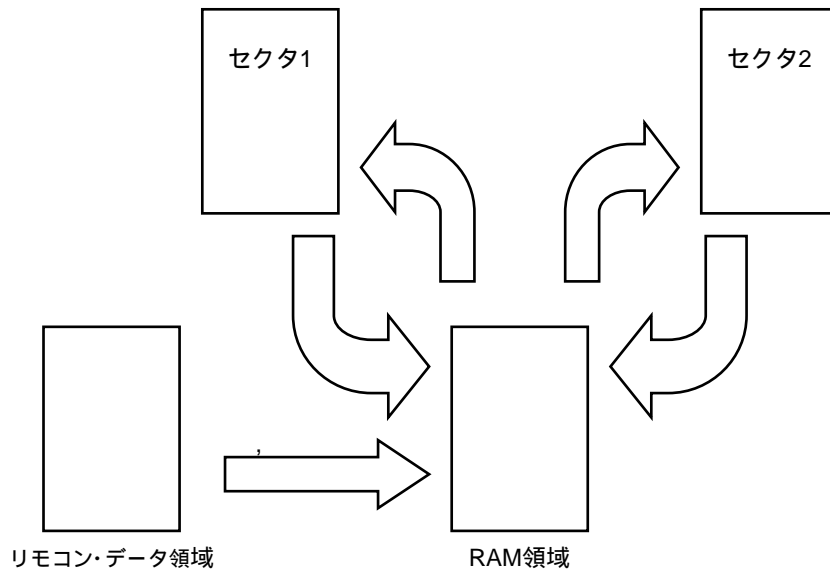
ブロック0,1にはセクタを設定することができません。それ以外の領域を指定してください。

注 メモリ・サイズやブロック番号の最大値は使用するマイコンにより異なります。詳しくはuPD179F1xxユーザーズ・マニュアルを参照してください。

3.3.2 機種データの書き換え

直前に書き込みを行ったセクタからプリセット・リモコン・データをRAM上に読み出し、書き換えるデータ領域のデータを書き換え、書き込み回数をカウントした後、もう一方のセクタにRAM上で書き換えたプリセット・リモコン・データを書き換えます。これを交互に実行することで、フラッシュ書き込み限界を軽減することが可能です。更に、フラグ領域と書き込み回数をチェックすることで、常に最新のプリセット・リモコン・データを書き込むことが可能となっています。

セクタ間のデータ書き換えの詳細については、次の図を参照してください。



セクタ1からRAM上にプリセット・リモコン・データを読み込みます。

RAM上で、書き換える機種データのリモコン・データのみ書き換え、書き込み回数をカウントします。

セルフ・プログラミングによりセクタ2のデータを書き換えます。

セクタ2からRAM上にプリセット・リモコン・データを読み込みます。

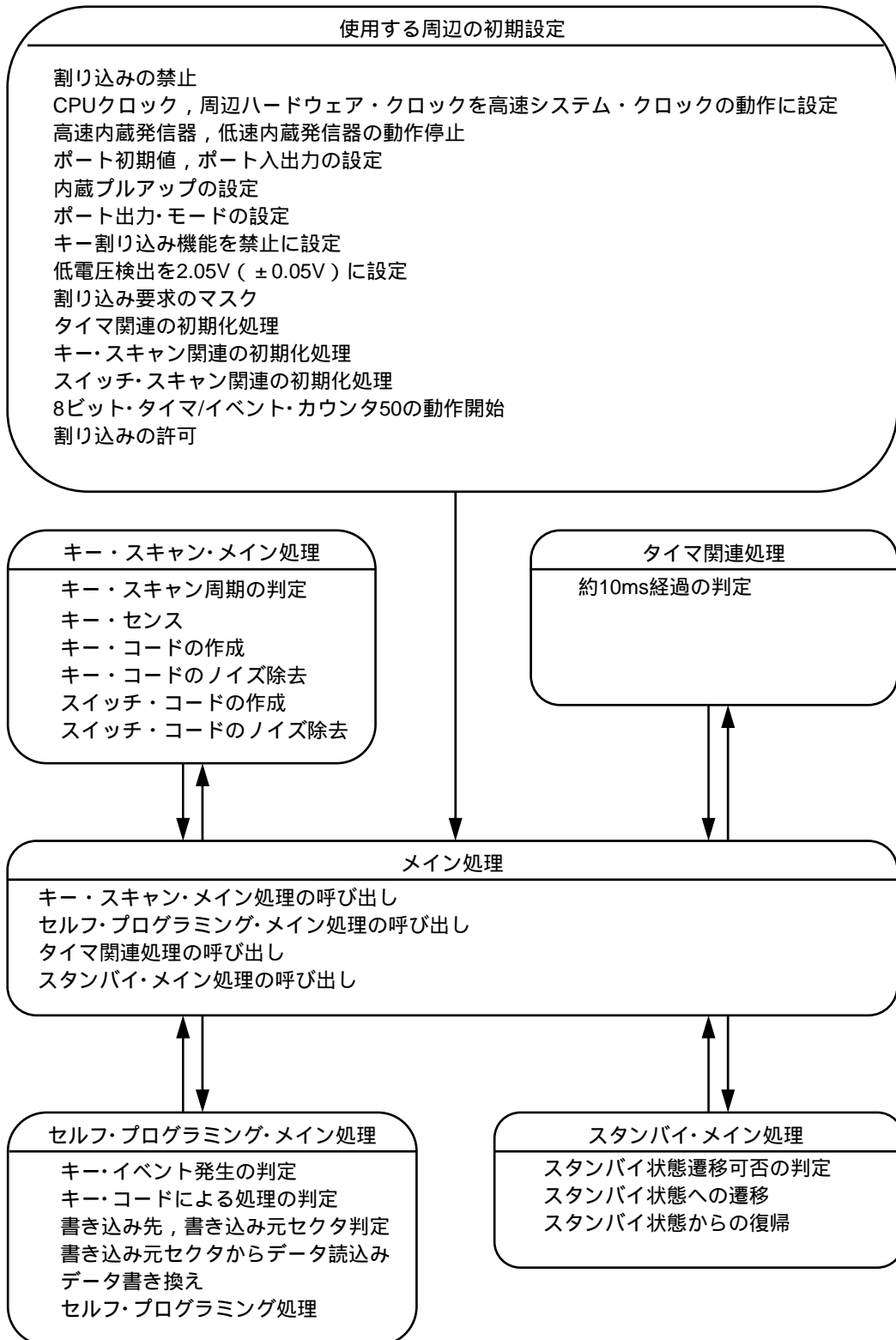
RAM上で、書き換える機種データのリモコン・データのみ書き換え、書き込み回数をカウントします。

セルフ・プログラミングによりセクタ1のデータを書き換えます。

3.4 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，ポート設定，キー割り込み機能の設定，低電圧検出機能の設定，その他各処理の初期化設定などを行います。

初期設定完了後，メイン処理へ分岐し，キー・スキャン処理，フラッシュ・セルフ・プログラミング処理，タイマ関連処理，スタンバイ処理の順番に繰り返し呼び出します。詳細については，次の状態遷移図（ステート・チャート）に示します。



3.4.1 キー・スキャン処理

キー・スキャン処理では、キー・スキャン要求が発生している場合、キー・スキャンとスイッチ・スキャンを実行します。キー・センス処理により入力されたキーを判別し、キー・コード作成処理によってキー・コードを生成します。キー・コードのノイズ除去処理によって得られたキー・コードが3回連続で一致したとき、キー・コードとして確定します。また、キー・コードがキー・オフまたは不正な多重押し以外であった場合、同時にキー・イベント要求が発生します。スイッチ・スキャンはスイッチ・ゲット処理によりスイッチの状態を取得し、スイッチ・コードのノイズ除去処理によって得られたスイッチ・コードが10回連続で一致したとき、スイッチ・コードとして確定します。

また、メイン処理1周でキー・オン・イベントの要求をクリアします。

3.4.2 フラッシュ・セルフ・プログラミング処理

フラッシュ・セルフ・プログラミング処理では、キー・イベント要求が発生し、かつキー入力が有効なキー多重押しであった場合のみセルフ・プログラミングを行います。2つのセクタのうちどちらが最新のプリセット・リモコン・データであるか判定するために、フラグ・チェック、書き込み回数カウンタの比較、デリミタ個数の比較を行います。フラグ・チェックでは、フラグ領域の4バイト全てが55Hである方を最新のプリセット・リモコン・データが保存されているセクタと判断し、フラグ領域の4バイト全てが00Hである方を古いプリセット・リモコン・データが保存されているセクタと判断します。それぞれのセクタのフラグを比較することにより、どちらのセクタが最新のプリセット・リモコン・データであるか判定します。フラグ・チェックでの判定に失敗した場合、書き込み回数カウンタの比較により、どちらのセクタが最新のプリセット・リモコン・データであるか判定します。これらの比較での判定に失敗した場合、両方のセクタに含まれるデリミタの個数により比較を行い、どちらのセクタが最新のプリセット・リモコン・データであるか判定します。以上の判定結果を元に、書き込み先セクタと書き込み元セクタを決定します。決定された書き込み元セクタから最新のプリセット・リモコン・データを書き込みバッファにコピーし、入力キーと機種選択スイッチの状態によって書き込みバッファのデータを書き換えます。また、書き込み回数をインクリメントします。その後、セルフ・プログラミング処理によって書き込みバッファのデータを書き込み先セクタに書き込みます。

フラッシュ・セルフ・プログラミングの動作電圧が $2.0V < V_{DD} < 3.6V$ であるため、セルフ・プログラミング処理の最初に電源検出レベルを $2.05V (\pm 0.05V)$ とした低電圧検出を行います。低電圧検出時はセルフ・プログラミング処理を実行せずにそのまま終了します。低電圧でない場合、FLMD0端子のプルアップを行い、セルフ・プログラミングを実行します。以降の処理の流れは基本的に「78K0マイクロコントローラセルフ・プログラミング・ライブラリ Type01 ユーザーズ・マニュアル」に記載されている手順と同様ですが、本サンプル・プログラム特有の処理として、ベリファイ後に「デリミタ・チェック処理」と「フラグ書き換え処理」が存在します。デリミタ・チェック処理では、書き込んだプリセット・リモコン・データに正しくデリミタが書き込まれているか確認を行います。また、フラグ書き換え処理では、書き込み先セクタのフラグ4バイトをすべて55Hに、書き込み元セクタのフラグ4バイトをすべて00Hに書き換えます。このフラグ書き換え処理により、最新のプリセット・リモコン・データを記憶するセクタのフラグが常に55Hになるようにしています。

3.4.3 タイマ関連処理

タイマ関連処理では、約10ms毎にキー・スキャン要求を発行します。（他に処理を追加した場合、この10msをベースにしたソフトウェア・タイマ・カウント処理をタイマ関連処理に追加してください。）

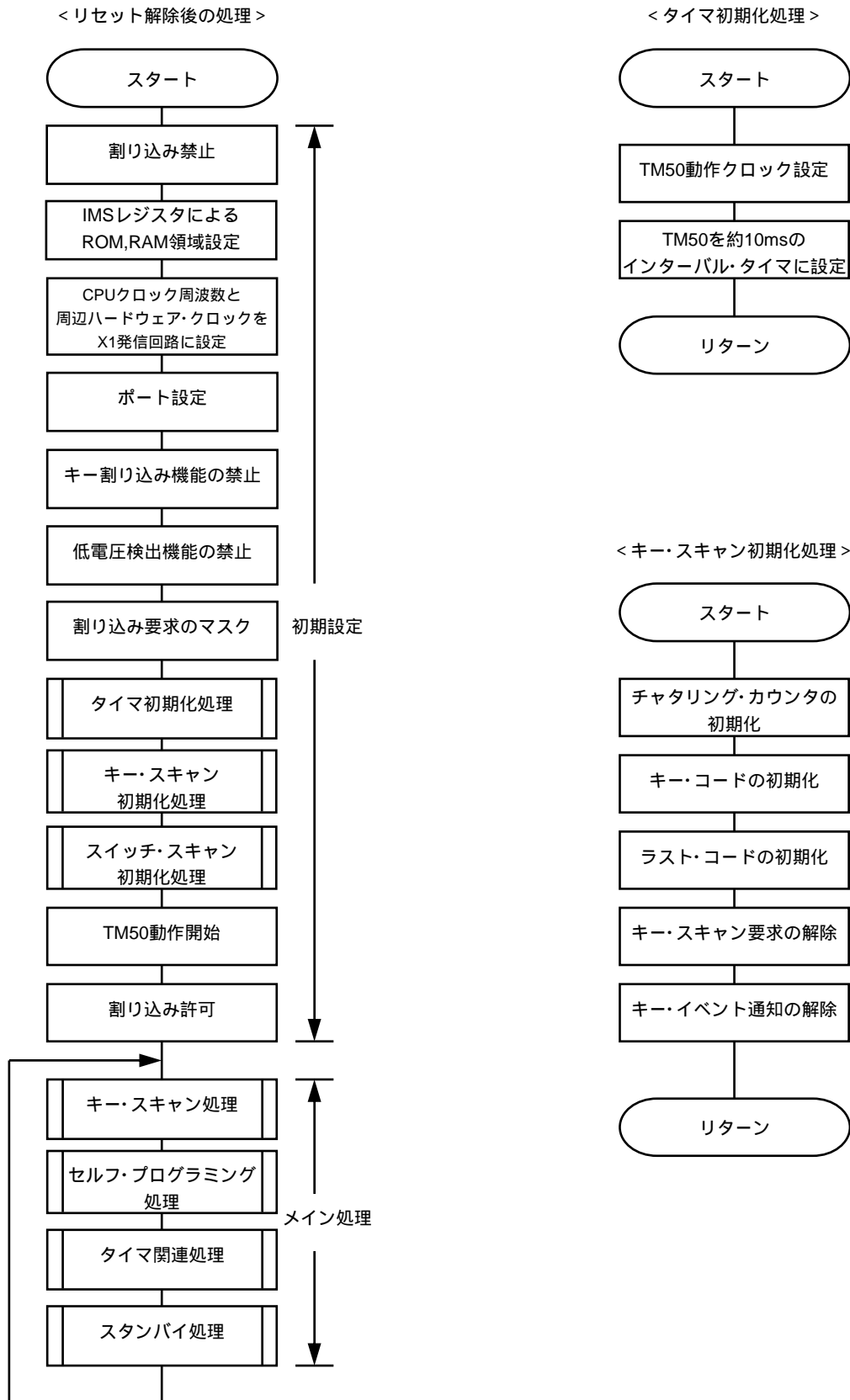
3.4.4 スタンバイ処理

スタンバイ処理では、最初にスタンバイ状態遷移可否の判定を行います。本サンプル・プログラムでは、キー・スキャン処理中の場合スタンバイ状態への遷移を不可能としています。（他に処理を追加した場合、必要に応じてスタンバイ可否判定を追加してください。）遷移可能である場合スタンバイ状態に遷移します。スタンバイ状態からの復帰はキー・リターン入力のみ可能です。スタンバイから復帰する際にキー・スキャン要求を発行します。

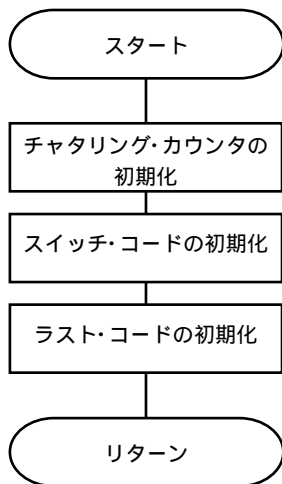
3.5 フロー・チャート

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。

3.5.1 初期化関連

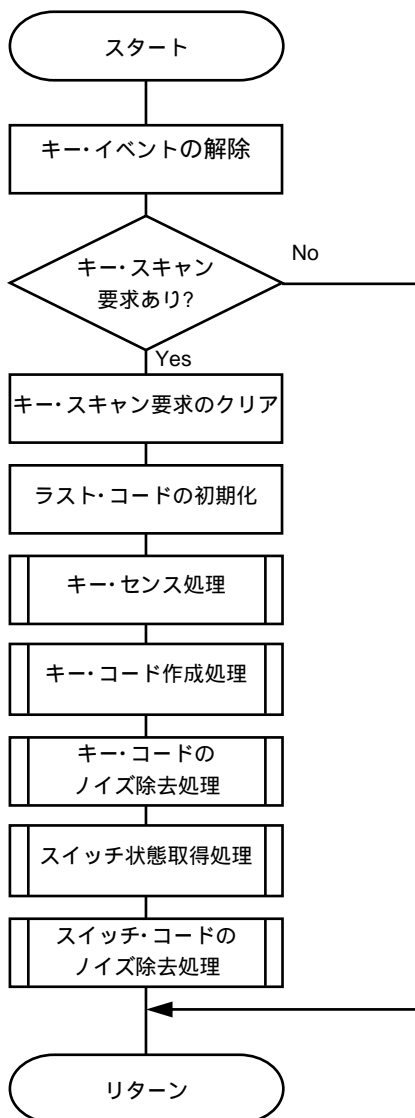


<スイッチ・スキャン初期化処理>



3.5.2 キー・スキャン , スイッチスキャン関連

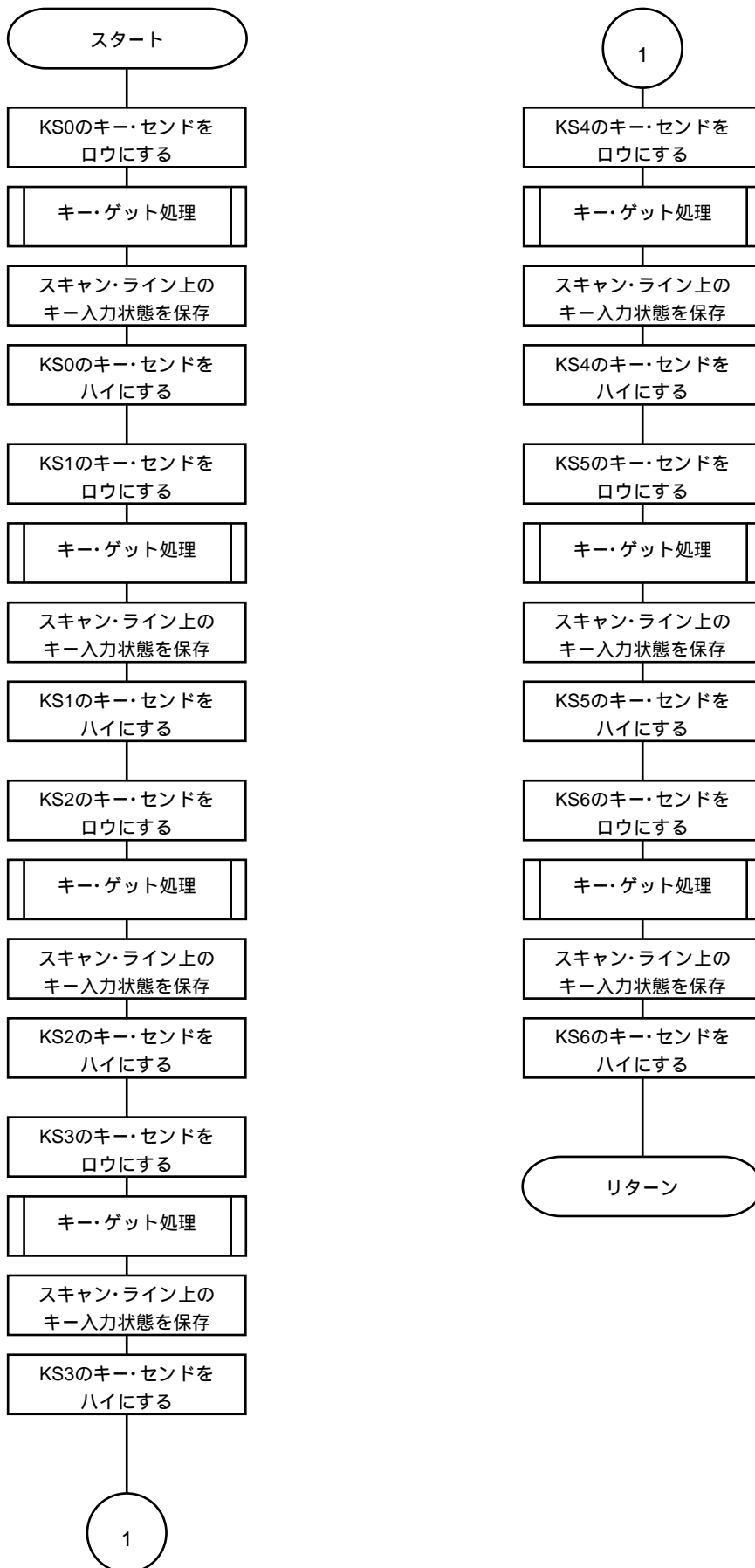
<キー・スキャン・メイン処理>



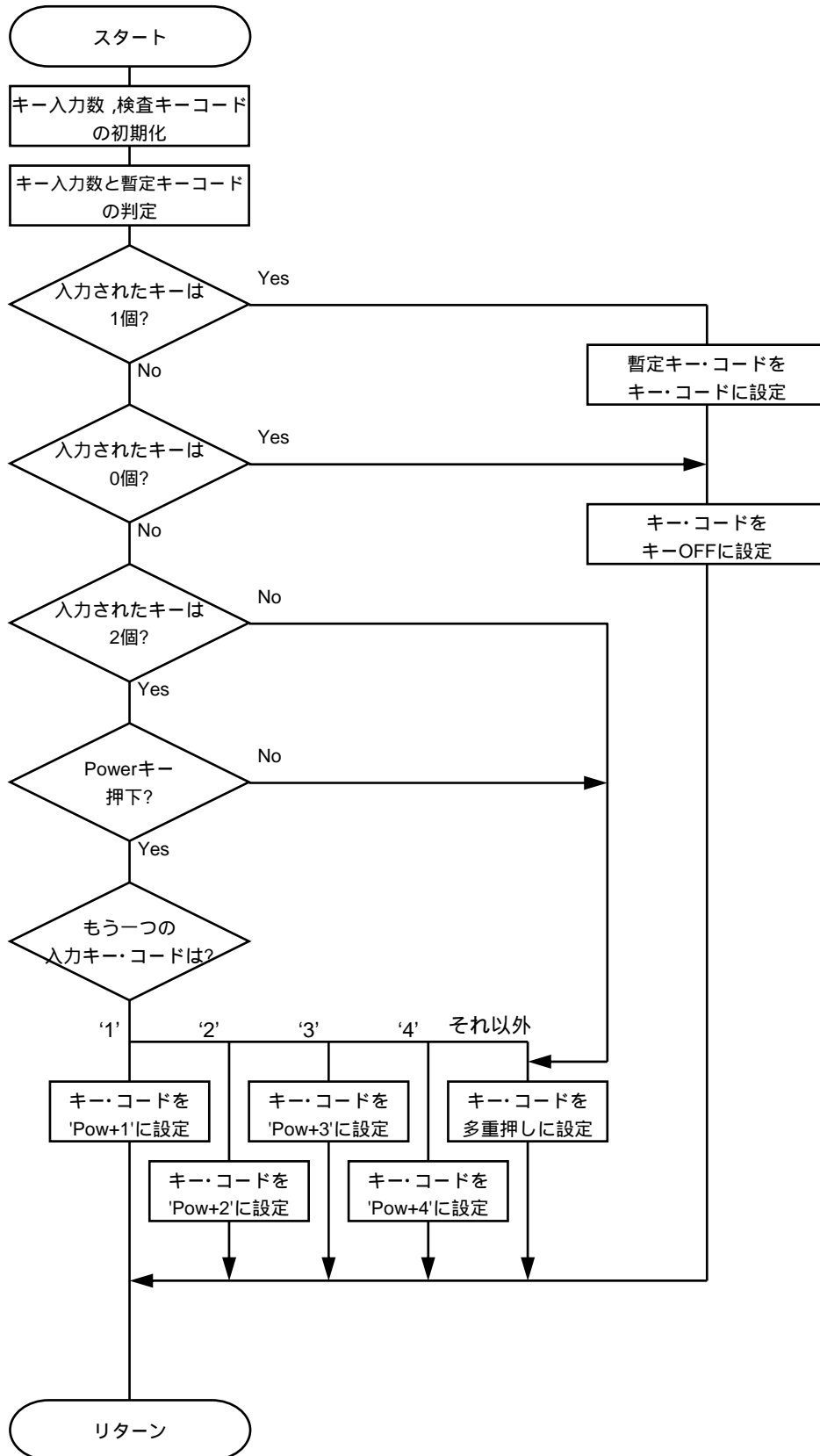
<キー・ゲット処理>



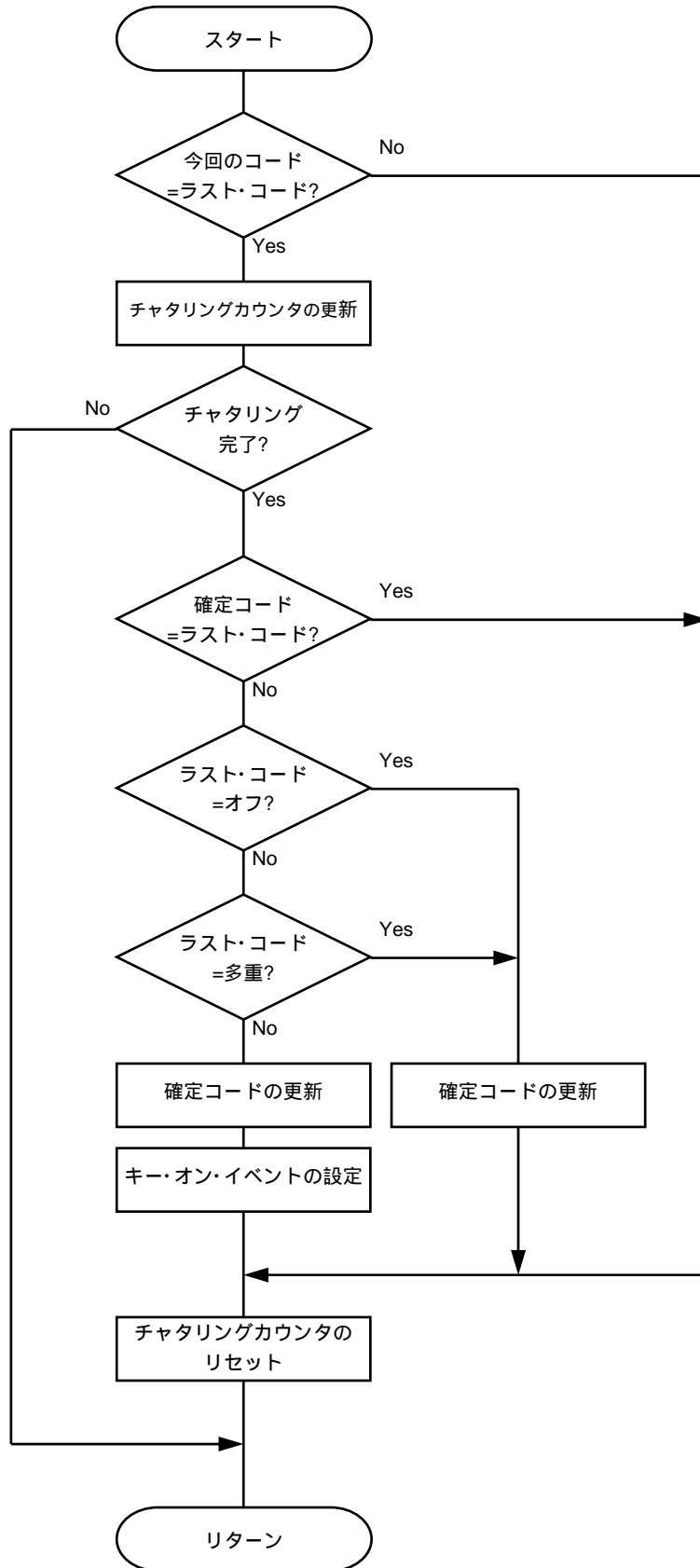
<キー・センス処理>



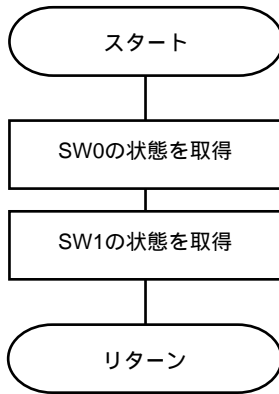
<キー・コード作成処理>



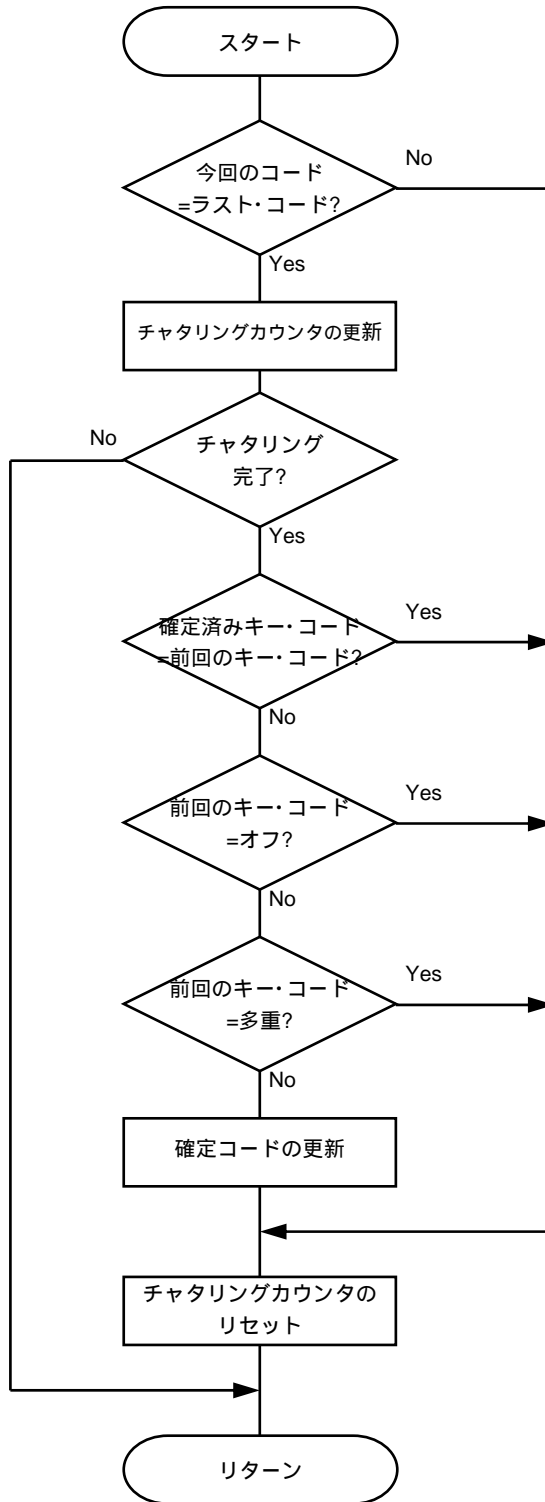
<キー・コードのノイズ除去処理



<スイッチ・ゲット処理>

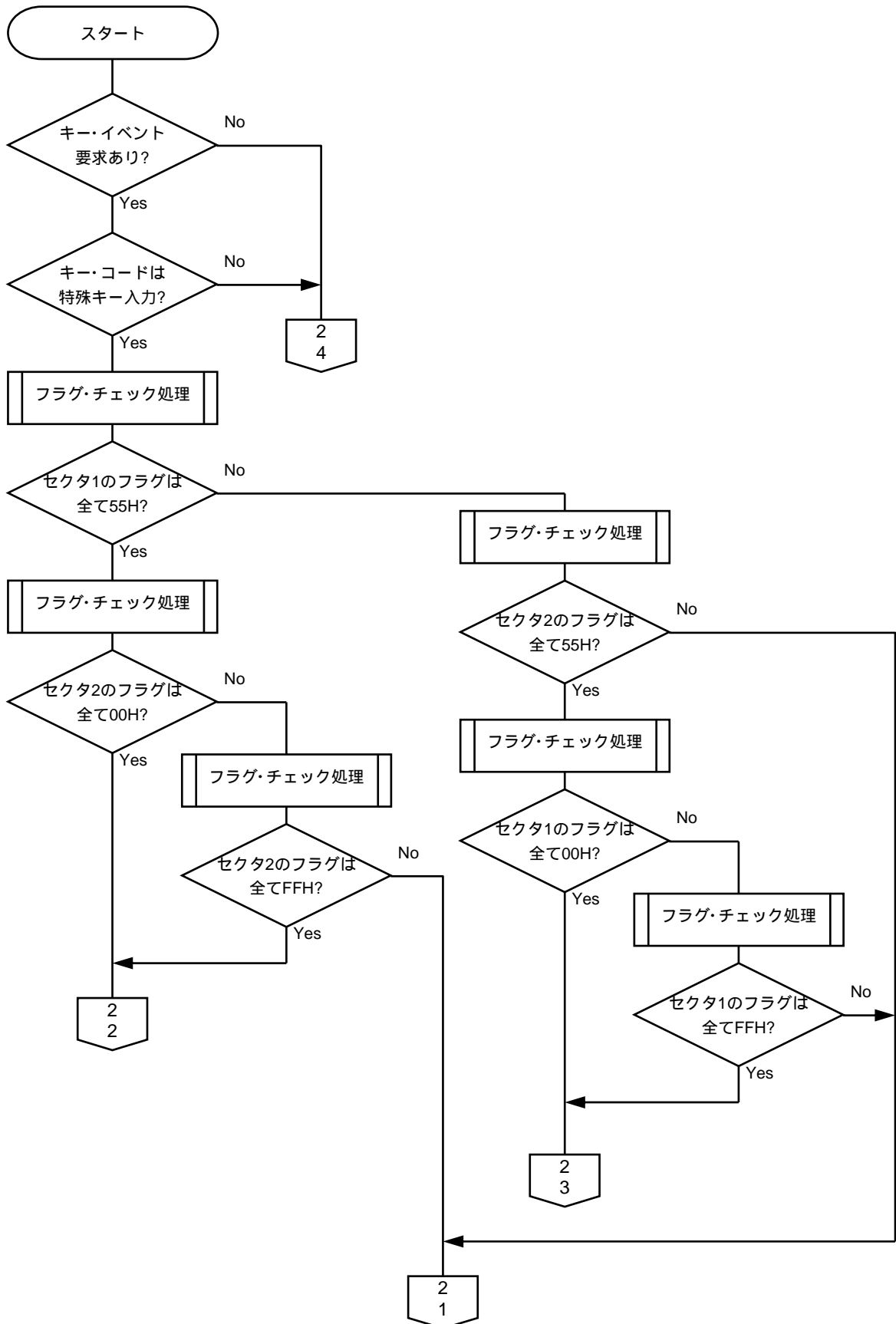


<スイッチ・コードのノイズ除去処理>

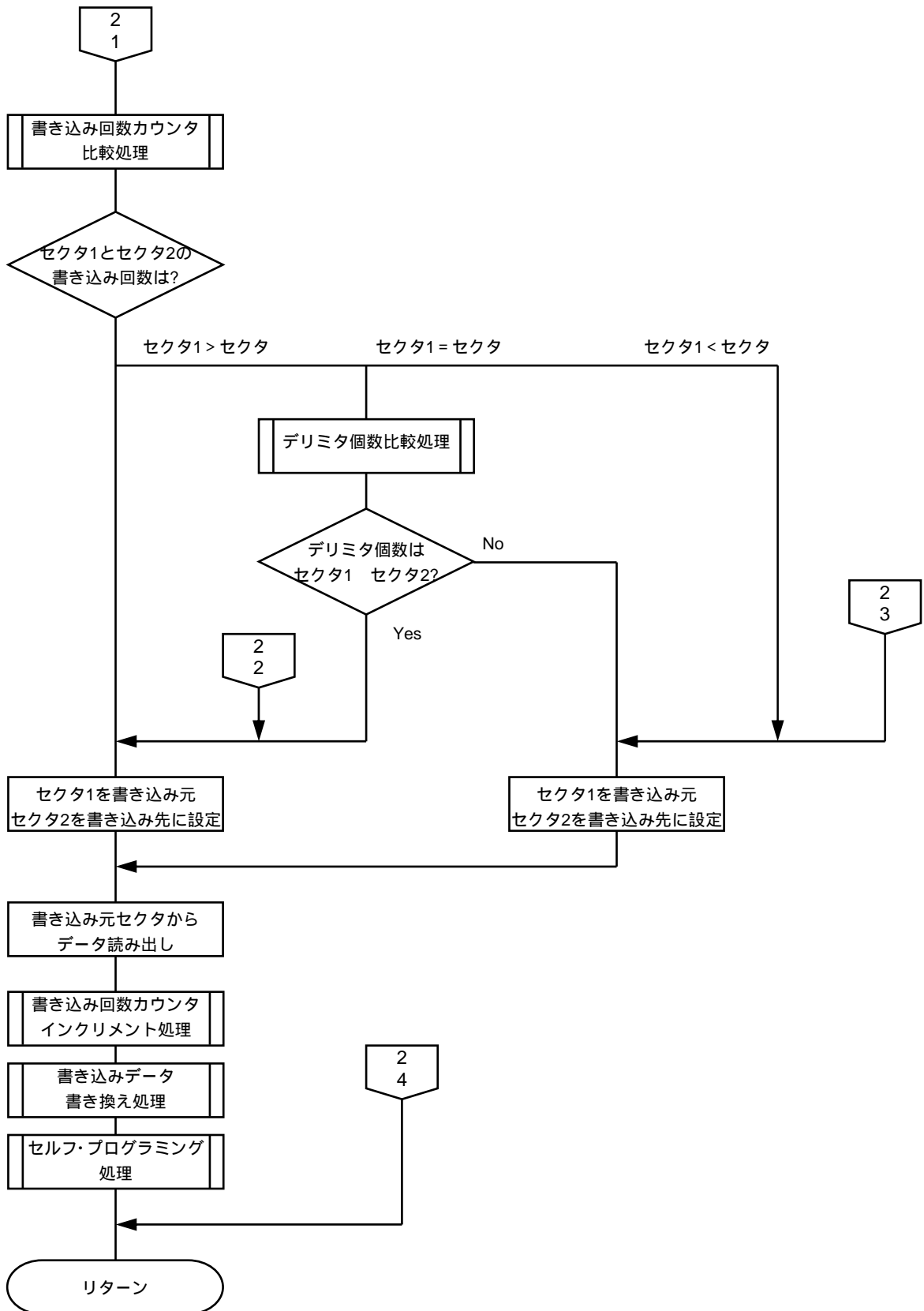


3.5.3 セルフ・プログラミング関連

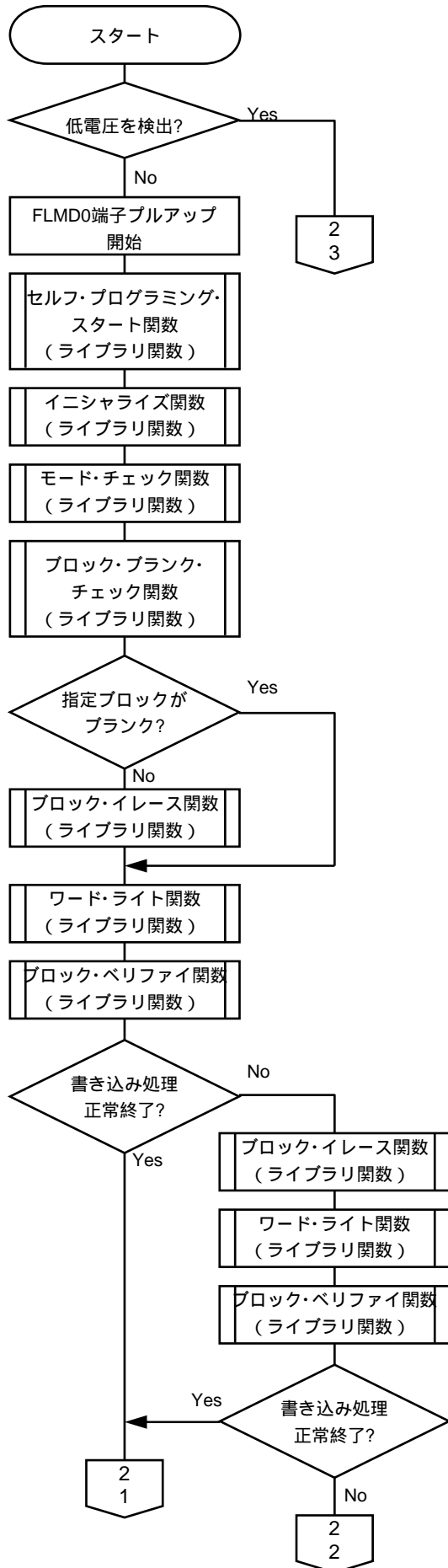
<セルフ・プログラミング・メイン処理>



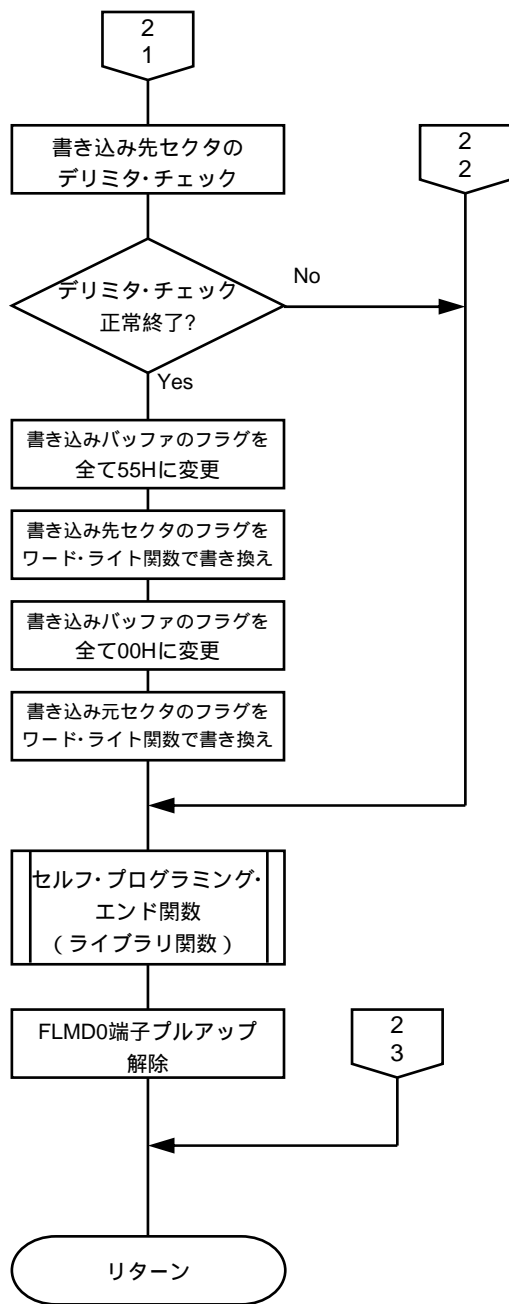
<セルフ・プログラミング・メイン処理>



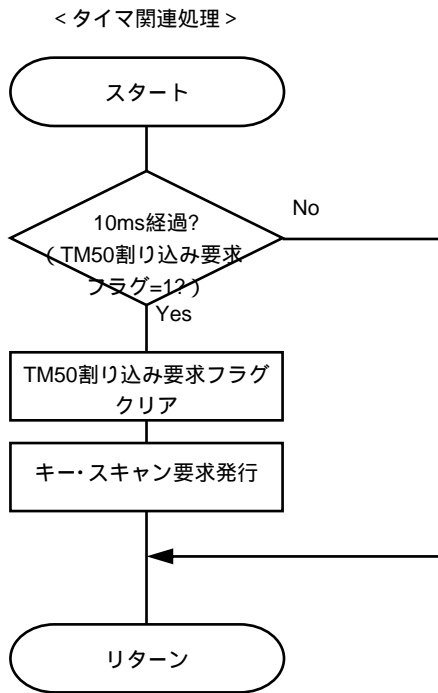
<セルフ・プログラミング処理>



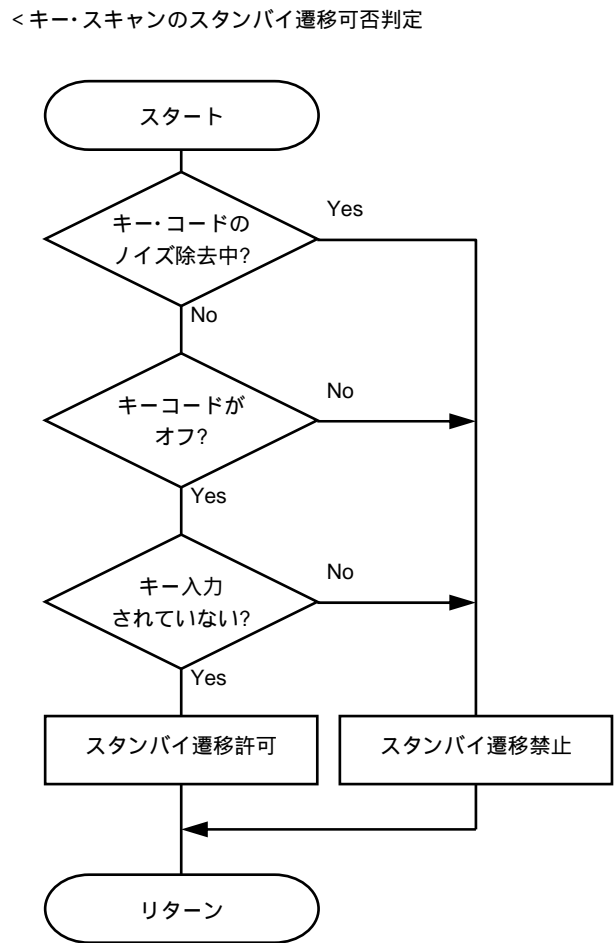
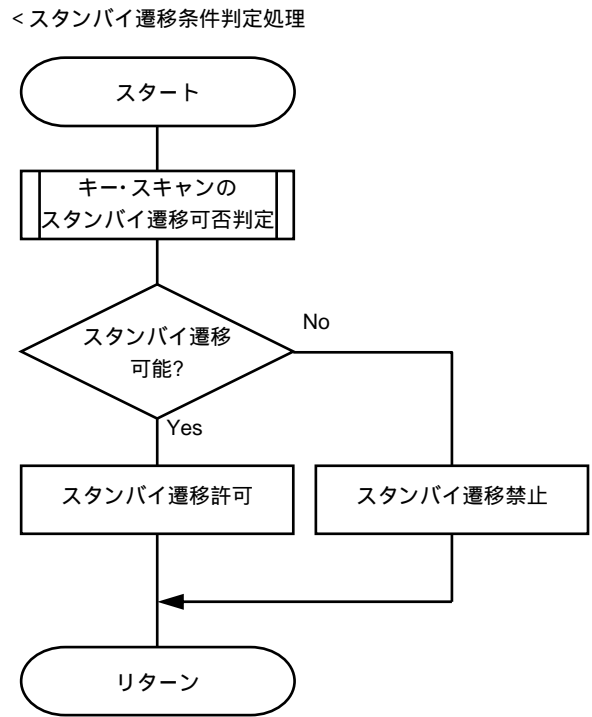
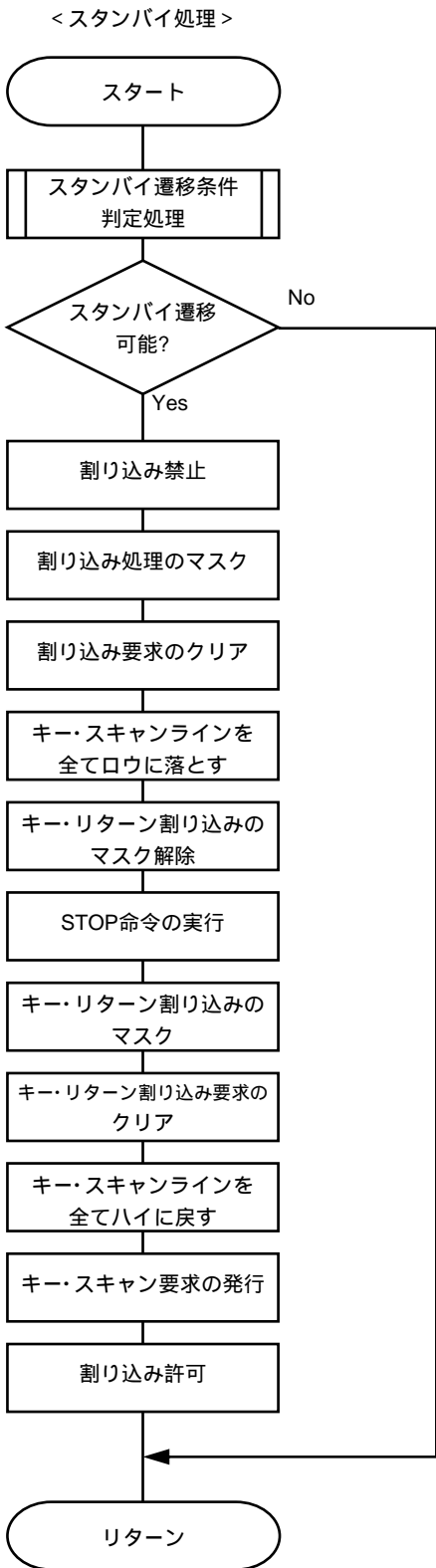
<セルフ・プログラミング処理>



3.5.4 タイマ関連処理



3.5.5 スタンバイ関連



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。

- ・ウォッチドッグ・タイマの動作
- ・低速内蔵発振器の動作
- ・オンチップ・デバッグの動作制御

第4章 処理説明

この章では、初期化処理、メイン処理、キー・スキャン処理、セルフ・プログラミング処理、タイマ関連処理、およびスタンバイ処理についてのソース上での処理内容を説明します。なお、AsmソースとCソースで処理内容に差異が無いため、Asmソースについてのみ処理説明を行います。

レジスタ設定方法等の詳細については、uPD179F1xxユーザーズ・マニュアルを参照してください。

4.1 初期化処理

4.1.1 初期化処理

初期化処理では、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。

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を選択してください。

スタック・ポインタの設定を行います。

クロック周波数の設定を行います。

ポート出力の設定を行います。

内蔵プルアップの設定を行います。

ポート出力モードの設定を行います。

キー割り込み機能の設定を行います。

低電圧検出機能の設定を行います。

割り込み要求をマスクします。

各モジュールの初期化処理を呼び出します。

8ビット・タイマ/イベント・カウンタ50 (TM50)の動作を開始します。^{注1}

割り込みを許可します。^{注1}

メインループ処理へ続きます。

```

*****
;
;*
;*
;*
;*
;*
;*
*****
IINIT:
-----+
DI                                     ;割り込み禁止
-----+
;+ レジスタバンク設定                +
-----+
SEL      RB0
-----+
;+ ROM/RAMサイズの設定                +
-----+
;uPD179F110はROM容量不足のため使用できません
;
; ;MOV      IMS,#42H                    ;uPD179F111使用時の設定
; ;MOV      IMS,#04H                    ;uPD179F112,uPD179F122使用時の設定
; ;MOV      IMS,#0C6H                   ;uPD179F113,uPD179F123使用時の設定
MOV      IMS,#0C8H                    ;uPD179F114,uPD179F124使用時の設定
-----+
;+ スタック・ポインタの設定          +
-----+
MOVW     SP,#STACKTOP                 ;スタック・ポインタを設定
-----+
;+ クロック周波数の設定                +
-----+
MOV      OSCCTL, #01000000B           ;クロック動作モード
;
; ||+++++----- <0>固定
; ++----- EXCLK/OSCSEL: X1発振モード
MOV      MOC, #00000000B              ;高速システム・クロックの動作制御
;
; |+++++----- <0>固定
; +----- MSTOP: X1発振回路動作
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)=fRH
; ||||| |      周辺ハードウェア・クロック(fPRS)=fRH
; ||||| +----- MCS: Read Only
; +++++----- 0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
MOV      RCM, #00000001B              ;内蔵発振器の動作モード設定
;
; |||||+----- RSTOP: 高速内蔵発振器の停止
; |||||+----- LSRSTOP: 低速内蔵発振器の停止
; |++++----- <0>固定
; +----- RSTS: Read Only
;
; 発信安定時間は使用する発振子により異なります。
; 発振子に最適な発信安定時間を選択して下さい。
MOV      OSTST, #00000001B           ;発振安定時間の選択

```

```

;+-----+
;+   ポート出力の初期化処理           +
;+-----+
SPOINI:
MOV   P0,   #00000000B
MOV   P1,   #00000000B
MOV   P2,   #11100000B
;   MOV   P3,   #00000000B           ;uPD179F12x使用時有効にして下さい
MOV   P12,  #00001000B
;+-----+
;+   内蔵プルアップの設定           +
;+-----+
MOV   PU0,  #00000000B
;   ;+++++----- PU00-07: 内蔵プルアップ抵抗を接続しない

MOV   PU1,  #11111111B
;   ;+++++----- PU10-17: 内蔵プルアップ抵抗を接続する

MOV   PU2,  #00011000B           ;uPD179F11x使用時有効にして下さい
;   MOV   PU2,  #00000000B           ;uPD179F12x使用時有効にして下さい

;   MOV   PU3,  #00111111B           ;uPD179F12x使用時有効にして下さい
;   ;*****----- PU10-17: 内蔵プルアップ抵抗を接続する
;   ;+++++----- <0>固定

MOV   PU12, #00000000B
;   ;||||+----- PU120: 内蔵プルアップ抵抗を接続しない
;   ;||||+----- <0>固定
;   ;|||+----- PU123: 内蔵プルアップ抵抗を接続しない
;   ;++++----- <0>固定
;+-----+
;+   ポート出力・モードの設定       +
;+-----+
MOV   POM0, #00001111B
;   ;|||++++----- POM00-03: N-ch O.D.
;   ;++++----- POM04-07: CMOS出力

MOV   POM1, #00000000B
;   ;+++++----- POM10-17: CMOS出力

MOV   POM2, #00011111B
;   ;||*++++----- POM20-22: N-ch O.D.
;   ;++++----- POM25-27: CMOS出力

;   MOV   POM3, #00111111B           ;uPD179F11x使用時有効にして下さい
;   ;+++++----- POM30-35: CMOS出力
;   ;++++----- <0>固定
;   MOV   POM3, #00000000B           ;uPD179F12x使用時有効にして下さい
;   ;+++++----- POM30-35: CMOS出力
;   ;++++----- <0>固定

MOV   POM12, #00000000B
;   ;||||+++----- POM120-122: CMOS出力
;   ;++++----- <0>固定
;+-----+
;+   ポート入出力モードの設定       +
;+-----+
MOV   PM0,  #11100000B
;   ;*|||++++----- KS3-6
;   ;||+----- 未使用
;   ;++----- SW0-1

MOV   PM1,  #11111111B
;   ;+++++----- KR0-7

MOV   PM2,  #00011000B
;   ;||*++++----- KS0-2
;   ;+++----- LED1-3

;   MOV   PM3,  #11111111B           ;uPD179F12x使用時有効にして下さい
;   ;*****

```



```

;+-----+
;+   キー割り込み機能の設定           +
;+-----+
MOV    KRML, #0000000B
;      +-----+----- KRM0-7: キー割り込み信号を検出しない
MOV    KRMH, #0000000B
;      |-----+----- KRM8-14: キー割り込み信号を検出しない
;      +-----+----- <0>固定
CLR1   KRIF0
; CLR1   KRIF1           ; (拡張キー使用時はこの処理を有効にしてください)
SET1   KRMK0
; SET1   KRMK1           ; (拡張キー使用時はこの処理を有効にしてください)

;+-----+
;+   低電圧検出機能の設定           +
;+-----+
SET1   LVIMK
MOV    LVIS, #00001110B           ;低電圧検出レベルの選択
;      |||+-----+----- LVIS3-0: 2.05 V±0.05 V
;      +-----+----- <0>固定
MOV    LVIM, #10000000B           ;低電圧検出レベルの選択
;      |||||+-----+----- LVIF: Read Only
;      |||||+-----+----- LVIMD: 内部割り込み発生
;      |||||+-----+----- LVISEL: VDDのレベルを検出
;      |-----+----- <0>固定
;      +-----+----- LVION: 低電圧検出動作許可

;+-----+
;+   割り込み要求のマスク           +
;+-----+
_INTMSK           ;全割り込み要因のマスク
_CLRIF           ;全割り込み要求フラグのクリア

;+-----+
;+   その他初期化処理の呼び出し     +
;+-----+
CALL   !STIMINI   ;タイマ初期化処理
CALL   !SKEYINI   ;キースキャン初期化処理
CALL   !SSWINI    ;スイッチスキャン初期化処理

SET1   TCE50      ;タイマ50動作開始
CLR1   TMIF50

EI           ;割り込み許可

BR     MMAIN

```

4.1.2 タイマ初期化処理

初期化処理では、次の動作を行います。

8ビット・タイマ/イベント・カウンタ50 (TM50)の初期設定を行います。

```
*****
;
;*
;
;*      タイマ関連の初期化処理      *
;
;*
;
;*****
STIMINI:
MOV     TCL50,#CITCL50      ;タイマ50のクロック設定
MOV     CR50,#CICR50
MOV     TMC50,#CITMC50     ;タイマ50の動作設定

RET
```

4.1.3 キー・スキャン初期化処理

キー・スキャン初期化処理では、次の動作を行います。

キー・スキャン関連の変数の初期設定を行います。

キー・スキャン要求、キー・イベント通知を解除します。

```
*****
;
;
;   キー・スキャン関連の初期化処理
;
;
*****
SKEYINI:
MOV     RCHTCNT,#CCHTCNT;チャタリングカウンタをリセット
MOV     RKEYCODE,#CKC_OFF      ;確定コード 【キーオフ】
MOV     RCHTCODE,#CKC_OFF      ;ラストコード 【キーオフ】

CLR1    FKEY_RQ                ;キー・スキャン要求解除
CLR1    FKEY_EV                ;キー・イベント通知解除

RET
```

4.1.4 スイッチ・スキャン初期化処理

初期化処理では、次の動作を行います。

スイッチ・スキャン関連の変数の初期設定を行います。

```
*****  
;*                                     *  
;*   スイッチスキャン関連の初期化処理   *  
;*                                     *  
*****  
SSWINI:  
MOV    RSWCHTCT,#CSWCHTCT    ;チャタリングカウンタをリセット  
MOV    RSWCODE,#CSW_TV      ;確定コード  【'TV'】  
MOV    RSWCHTCD,#CSW_TV     ;ラストコード  【'TV'】  
  
RET
```

4.2 メイン処理

メイン処理では、次の動作を行います。

各モジュールのメイン処理を順次呼び出します。

```

;+*****+
;+
;+
;+   メイン処理
;+
;+
;+-----+
MMAIN:
CALL   !MKYSCAN      ;キースキャン処理呼び出し
CALL   !MSELF        ;Flash書き込み処理呼び出し
CALL   !MTIMER       ;タイマ処理呼び出し
CALL   !MSTB         ;スタンバイ処理呼び出し

BR     MMAIN

```

4.3 キー・スキャン処理

4.3.1 キー・スキャン・メイン処理

キー・スキャン・メイン処理では、次の動作を行います。

キー・イベント通知を解除します。

キー・スキャン要求が発生しているとき、キー・スキャン要求を解除して の処理に移行します。未発生
のときは に移行します。

キー・センス処理を実行します。

キー・コード作成処理を実行します。

キー・コード・ノイズ除去を実行します。

スイッチ・コード作成処理を実行します。

スイッチ・コード・ノイズ除去を実行します。

キー・スキャン・メイン処理を終了します。

```

*****
;
;*
;
;*
;
;*      キー・スキャン・スイッチ・スキャン・メイン処理
;*
;*
;*
;
;*****
MKYSCAN:
  CLR1      FKEY_EV          ;メイン1周でイベントをクリア
;
  BTCLR     FKEY_RQ,$LKS200  ;キー・スキャン要求有り? Yes,
  BR        LKS900          ;No,
LKS200:
  CALL     !SKYSENCE        ;キー・センス
;
  CALL     !SMKCODE         ;コード作成
;
  CALL     !SRMCAHT         ;ノイズ除去
;
;
  CALL     !SSWGET          ;スイッチ状態取得
;
  CALL     !SSWCAHT         ;ノイズ除去
;
;
LKS900:
  RET

```

4.3.2 キー・センス処理

キー・センス処理では、次の動作を行います。

キー・センス・ライン0をLOWに下げます。

キー・ゲット処理を実行し、キー・リターン値を取得します。

キー・センス・ライン0をHIに上げます。

キー・センス・ライン1~6 (uPD179F12xは1~7) について、 ~ と同等の処理を実行します。

キー・センス処理を終了します。

また、キー・ゲット処理では、次の動作を行います。

電圧安定の間ウェイトを行います。

キー・リターン値を取得します。

キー・ゲット処理を終了します。

```

*****
,*
,*   キーセンス
,*
,*
*****
,*   [O]: RKEYBUF[n]   ... キーマップ
,*
*****
SKYSENCE:
CLR1   PKS0           ;キーセンド# 0
CALL   !SKYGET
MOV    RKEYBUF,A
SET1   PKS0

CLR1   PKS1           ;キーセンド# 1
CALL   !SKYGET
MOV    RKEYBUF+1,A
SET1   PKS1

CLR1   PKS2           ;キーセンド# 2
CALL   !SKYGET
MOV    RKEYBUF+2,A
SET1   PKS2

CLR1   PKS3           ;キーセンド# 3
CALL   !SKYGET
MOV    RKEYBUF+3,A
SET1   PKS3

CLR1   PKS4           ;キーセンド# 4
CALL   !SKYGET
MOV    RKEYBUF+4,A
SET1   PKS4

CLR1   PKS5           ;キーセンド# 5
CALL   !SKYGET
MOV    RKEYBUF+5,A
SET1   PKS5

CLR1   PKS6           ;キーセンド# 6
CALL   !SKYGET
MOV    RKEYBUF+6,A
SET1   PKS6

;-- uPD179F12x使用時は有効にしてください --
$_IF (0)
CLR1   PKS7           ;キーセンド# 7
CALL   !SKYGET
MOV    RKEYBUF+7,A
SET1   PKS7
$ENDIF
RET

*****
,*
,*   キーリターンの値を取得
,*
,*
*****
,*   [O]: Areg   ... キーリターンの値
,*
*****
SKYGET:
MOV    B,#21           ;動作クロック4MHz時15usウェイト (電圧安定時間)
JKSWA100:
NOP
DBNZ   B,$JKSWA100    ;電圧安定までのウェイト

MOV    A,PKEYRET       ;キーリターンポートから, データ取得
XOR    A,#11111111B   ;( 負論理 )

```


4.3.3 キー・コード作成

キー・コード作成処理では、次の動作を行います。

キー・リターン・バッファのビット0が1か0かチェックします。0であった場合は の処理に移行します。

1であった場合は の処理に移行します。

キーが入力されたと判断し、アクティブなキーの数をカウントします。また、暫定キー・コードを検査中のキーにします。

検査中のキーを更新します。

ビット1~7について、 ~ と同等の処理を実行します。

チェックするキー・リターン・バッファを更新します。全てのキー・リターン・バッファがチェック済みである場合は に移行します。未チェックのキー・リターン・バッファがある場合は に移行します。

入力されたキーが1個かどうかチェックします。1個であった場合は の処理に移行します。それ以外の場合は の処理に移行します。

キー・コードを暫定キー・コードに設定します。

入力されたキーが0個かどうかチェックします。0個であった場合は の処理に移行します。それ以外の場合は の処理に移行します。

キー・コードをキー・オフに設定します。

入力されたキーが2個かどうかチェックします。2個であった場合は の処理に移行します。それ以外の場合は の処理に移行します。

Powキーが押されているかチェックします。押されている場合は の処理に移行します。押されていない場合は の処理に移行します。

'1'キー~'4'キーが押されているかチェックします。押されている場合は対応したキー・コード('Pow+1'~'Pow+4')を設定して の処理に移行します。それ以外の場合は の処理に移行します。

キー・コードをキー多重押しに設定します。

キー・コード作成処理を終了します。


```

JKMC260:
  INC      B

  MOV      A,[HL]
  BF      A.3,$JKMC280          ;ビット3 = 1 ? No,
  INC      D
  MOV      A,B
  MOV      E,A
JKMC280:
  INC      B

  MOV      A,[HL]
  BF      A.4,$JKMC300          ;ビット4 = 1 ? No,
  INC      D
  MOV      A,B
  MOV      E,A
JKMC300:
  INC      B

  MOV      A,[HL]
  BF      A.5,$JKMC320          ;ビット5 = 1 ? No,
  INC      D
  MOV      A,B
  MOV      E,A
JKMC320:
  INC      B

  MOV      A,[HL]
  BF      A.6,$JKMC340          ;ビット6 = 1 ? No,
  INC      D
  MOV      A,B
  MOV      E,A
JKMC340:
  INC      B

  MOV      A,[HL]
  BF      A.7,$JKMC360          ;ビット7 = 1 ? No,
  INC      D
  MOV      A,B
  MOV      E,A
JKMC360:
  INC      B

  INCW     HL                    ;次のワークエリアをさす
  DBNZ    C,$JKMC200            ;全エリア終了? No,

;*****
;*      アクティブなキー数の検査 & 今回のコード決定      *
;*****
;JKMC700:
;  MOV     A,D
;  CMP     A,#1                    ;キー入力の一つのみ?
;  BNZ    $JKMC720                ; No,
;  MOV     A,E                    ; Yes,作成したキーコードを返す
;  BR     JKMC900
JKMC720:
  CMP     A,#0                    ;キーの入力無し?
  BNZ    $JKMC760                ; No,
  MOV     A,#CKC_OFF              ; Yes,【オフコード】を返す
  BR     JKMC900
JKMC760:
  CMP     A,#2                    ;キー入力は2個?
  BNZ    $JKMC890                ; No,

```

```
*****
;
;*   アクティブなキー数が2個の場合   *
*****
;-- 有効な多重入力をチェックする --
BF      FTMP_POW,$JKMC890          ;POWER KEY ON ? No,

;-- 以降別の有効な多重キーの対応するビットをチェックする --
BF      FTMP_KY1,$JKMC820
MOV     A,#CKC_F1                  ;'Pow+1'キーコードを返す
BR      JKMC900
JKMC820:
BF      FTMP_KY2,$JKMC830
MOV     A,#CKC_F2                  ;'Pow+2'キーコードを返す
BR      JKMC900
JKMC830:
BF      FTMP_KY3,$JKMC840
MOV     A,#CKC_F3                  ;'Pow+3'キーコードを返す
BR      JKMC900
JKMC840:
BF      FTMP_KY4,$JKMC890
MOV     A,#CKC_F4                  ;'Pow+4'キーコードを返す
BR      JKMC900

;-- 不正な多重入力 --
JKMC890:
MOV     A,#CKC_MLT                 ;【多重】キーコードを返す

JKMC900:
RET
```

4.3.4 キー・コード・ノイズ除去

キー・コード・ノイズ除去処理では、次の動作を行います。

今回取得したキー・コードと前回取得したキー・コードが同じかチェックします。同じであった場合は の処理に移行します。異なる場合は の処理に移行します。

前回取得したキー・コードを今回取得したキー・コードに設定し、 の処理へ移行します。

チャタリング・カウンタをカウントします。また、同時にノイズ除去が完了したか確認します。ノイズ除去が完了している場合は の処理に移行します。ノイズ除去が未完了の場合は の処理に移行します。

確定済みのキー・コードと前回取得したキー・コードが同じかチェックします。同じであった場合は の処理に移行します。異なる場合は の処理に移行します。

確定済みのキー・コードが「キー・多重押し」かどうかチェックします。「キー・多重押し」であった場合は の処理に移行します。「キー・多重押し」でない場合は の処理に移行します。

前回取得したキー・コードが「キー・オフ」かどうかチェックします。「キー・オフ」であった場合は の処理に移行します。「キー・オフ」でない場合は の処理に移行します。

確定済みのキー・コードを「キー・オフ」に設定し、 の処理に移行します。

前回取得したキー・コードが「キー・オフ」かどうかチェックします。「キー・オフ」であった場合は の処理に移行します。「キー・オフ」でない場合は の処理に移行します。

前回取得したキー・コードが「キー・多重押し」かどうかチェックします。「キー・多重押し」であった場合は の処理に移行します。「キー・多重押し」でない場合は の処理に移行します。

確定済みのキー・コードを前回取得したキー・コードに設定し、 の処理に移行します。

確定済みのキー・コードを前回取得したキー・コードに設定します。また、キー・イベント要求を発生させます。

チャタリング・カウンタをリセットします。

キー・コード・ノイズ除去処理を終了します。

```

*****
;
;*
;*
;* キーコードの確定処理
;*
;*
*****
;* [I] : Areg ... 今回のコード
;* [IO]: RCHTCODE ... ラストコード
;* [IO]: RKEYCODE ... 確定コード
;* [O] : FKEY_EV ... キーイベント通知フラグ
*****
SRMCAHT:
    CMP    A,RCHTCODE      ;今回 = ラスト?
    BZ     $JRC200         ; Yes,

    MOV    RCHTCODE,A      ;ラストコードを更新
    BR     JRC800         ;カウンタをリセット

JRC200:
    DBNZ   RCHTCNT,$JRC900 ;ノイズ除去完了? No,

    MOV    A,RCHTCODE
    CMP    A,RKEYCODE      ;確定コード = ラストコード?
    BZ     $JRC800        ; Yes,

*****
;
;* 「多重」の場合の判定
;*
*****
    CMP    RKEYCODE,#CKC_MLT ;確定コードは「多重」?
    BNZ    $JRC400         ; No,
    CMP    RCHTCODE,#CKC_OFF ;「多重」から「オフ」への変化?
    BNZ    $JRC800        ; No,

    MOV    RKEYCODE,#CKC_OFF ; Yes,
    BR     JRC800

*****
;
;* その他の場合の判定
;*
*****
JRC400:
    CMP    RCHTCODE,#CKC_OFF ;ラストコードが「オフ」?
    BZ     $JRC420         ; Yes,
    CMP    RCHTCODE,#CKC_MLT ;ラストコードが「多重」?
    BNZ    $JRC600        ; No,

JRC420:
    MOV    A,RCHTCODE      ;確定コードを更新 (オフ | 多重)
    MOV    RKEYCODE,A
    BR     JRC800

JRC600:
    MOV    A,RCHTCODE      ;確定コードを更新
    MOV    RKEYCODE,A
    SET1   FKEY_EV        ;キーオンイベントを発行

JRC800:
    MOV    RCHTCNT,#CCHTCNT ;カウンタをリセット

JRC900:

```

4.3.5 スイッチ・コード作成

スイッチ・コード作成処理では、次の動作を行います。

SW0,SW1の状態を取得します。

```
*****
;
;*
;
;*   スイッチ・コードを取得
;
;*
;
;*****
;* [O]: Areg ... スイッチ・コード
;*****
SSWGET:
  MOV     A,#0
;-- スイッチ・コード生成 --
  MOV1    CY,PSW0           ;SW0の状態を取得
  MOV1    A,0,CY
  MOV1    CY,PSW1           ;SW1の状態を取得
  MOV1    A,1,CY
  RET
```

4.3.6 スイッチ・コード・ノイズ除去

スイッチ・コード・ノイズ除去処理では、次の動作を行います。

今回取得したスイッチ・コードと前回取得したスイッチ・コードが同じかチェックします。同じであった場合は の処理に移行します。異なる場合は の処理に移行します。

前回取得したスイッチ・コードを今回取得したスイッチ・コードに設定し、 の処理へ移行します。

今回取得したスイッチ・コードと確定済みのスイッチ・コードが同じかチェックします。同じであった場合は の処理に移行します。異なる場合は の処理に移行します。

チャタリング・カウンタをカウントします。また、同時にノイズ除去が完了したか確認します。ノイズ除去が完了している場合は の処理に移行します。ノイズ除去が未完了の場合は の処理に移行します。

確定済みのスイッチ・コードと前回取得したスイッチ・コードが同じかチェックします。同じであった場合は の処理に移行します。異なる場合は の処理に移行します。

確定済みのスイッチ・コードを前回取得したスイッチ・コードに設定します。

チャタリング・カウンタをリセットします。

スイッチ・コード・ノイズ除去処理を終了します。

```

;*****
;*
;*
;*
;*   スイッチコードの確定処理
;*
;*
;*
;*****
;*   [I] : Areg   ... 今回のコード
;*   [IO]: RSWCHTCD ... ラストコード
;*   [IO]: RSWCODE  ... 確定コード
;*****
SSWCAHT:
    CMP     A,RSWCHTCD      ;今回 = ラスト?
    BZ     $JSWCH200       ; Yes,

    MOV     RSWCHTCD,A      ; No, ラストコードを更新
    BR     JSWCH800        ; カウンタをリセット

JSWCH200:
    CMP     A,RSWCODE       ;今回 = 確定コード?
    BZ     $JSWCH800       ; Yes,

    DBNZ   RSWCHTCT,$JSWCH900 ; ノイズ除去完了? No,

    MOV     A,RSWCHTCD
    CMP     A,RSWCODE       ;確定コード = ラストコード?
    BZ     $JSWCH800       ; Yes,

    MOV     A,RSWCHTCD      ;確定コードを更新
    MOV     RSWCODE,A

JSWCH800:
    MOV     RSWCHTCT,#CSWCHTCT ; カウンタをリセット

JSWCH900:
    RET
    
```


4.4 セルフ・プログラミング処理

4.4.1 セクタ領域の定義

セクタ領域の定義では、次の動作を行います。

セクタ1領域を定義します。最初の4バイトすべてを55H、次の4バイトすべてを00Hと定義します。また、残りの1016バイトすべてをFFHと定義します。

セクタ2領域を定義します。1024バイトすべてをFFHと定義します。

```

*****
;
;*
;          セクタ領域の定義
;*
;
;*****
XSEC1      CSEG      AT      CSECT1 ;セクタ 1 領域のデータテーブル
TSECT1:
  DB      55H,55H,55H,55H          ;セクタ 1 領域のフラグ初期値
  DB      00H,00H,00H,00H          ;セクタ 1 領域のカウンタ初期値
  rept    1016
  db      0ffh
  endm

XSEC2      CSEG      AT      CSECT2 ;セクタ 2 領域のデータテーブル
TSECT2:
  rept    1024
  db      0ffh
  endm
    
```

4.4.2 セルフ・プログラミング・メイン処理

セルフ・プログラミング・メイン処理では、次の動作を行います。

キー・イベントが発生しているかチェックします。発生している場合は の処理に移行します。発生していない場合は の処理に移行します。

キー・コードが「特殊キー('Pow+1' ~ 'Pow+4')」かどうか判定します。「特殊キー('Pow+1' ~ 'Pow+4')」である場合は の処理に移行します。それ以外の場合は の処理に移行します。

フラグ・チェック処理を実行し、セクタ1のフラグすべてが55Hかチェックします。すべて55Hであった場合 の処理へ移行します。それ以外の場合は の処理に移行します。

フラグ・チェック処理を実行し、セクタ2のフラグすべてが00Hかチェックします。すべて00Hであった場合 の処理へ移行します。それ以外の場合は の処理に移行します。

フラグ・チェック処理を実行し、セクタ2のフラグすべてがFF0Hかチェックします。すべてFFHであった場合 の処理へ移行します。それ以外の場合は の処理に移行します。

フラグ・チェック処理を実行し、セクタ2のフラグすべてが55Hかチェックします。すべて55Hであった場合 の処理へ移行します。それ以外の場合は の処理に移行します。

フラグ・チェック処理を実行し、セクタ1のフラグすべてが00Hかチェックします。すべて00Hであった場合 の処理へ移行します。それ以外の場合は の処理に移行します。

フラグ・チェック処理を実行し、セクタ1のフラグすべてがFFHかチェックします。すべてFFHであった場合 の処理へ移行します。それ以外の場合は の処理に移行します。

カウンタ比較処理を実行します。書き込み回数が「セクタ1 < セクタ2」であった場合 の処理に移行します。書き込み回数が「セクタ1 > セクタ2」であった場合 の処理に移行します。また、書き込み回数が「セクタ1 = セクタ2」であった場合 の処理に移行します。

デリミタ個数比較処理を実行します。デリミタ個数が「セクタ1 < セクタ2」であった場合 の処理に移行します。デリミタ個数が「セクタ1 = セクタ2」であった場合 の処理に移行します。

書き込み先セクタをセクタ1に、書き込み元セクタをセクタ2に設定します。また、 の処理に移行します。

書き込み先セクタをセクタ2に、書き込み元セクタをセクタ1に設定します。また、 の処理に移行します。

書き込み元セクタから書き込みバッファへプリセット・リモコン・データをコピーします。

コピーするプリセット・リモコン・データのアドレスを更新します。すべてのプリセット・リモコン・データのコピーが完了した場合は の処理に移行します。

カウンタ領域のインクリメント処理を実行します。

機種データ書き換え処理を実行します。

セルフ・プログラミング処理を実行します。

セルフ・プログラミング・メイン処理を終了します。

```

;*****
;
;*                               *
;*   Self Programmingのメイン処理   *
;*                               *
;*****
MSELF:
BT    FKEY_EV,$LSELF050        ;キーイベントあり? ,Yes
BR    LSELF900                  ; No,Return
LSELF050:
CMP   RKEYCODE,#CKC_F1         ;キーコードは特殊キー-1?
BZ    $LSELF100                 ; Yes,
CMP   RKEYCODE,#CKC_F2         ;キーコードは特殊キー-2?
BZ    $LSELF100                 ; Yes,
CMP   RKEYCODE,#CKC_F3         ;キーコードは特殊キー-3?
BZ    $LSELF100                 ; Yes,
CMP   RKEYCODE,#CKC_F4         ;キーコードは特殊キー-4?
BZ    $LSELF100                 ; Yes,
BR    LSELF900                  ; No, return

;***** *****
;*   フラグによる書き込みセクタ判定   *
;***** *****
LSELF100:
;-- セクタ1のフラグが55Hか確認 --
MOVW  HL,#TSECT1
MOV   A,#55H
CALL  !SCHKFLG                  ;セクタ1のフラグが55h:55h:55h:55h ?
BNZ   $LSELF150                 ; No,

; Yes,セクタ2のフラグを確認
MOVW  HL,#TSECT2
MOV   A,#00H
CALL  !SCHKFLG                  ;セクタ2のフラグが00h:00h:00h:00h ?
BZ    $LSELF310                 ; Yes,セクタ2が書き込み先セクタ

MOVW  HL,#TSECT2                 ; No,
MOV   A,#0FFH
CALL  !SCHKFLG                  ;セクタ2のフラグがffh:ffh:ffh:ffh ?
BZ    $LSELF310                 ; Yes,セクタ2が書き込み先セクタ

BR    LSELF200                  ; No,カウント回数で判別

;-- セクタ2のフラグが55Hか確認 --
LSELF150:
MOVW  HL,#TSECT2
MOV   A,#55H
CALL  !SCHKFLG                  ;セクタ2のフラグが55h:55h:55h:55h ?
BNZ   $LSELF200                 ; No,

; Yes,セクタ1のフラグを確認
MOVW  HL,#TSECT1
MOV   A,#00H
CALL  !SCHKFLG                  ;セクタ1のフラグが00h:00h:00h:00h ?
BZ    $LSELF300                 ; Yes,セクタ1が書き込み先セクタ

MOVW  HL,#TSECT1                 ; No,
MOV   A,#0FFH
CALL  !SCHKFLG                  ;セクタ1のフラグがffh:ffh:ffh:ffh ?
BZ    $LSELF300                 ; Yes,セクタ1が書き込み先セクタ

;-- カウント回数で判別 --
LSELF200:
CALL  !SCHKCNT                  ;セクタ1とセクタ2のカウント回数比較
BZ    $LSELF250                 ; セクタ1=セクタ2,デリミタ・チェック

```

```

;-- デリミタの個数で判別 --
LSELF250:
CALL    !SCHDMALL          ;デリミタ個数 セクタ1 セクタ2?
BNC     $LSELF310         ; Yes,セクタ2が書き込み先セクタ
;
BC      $LSELF300         ; No,セクタ1が書き込み先セクタ

LSELF300:
MOVW    RWRADDR,#TSECT1   ;書き込み先セクタを1に設定
MOVW    RRDADDR,#TSECT2   ;書き込み元セクタを2に設定
BR      LSELF400

LSELF310:
MOVW    RWRADDR,#TSECT2   ;書き込み先セクタを2に設定
MOVW    RRDADDR,#TSECT1   ;書き込み元セクタを1に設定

;*****
;*   最新データ読み出し   *
;*****
LSELF400:
MOVW    AX,RRDADDR        ;書き込み元アドレス
MOVW    HL,AX
MOVW    DE,#RBUF          ;書き込み先アドレス
MOV     B,#CBUFSIZ        ;転送バイト数

LSELF420:
MOV     A,[HL]
MOV     [DE],A

INCW    HL
INCW    DE
DBNZ    B,$LSELF420       ;全データ読み込み完了? No,

;*****
;*   バッファデータ書き換え   *
;*****
CALL    !SCNTINC          ;カウンタ部分をインクリメント
CALL    !SBUFW           ;データ部分書き換え

;*****
;*   flash書き込み   *
;*****
CALL    !SELF            ;セルフ・プログラミング処理

;必要に応じてRWRSTATの値による処理を追加してください

LSELF900:

```

4.4.3 フラグ・チェック処理

セルフ・プログラミング・メイン処理では、次の動作を行います。

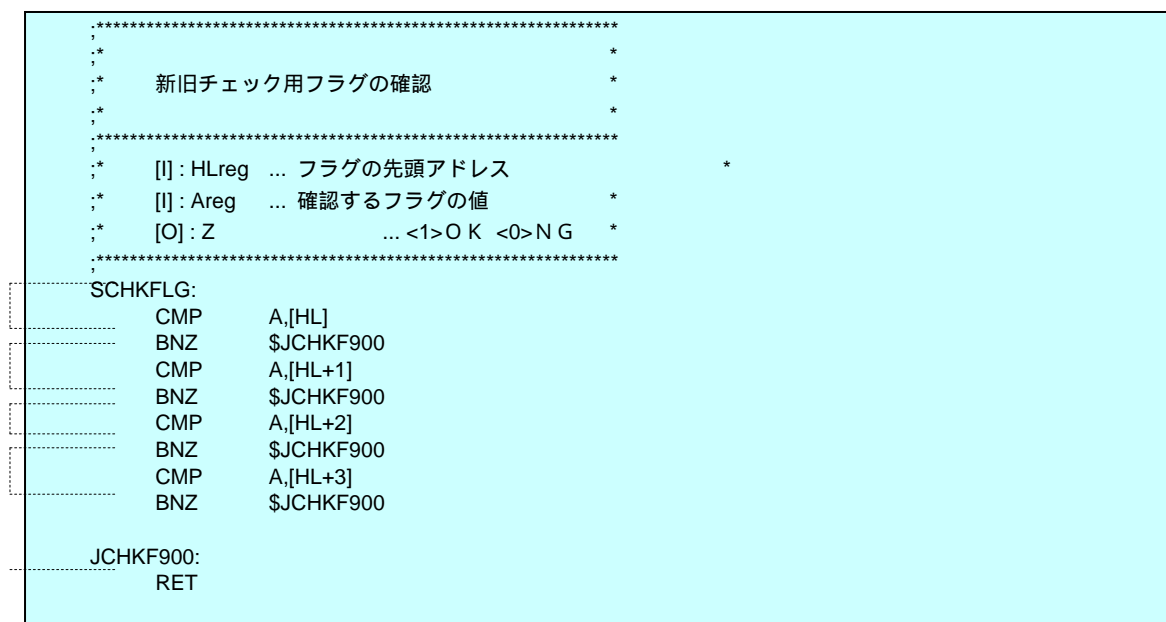
フラグの最上位バイトとチェックする値を比較します。同じだった場合は の処理に移行します。それ以外の場合は の処理に移行します。

フラグの上位2バイト目とチェックする値を比較します。同じだった場合は の処理に移行します。それ以外の場合は の処理に移行します。

フラグの上位3バイト目とチェックする値を比較します。同じだった場合は の処理に移行します。それ以外の場合は の処理に移行します。

フラグの最下位バイトとチェックする値を比較します。

フラグ・チェック処理を終了します。



4.4.4 カウンタ比較処理

カウンタ比較処理では、次の動作を行います。

セクタ1とセクタ2のカウンタについて、最上位バイトから大小を比較します。値が等しかった場合はの処理に移行します。値が異なる場合は の処理に移行します。

比較するカウンタ位置を更新します。カウンタのすべての桁が比較済みの場合は の処理に移行します。それ以外の場合は の処理に移行します。

セクタ1とセクタ2のカウンタは等しいと設定します。

カウンタ比較処理を終了します。

```

*****
;*
;*      書き込み回数チェック
;*
*****
;*      [O]: Z          ... <1>セクタ1 = セクタ2
;*                  <0>セクタ1  セクタ2
;*      [O]: CY        ... <1>セクタ1 < セクタ2
;*                  <0>セクタ1 > セクタ2
;*
;*                  Z=0のときCYが有効
*****
SCHKCNT:
    MOVW    HL,#TSECT2+4      ;セクタ2のカウンタの先頭アドレス
    MOVW    DE,#TSECT1+4     ;セクタ1のカウンタの先頭アドレス

    MOV     B,#4

JCHKC100:
    ;-- 最上位byteから大小比較 --
    MOV     A,[DE]
    CMP     A,[HL]           ;セクタ1 = セクタ2?
    BNZ     $JCHKC900       ; No,

    ;-- 比較位置更新 --
    INCW    HL
    INCW    DE

    DBNZ    B,$JCHKC100     ;比較完了? No,
    SET1    Z               ; Yes,セクタ1 = セクタ2

JCHKC900:

```

4.4.5 デリミタ個数比較処理

デリミタ個数比較処理では、次の動作を行います。

セクタ1のTV用機種データ領域にデリミタが存在するかチェックします。デリミタが存在する場合はカウントします。

セクタ1のVHS用機種データ領域，DVD1用機種データ領域，DVD2用機種データ領域について， と同様の処理を行います。

セクタ2のTV用機種データ領域にデリミタが存在するかチェックします。デリミタが存在する場合はカウントします。

セクタ2のVHS用機種データ領域，DVD1用機種データ領域，DVD2用機種データ領域について， と同様の処理を行います。

セクタ1とセクタ2のデリミタ個数を比較します。

デリミタ個数比較処理を終了します。

```

;*****
;
;*
;*   セクタ1,2のデリミタの個数をチェック
;*
;*
;*****
;*   [O]: CY ... <1>デリミタ個数 セクタ1<セクタ2
;*
;*           <0>デリミタ個数 セクタ1 セクタ2
;*
;*****
SCHDMALL:
MOV     B,#0                ;セクタ1のデリミタ個数カウント用
MOV     C,#0                ;セクタ2のデリミタ個数カウント用

;-- セクタ1のデリミタの個数をチェック --
MOV     A,#CDELIMIT        ;デリミタ値
MOVW    HL,#TSECT1         ;セクタ1の先頭アドレス

;JCDMA100:
CMP     A,[HL+38]           ;TV用機種データのデリミタ・チェックOK?
BZ      $JCDMA120           ; No,
INC     B                   ; Yes,
;JCDMA120:
CMP     A,[HL+70]           ;VHS用機種データのデリミタ・チェックOK?
BZ      $JCDMA140           ; No,
INC     B                   ; Yes,
;JCDMA140:
CMP     A,[HL+102]          ;DVD1用機種データのデリミタ・チェックOK?
BZ      $JCDMA160           ; No,
INC     B                   ; Yes,
;JCDMA160:
CMP     A,[HL+134]          ;DVD2用機種データのデリミタ・チェックOK?
BZ      $JCDMA200           ; No,
INC     B                   ; Yes,

;-- セクタ2のデリミタの個数をチェック --
;JCDMA200:
MOVW    HL,#TSECT2         ;セクタ2の先頭アドレス

CMP     A,[HL+38]           ;TV用機種データのデリミタ・チェックOK?
BZ      $JCDMA220           ; No,
INC     C                   ; Yes,
;JCDMA220:
CMP     A,[HL+70]           ;VHS用機種データのデリミタ・チェックOK?
BZ      $JCDMA240           ; No,
INC     C                   ; Yes,
;JCDMA240:
CMP     A,[HL+102]          ;DVD1用機種データのデリミタ・チェックOK?
BZ      $JCDMA260           ; No,
INC     C                   ; Yes,
;JCDMA260:
CMP     A,[HL+134]          ;DVD2用機種データのデリミタ・チェックOK?
BZ      $JCDMA500           ; No,
INC     C                   ; Yes,

;-- セクタ1とセクタ2のデリミタの個数を比較 --
;JCDMA500:
MOV     A,B
SUB     A,C

```


4.4.6 カウンタ領域のインクリメント処理

領域のインクリメント処理では、次の動作を行います。

カウンタの最下位バイトをインクリメントします。

下位バイトからの桁上りを加算します。

領域のインクリメント処理を終了します。

```

*****
;*
;*   カウンタ更新処理
;*
*****
SCNTINC:
;-- カウンタの最下位byteをインクリメント --
MOVW   HL,#RBUF+4           ;カウンタの先頭アドレス

MOV     A,#1
ADD     A,[HL+3]
MOV     [HL+3],A
MOV     A,#0
ADDC   A,[HL+2]
MOV     [HL+2],A
MOV     A,#0
ADDC   A,[HL+1]
MOV     [HL+1],A
MOV     A,#0
ADDC   A,[HL]
MOV     [HL],A

RET

```

4.4.7 機種データ書き換え処理

機種データ書き換え処理では、次の動作を行います。

書き換えるデータの機種を判定します。機種データがTV,VHS,DVD1,DVD2のいずれかである場合は の処理に移行します。そうでない場合は の処理に移行します。

選択した機種のプリセット・リモコン・データが格納されたアドレスの先頭アドレスをセットします。また、書き込みバッファのデータ書き換え位置をセットします。

書き換えるデータのメーカーを判定します。機種データがTV,VHS,DVD1,DVD2のいずれかである場合は の処理に移行します。そうでない場合は の処理に移行します。

プリセット・リモコン・データの格納されたアドレスを計算します。

プリセット・リモコン・データの格納されたアドレスをセットします。

書き換えデータ数を設定します。

書き込みバッファのデータ書き換えを実行します。

データ書き換え位置を更新します。

データ書き換えが完了したか判定します。完了した場合は の処理に移行します。完了していない場合は の処理に移行します。

機種データのデリミタ位置にデリミタ・データを書き込みます。

機種データ書き換え処理を終了します。

```

;*****
;
;*
;*   書き込みバッファの機種データ更新
;*
;*
;*****
SBUFW:
    CMP    RSWCODE,#CSW_TV          ;機種データはTV?
    BZ     $JBUF100                  ; Yes,
    CMP    RSWCODE,#CSW_VHS        ;機種データはVHS?
    BZ     $JBUF110                  ; Yes,
    CMP    RSWCODE,#CSW_DVD1       ;機種データはDVD1?
    BZ     $JBUF120                  ; Yes,
    CMP    RSWCODE,#CSW_DVD2       ;機種データはDVD2?
    BZ     $JBUF130                  ; Yes,

    BR     JBUF900                    ; No,retutn

JBUF100:
    MOVW   HL,#TRMDAT1X              ;TVのデータ・テーブルをセット
    MOVW   DE,#RBUF+8                ;書き込みバッファのデータ書き換え位置を決定
    BR     JBUF200

JBUF110:
    MOVW   HL,#TRMDAT2X              ;Videoのデータ・テーブルをセット
    MOVW   DE,#RBUF+40               ;書き込みバッファのデータ書き換え位置を決定
    BR     JBUF200

JBUF120:
    MOVW   HL,#TRMDAT3X              ;DVD1のデータ・テーブルをセット
    MOVW   DE,#RBUF+72               ;書き込みバッファのデータ書き換え位置を決定
    BR     JBUF200

JBUF130:
    MOVW   HL,#TRMDAT4X              ;DVD2のデータ・テーブルをセット
    MOVW   DE,#RBUF+104              ;書き込みバッファのデータ書き換え位置を決定
    BR     JBUF200

JBUF200:
    CMP    RKEYCODE,#CKC_F1          ;キーコードは'Pow+1'?
    BZ     $JBUF330                  ; Yes,
    CMP    RKEYCODE,#CKC_F2          ;キーコードは'Pow+2'?
    BZ     $JBUF320                  ; Yes,
    CMP    RKEYCODE,#CKC_F3          ;キーコードは'Pow+3'?
    BZ     $JBUF310                  ; Yes,
    CMP    RKEYCODE,#CKC_F4          ;キーコードは'Pow+4'?
    BZ     $JBUF300                  ; Yes,

    BR     JBUF900                    ; No,retutn

;-- キーコードに応じたアドレスに更新 --
JBUF300:
    INCW   HL                          ;HL+6
    INCW   HL
JBUF310:
    INCW   HL                          ;HL+4
    INCW   HL
JBUF320:
    INCW   HL                          ;HL+2
    INCW   HL
JBUF330:
    INCW   HL                          ;HL+0

;-- 書き換えるデータのアドレスを格納 --
    MOV    A,[HL]
    MOV    X,A
    MOV    A,[HL+1]
    MOVW   HL,AX

```

```
..... MOV      B,#32                ;機種データ長
JBUFW500:
..... ;-- 機種データ書き換え --
..... MOV      A,[HL]
..... MOV      [DE],A
.....
..... INCW     HL                    ;書き換え位置更新
..... INCW     DE
.....
..... DBNZ     B,$JBUFW500          ;書き換え完了? No,
.....
..... ;-- 機種データにデリミタ書き込み --
..... DECW     DE
..... DECW     DE
.....
..... MOV      A,#CDELIMIT
..... MOV      [DE],A
.....
..... JBUFW900:
..... RET
```

4.4.8 フラッシュ書き込み処理

フラッシュ書き込み処理では、次の動作を行います。

低電圧検出フラグ(LVIF)を使用し、VDDが2.05V (±0.05V) 以下かどうか判定します。低電圧状態である場合は、エラー・コードを設定し処理 に移行します。低電圧状態でない場合は処理 に移行します。FLMD0端子をプルアップします。

セルフ・プログラミング・ライブラリのセルフ・プログラミング・スタート関数を呼び出します。

エントリRAMを設定し、セルフ・プログラミング・ライブラリのイニシャライズ関数を呼び出します。

セルフ・プログラミング・ライブラリのモード・チェック関数を呼び出します。FLMD0端子がHIの場合は処理 に移行します。FLMD0端子がLOWの場合は処理 に行します。

書き込み先セクタがどちらかチェックを行い、書き込み先セクタのブロック番号をセットします。

書き込み先セクタのバンク番号をセットします。注

セルフ・プログラミング・ライブラリのブロック・ブランク・チェック関数を呼び出します。

指定ブロックがブランクの場合は処理 に移行します。指定ブロックが既に書き込まれている場合は処理 に移行します。ブロック・ブランク・チェック関数の実行中に割り込みが発生した場合は処理 に移行します。エラーが発生した場合は処理 に移行します。

ブロック・イレース処理を呼び出します。正常消去成功の場合は処理 に移行します。正常消去失敗の場合は処理 に移行します。

ワード・ライト処理を呼び出します。正常書き込み成功の場合は処理 に移行します。正常書き込み失敗の場合は処理 に移行します。

ブロック・ベリファイ処理を呼び出します。正常ベリファイ成功の場合は処理 に移行します。正常ベリファイ失敗の場合は処理 に行します。

ブロック・イレース処理を呼び出します。正常消去成功の場合は処理 に移行します。正常消去失敗の場合は処理 に行します。

ワード・ライト処理を呼び出します。正常書き込み成功の場合は処理 に移行します。正常書き込み失敗の場合は処理 移行します。

ブロック・ベリファイ処理を呼び出します。正常ベリファイ成功の場合は処理 に移行します。正常ベリファイ失敗の場合は処理 移行します。

デリミタ・チェック処理を呼び出し、書き込み先セクタにおけるデータ書き換え箇所にデリミタが書き込まれているかチェックします。デリミタが書き込まれている場合は処理 に移行します。デリミタが書き込まれていない場合は処理 移行します。

書き込み先セクタのフラグを55Hに書き換えます。

書き込み元セクタのフラグを00Hに書き換えます。

セルフ・プログラミング・ライブラリのセルフ・プログラミング・エンド関数を呼び出します。また、FLMD0端子のプルアップを解除します。

フラッシュ書き込み処理を終了します。

注1. uPD179F1xxマイクロ・コンピュータにはROMバンクが存在しないため、0固定になります。

```

;*****
;
;*
; Self Programming処理
;*
;*****
;* [I]: RWRADDR ... 書き込み先セクタの先頭アドレス *
;* [I]: RRDADDR ... 書き込み元セクタの先頭アドレス *
;* [O]: RWRSTAT ... 書き込みステータス *
;*****
SSELF:
;-- LVI検出 --
BF LVIF,$JSELF050 ;VDD < 2.05V? No,
MOV RWRSTAT,#CLVIERR ;Yes,エラーコード設定
BR JSELF900

JSELF050:
;-- FLMD0端子のプルアップ --
MOV FPCTL, #00000001B ;FLMD0端子プルアップ/プルダウン制御
; |||||+----- FLMDPUP: プルアップ
; ++++++----- <0>固定

MOV FPEN, #00000001B ;FLMD0端子プルアップ/プルダウン許可
; |||||+----- FPEN0: 許可
; ++++++----- <0>固定

;*****
;*
;* データ書き込み処理
;*
;*****
;-- セルフ・プログラミング・スタート関数 --
DI
CALL !_FlashStart
EI

;-- イニシャライズ関数 --
MOVW AX,#RENTRAM ;エントリRAMの先頭アドレスを引数とする
CALL !_FlashEnv

;-- モード・チェック関数 --
CALL !_CheckFLMD
MOV RWRSTAT,A ;ステータスの格納
CMP A,#CNMLEND ;FLMD0端子がHiレベル?
BZ $JSELF100 ;Yes,
BR JSELF900 ;No,return

;-- ブロック・ブランク・チェック関数 --
JSELF100:
MOVW AX,RWRADDR
CMPW AX,#TSECT1
BNZ $JSELF110
MOV B,#CSEC1BLK ;チェックするブロックの番号
BR JSELF150
JSELF110:
MOV B,#CSEC2BLK ;チェックするブロックの番号
JSELF150:
MOV A,#0 ;バンク非搭載なので0固定

DI
CALL !_FlashBlockBlankCheck ;書き込み先セクタのブランク・チェック
MOV RWRSTAT,A ;ステータスの格納
EI
CMP A,#CNMLEND ;指定ブロックがブランク?
BZ $JSELF250 ;Yes,
CMP A,#CBLCERR ;指定ブロックがブランクではない?
BZ $JSELF200 ;Yes,
CMP A,#CINTVL ;処理中に割り込み発生?
BZ $JSELF100 ;Yes,再処理

```

```

JSELF200:
CALL    !SBERS                ;ブロック・イレース処理
BNZ     $JSELF300            ; 正常消去完了? No,

JSELF250:
CALL    !SWWRITE             ; Yes,ワード・ライト処理
BNZ     $JSELF300            ; 正常書き込み完了? No,

CALL    !SBVRF               ; Yes,ベリファイ処理
BZ      $JSELF600            ; 正常書き込み完了? Yes,

;-- 再度ブロック・イレース処理からやり直し --

JSELF300:
CALL    !SBERS                ;ブロック・イレース処理
BNZ     $JSELF900            ; 正常消去完了? No,return

CALL    !SWWRITE             ; Yes,ワード・ライト処理
BNZ     $JSELF900            ; 正常書き込み完了? No,return

CALL    !SBVRF               ; Yes,ベリファイ処理
BNZ     $JSELF900            ; 正常書き込み完了? No,return

;*****
;*   デリミタ・チェック処理   *
;*****
JSELF600:
;-- 書き込み先セクタのデリミタ・チェック --
MOVW    AX,RWRADDR           ;書き込み開始アドレスの格納
MOVW    HL,AX

CALL    !SCHKDLM             ;デリミタ・チェック
BZ      $JSELF700            ;デリミタ・チェック正常終了? Yes,
MOV     RWRSTAT,#CDLMERR     ; No,エラーコード格納
BR      JSELF900             ;return

;*****
;*   新規フラグ書換え処理   *
;*****
JSELF700:
MOV     A,#55H               ;フラグを55Hに書き換える
CALL    !SSETFLG             ;フラグ部分書き換え

MOVW    AX,RWRADDR           ;書き込み開始アドレスの格納
MOVW    RWADDR,AX

MOV     RWBANK,#0            ;バンク非搭載なので0 固定

MOVW    AX,#RWADDR           ;書き込み開始アドレス・バンク番号をもつアドレス
MOV     B,#4/4               ;書き込みデータ数(フラグバイト数/ワード)
MOVW    HL,#RBUF              ;書き込み開始位置(フラグの先頭アドレス)の格納

DI
CALL    !FlashWordWrite      ;フラグ書き換え
    
```

```

*****
;
;*   旧データフラグ書換え処理   *
;
*****
JSELF800:
;-- 書き込み元セクタのフラグ書き換え --
MOV     A,#00H                ;フラグを00Hに書き換える
CALL    !SSETFLG              ;書き込みバッファのフラグ部分書き換え

MOVW    AX,RRDADDR            ;書き込み開始アドレスの格納
MOVW    RWADDR,AX

MOV     RWBANK, #0            ;バンク非搭載なので0固定

MOVW    AX,#RWADDR            ;書き込み開始アドレス・バンク番号をもつアドレス
MOV     B,#4/4                ;書き込みデータ数(フラグバイト数/ワード)
MOVW    HL,#RBUF              ;書き込み開始位置(フラグの先頭アドレス)の格納

DI
CALL    !_FlashWordWrite      ;フラグ書き換え
MOV     RWRSTAT,A            ;ステータスの格納
EI

JSELF900:
;-- セルフ・プログラミング・エンド関数 --
DI
CALL    !_FlashEnd
EI

;-- FLMD0端子のプルアップ解除 --
MOV     FPEN, #00000000B      ;FLMD0端子プルアップ/プルダウン許可
;          |||||+----- FPEN0: 禁止
;          ++++++----- <0>固定

MOV     FPCTL, #00000000B     ;FLMD0端子プルアップ/プルダウン制御
;          |||||+----- FLMDPUP: プルダウン
;          ++++++----- <0>固定

```


4.4.9 ブロック・イレース処理

ブロック・イレース処理では、次の動作を行います。

書き込み先セクタがどちらかチェックを行い、書き込み先セクタのブロック番号をセットします。

書き込み先セクタのバンク番号をセットします。^{注1}

セルフ・プログラミング・ライブラリのブロック・イレース関数を呼び出します。

ブロック・イレース関数実行中に割り込みが発生したかチェックします。割り込みが発生した場合はの処理からやり直します。

ブロック・イレース関数が正常終了したかチェックします。

ブロック・イレース処理を終了します。

注1. uPD179F1xxマイクロ・コンピュータにはROMバンクが存在しないため、0固定になります。

```

*****
;
;*
;*   ブロック・イレース処理
;*
;*
*****
;*   [I] : RWRADDR      ... 書き込み先セクタの先頭アドレス      *
;*   [O] : RWRSTAT     ... 書き込みステータス                  *
;*   [O] : Z           ... <1>正常消去完了                      *
;*                   ... <0>消去失敗                          *
*****
SBERS:
;-- ブロック・イレース関数 --
MOVW   AX,RWRADDR
CMPW   AX,#TSECT1
BNZ    $JBERS200
MOV    B,#CSEC1BLK          ;消去するブロック番号の設定
BR     JBERS250
JBERS200:
MOV    B,#CSEC2BLK          ;消去するブロック番号の設定
JBERS250:
MOV    A,#0                 ;消去するブロックのバンク番号

DI
CALL   !_FlashBlockErase    ;書き込み先セクタの消去
MOV    RWRSTAT,A           ;ステータスの格納
EI
CMP    A,#CINTVL           ;処理中に割り込み発生？
BZ     $SBERS              ; Yes,再処理
CMP    A,#CNMLEND          ;正常消去完了？
; BZ     $JBERS900         ; Yes,
; CMP    A,#CPRTEERR       ;指定ブロックがプロテクトされている？
; BZ     $JBERS900         ; Yes,return
; CMP    A,#CERSERR        ;消去中にエラーが発生した？
; BZ     $JBERS900         ; Yes,return
; CMP    A,#CPRMERR        ;バンク番号,ブロック番号設定ミス？
; BZ     $JBERS900         ; Yes,return

;JBERS900:
DEF

```

4.4.10 ワード・ライト処理

ワード・ライト処理では、次の動作を行います。

書き込み先セクタの書き込み開始アドレスを構造体にセットします。

書き込み先セクタのバンク番号を構造体にセットします。^{注1}

書き込み開始アドレス、バンク番号をセットした構造体の先頭アドレスと、書き込みデータ数(ワード単位)をセットします。^{注2}

セルフ・プログラミング・ライブラリのワード・ライト関数を呼び出します。

ワード・ライト関数実行中に割り込みが発生したかチェックします。割り込みが発生した場合は の処理からやり直します。

ワード・ライト関数が正常終了したかチェックします。

ワード・ライト処理を終了します。

注1. uPD179F1xxマイクロ・コンピュータにはROMバンクが存在しないため、0固定になります。

注2. 1ワード=4バイトです。

```

;*****
;
;*
;*   ワード・ライト処理
;*
;*
;*****
;*   [I]: RWRADDR    ... 書き込み先セクタの先頭アドレス      *
;*   [I]: RBUF      ... 書き込みバッファの先頭アドレス        *
;*   [O]: RWRSTAT   ... 書き込みステータス                    *
;*   [O]: Z          ... <1>正常書き込み完了                  *
;*
;*   ... <0>書き込み失敗                                      *
;*****
;
;SWWRITE:
;-- ワード・ライト関数 --
MOVW   AX,RWRADDR      ;書き込み先セクタの先頭アドレス格納
ADDW   AX,#4           ;カウンタの先頭アドレス格納
MOVW   RWADDR,AX
;
MOV    RWBANK, #0      ;バンク非搭載なので0 固定
;
MOVW   AX,#RWADDR      ;書き込み開始アドレス・バンク番号をもつアドレス
MOV    B,#(CBUFsiz-4)/4 ;書き込みデータ数( (全データ数-フラグバイト数)/ワー
;   ド数)
MOVW   HL,#RBUF+4      ;書き込み開始位置(カウンタの先頭アドレス)の格納
DI
CALL   !_FlashWordWrite ;書き込み先セクタにデータ書き込み
MOV    RWRSTAT,A       ;ステータスの格納
EI
CMP    A,#CINTVL       ;処理中に割り込み発生?
BZ     $SWWRITE        ; Yes,再処理
CMP    A,#CNMLEND      ;正常書き込み完了?
;   BZ     $JWWRT400    ; Yes,
;   CMP    A,#CPRTErr   ;指定ブロックがプロテクトされている?
;   BZ     $JWWRT900    ; Yes,return
;   CMP    A,#CWRTErr   ;書き込みエラー?
;   BZ     $JWWRT900    ; Yes,return
;   CMP    A,#CPRMErr   ;パラメータ・エラー?
;   BZ     $JWWRT900    ; Yes,return
;

```

4.4.11 ブロック・ベリファイ処理

ブロック・ベリファイ処理では、次の動作を行います。

書き込み先セクタがどちらかチェックを行い、書き込み先セクタのブロック番号をセットします。

書き込み先セクタのバンク番号をセットします。^{注1}

セルフ・プログラミング・ライブラリのブロック・ベリファイ関数を呼び出します。

ブロック・ベリファイ関数実行中に割り込みが発生したかチェックします。割り込みが発生した場合は
の処理からやり直します。

ブロック・ベリファイ関数が正常終了したかチェックします。

ブロック・ベリファイ処理を終了します。

注1. uPD179F1xxマイクロ・コンピュータにはROMバンクが存在しないため、0固定になります。

```

;*****
;
; *
; *   ブロック・ベリファイ処理   *
; *                               *
; *                               *
;*****
; [I]: RWRADDR    ... 書き込み先セクタの先頭アドレス   *
; [O]: RWRSTAT   ... 書き込みステータス                 *
; [O]: Z         ... <1>ベリファイ完了                 *
;               ... <0>ベリファイ失敗                 *
;*****
SBVRF:
;-- ブロック・ベリファイ関数 --
MOVW    AX,RWRADDR
CMPW    AX,#TSECT1

    BNZ    $JBVRF200
    MOV    B,#CSEC1BLK           ;ベリファイするブロック番号の設定
    BR     JBVRF250

JBVRF200:
    MOV    B,#CSEC2BLK           ;ベリファイするブロック番号の設定
JBVRF250:
    MOV    A,#0                  ;ベリファイするブロックのバンク番号

    DI
    CALL   !_FlashBlockVerify    ;書き込み先セクタのベリファイ
    MOV    RWRSTAT,A            ;ステータスの格納
    EI
    CMP    A,#CINTVL            ;処理中に割り込み発生?
    BZ     $SBVRF                ; Yes,再処理
    CMP    A,#CNMLEND           ;ベリファイ完了?
;    BZ     $JBVRF900            ; Yes,
;    CMP    A,#CIVRFERR          ;内部ベリファイ・エラー?
;    BZ     $JBVRF900            ; Yes,return
;    CMP    A,#CPRMERR           ;バンク番号,ブロック番号設定ミス?
;    BZ     $JBVRF900            ; Yes,return

;JBVRF900:
    RET

```

4.4.12 デリミタ・チェック処理

デリミタ・チェック処理では、次の動作を行います。

デリミタ・チェックするデータの機種を判定します。機種データがTVの場合は の処理に移行します。VHSの場合は の処理に移行します。DVD1の場合は の処理に移行します。DVD2の場合は の処理に移行します。それ以外の場合は の処理に移行します。

TV用機種データのデリミタ・チェックを行います。処理実行後、 の処理に移行します。

VHS用機種データのデリミタ・チェックを行います。処理実行後、 の処理に移行します。

DVD1用機種データのデリミタ・チェックを行います。処理実行後、 の処理に移行します。

DVD2用機種データのデリミタ・チェックを行います。処理実行後、 の処理に移行します。

デリミタ・チェック処理を終了します。

```

*****
;
;*
;*   書き込み箇所のデリミタ・チェック
;*
;*
*****
;*   [I] : HLreg           ... セクタの先頭アドレス           *
;*   [I] : RSWCODE        ... スイッチ・データ               *
;*   [O] : Z               ... <1>デリミタ・チェックOK        *
;*                       ... <0>デリミタ・チェックNG        *
*****
SCHKDLM:
;-- 書き込んだ機種データにデリミタがあるかチェック --
MOV     A,#CDELIMIT           ;デリミタ値

      CMP     RSWCODE,#CSW_TV       ;書き込んだ機種データはTV?
      BZ      $JCHKD100            ; Yes,
      CMP     RSWCODE,#CSW_VHS     ;書き込んだ機種データはVHS?
      BZ      $JCHKD200            ; Yes,
      CMP     RSWCODE,#CSW_DVD1    ;書き込んだ機種データはDVD1?
      BZ      $JCHKD300            ; Yes,
      CMP     RSWCODE,#CSW_DVD2    ;書き込んだ機種データはDVD2?
      BZ      $JCHKD400            ; Yes,

      BR     JCHKD900

JCHKD100:
      CMP     A,[HL+38]             ;TV用機種データのデリミタ・チェック
      BR     JCHKD900

JCHKD200:
      CMP     A,[HL+70]             ;VHS用機種データのデリミタ・チェック
      BR     JCHKD900

JCHKD300:
      CMP     A,[HL+102]            ;DVD1用機種データのデリミタ・チェック
      BR     JCHKD900

JCHKD400:
      CMP     A,[HL+134]            ;DVD2用機種データのデリミタ・チェック
      BR     JCHKD900

JCHKD900:

```

4.4.13 セルフ・プログラミング・エンド処理

セルフ・プログラミング・エンド処理では、次の動作を行います。

セルフ・プログラミング・ライブラリのセルフ・プログラミング・エンド関数を呼び出します。

FLMD0端子のプルアップを解除します。

```

*****
;
;*
;
;*
;*   セルフ・プログラミング・エンド処理   *
;*
;*
;*
;*****

JSELF900:

----- ;-- セルフ・プログラミング・エンド関数 --
DI
CALL    !_FlashEnd
EI

----- ;-- FLMD0端子のプルアップ解除 --
MOV     FPEN,    #00000000B          ;FLMD0端子プルアップ/プルダウン許可
;                                     |||||+----- FPEN0: 禁止
;                                     ++++++----- <0>固定

MOV     FPCTL,   #00000000B          ;FLMD0端子プルアップ/プルダウン制御
;                                     |||||+----- FLMDPUP: プルダウン
;                                     ++++++----- <0>固定

-----
RET

```


4.5 タイマ関連処理

タイマ関連処理では、次の動作を行います。

約10ms経過したかどうかチェックします。約10ms経過している場合は の処理に遷移します。約10ms経過していない場合は の処理に分岐します。

TM50割り込み要求フラグをクリアし、キー・スキャン要求を発生させます。

タイマ関連処理を終了します。

```

*****
;
;*
;
;*
;*      タイマ関連   メイン処理
;*
;*
;*
;*****
MTIMER:
BF      TMIF50,$LTM900      ;10ms経過? No,
;
CLR1    TMIF50              ; Yes,
SET1    FKEY_RQ             ;キー・スキャン要求発行

;-- 必要に応じて10msを元にしたソフトウェア・タイマ・カウント処理を追加してください --

LTM900:
RET

```

4.6 スタンバイ処理

4.6.1 スタンバイ・メイン処理

スタンバイ・メイン処理では、次の動作を行います。

スタンバイ遷移可否判定処理を呼び出し、スタンバイ状態への遷移が可能かチェックします。可能な場合は の処理に移行します。不可能な場合は の処理に移行します。

すべての割り込みを禁止します。

割り込みをマスクします。

割り込み発生フラグをクリアします。

キー・センド・ラインをLOWに落とします。

キー・リターン割り込みのみ有効にします。

STOP命令を実行し、スタンバイ状態にします。

スタンバイ状態からの復帰のためにウェイトします。

キー・リターン割り込みを禁止します。

キー・センド・ラインをHIに戻します。

キー・リターン割り込み発生フラグをクリアします。

キー・スキャン要求を発生させます。

すべての割り込みを許可します。

スタンバイ・メイン処理を終了します。

4.6.2 スタンバイ遷移可否判定処理

スタンバイ遷移可否判定処理では、次の動作を行います。

キー・スキャン状態判定処理を呼び出し、スタンバイ状態への遷移が可能かチェックします。可能な場合は の処理に移行します。不可能な場合は の処理に移行します。注1

スタンバイ遷移可能とし、スタンバイ遷移可否判定処理を終了します。

スタンバイ遷移不可能とし、スタンバイ遷移可否判定処理を終了します。

注1. 他の処理（リモコン・データ送信など）を追加した場合、スタンバイ遷移可否判定を追加する必要があります。

```

*****
;*                                     *
;*                                     *
;*                                     *
;*   スタンバイ遷移条件の判定         *
;*                                     *
;*                                     *
;*                                     *
*****
;*   [O]: Z           ... <1>O K       *
;*                                     *
;*                                     *
;*                                     *
*****
SCHKSTB:
CALL    !SKEYSTB           ;キーはスタンバイ可能？
BNZ     $JCKS900           ;No,

;-- 他の処理を追加した時はスタンバイ遷移条件を追加してください --

JCKS800:
SET1    Z                 ;スタンバイ遷移可
RET

JCKS900:
CLR1    Z                 ;スタンバイ遷移不可
RET
    
```

4.6.3 キー・スキャン状態判定処理

キー・スキャン状態判定処理では、次の動作を行います。

キー・スキャン状態がキー・ノイズ除去中かどうかチェックします。キー・ノイズ除去中の場合は の処理に移行します。キー・ノイズ除去中でない場合は の処理に移行します。

確定済みキー・コードが「キー・オフ」かどうかチェックします。「キー・オフ」でない場合は の処理に移行します。「キー・オフ」の場合は の処理に移行します。

前回取得したキー・コードが「キー・オフ」かどうかチェックします。「キー・オフ」でない場合は の処理に移行します。「キー・オフ」の場合は の処理に移行します。

スタンバイ遷移可能とし、キー・スキャン状態判定処理を終了します。

スタンバイ遷移不可能とし、キー・スキャン状態判定処理を終了します。

```

*****
;
;*
;*   キー・スキャンのスタンバイ遷移可否判定   *
;*
;*
*****
;*   [O]: Z           ... <1> O K           *
;*
;*                   <0> N G           *
;*
*****
SKEYSTB:
-----
CMP     RCHTCNT,#CCHTCNT           ;チャタリングカウント中?
-----
BNZ     $JKSB900                   ; Yes,
-----
CMP     RKEYCODE,#CKC_OFF          ;キーコードはオフ?
-----
BNZ     $JKSB900                   ; No,
-----
CMP     RCHTCODE,#CKC_OFF          ;キーは完全にオフ?
-----
BNZ     $JKSB900                   ; No,
-----
JKSB800:
-----
SET1    Z                           ;スタンバイ遷移可
-----
RET
-----
JKSB900:
-----
CLR1    Z                           ;スタンバイ遷移不可
-----
RET

```

第5章 関連資料

資料名		和文 / 英文
μPD179F11x, 179F12x ユーザーズ・マニュアル		PDF
78K/0シリーズ 命令編 ユーザーズ・マニュアル		PDF
RA78K0 アセンブラ・パッケージ ユーザーズ・マニュアル	言語編	PDF
	操作編	PDF
CC78K0 Cコンパイラ ユーザーズ・マニュアル	言語編	PDF
	操作編	PDF
PM plus ユーザーズ・マニュアル		PDF
セルフ・プログラミング・ライブラリ Type01 ユーザーズ・マニュアル		PDF

付録A プログラム・リスト

プログラム・リスト例として、uPD179F11x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)

```
*****
;*
;*
;* RRRRRRR EEEEEEE NN NN EEEEEEE SSSSSS AA SSSSSS *
;* RR RR EE NNNNN NN EE SS AAAA SS *
;* RRRRRRR EEEEEEE NN NN NN EEEEEEE SSSSSS AA AA SSSSS *
;* RR RR EE NN NNNN EE SS AAAAAAA SS *
;* RR RR EEEEEEE NN NN EEEEEEE SSSSSS AA AA SSSSSS *
;*
;*-----*
.*
.*
;* User : Renesas Electronics Corporation. *
.*
.*
;* System : Preset Remocon Flash Self Programming *
.*
.*
.* CPU : uPD179F1xx *
.*
.*
;* Last update : 2010/12/10 *
.*
.*
;* File name : main.asm *
.*
.*
*****
$include (SelfLib.inc)

;+-----+
;+ +
;+ +
;+ ベクタ・テーブル定義 +
;+ +
;+ +
;+-----+
XVCT1 CSEG AT 0000H
      DW IINIT ;0000H RESET入力,POC,LVI,WDT
      DW 0FFFFH ;0002H 0FFFFH (OCD予約)
      DW IINIT ;0004H INTLVI
      DW IINIT ;0006H INTP0
```

```

DW    IINIT                ;0008H INTP1
DW    IINIT                ;000AH INTP2
DW    IINIT                ;000CH INTP3
DW    IINIT                ;000EH INTP4
DW    IINIT                ;0010H INTP5
DW    IINIT                ;0012H INTSRE6
DW    IINIT                ;0014H INTSR6
DW    IINIT                ;0016H INTST6
DW    IINIT                ;0018H --
DW    IINIT                ;001AH INTTMH1
DW    IINIT                ;001CH INTTMH0
DW    IINIT                ;001EH INTTM50
DW    IINIT                ;0020H INTTM000
DW    IINIT                ;0022H INTTM010
DW    IINIT                ;0024H --
DW    IINIT                ;0026H --
DW    IINIT                ;0028H --
DW    IINIT                ;002AH INTTM51
DW    IINIT                ;002CH INTKR0
DW    IINIT                ;002EH INTKR1
DW    IINIT                ;0030H --
DW    IINIT                ;0032H --
DW    IINIT                ;0034H --
DW    IINIT                ;0036H --
DW    IINIT                ;0038H --
DW    IINIT                ;003AH --
DW    IINIT                ;003CH --
DW    IINIT                ;003EH BRK

```

```

;+-----+

```

```

;+          +
;+          +
;+  CALLTテーブル定義  +
;+          +
;+          +

```

```

;+-----+

```

```

XCALT      CSEG  AT      0040H
DW    0FFFFH      ;0040H 未使用
DW    0FFFFH      ;0042H 未使用
DW    0FFFFH      ;0044H 未使用
DW    0FFFFH      ;0046H 未使用
DW    0FFFFH      ;0048H 未使用
DW    0FFFFH      ;004AH 未使用

```



```

DW 0FFFFH ;004CH 未使用
DW 0FFFFH ;004EH 未使用
DW 0FFFFH ;0050H 未使用
DW 0FFFFH ;0052H 未使用
DW 0FFFFH ;0054H 未使用
DW 0FFFFH ;0056H 未使用
DW 0FFFFH ;0058H 未使用
DW 0FFFFH ;005AH 未使用
DW 0FFFFH ;005CH 未使用
DW 0FFFFH ;005EH 未使用
DW 0FFFFH ;0060H 未使用
DW 0FFFFH ;0062H 未使用
DW 0FFFFH ;0064H 未使用
DW 0FFFFH ;0066H 未使用
DW 0FFFFH ;0068H 未使用
DW 0FFFFH ;006AH 未使用
DW 0FFFFH ;006CH 未使用
DW 0FFFFH ;006EH 未使用
DW 0FFFFH ;0070H 未使用
DW 0FFFFH ;0072H 未使用
DW 0FFFFH ;0074H 未使用
DW 0FFFFH ;0076H 未使用
DW 0FFFFH ;0078H 未使用
DW 0FFFFH ;007AH 未使用
DW 0FFFFH ;007CH 未使用
DW 0FFFFH ;007EH (OCD予約)

```

```

;+-----+

```

```

;+ +

```

```

;+ +

```

```

;+ I/Oポート名定義(uPD179F11x用) +

```

```

;+ +

```

```

;+ +

```

```

;+-----+

```

```

;Description

```

```

;[No] Pin number [IO] I/O direction
; I : Input
; O : Output
; X : Both
; A : Analog(A/DC) In
; - : (don't care)
;
;[Ac] Active level, edge [In] Initial Output Value

```

```

;      H : High Level                H : Output High level
;      L : Low Level                 L : Output Low level
;      X : Both Level                - : (don't care or Input Pin)
;      U : Up Edge
;      D : Down Edge
;      E : Both Edge
;      - : (don't care)
;

```

```

;[ST]  Level at STOP mode

```

```

;      I : Input mode
;      H : Output High level
;      L : Output Low level
;      - : (don't care)
;
;
;

```

30pin SSOP -----+

```

;-----+
;Alias Name      Orig.                ;[No] [IO][Ac][In][ST]

PLED1            EQU    P2.5                ;[ 1] [O] [L] [H] [-] ;LED #1
PLED2            EQU    P2.6                ;[ 2] [O] [L] [H] [-] ;LED #2
PLED3            EQU    P2.7 ;INTP4        ;[ 3] [O] [L] [H] [-] ;LED #3
;--            EQU    P0.4 ;INTP3/OCD1A    ;[ 4] [I] [-] [-] [-] ;OCD1A
PSW0             EQU    P0.5 ;INTP2/OCD1B   ;[ 5] [I] [X] [-] [-] ;Switch #0
PSW1             EQU    P0.6 ;TI50/TO50/INTP1 ;[ 6] [I] [X] [-] [-] ;Switch #1
;--            EQU    P12.0 ;EXLVI/INTP0    ;[ 7] [-] [ ] [ ] [-] ;
;--            EQU    P0.7 ;REM/TOH1        ;[ 8] [-] [ ] [ ] [-] ;REM
;--            EQU    Vdd                    ;[ 9] [-] [ ] [ ] [-] ;Power (3.0V)
;--            EQU    X2 ;P122/EXCLK/OCD0B ;[10] [-] [ ] [ ] [-] ;Main Clock in
;--            EQU    X1 ;P121/OCD0A        ;[11] [-] [ ] [ ] [-] ;Main Clock in
;--            EQU    Vss                    ;[12] [-] [ ] [ ] [-] ;Ground
;--            EQU    REGC                    ;[13] [ ] [ ] [ ] [-] ;REG C
;--            EQU    FLMD0                    ;[14] [ ] [H] [ ] [-] ;FLMD0 control output
;--            EQU    RESET ;P123/KR8/INTP5 ;[15] [I] [U] [-] [-] ;RESET in
;-----+

```

```

;Alias Name      Orig.                ;[No] [IO][Ac][In][ST]

PKR0             EQU    P1.0 ;KR0            ;[16] [I] [L] [-] [-] ;Key Matrix Return #0
PKR1             EQU    P1.1 ;KR1            ;[17] [I] [L] [-] [-] ;Key Matrix Return #1
PKR2             EQU    P1.2 ;KR2            ;[18] [I] [L] [-] [-] ;Key Matrix Return #2
PKR3             EQU    P1.3 ;KR3            ;[19] [I] [L] [-] [-] ;Key Matrix Return #3
PKR4             EQU    P1.4 ;KR4            ;[20] [I] [L] [-] [-] ;Key Matrix Return #4

```

```

PKR5      EQU    P1.5 ;KR5                ;[21] [I] [L] [-] [-] ;Key Matrix Return #5
PKR6      EQU    P1.6 ;KR6                ;[22] [I] [L] [-] [-] ;Key Matrix Return #6
PKR7      EQU    P1.7 ;KR7                ;[23] [I] [L] [-] [-] ;Key Matrix Return #7
PKS0      EQU    P0.0 ;TI51/TO51          ;[24] [O] [L] [H] [L] ;Key Scan #0
PKS1      EQU    P0.1 ;TOH0               ;[25] [O] [L] [H] [L] ;Key Scan #1
PKS2      EQU    P0.2 ;TI000/TxD6 ;[26] [O] [L] [H] [L] ;Key Scan #2
PKS3      EQU    P0.3 ;TI010/TO00/RxD6 ;[27] [O] [L] [H] [L] ;Key Scan #3
PKS4      EQU    P2.0                     ;[28] [O] [L] [H] [L] ;Key Scan #4
PKS5      EQU    P2.1                     ;[29] [O] [L] [H] [L] ;Key Scan #5
PKS6      EQU    P2.2                     ;[30] [O] [L] [H] [L] ;Key Scan #6

```

```

PKEYRET      EQU    P1

```

```

;+-----+
;+
;+
;+   リモコン用データ・テーブル
;+
;+
;+-----+

```

XMDL CSEG UNITP

TMODEL:

```

    DW   TRMDAT1X   ;TV用テーブル・データ先頭アドレス
    DW   TRMDAT2X   ;Video用テーブル・データ先頭アドレス
    DW   TRMDAT3X   ;DVD1用テーブル・データ先頭アドレス
    DW   TRMDAT4X   ;DVD2用テーブル・データ先頭アドレス

```

XRDT CSEG UNITP

TRMDAT1X: ;TV用テーブル・データ先頭アドレス

```

    DW   TRMDAT10   ;メーカ1
    DW   TRMDAT11   ;メーカ2
    DW   TRMDAT12   ;メーカ3
    DW   TRMDAT13   ;メーカ4
;   DW   TRMDAT14   ;予備
;   DW   TRMDAT15   ;予備
;   DW   TRMDAT16   ;予備
;   DW   TRMDAT17   ;予備
;   DW   TRMDAT18   ;予備
;   DW   TRMDAT19   ;予備

```

TRMDAT2X: ;Video用テーブル・データ先頭アドレス

```

    DW   TRMDAT20   ;メーカ1

```

```

    DW    TRMDAT21    ;メーカ2
    DW    TRMDAT22    ;メーカ3
    DW    TRMDAT23    ;メーカ4
;   DW    TRMDAT24    ;予備
;   DW    TRMDAT25    ;予備
;   DW    TRMDAT26    ;予備
;   DW    TRMDAT27    ;予備
;   DW    TRMDAT28    ;予備
;   DW    TRMDAT29    ;予備
TRMDAT3X:                ;DVD1用テーブル・データ先頭アドレス
    DW    TRMDAT30    ;メーカ1
    DW    TRMDAT31    ;メーカ2
    DW    TRMDAT32    ;メーカ3
    DW    TRMDAT33    ;メーカ4
;   DW    TRMDAT34    ;予備
;   DW    TRMDAT35    ;予備
;   DW    TRMDAT36    ;予備
;   DW    TRMDAT37    ;予備
;   DW    TRMDAT38    ;予備
;   DW    TRMDAT39    ;予備
TRMDAT4X:                ;DVD2用テーブル・データ先頭アドレス
    DW    TRMDAT40    ;メーカ1
    DW    TRMDAT41    ;メーカ2
    DW    TRMDAT42    ;メーカ3
    DW    TRMDAT43    ;メーカ4
;   DW    TRMDAT44    ;予備
;   DW    TRMDAT45    ;予備
;   DW    TRMDAT46    ;予備
;   DW    TRMDAT47    ;予備
;   DW    TRMDAT48    ;予備
;   DW    TRMDAT49    ;予備

XPRDAT    CSEG
TRMDAT10: ;DB    (32)    ;dummy data
    DB    097H,040H,04AH,010H,03EH,0A2H,064H,042H
    DB    06BH,018H,046H,00DH,0F6H,0EEH,09EH,0B7H
    DB    022H,02AH,0E8H,01AH,073H,012H,0E5H,047H
    DB    0BFH,0EEH,049H,06FH,088H,00FH,02AH,004H

TRMDAT11: ;DB    (32)    ;dummy data
    DB    000H,001H,002H,003H,004H,005H,006H,007H
    DB    008H,009H,00AH,00BH,00CH,00DH,00EH,00FH
    DB    010H,011H,012H,013H,014H,015H,016H,017H

```

```

DB      018H,019H,01AH,01BH,01CH,01DH,01EH,01FH

TRMDAT12: ;DB      (32)      ;dummy data
DB      020H,021H,022H,023H,024H,025H,026H,027H
DB      028H,029H,02AH,02BH,02CH,02DH,02EH,02FH
DB      030H,031H,032H,033H,034H,035H,036H,037H
DB      038H,039H,03AH,03BH,03CH,03DH,03EH,03FH

TRMDAT13: ;DB      (32)      ;dummy data
DB      040H,041H,042H,043H,044H,045H,046H,047H
DB      048H,049H,04AH,04BH,04CH,04DH,04EH,04FH
DB      050H,051H,052H,053H,054H,055H,056H,057H
DB      058H,059H,05AH,05BH,05CH,05DH,05EH,05FH
;TRMDAT14: ;予備
;   DB      (32)      ;dummy data
;TRMDAT15: ;予備
;   DB      (32)      ;dummy data
;TRMDAT16: ;予備
;   DB      (32)      ;dummy data
;TRMDAT17: ;予備
;   DB      (32)      ;dummy data
;TRMDAT18: ;予備
;   DB      (32)      ;dummy data
;TRMDAT19: ;予備
;   DB      (32)      ;dummy data

TRMDAT20: ;DB      (32)      ;dummy data
DB      060H,061H,062H,063H,064H,065H,066H,067H
DB      068H,069H,06AH,06BH,06CH,06DH,06EH,06FH
DB      070H,071H,072H,073H,074H,075H,076H,077H
DB      078H,079H,07AH,07BH,07CH,07DH,07EH,07FH

TRMDAT21: ;DB      (32)      ;dummy data
DB      080H,081H,082H,083H,084H,085H,086H,087H
DB      088H,089H,08AH,08BH,08CH,08DH,08EH,08FH
DB      090H,091H,092H,093H,094H,095H,096H,097H
DB      098H,099H,09AH,09BH,09CH,09DH,09EH,09FH

TRMDAT22: ;DB      (32)      ;dummy data
DB      0A0H,0A1H,0A2H,0A3H,0A4H,0A5H,0A6H,0A7H
DB      0A8H,0A9H,0AAH,0ABH,0ACH,0ADH,0AEH,0AFH
DB      0B0H,0B1H,0B2H,0B3H,0B4H,0B5H,0B6H,0B7H
DB      0B8H,0B9H,0BAH,0BBH,0BCH,0BDH,0BEH,0BFH

```

```

TRMDAT23: ;DB (32) ;dummy data
          DB 0C0H,0C1H,0C2H,0C3H,0C4H,0C5H,0C6H,0C7H
          DB 0C8H,0C9H,0CAH,0CBH,0CCH,0CDH,0CEH,0CFH
          DB 0D0H,0D1H,0D2H,0D3H,0D4H,0D5H,0D6H,0D7H
          DB 0D8H,0D9H,0DAH,0DBH,0DCH,0DDH,0DEH,0DFH
;TRMDAT24: ;予備
;   DB (32) ;dummy data
;TRMDAT25: ;予備
;   DB (32) ;dummy data
;TRMDAT26: ;予備
;   DB (32) ;dummy data
;TRMDAT27: ;予備
;   DB (32) ;dummy data
;TRMDAT28: ;予備
;   DB (32) ;dummy data
;TRMDAT29: ;予備
;   DB (32) ;dummy data

TRMDAT30: ;DB (32) ;dummy data
          DB 0E0H,0E1H,0E2H,0E3H,0E4H,0E5H,0E6H,0E7H
          DB 0E8H,0E9H,0EAH,0EBH,0ECH,0EDH,0EEH,0EFH
          DB 0F0H,0F1H,0F2H,0F3H,0F4H,0F5H,0F6H,0F7H
          DB 0F8H,0F9H,0FAH,0FBH,0FCH,0FDH,0FEH,0FFH

TRMDAT31: ;DB (32) ;dummy data
          DB 010H,011H,012H,013H,014H,015H,016H,017H
          DB 018H,019H,01AH,01BH,01CH,01DH,01EH,01FH
          DB 020H,021H,022H,023H,024H,025H,026H,027H
          DB 028H,029H,02AH,02BH,02CH,02DH,02EH,02FH

TRMDAT32: ;DB (32) ;dummy data
          DB 030H,031H,032H,033H,034H,035H,036H,037H
          DB 038H,039H,03AH,03BH,03CH,03DH,03EH,03FH
          DB 040H,041H,042H,043H,044H,045H,046H,047H
          DB 048H,049H,04AH,04BH,04CH,04DH,04EH,04FH

TRMDAT33: ;DB (32) ;dummy data
          DB 050H,051H,052H,053H,054H,055H,056H,057H
          DB 058H,059H,05AH,05BH,05CH,05DH,05EH,05FH
          DB 060H,061H,062H,063H,064H,065H,066H,067H
          DB 068H,069H,06AH,06BH,06CH,06DH,06EH,06FH
;TRMDAT34: ;予備

```

```

;   DB   (32)           ;dummy data
;TRMDAT35:           ;予備

;   DB   (32)           ;dummy data
;TRMDAT36:           ;予備

;   DB   (32)           ;dummy data
;TRMDAT37:           ;予備

;   DB   (32)           ;dummy data
;TRMDAT38:           ;予備

;   DB   (32)           ;dummy data
;TRMDAT39:           ;予備

;   DB   (32)           ;dummy data

TRMDAT40: ;DB   (32)   ;dummy data
          DB   070H,071H,072H,073H,074H,075H,076H,077H
          DB   078H,079H,07AH,07BH,07CH,07DH,07EH,07FH
          DB   080H,081H,082H,083H,084H,085H,086H,087H
          DB   088H,089H,08AH,08BH,08CH,08DH,08EH,08FH

TRMDAT41: ;DB   (32)   ;dummy data
          DB   090H,091H,092H,093H,094H,095H,096H,097H
          DB   098H,099H,09AH,09BH,09CH,09DH,09EH,09FH
          DB   0A0H,0A1H,0A2H,0A3H,0A4H,0A5H,0A6H,0A7H
          DB   0A8H,0A9H,0AAH,0ABH,0ACH,0ADH,0AEH,0AFH

TRMDAT42: ;DB   (32)   ;dummy data
          DB   0B0H,0B1H,0B2H,0B3H,0B4H,0B5H,0B6H,0B7H
          DB   0B8H,0B9H,0BAH,0BBH,0BCH,0BDH,0BEH,0BFH
          DB   0C0H,0C1H,0C2H,0C3H,0C4H,0C5H,0C6H,0C7H
          DB   0C8H,0C9H,0CAH,0CBH,0CCH,0CDH,0CEH,0CFH

TRMDAT43: ;DB   (32)   ;dummy data
          DB   0D0H,0D1H,0D2H,0D3H,0D4H,0D5H,0D6H,0D7H
          DB   0D8H,0D9H,0DAH,0DBH,0DCH,0DDH,0DEH,0DFH
          DB   0E0H,0E1H,0E2H,0E3H,0E4H,0E5H,0E6H,0E7H
          DB   0E8H,0E9H,0EAH,0EBH,0ECH,0EDH,0EEH,0EFH
;TRMDAT44:           ;予備

;   DB   (32)           ;dummy data
;TRMDAT45:           ;予備

;   DB   (32)           ;dummy data
;TRMDAT46:           ;予備

;   DB   (32)           ;dummy data
;TRMDAT47:           ;予備

;   DB   (32)           ;dummy data

```

```

;TRMDAT48:                ;予備
;   DB   (32)             ;dummy data
;TRMDAT49:                ;予備
;   DB   (32)             ;dummy data

;+-----+
;+                                     +
;+                                     +
;+   スタック領域の確保               +
;+                                     +
;+                                     +
;+-----+
DSTK DSEG  AT    0FE84H
STACKEND:
   DS    3CH           ;スタック領域を60バイト確保
STACKTOP:              ;スタック領域の先頭アドレス

;+-----+
;+                                     +
;+   セクタ1,2のアドレス定義           +
;+                                     +
;+-----+
CSECT1      EQU   1800H   ;ブロック6先頭アドレス
CSECT2      EQU   1C00H   ;ブロック7先頭アドレス

CSEC1BLK    EQU   6       ;ブロック6
CSEC2BLK    EQU   7       ;ブロック7

;+-----+
;+                                     +
;+   内蔵RAMの先頭アドレス定義         +
;+                                     +
;+-----+
CRAMTOP      EQU   0FD00H ;R A M先頭アドレス(uPD179F111対応)

;+-----+
;+                                     +
;+                                     +
;+   キーコード定義                   +
;+                                     +

```



```

;+
;-----+
;   return
;   +-----+
;   | KR7(P17) <=|   |   |   |   |   |   |
;   +-----+
;   | KR6(P16) <=|   |   |   |   |   |   |
;   +-----+
;   | KR5(P15) <=|   |   |   |   |   |   |
;   +-----+
;   | KR4(P14) <=|  '4'|   |   |   |   |
;   +-----+
;   | KR3(P13) <=|  '3'|   |   |   |   |
;   +-----+
;   | KR2(P12) <=|  '2'|   |   |   |   |
;   +-----+
;   | KR1(P11) <=|  '1'|   |   |   |   |
;   +-----+
;   | KR0(P10) <=| Pow|   |   |   |   |
;   +-----+
;           | ^ | ^ | ^ | ^ | ^ | ^ | ^ |
;           | KS0 | KS1 | KS2 | KS3 | KS4 | KS5 | KS6 |
;           |(P00)|(P01)|(P02)|(P03)|(P20)|(P21)|(P22)|send
;           +-----+
;

```

CKC_OFF	EQU	000H	;キーオフ
CKC_MLT	EQU	0FFH	;多重
CKC_POW	EQU	1	;POW
CKC_1	EQU	10	;'1'
CKC_2	EQU	11	;'2'
CKC_3	EQU	12	;'3'
CKC_4	EQU	13	;'4'
CKC_F1	EQU	80H	;'Pow+1'
CKC_F2	EQU	81H	;'Pow+2'
CKC_F3	EQU	82H	;'Pow+3'
CKC_F4	EQU	83H	;'Pow+4'

```

;-----+
;+
;

```

```

;+
;+   スイッチコード定義
;+
;+
;+
;+-----+
;   +-----+
;   | SW1(P06) | SW0(P05) | CODE |
;   +-----+
;   |   OFF   |   OFF   | 'TV' |
;   +-----+
;   |   OFF   |   ON    | 'Video'|
;   +-----+
;   |   ON    |   OFF   | 'DVD1' |
;   +-----+
;   |   ON    |   ON    | 'DVD2' |
;   +-----+
;
;
;

```

```

CSW_TV      EQU  0      ;'TV'
CSW_VHS     EQU  1      ;'Video'
CSW_DVD1    EQU  2      ;'DVD1'
CSW_DVD2    EQU  3      ;'DVD2'

```

```

;+-----+
;+
;+
;+   定数定義
;+
;+
;+
;+-----+
;+-----+
;+   セルフ・プログラミング関連
;+-----+
;+-----+
CDELMIT     EQU  00H    ;デリミタ値

```

-- 書き込みバッファサイズ定義 --

```

CBUFSIZ     EQU  4+4+(32*4) ;Flag(4) + Counter(4) + Data(32)*4 = 136

```

```

;+-----+
;+   タイマ関連
;+

```

```

;+-----+
CITCL50      EQU    00000110B      ;タイマ50のクロック選択
;
;          |||||
;          |||||+++----- TCL502-500:fPRS/2^8
;          |||||                (fPRS = 4MHz時、1周期64.1us)
;          +++++----- <0>固定

CICR50      EQU    155              ;(155+1)*64.1 = 9.999ms

CITMC50      EQU    00000000B
;
;          |||||
;          |||||+----- TOE50:タイマ50出力禁止
;          |||||+----- TMC501:タイマF/F反転動作禁止
;          |||++----- LVS50/LVR50:タイマ出力F/Fは変化しない
;          ||++----- <0>固定
;          |+----- TMC506:TM50とCR50の一致でクリア&スタート
;          +----- TCE50:タイマ50動作停止

```

```

;+-----+
;+
;+
;+   マクロ定義
;+
;+
;+
;+-----+
;+-----+
;+
;+   全割り込み要因のマスク
;+
;+
;+-----+

```

```

_INTMSK      macro
            mov    MK0L,#0ffh
            mov    MK0H,#0ffh
            mov    MK1L,#0ffh
            endm

```

```

;+-----+
;+
;+   全割り込み要求フラグのクリア
;+
;+
;+-----+
_CLRIF      macro

```

```

mov    IF0L,#00h
mov    IF0H,#00h
mov    IF1L,#00h
endm

```

```

;+-----+
;+                                     +
;+                                     +
;+ 変数定義                             +
;+                                     +
;+                                     +
;+-----+

```

```

;+-----+
;+ キー・スキャン関連                 +
;+-----+

```

```

DKYEBUF      DSEG  SADDR
RKEYBUF:     DS    010H      ;Powキーリターン格納ワークエリア
FTMP_POW     EQU   RKEYBUF.0  ;Powキーリターン格納位置
FTMP_KY1     EQU   (RKEYBUF+1).1 ;'1'キーリターン格納位置
FTMP_KY2     EQU   (RKEYBUF+1).2 ;'2'キーリターン格納位置
FTMP_KY3     EQU   (RKEYBUF+1).3 ;'3'キーリターン格納位置
FTMP_KY4     EQU   (RKEYBUF+1).4 ;'4'キーリターン格納位置

```

-- キー関連フラグエリア --

```

DKEYS        DSEG  SADDR
RKEYST:      DS    1          ;
FKEY_RQ      EQU   RKEYST.7   ;キー・スキャン要求
FKEY_EV      EQU   RKEYST.6   ;キーイベント通知フラグ
;            EQU   RKEYST.5
;            EQU   RKEYST.4
;            EQU   RKEYST.3
;            EQU   RKEYST.2
;            EQU   RKEYST.1
;            EQU   RKEYST.0

```

-- キーコード関連 --

```

RKEYCODE:    DS    1          ;確定コード
RCHTCODE:    DS    1          ;ラストコード

RCHTCNT:     DS    1          ;チャタリングカウンタ

```

CCHTCNT EQU 3 ;3回

;- スイッチコード関連 -;

DSW DSEG SADDR
 RSWCODE: DS 1 ;確定コード
 RSWCHTCD: DS 1 ;ラストコード

 RSWCHTCT: DS 1 ;チャタリングカウンタ
 CSWCHTCT EQU 10 ;10回

;+-----+

;+ セルフ・プログラミング関連 +

;+-----+

DENT DSEG AT CRAMTOP
 RENTRAM: DS 100 ;ライブラリ用エントリRAM領域
 RBUF: DS CBUFSIZ ;書き込みバッファ

DSELF DSEG SADDRP
 RWADDR: DS 2 ;書き込み開始アドレス
 RWBANK: DS 1 ;書き込みブロックのバンク番号

DSTAT DSEG SADDR
 RWRSTAT: DS 1 ;ステータス

;- セルフ処理のエラー・ステータス (ライブラリのリターン・コード) -;

CNMLEND EQU 00H ;正常終了
 CFLMDERR EQU 01H ;FLMDエラー
 CPRMERR EQU 05H ;パラメータ・エラー
 CPRTERR EQU 10H ;プロテクト・エラー
 CERSERR EQU 1AH ;消去エラー
 CBLCERR EQU 1BH ;ブランク・チェック・エラー
 CIVRFERR EQU 1BH ;内部ペリファイ・エラー
 CWRRTERR EQU 1CH ;書き込みエラー
 CVRFERR EQU 1DH ;ペリファイ(MRG12)エラー
 CBLKERR EQU 1EH ;ブランク・エラー
 CINTVL EQU 1FH ;割り込み発生
 CREDERR EQU 20H ;リード・エラー

;- 独自のエラー・ステータス -;

CLVIERR EQU 80H ;LVI検出
 CDLMERR EQU 81H ;デリミタ・エラー

```

DSLFADDR  DSEG  SADDRP
RWRADDR:  DS    2          ;書き込み先セクタの先頭アドレス
RRDADDR:  DS    2          ;書き込み元セクタの先頭アドレス
    
```

```

;*****
;
;*
;
;*      リセット解除後の初期化処理
;*
;*
;*
;*****
;
    
```

```
XINIT CSEG AT 00190H
```

```
IINIT:
```

```
DI ;割り込み禁止
```

```

;+-----+
;+ レジスタバンク設定 +
;+-----+
    
```

```
SEL RB0
```

```

;+-----+
;+ ROM/RAMサイズの設定 +
;+-----+
    
```

```
; uPD179F110はROM容量不足のため使用できません
```

```

; MOV IMS,#42H ; uPD179F111使用時の設定
; MOV IMS,#04H ; uPD179F112,uPD179F122使用時の設定
; MOV IMS,#0C6H ; uPD179F113,uPD179F123使用時の設定
MOV IMS,#0C8H ; uPD179F114,uPD179F124使用時の設定
    
```

```

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

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

```

;+-----+
;+ クロック周波数の設定 +
;+-----+
    
```

```
MOV OSCCTL, #0100000B ;クロック動作モード
```

```

;          ||+++++----- <0>固定
;          ++----- EXCLK/OSCSEL: X1発振モード

MOV   MOC, #0000000B    ;高速システム・クロックの動作制御
;          |+++++----- <0>固定
;          +----- MSTOP: X1発振回路動作

MOV   MCM, #0000000B    ;供給クロック選択
;          |||||+----- XSEL/MCM0:
;          ||||| |      メイン・システム・クロック(fXP)=fRH
;          ||||| |      周辺ハードウェア・クロック(fPRS)=fRH
;          ||||| +----- MCS: Read Only
;          +++++----- 0固定

MOV   PCC, #0000000B    ;CPUクロック(fCPU)の選択
;          |||||+++----- CSS/PCC2/PCC1/PCC0: CPUクロック(fCPU)=fXP
;          +++++----- <0>固定

MOV   RCM, #0000001B    ;内蔵発振器の動作モード設定
;          |||||+----- RSTOP: 高速内蔵発振器の停止
;          |||||+----- LSRSTOP: 低速内蔵発振器の停止
;          |+++++----- <0>固定
;          +----- RSTS: Read Only

```

; 発信安定時間は使用する発振子により異なります。
; 発振子に最適な発信安定時間を選択して下さい。

```

MOV   OST2, #0000001B    ;発信安定時間の選択
;          |||||+++----- OST2-0: 510us
;          +++++----- <0>固定

```

```

*****
;
.*          *
;
.*          *
;
.*   ポートの初期化処理          *
;
.*          *
;
.*          *
;
*****

```

SPORTINI:

```

;+-----+
.*   ポート出力の初期化処理          *
;+-----+

```

SPOINI:

```

MOV   P0,   #00000000B
MOV   P1,   #00000000B
MOV   P2,   #11100000B
; MOV   P3,   #00000000B   ;uPD179F12x使用時有効にして下さい
MOV   P12,  #00001000B

;+-----+
;+   内蔵プルアップの設定           +
;+-----+

MOV   PU0,  #00000000B
;          +----- PU00-07: 内蔵プルアップ抵抗を接続しない

MOV   PU1,  #11111111B
;          +----- PU10-17: 内蔵プルアップ抵抗を接続する

MOV   PU2,  #00011000B   ;uPD179F11x使用時有効にして下さい
; MOV   PU2,  #00000000B   ;uPD179F12x使用時有効にして下さい

; MOV   PU3,  #00111111B   ;uPD179F12x使用時有効にして下さい
;          ||*****----- PU10-17: 内蔵プルアップ抵抗を接続する
;          ++----- <0>固定

MOV   PU12, #00001000B
;          |||||+----- PU120: 内蔵プルアップ抵抗を接続しない
;          ||||+----- <0>固定
;          |||+----- PU123: 内蔵プルアップ抵抗を接続する
;          ++++----- <0>固定

;+-----+
;+   ポート出力モードの設定         +
;+-----+

MOV   POM0, #00001111B
;          |||++++----- POM00-03: N-ch O.D.
;          ++++----- POM04-07: CMOS出力

MOV   POM1, #00000000B
;          +----- POM10-17: CMOS出力

MOV   POM2, #00011111B
;          |||*++++----- POM20-22: N-ch O.D.
;          +++----- POM25-27: CMOS出力

; MOV   POM3, #00111111B   ;uPD179F11x使用時有効にして下さい

```



```

;          ||+++++----- POM30-35: CMOS出力
;          ++----- <0>固定
;  MOV    POM3, #0000000B      ;uPD179F12x使用時有効にして下さい
;          ||+++++----- POM30-35: CMOS出力
;          ++----- <0>固定

MOV    POM12, #0000000B
;          ||||++++----- POM120-122: CMOS出力
;          +++++----- <0>固定

;+-----+
;+   ポート入出力モードの設定           +
;+-----+
MOV    PM0,  #11100000B
;          *|||++++----- KS3-6
;          ||+----- 未使用
;          ++----- SW0-1

MOV    PM1,  #11111111B
;          ++++++----- KR0-7

MOV    PM2,  #00011000B
;          |||*++++----- KS0-2
;          +++----- LED1-3

;  MOV    PM3,  #11111111B      ;uPD179F12x使用時有効にして下さい
;          ||*****
;          ++----- <1>固定

MOV    PM12, #11111110B
;          |||||+----- 未使用
;          ||||+----- X1/X2
;          +++++----- <1>固定

MOV    RSTMASK,#00000000B      ;RESET端子有効、兼用機能無効

;  MOV    RSTMASK,#00001000B      ;KR8使用時はこちらを有効にして下さい
;          ||||+----- <0>固定
;          |||+----- RSTM: 0...兼用機能無効
;          +++++----- <0>固定

;+-----+
;+   キー割り込み機能の設定           +
;+-----+

```

```

;+-----+
MOV   KRML, #0000000B
;
;          +----- KRM0-7: キー割り込み信号を検出しない

MOV   KRMH, #0000000B
;
;          |----- KRM8-14: キー割り込み信号を検出しない
;          +----- <0>固定

CLR1  KRIF0
;
CLR1  KRIF1          ;(拡張キー使用時はこの処理を有効にしてください)

SET1  KRMK0
;
SET1  KRMK1          ;(拡張キー使用時はこの処理を有効にしてください)

;+-----+
;+  低電圧検出機能の設定          +
;+-----+
SET1  LVIMK

MOV   LVIS, #00001110B   ;低電圧検出レベルの選択
;
;          |||+----- LVIS3-0: 2.05 V±0.05 V
;          +----- <0>固定

MOV   LVIM, #10000000B   ;低電圧検出, 動作モードを設定
;
;          |||||+----- LVIF: Read Only
;          |||||+----- LVIMD: 内部割り込み発生
;          ||||+----- LVISEL: VDDのレベルを検出
;          |----- <0>固定
;          +----- LVION: 低電圧検出動作許可

;+-----+
;+  割り込み要求のマスク          +
;+-----+
_INTMSK          ;全割り込み要因のマスク
_CLRIF          ;全割り込み要求フラグのクリア

;+-----+
;+  その他初期化処理の呼び出し    +
;+-----+
CALL  !STIMINI    ;タイマ初期化処理
CALL  !SKEYINI    ;キースキャン初期化処理
CALL  !SSWINI     ;スイッチスキャン初期化処理

```

```

SET1  TCE50          ;タイマ50動作開始
CLR1  TMIF50

EI          ;割り込み許可

BR     MMMAIN
    
```

```

;+-----+
;+                                     +
;+                                     +
;+ キー・スキャン、スイッチ・スキャン関連 +
;+                                     +
;+                                     +
;+-----+
    
```

XSW CSEG

```

*****
;
.*                                     *
;                                     *
;*   スイッチスキャン関連の初期化処理   *
.*                                     *
;                                     *
*****
    
```

SSWINI:

```

MOV   RSWCHTCT,#CSWCHTCT    ;チャタリングカウンタをリセット
MOV   RSWCODE,#CSW_TV      ;確定コード  【'TV'】
MOV   RSWCHTCD,#CSW_TV     ;ラストコード  【'TV'】

RET
    
```

```

*****
;
.*                                     *
;                                     *
.*                                     *
;                                     *
;*   スイッチ・コードを取得             *
.*                                     *
;                                     *
.*                                     *
;                                     *
*****
;*   [O]: Areg      ... スイッチ・コード   *
;                                     *
*****
    
```

SSWGET:

```

MOV   A,#0

;-- スイッチ・コード生成 --
MOV1  CY,PSW0              ;SW0の状態を取得
MOV1  A.0,CY
    
```

```

MOV1  CY,PSW1                ;SW1の状態を取得
MOV1  A.1,CY

RET

```

```

;*****
;
;*                               *
;                               *
;*   スイッチコードの確定処理   *
;                               *
;*                               *
;*****
;*   [] : Areg ... 今回のコード   *
;*   [IO]: RSWCHTCD... ラストコード *
;*   [IO]: RSWCODE ... 確定コード *
;*****
SSWCAHT:
    CMP  A,RSWCHTCD          ;今回 = ラスト?
    BZ   $JSWCH200           ; Yes,

    MOV  RSWCHTCD,A          ; No,ラストコードを更新
    BR   JSWCH800            ;カウンタをリセット

JSWCH200:
    CMP  A,RSWCODE           ;今回 = 確定コード?
    BZ   $JSWCH800           ; Yes,

    DBNZ RSWCHTCT,$JSWCH900 ;ノイズ除去完了? No,

    MOV  A,RSWCHTCD
    CMP  A,RSWCODE           ;確定コード = ラストコード?
    BZ   $JSWCH800           ; Yes,

    MOV  A,RSWCHTCD          ;確定コードを更新
    MOV  RSWCODE,A

JSWCH800:
    MOV  RSWCHTCT,#CSWCHTCT  ;カウンタをリセット

JSWCH900:
    RET

```

XKEY CSEG

```

*****
;
.*
;
.*      キースキャン関連の初期化処理      *
;
.*
;
*****
;

```

SKEYINI:

```

MOV   RCHTCNT,#CCHTCNT  ;チャタリングカウンタをリセット
MOV   RKEYCODE,#CKC_OFF ;確定コード  【キーオフ】
MOV   RCHTCODE,#CKC_OFF ;ラストコード  【キーオフ】

MOV   RKEYST,#00000000B ;FKEY_RQ,FKEY_EV クリア
                        ;キースキャン要求解除
                        ;キーイベント通知解除

```

RET

```

*****
;
.*
;
.*      キースキャンのスタンバイ遷移可否判定      *
;
.*
;
*****
;
.*      [O] : Z          ... <1> O K          *
.*
.*      <0> N G          *
;
*****
;

```

SKEYSTB:

```

CMP   RCHTCNT,#CCHTCNT  ;チャタリングカウント中？
BNZ   $JKSB900          ; Yes,
CMP   RKEYCODE,#CKC_OFF ;キーコードはオフ？
BNZ   $JKSB900          ; No,
CMP   RCHTCODE,#CKC_OFF ;キーは完全にオフ？
BNZ   $JKSB900          ; No,

```

JKSB800:

```

SET1  Z                ;スタンバイ遷移可
RET

```

JKSB900:

```

CLR1  Z                ;スタンバイ遷移不可

RET

```

```

*****
;

```

```

.*
;
.*
;
.*   キーリターンの値を取得
.*
.*
;
.*
;
*****
;
.*   [O] : Areg      ... キーリターンの値
.*
*****
;
SKYGET:
    MOV    B,#21                ;動作クロック4MHz時15usウェイト（電圧安定時間）

JKSWA100:
    NOP
    DBNZ   B,$JKSWA100         ;電圧安定までのウェイト

    MOV    A,PKEYRET           ;キーリターンポートから、データ取得
    XOR    A,#11111111B       ;（負論理）

    RET

*****
;
.*
;
.*
;
.*   キーセンス
.*
.*
;
.*
;
*****
;
.*   [O] : RKEYBUF[n]    ... キーマップ
.*
*****
;
SKYSENCE:

    CLR1   PKS0                ;キーセンド # 0
    CALL   !SKYGET
    MOV    RKEYBUF,A
    SET1   PKS0

    CLR1   PKS1                ;キーセンド # 1
    CALL   !SKYGET
    MOV    RKEYBUF+1,A
    SET1   PKS1

    CLR1   PKS2                ;キーセンド # 2
    CALL   !SKYGET

```

```

MOV   RKEYBUF+2,A
SET1  PKS2

CLR1  PKS3           ;キーセンド # 3
CALL  !SKYGET
MOV   RKEYBUF+3,A
SET1  PKS3

CLR1  PKS4           ;キーセンド # 4
CALL  !SKYGET
MOV   RKEYBUF+4,A
SET1  PKS4

CLR1  PKS5           ;キーセンド # 5
CALL  !SKYGET
MOV   RKEYBUF+5,A
SET1  PKS5

CLR1  PKS6           ;キーセンド # 6
CALL  !SKYGET
MOV   RKEYBUF+6,A
SET1  PKS6

;-- uPD179F12x使用時は有効にしてください --
$_IF (0)
CLR1  PKS7           ;キーセンド # 7
CALL  !SKYGET
MOV   RKEYBUF+7,A
SET1  PKS7
$ENDIF
RET

```

```

*****
;
.*                                     *
;                                     *
.*                                     *
;   キーコード作成                     *
.*                                     *
;                                     *
.*                                     *
;                                     *
*****
;
.*   [I] : RKEYBUF[n]... キーマップ     *
.*   [O] : Areg      ... キーコード     *
.*   [W] : Breg      ... 検査中のキーコード *

```



```

INC    B

MOV    A,[HL]
BF     A.2,$JKMC260      ;ビット 2 = 1 ? No,
INC    D
MOV    A,B
MOV    E,A
JKMC260:
INC    B

MOV    A,[HL]
BF     A.3,$JKMC280      ;ビット 3 = 1 ? No,
INC    D
MOV    A,B
MOV    E,A
JKMC280:
INC    B

MOV    A,[HL]
BF     A.4,$JKMC300      ;ビット 4 = 1 ? No,
INC    D
MOV    A,B
MOV    E,A
JKMC300:
INC    B

MOV    A,[HL]
BF     A.5,$JKMC320      ;ビット 5 = 1 ? No,
INC    D
MOV    A,B
MOV    E,A
JKMC320:
INC    B

MOV    A,[HL]
BF     A.6,$JKMC340      ;ビット 6 = 1 ? No,
INC    D
MOV    A,B
MOV    E,A
JKMC340:
INC    B

MOV    A,[HL]

```

```

BF      A,7,$JKMC360      ;ビット7 = 1 ? No,
INC     D
MOV     A,B
MOV     E,A
JKMC360:
INC     B

INCW    HL                ;次のワークエリアをさす
DBNZ   C,$JKMC200        ;全エリア終了? No,

;*****
;
;*      アクティブなキー数の検査 & 今回のコード決定      *
;*****
;
;JKMC700:
MOV     A,D
CMP     A,#1              ;キー入力は一つのみ?
BNZ    $JKMC720          ; No,
MOV     A,E                ; Yes,作成したキーコードを返す
BR     JKMC900

JKMC720:
CMP     A,#0              ;キーの入力無し?
BNZ    $JKMC760          ; No,
MOV     A,#CKC_OFF        ; Yes,【オフコード】を返す
BR     JKMC900

JKMC760:
CMP     A,#2              ;キー入力は2個?
BNZ    $JKMC890          ; No,

;*****
;
;*      アクティブなキー数が2個の場合      *
;*****
;
;-- 有効な多重入力をチェックする --
BF     FTMP_POW,$JKMC890 ;POWER KEY ON ? No,

;-- 以降別の有効な多重キーの対応するビットをチェックする --
BF     FTMP_KY1,$JKMC820
MOV    A,#CKC_F1          ;'Pow+1'キーコードを返す
BR     JKMC900

JKMC820:
BF     FTMP_KY2,$JKMC830
MOV    A,#CKC_F2          ;'Pow+2'キーコードを返す
BR     JKMC900

```

```

JKMC830:
    BF    FTMP_KY3,$JKMC840
    MOV   A,#CKC_F3          ;'Pow+3'キーコードを返す
    BR    JKMC900

JKMC840:
    BF    FTMP_KY4,$JKMC890
    MOV   A,#CKC_F4          ;'Pow+4'キーコードを返す
    BR    JKMC900

    ;-- 不正な多重入力 --

JKMC890:
    MOV   A,#CKC_MLT        ;【多重】キーコードを返す

JKMC900:
    RET

*****
;
;*
;
;*      キーコードの確定処理
;*
;*
;*
*****
;
;*  [I] : Areg ... 今回のコード
;*  [IO]: RCHTCODE ... ラストコード
;*  [IO]: RKEYCODE ... 確定コード
;*  [O] : FKEY_EV ... キーイベント通知フラグ
*****
;

SRMCAHT:

    CMP   A,RCHTCODE        ;今回 = ラスト?
    BZ    $JRC200           ; Yes,

    MOV   RCHTCODE,A        ;ラストコードを更新
    BR    JRC800            ;カウンタをリセット

JRC200:
    DBNZ  RCHTCNT,$JRC900   ;ノイズ除去完了? No,

    MOV   A,RCHTCODE
    CMP   A,RKEYCODE        ;確定コード = ラストコード?
    BZ    $JRC800           ; Yes,

```

```

.*****
;
;*   「多重」の場合の判定           *
.*****
;
    CMP   RKEYCODE,#CKC_MLT ;確定コードは「多重」？
    BNZ   $JRC400           ; No,
    CMP   RCHTCODE,#CKC_OFF ;「多重」から「オフ」への変化？
    BNZ   $JRC800           ; No,

    MOV   RKEYCODE,#CKC_OFF ; Yes,
    BR    JRC800

.*****
;
;*   その他の場合の判定           *
.*****
;
JRC400:
    CMP   RCHTCODE,#CKC_OFF ;ラストコードが「オフ」？
    BZ    $JRC420           ; Yes,
    CMP   RCHTCODE,#CKC_MLT ;ラストコードが「多重」？
    BNZ   $JRC600           ; No,

JRC420:
    MOV   A,RCHTCODE        ;確定コードを更新（オフ | 多重）
    MOV   RKEYCODE,A
    BR    JRC800

JRC600:
    MOV   A,RCHTCODE        ;確定コードを更新
    MOV   RKEYCODE,A
    SET1  FKEY_EV           ;キーオンイベントを発行

JRC800:
    MOV   RCHTCNT,#CCHTCNT ;カウンタをリセット

JRC900:
    RET

.*****
;
;*                               *
;                               *
;*                               *
;*   キースキャン・スイッチスキャンメイン処理           *
;*                               *
;*                               *
;*                               *
;

```

```

*****
;
MKYSCAN:
    CLR1  FKEY_EV                ;メイン 1 周でイベントをクリア

    BTCLR FKEY_RQ,$LKS200        ;キースキャン要求有り? Yes,
    BR    LKS900                ; No,

LKS200:
    CALL !SKYSENCE              ;キーセンス
    CALL !SMKCODE               ;コード作成
    CALL !SRMCAHT               ;ノイズ除去

    CALL !SSWGET                ;スイッチ状態取得
    CALL !SSWCAHT               ;ノイズ除去

```

```

LKS900:
    RET

```

```

;+-----+
;+                +
;+                +
;+   セルフ・プログラミング関連           +
;+                +
;+                +
;+-----+

```

```

*****
;
.*
;                *
.*   セクタ領域の定義                       *
;
.*                *
;
*****

```

```

XSEC1      CSEG  AT      CSECT1      ;セクタ 1 領域のデータテーブル
TSECT1:
    DB      55H,55H,55H,55H          ;セクタ 1 領域のフラグ初期値
    DB      00H,00H,00H,00H          ;セクタ 1 領域のカウンタ初期値
    rept    1016
    db      Offh
    endm

```

```

XSEC2      CSEG  AT      CSECT2      ;セクタ 2 領域のデータテーブル
TSECT2:
    rept    1024
    db      Offh
    endm

```

```

XSELF      CSEG
;*****
;
;*          *
;          *
;* 新旧チェック用フラグの確認          *
;          *
;*          *
;*****
;* [I] : HLreg      ... フラグの先頭アドレス          *
;* [I] : Areg ... 確認するフラグの値          *
;* [O] : Z          ... <1>O K <0>N G          *
;*****
SCHKFLG:
    CMP    A,[HL]
    BNZ    $JCHKF900
    CMP    A,[HL+1]
    BNZ    $JCHKF900
    CMP    A,[HL+2]
    BNZ    $JCHKF900
    CMP    A,[HL+3]
    BNZ    $JCHKF900

JCHKF900:
    RET

;*****
;*          *
;* 書き込み回数チェック          *
;*          *
;*****
;* [O] : Z          ... <1>セクタ1 = セクタ2          *
;*          *          *          *
;*          *          *          *          *
;* [O] : CY ... <1>セクタ1 < セクタ2          *
;*          *          *          *          *
;*          *          *          *          *
;*          *          *          *          *
;*          *          *          *          *
;*          *          *          *          *
;*          *          *          *          *
;*          *          *          *          *
;*          *          *          *          *
;*          *          *          *          *
;*          *          *          *          *
;*          *          *          *          *
;*****
SCHKCNT:
    MOVW   HL,#TSECT2+4      ;セクタ2のカウンタの先頭アドレス
    MOVW   DE,#TSECT1+4     ;セクタ1のカウンタの先頭アドレス

    MOV    B,#4

JCHKC100:
    ;-- 最上位byteから大小比較 --

```

```

MOV   A,[DE]
CMP   A,[HL]           ;セクタ1 = セクタ2?
BNZ   $JCHKC900        ; No,

;-- 比較位置更新 --
INCW  HL
INCW  DE

DBNZ  B,$JCHKC100     ;比較完了? No,
SET1  Z                ; Yes,セクタ1 = セクタ2

```

JCHKC900:

```
RET
```

```

*****
;
.*                                     *
;                                     *
;*   書き込み箇所のデリミタ・チェック   *
;                                     *
.*                                     *
;                                     *
*****
;*   [I] : HLreg      ... セクタの先頭アドレス   *
;*   [I] : RSWCODE   ... スイッチ・データ       *
;*   [O] : Z         ... <1>デリミタ・チェックOK   *
;*                   <0>デリミタ・チェックNG   *
*****
;

```

SCHKDLM:

```

;-- 書き込んだ機種データにデリミタがあるかチェック --
MOV   A,#CDELMIT      ;デリミタ値

CMP   RSWCODE,#CSW_TV      ;書き込んだ機種データはTV?
BZ    $JCHKD100           ; Yes,
CMP   RSWCODE,#CSW_VHS    ;書き込んだ機種データはVHS?
BZ    $JCHKD200           ; Yes,
CMP   RSWCODE,#CSW_DVD1   ;書き込んだ機種データはDVD1?
BZ    $JCHKD300           ; Yes,
CMP   RSWCODE,#CSW_DVD2   ;書き込んだ機種データはDVD2?
BZ    $JCHKD400           ; Yes,

BR    JCHKD900

```

JCHKD100:

```

CMP   A,[HL+38]          ;TV用機種データのデリミタ・チェック
BR    JCHKD900

```

JCHKD200:

CMP A,[HL+70] ;VHS用機種データのデリミタ・チェック
BR JCHKD900

JCHKD300:

CMP A,[HL+102] ;DVD1用機種データのデリミタ・チェック
BR JCHKD900

JCHKD400:

CMP A,[HL+134] ;DVD2用機種データのデリミタ・チェック
BR JCHKD900

JCHKD900:

RET

```

*****
;
;*
;
;*   セクタ1,2のデリミタの個数をチェック
;*
;*
;
*****
;
;*   [O]: CY ... <1>デリミタ個数 セクタ1<セクタ2
;*
;*           <0>デリミタ個数 セクタ1 セクタ2
;
*****
;

```

SCHDMALL:

MOV B,#0 ;セクタ1のデリミタ個数カウント用
MOV C,#0 ;セクタ2のデリミタ個数カウント用

-- セクタ1のデリミタの個数をチェック --

MOV A,#CDELIMIT ;デリミタ値
MOVW HL,#TSECT1 ;セクタ1の先頭アドレス

;JCDMA100:

CMP A,[HL+38] ;TV用機種データのデリミタ・チェックOK?
BZ \$JCDMA120 ; No,
INC B ; Yes,

JCDMA120:

CMP A,[HL+70] ;VHS用機種データのデリミタ・チェックOK?
BZ \$JCDMA140 ; No,
INC B ; Yes,

JCDMA140:

CMP A,[HL+102] ;DVD1用機種データのデリミタ・チェックOK?
BZ \$JCDMA160 ; No,
INC B ; Yes,

JCDMA160:

```

CMP   A,[HL+134]           ;DVD2用機種データのデリミタ・チェックOK?
BZ    $JCDMA200           ; No,
INC   B                   ; Yes,
    
```

;- セクタ2のデリミタの個数をチェック --

JCDMA200:

```

MOVW  HL,#TSECT2         ;セクタ2の先頭アドレス

CMP   A,[HL+38]         ;TV用機種データのデリミタ・チェックOK?
BZ    $JCDMA220         ; No,
INC   C                 ; Yes,
    
```

JCDMA220:

```

CMP   A,[HL+70]         ;VHS用機種データのデリミタ・チェックOK?
BZ    $JCDMA240         ; No,
INC   C                 ; Yes,
    
```

JCDMA240:

```

CMP   A,[HL+102]        ;DVD1用機種データのデリミタ・チェックOK?
BZ    $JCDMA260         ; No,
INC   C                 ; Yes,
    
```

JCDMA260:

```

CMP   A,[HL+134]        ;DVD2用機種データのデリミタ・チェックOK?
BZ    $JCDMA500         ; No,
INC   C                 ; Yes,
    
```

;- セクタ1とセクタ2のデリミタの個数を比較 --

JCDMA500:

```

MOV   A,B
SUB   A,C

RET
    
```

```

*****
;
;*                               *
;                               *
;*   フラグ書換え処理           *
;                               *
;*                               *
*****
;
;*   [] : Areg ... 書き換えるフラグの値(55Hor00H)   *
*****
;
    
```

SSETFLG:

```

;- 書き込みバッファのフラグ部分を書き換える --
MOV   !RBUF,A
    
```

```
MOV !RBUF+1,A
MOV !RBUF+2,A
MOV !RBUF+3,A
```

```
RET
```

```
*****
;
;*
; *
;* カウンタ更新処理 *
;*
;* *
*****
;
```

SCNTINC:

;- カウンタの最下位byteをインクリメント -;

```
MOVW HL,#RBUF+4 ;カウンタの先頭アドレス
```

```
MOV A,#1
ADD A,[HL+3]
MOV [HL+3],A
MOV A,#0
ADDC A,[HL+2]
MOV [HL+2],A
MOV A,#0
ADDC A,[HL+1]
MOV [HL+1],A
MOV A,#0
ADDC A,[HL]
MOV [HL],A
```

```
RET
```

```
*****
;
;*
; *
;* 書き込みバッファの機種データ更新 *
;*
;* *
*****
;
```

SBUFV:

```
CMP RSWCODE,#CSW_TV ;機種データはTV?
BZ $JBUFV100 ; Yes,
CMP RSWCODE,#CSW_VHS ;機種データはVHS?
BZ $JBUFV110 ; Yes,
CMP RSWCODE,#CSW_DVD1 ;機種データはDVD1?
BZ $JBUFV120 ; Yes,
CMP RSWCODE,#CSW_DVD2 ;機種データはDVD2?
```

```

BZ    $JBUFW130          ; Yes,

BR    JBUFW900          ; No,retutn

JBUFW100:
MOVW  HL,#TRMDAT1X      ;TVのデータ・テーブルをセット
MOVW  DE,#RBUF+8        ;書き込みバッファのデータ書き換え位置を決定
BR    JBUFW200

JBUFW110:
MOVW  HL,#TRMDAT2X      ;Videoのデータ・テーブルをセット
MOVW  DE,#RBUF+40       ;書き込みバッファのデータ書き換え位置を決定
BR    JBUFW200

JBUFW120:
MOVW  HL,#TRMDAT3X      ;DVD1のデータ・テーブルをセット
MOVW  DE,#RBUF+72       ;書き込みバッファのデータ書き換え位置を決定
BR    JBUFW200

JBUFW130:
MOVW  HL,#TRMDAT4X      ;DVD2のデータ・テーブルをセット
MOVW  DE,#RBUF+104      ;書き込みバッファのデータ書き換え位置を決定
; BR    JBUFW200

JBUFW200:
CMP   RKEYCODE,#CKC_F1  ;キーコードは'Pow+1'?
BZ    $JBUFW330          ; Yes,
CMP   RKEYCODE,#CKC_F2  ;キーコードは'Pow+2'?
BZ    $JBUFW320          ; Yes,
CMP   RKEYCODE,#CKC_F3  ;キーコードは'Pow+3'?
BZ    $JBUFW310          ; Yes,
CMP   RKEYCODE,#CKC_F4  ;キーコードは'Pow+4'?
BZ    $JBUFW300          ; Yes,

BR    JBUFW900          ; No,retutn

;-- キーコードに応じたアドレスに更新 --
JBUFW300:
;HL+6
INCW  HL
INCW  HL
JBUFW310:
;HL+4
INCW  HL
INCW  HL
JBUFW320:
;HL+2
INCW  HL
INCW  HL

```

```

JBUF330:                                ;HL+0

    ;-- 書き換えるデータのアドレスを格納 --
    MOV  A,[HL]
    MOV  X,A
    MOV  A,[HL+1]
    MOVW HL,AX

    MOV  B,#32                            ;機種データ長
JBUF500:
    ;-- 機種データ書き換え --
    MOV  A,[HL]
    MOV  [DE],A

    INCW HL                                ;書き換え位置更新
    INCW DE

    DBNZ B,$JBUF500                       ;書き換え完了? No,

    ;-- 機種データにデリミタ書き込み --
    DECW DE
    DECW DE

    MOV  A,#CDELIMIT
    MOV  [DE],A

JBUF900:
    RET

;*****
;
;*                                     *
;* Self Programming処理                 *
;*                                     *
;*****
;* [I] : RWRADDR ... 書き込み先セクタの先頭アドレス *
;* [I] : RRDADDR ... 書き込み元セクタの先頭アドレス *
;* [O] : RWRSTAT ... 書き込みステータス           *
;*****
SSELF:
    ;-- LVI検出 --
    BF   LVIF,$JSELF050                   ;VDD < 2.05V ? No,

```

```

MOV   RWRSTAT,#CLVIERR   ; Yes,エラーコード設定
BR    JSELF900

JSELF050:
;-- FLMD0端子のプルアップ --
MOV   FPCTL, #00000001B   ;FLMD0端子プルアップ/プルダウン制御
;
;          |||||+----- FLMDPUP: プルアップ
;          ++++++----- <0>固定

MOV   FPEN, #00000001B   ;FLMD0端子プルアップ/プルダウン許可
;
;          |||||+----- FPEN0: 許可
;          ++++++----- <0>固定

;*****
;
;*   データ書き込み処理   *
;*****
;
;-- セルフ・プログラミング・スタート関数 --
DI
CALL  !_FlashStart
EI

;-- イニシャライズ関数 --
MOVW  AX,#RENTRAM        ;エントリRAMの先頭アドレスを引数とする
CALL  !_FlashEnv

;-- モード・チェック関数 --
CALL  !_CheckFLMD
MOV   RWRSTAT,A         ;ステータスの格納
CMP   A,#CNMLEND        ;FLMD0端子がHiレベル?
BZ    $JSELF100         ; Yes,
BR    JSELF900          ; No,return

;-- ブロック・ブランク・チェック関数 --
JSELF100:
MOVW  AX,RWRADDR
CMPW  AX,#TSECT1
BNZ   $JSELF110
MOV   B,#CSEC1BLK       ;チェックするブロックの番号
BR    JSELF150
JSELF110:
MOV   B,#CSEC2BLK       ;チェックするブロックの番号

JSELF150:

```

```

MOV   A,#0                ;バンク非搭載なので0固定

DI
CALL  !_FlashBlockBlankCheck ;書き込み先セクタのブランク・チェック
MOV   RWRSTAT,A          ;ステータスの格納
EI
CMP   A,#CNMLEND         ;指定ブロックがブランク？
BZ    $JSELF250           ; Yes,
CMP   A,#CBLCERR         ;指定ブロックがブランクではない？
BZ    $JSELF200           ; Yes,
CMP   A,#CINTVL          ;処理中に割り込み発生？
BZ    $JSELF100          ; Yes,再処理
;    CMP   A,#CPRMERR     ;バンク番号,ブロック番号設定ミス？
;    BZ    $JSELF900      ; Yes,return

BR    JSELF900           ; Default,return

JSELF200:
CALL  !SBERS             ;ブロック・イレース処理
BNZ   $JSELF300         ; 正常消去完了？ No,

JSELF250:
CALL  !SWWRITE          ; Yes,ワード・ライト処理
BNZ   $JSELF300         ; 正常書き込み完了？ No,

CALL  !SBVRF            ; Yes,ベリファイ処理
BZ    $JSELF600         ; 正常書き込み完了？ Yes,

;-- 再度ブロック・イレース処理からやり直し --

JSELF300:
CALL  !SBERS             ;ブロック・イレース処理
BNZ   $JSELF900         ; 正常消去完了？ No,return

CALL  !SWWRITE          ; Yes,ワード・ライト処理
BNZ   $JSELF900         ; 正常書き込み完了？ No,return

CALL  !SBVRF            ; Yes,ベリファイ処理
BNZ   $JSELF900         ; 正常書き込み完了？ No,return

;*****
;
;*   デリミタ・チェック処理           *
;*****
;*****
JSELF600:
;-- 書き込み先セクタのデリミタ・チェック --

```

```

MOVW AX,RWRADDR      ;書き込み開始アドレスの格納
MOVW HL,AX

CALL !SCHKDLM        ;デリミタ・チェック
BZ    $JSELF700      ;デリミタ・チェック正常終了? Yes,
MOV   RWRSTAT,#CDLMERR ;No,エラーコード格納
BR    JSELF900       ;return

```

;* 新規フラグ書換え処理 *

JSELF700:

```

MOV   A,#55H          ;フラグを55Hに書き換える
CALL  !SSETFLG        ;フラグ部分書き換え

MOVW  AX,RWRADDR     ;書き込み開始アドレスの格納
MOVW  RWADDR,AX

MOV   RWBANK, #0      ;バンク非搭載なので0固定

MOVW  AX,#RWADDR     ;書き込み開始アドレス・バンク番号をもつアドレス
MOV   B,#4/4         ;書き込みデータ数(フラグバイト数/ワード)
MOVW  HL,#RBUF        ;書き込み開始位置(フラグの先頭アドレス)の格納

DI
CALL  !_FlashWordWrite ;フラグ書き換え
MOV   RWRSTAT,A      ;ステータスの格納
EI

```

;* 旧データフラグ書換え処理 *

JSELF800:

```

;-- 書き込み元セクタのフラグ書き換え --
MOV   A,#00H          ;フラグを00Hに書き換える
CALL  !SSETFLG        ;書き込みバッファのフラグ部分書き換え

MOVW  AX,RRDADDR     ;書き込み開始アドレスの格納
MOVW  RWADDR,AX

MOV   RWBANK, #0      ;バンク非搭載なので0固定

MOVW  AX,#RWADDR     ;書き込み開始アドレス・バンク番号をもつアドレス

```

```

MOV    B,#4/4                ;書き込みデータ数(フラグバイト数/ワード)
MOVW   HL,#RBUF              ;書き込み開始位置(フラグの先頭アドレス)の格納

DI

CALL   !_FlashWordWrite      ;フラグ書き換え
MOV    RWRSTAT,A            ;ステータスの格納

EI

```

JSELF900:

;- セルフ・プログラミング・エンド関数 -;

```

DI
CALL   !_FlashEnd
EI

```

;- FLMD0端子のプルアップ解除 -;

```

MOV    FPEN, #00000000B      ;FLMD0端子プルアップ/プルダウン許可
;
;          |||||+----- FPEN0: 禁止
;          ++++++----- <0>固定

```

```

MOV    FPCTL, #00000000B     ;FLMD0端子プルアップ/プルダウン制御
;
;          |||||+----- FLMDPUP: プルダウン
;          ++++++----- <0>固定

```

RET

```

*****
;
;*
;
;*   ブロック・イレース処理
;*
;*
;
*****
;
;*   [I]: RWRADDR ... 書き込み先セクタの先頭アドレス      *
;*   [O]: RWRSTAT ... 書き込みステータス                  *
;*   [O]: Z      ... <1>正常消去完了                      *
;*
;*               ... <0>消去失敗                          *
;
*****

```

SBERS:

;- ブロック・イレース関数 -;

```

MOVW   AX,RWRADDR
CMPW   AX,#TSECT1
BNZ    $JBERS200
MOV    B,#CSEC1BLK          ;消去するブロック番号の設定

```



```

BR    JBERS250
JBERS200:
MOV   B,#CSEC2BLK      ;消去するブロック番号の設定
JBERS250:
MOV   A,#0             ;消去するブロックのバンク番号

DI
CALL  !_FlashBlockErase ;書き込み先セクタの消去
MOV   RWRSTAT,A       ;ステータスの格納
EI
CMP   A,#CINTVL       ;処理中に割り込み発生?
BZ    $SBERS           ; Yes,再処理
CMP   A,#CNMLEND      ;正常消去完了?
; BZ    $JBERS900      ; Yes,
; CMP   A,#CPRTErr     ;指定ブロックがプロテクトされている?
; BZ    $JBERS900      ; Yes,return
; CMP   A,#CERSERR     ;消去中にエラーが発生した?
; BZ    $JBERS900      ; Yes,return
; CMP   A,#CPRMERR     ;バンク番号,ブロック番号設定ミス?
; BZ    $JBERS900      ; Yes,return

;JBERS900:
RET

;*****
;*                                     *
;*   ワード・ライト処理                                     *
;*                                     *
;*****
;* [I] : RWRADDR ... 書き込み先セクタの先頭アドレス      *
;* [I] : RBUF    ... 書き込みバッファの先頭アドレス      *
;* [O] : RWRSTAT ... 書き込みステータス                  *
;* [O] : Z       ... <1>正常書き込み完了                  *
;*               ... <0>書き込み失敗                    *
;*****

SWWRITE:
;-- ワード・ライト関数 --
MOVW  AX,RWRADDR      ;書き込み先セクタの先頭アドレス格納
ADDW  AX,#4           ;カウンタの先頭アドレス格納
MOVW  RWADDR,AX

MOV   RWBANK, #0      ;バンク非搭載なので0固定

```

```

MOVW AX,#RWADDR      ;書き込み開始アドレス・バンク番号をもつアドレス
MOV  B,#(CBUFsiz-4)/4 ;書き込みデータ数( (全データ数-フラグバイト数)/ワード数)
MOVW HL,#RBUF+4      ;書き込み開始位置(カウンタの先頭アドレス)の格納
DI
CALL !_FlashWordWrite ;書き込み先セクタにデータ書き込み
MOV  RWRSTAT,A       ;ステータスの格納
EI
CMP  A,#CINTVL       ;処理中に割り込み発生?
BZ   $SWWRITE        ; Yes,再処理
CMP  A,#CNMLEND      ;正常書き込み完了?
; BZ   $JWWRT400      ; Yes,
; CMP  A,#CPRTErr    ;指定ブロックがプロテクトされている?
; BZ   $JWWRT900     ; Yes,return
; CMP  A,#CWRTErr    ;書き込みエラー?
; BZ   $JWWRT900     ; Yes,return
; CMP  A,#CPRMERR    ;パラメータ・エラー?
; BZ   $JWWRT900     ; Yes,return

```

;JWWRT900:

RET

```

*****
;
;*                               *
;                               *
;*   ブロック・ベリファイ処理   *
;                               *
;*                               *
;                               *
*****
;*   [I] : RWRADDR  ... 書き込み先セクタの先頭アドレス   *
;*   [O] : RWRSTAT  ... 書き込みステータス               *
;*   [O] : Z        ... <1>ベリファイ完了                 *
;*                               ... <0>ベリファイ失敗     *
;                               *
*****

```

SBVRF:

;- ブロック・ベリファイ関数 -;

MOVW AX,RWRADDR

CMPW AX,#TSECT1

BNZ \$JBVRF200

MOV B,#CSEC1BLK ;ベリファイするブロック番号の設定

BR JBVRF250

JBVRF200:

MOV B,#CSEC2BLK ;ベリファイするブロック番号の設定

JBVRF250:

```

MOV    A,#0                ;ベリファイするブロックのバンク番号

DI
CALL   !_FlashBlockVerify ;書き込み先セクタのベリファイ
MOV    RWRSTAT,A          ;ステータスの格納
EI
CMP    A,#CINTVL          ;処理中に割り込み発生?
BZ     $SBVRF              ; Yes,再処理
CMP    A,#CNMLEND         ;ベリファイ完了?
; BZ     $JBVRF900         ; Yes,
; CMP    A,#CIVRFERR       ;内部ベリファイ・エラー?
; BZ     $JBVRF900         ; Yes,return
; CMP    A,#CPRMERR        ;バンク番号,ブロック番号設定ミス?
; BZ     $JBVRF900         ; Yes,return

;JBVRF900:
RET

```

```

;*****
;
;*                               *
;*   Self Programmingのメイン処理   *
;*                               *
;*                               *
;*****
MSELF:
BT     FKEY_EV,$LSELF050    ;キーイベントあり? ,Yes
BR     LSELF900             ; No,Return
LSELF050:
CMP    RKEYCODE,#CKC_F1    ;キーコードは特殊キー1?
BZ     $LSELF100            ; Yes,
CMP    RKEYCODE,#CKC_F2    ;キーコードは特殊キー2?
BZ     $LSELF100            ; Yes,
CMP    RKEYCODE,#CKC_F3    ;キーコードは特殊キー3?
BZ     $LSELF100            ; Yes,
CMP    RKEYCODE,#CKC_F4    ;キーコードは特殊キー4?
BZ     $LSELF100            ; Yes,
BR     LSELF900             ; No, return

```

```

;*****
;*   フラグによる書き込みセクタ判定   *
;*****
LSELF100:
;-- セクタ1のフラグが55Hか確認 --

```

```

MOVW HL,#TSECT1
MOV   A,#55H
CALL  !SCHKFLG           ;セクタ1のフラグが55h:55h:55h:55h ?
BNZ   $LSELF150         ; No,

```

; Yes,セクタ2のフラグを確認

```

MOVW HL,#TSECT2
MOV   A,#00H
CALL  !SCHKFLG           ;セクタ2のフラグが00h:00h:00h:00h ?
BZ    $LSELF310         ; Yes,セクタ2が書き込み先セクタ

```

```

MOVW HL,#TSECT2         ; No,
MOV   A,#0FFH
CALL  !SCHKFLG           ;セクタ2のフラグがffh:ffh:ffh:ffh ?
BZ    $LSELF310         ; Yes,セクタ2が書き込み先セクタ

```

```

BR    LSELF200          ; No,カウント回数で判別

```

-- セクタ2のフラグが55Hか確認 --

LSELF150:

```

MOVW HL,#TSECT2
MOV   A,#55H
CALL  !SCHKFLG           ;セクタ2のフラグが55h:55h:55h:55h ?
BNZ   $LSELF200         ; No,

```

; Yes,セクタ1のフラグを確認

```

MOVW HL,#TSECT1
MOV   A,#00H
CALL  !SCHKFLG           ;セクタ1のフラグが00h:00h:00h:00h ?
BZ    $LSELF300         ; Yes,セクタ1が書き込み先セクタ

```

```

MOVW HL,#TSECT1         ; No,
MOV   A,#0FFH
CALL  !SCHKFLG           ;セクタ1のフラグがffh:ffh:ffh:ffh ?
BZ    $LSELF300         ; Yes,セクタ1が書き込み先セクタ

```

-- カウント回数で判別 --

LSELF200:

```

CALL  !SCHKCNT           ;セクタ1とセクタ2のカウント回数比較
BZ    $LSELF250         ; セクタ1 = セクタ2,デリミタ・チェック
BC    $LSELF300         ; セクタ1 < セクタ2,書き込み先セクタを1に設定
BR    LSELF310         ; セクタ1 > セクタ2,書き込み先セクタを2に設定

```

-- デリミタの個数で判別 --

```

LSELF250:
    CALL  !SCHDMALL          ;デリミタ個数 セクタ1 セクタ2 ?
    BNC   $LSELF310         ; Yes,セクタ2が書き込み先セクタ
;    BC    $LSELF300        ; No,セクタ1が書き込み先セクタ

LSELF300:
    MOVW  RWRADDR,#TSECT1   ;書き込み先セクタを1に設定
    MOVW  RRDADDR,#TSECT2   ;書き込み元セクタを2に設定
    BR    LSELF400

LSELF310:
    MOVW  RWRADDR,#TSECT2   ;書き込み先セクタを2に設定
    MOVW  RRDADDR,#TSECT1   ;書き込み元セクタを1に設定

;*****
;
;*   最新データ読み出し   *
;*****
;

LSELF400:
    MOVW  AX,RRDADDR        ;書き込み元アドレス
    MOVW  HL,AX
    MOVW  DE,#RBUF         ;書き込み先アドレス
    MOV   B,#CBUFSIZ       ;転送バイト数

LSELF420:
    MOV   A,[HL]
    MOV   [DE],A

    INCW  HL
    INCW  DE
    DBNZ  B,$LSELF420      ;全データ読み込み完了? No,

;*****
;
;*   パッファデータ書き換え   *
;*****
;

    CALL  !SCNTINC         ;カウンタ部分をインクリメント
    CALL  !SBUFW          ;データ部分書き換え

;*****
;
;*   flash書き込み   *
;*****
;

    CALL  !SSELF          ;セルフ・プログラミング処理

;必要に応じてRWRSTATの値による処理を追加してください

```

LSELF900:

RET

XTIM CSEG

```
;+-----+
```

```
;+                                     +
```

```
;+                                     +
```

```
;+   タイマ関連処理                       +
```

```
;+                                     +
```

```
;+                                     +
```

```
;+-----+
```

```
*****
```

```
.*                                     *
```

```
.*                                     *
```

```
.*   タイマ関連の初期化処理                 *
```

```
.*                                     *
```

```
.*                                     *
```

```
*****
```

STIMINI:

```
MOV   TCL50,#CITCL50           ;タイマ50のクロック設定
```

```
MOV   CR50,#CICR50
```

```
MOV   TMC50,#CITMC50           ;タイマ50の動作設定
```

RET

```
*****
```

```
.*                                     *
```

```
.*                                     *
```

```
.*   タイマ関連   メイン処理                 *
```

```
.*                                     *
```

```
.*                                     *
```

```
*****
```

MTIMER:

```
BF    TMIF50,$LTM900           ;10ms経過? No,
```

```
CLR1  TMIF50                   ; Yes,
```

```
SET1  FKEY_RQ                  ;キースキャン要求発行
```

;-- 必要に応じて10msを元にしたソフトウェア・タイマ・カウント処理を追加してください --

LTM900:

RET

XSTB CSEG

```

;+-----+
;+                               +
;+                               +
;+   スタンバイ処理             +
;+                               +
;+                               +
;+-----+

```

```

*****
;
.*                               *
;
.*                               *
;
.*   スタンバイ遷移条件の判定     *
;
.*                               *
;
.*                               *
;
*****
;
.*   [O] : Z           ... <1>O K   *
;
.*                               *
.*           <0>N G               *
;
*****
;

```

SCHKSTB:

```

CALL  !SKEYSTB           ;キーはスタンバイ可能?
BNZ   $JCKS900          ;No,

```

;-- 他の処理を追加した時はスタンバイ遷移条件を追加してください --

JCKS800:

```

SET1  Z                 ;スタンバイ遷移可
RET

```

JCKS900:

```

CLR1  Z                 ;スタンバイ遷移不可
RET

```

```

*****
;
.*                               *
;
.*                               *
;
.*                               *
;
.*   スタンバイ制御   メイン処理     *
;
.*                               *
;
.*                               *
;
.*                               *
;
*****
;

```

MSTB:

```
CALL  !SCHKSTB          ;スタンバイに入る条件を満たす？
BNZ   $LSTB900         ; No,

DI
_INTMSK
_CLRIF

CLR1  PKS0             ;キースキャンラインをLOWに落としておく
CLR1  PKS1
CLR1  PKS2
CLR1  PKS3
CLR1  PKS4
CLR1  PKS5
CLR1  PKS6
; CLR1  PKS7          ;uPD179F12x使用時有効にして下さい

CLR1  KRMK0           ;キーリターン検出のみでスタンバイ解除
; CLR1  KRMK1         ;KR8使用時、uPD179F12x使用時は有効にして下さい

MOV   KRML, #11111111B
; MOV  KRMH, #00000001B ;KR8使用時でuPD179F11x使用時は有効にして下さい
; MOV  KRMH, #11111110B ;KR8未使用時でuPD179F12x使用時は有効にして下さい
; MOV  KRMH, #11111111B ;KR8使用時、uPD179F12x使用時は有効にして下さい

STOP
NOP
NOP
NOP
NOP

; SET1  KRMK0
; SET1  KRMK1         ;KR8使用時、uPD179F12x使用時は有効にして下さい

MOV   KRML, #00000000B
; MOV  KRMH, #00000000B ;KR8使用時は有効にして下さい
; MOV  KRMH, #00000001B ;KR8未使用時でuPD179F12x使用時は有効にして下さい

SET1  PKS0           ;キースキャンラインをHIに戻す
SET1  PKS1
SET1  PKS2
SET1  PKS3
SET1  PKS4
SET1  PKS5
SET1  PKS6
```



```

;   SET1   PKS7           ;uPD179F12x使用時有効にしてください
      CLR1   KRIF0
;   CLR1   KRIF1         ;KR8使用時、uPD179F12x使用時は有効にしてください

      SET1   FKEY_RQ           ;キースキャン要求発行

      EI
LSTB900:
      RET

```

```

XMAIN      CSEG
;+-----+
;+                +
;+                +
;+   メイン処理   +
;+                +
;+                +
;+-----+

```

```

MMAIN:

      CALL  !MKYSCAN       ;キースキャン処理呼び出し
      CALL  !MSELF        ;Flash書き込み処理呼び出し
      CALL  !MTIMER        ;タイマ処理呼び出し
      CALL  !MSTB         ;スタンバイ処理呼び出し

      BR    MMAIN

```

end

付録B 改版履歴

版 数	発行年月	改版箇所	改版内容
第1版	December 2010	-	-