

お客様各位

カタログ等資料中の旧社名の扱いについて

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

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

2010年4月1日

ルネサスエレクトロニクス株式会社

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

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

ご注意書き

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

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

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

1. M25P シリーズのシリアル Flash 制御方法

1.1 動作概要

M16C 内蔵のクロック同期形シリアル I/O を使って、M25P シリーズのシリアル Flash 制御を実現します。
 サンプルプログラムでは、以下の制御を行っています。

- シリアル Flash の S#端子を M16C の Port に接続し、M16C 汎用ポート出力で、制御する。
- データの入出力を、クロック同期形シリアル I/O (内部クロックを使用) で、制御する。
 クロック同期形シリアル I/O を割り当てる際には、高速動作させるため、CMOS 出力可能な端子割り当てと CMOS 出力設定をしてください。
 送信制御は、送信バッファの空きを検出して、割り込みを使用せずに、送信割り込み要求ビットを利用して、います。したがって、割り込み関連を以下のように設定しています。
 — 割り込み優先レベルを”000b”のレベル 0 (割り込み禁止) に設定。
 — 送信割り込み要因選択ビットを”0:送信バッファ空”に設定。(DMA 使用時には、DMA 要因を”UART 送信割り込み要求 “に設定。)
- オプションで、DMAC を使った転送制御する。

使用可能なクロック周波数は、MCU とシリアル Flash の各データシートを参照し、設定してください。

以下に接続方法を示します。

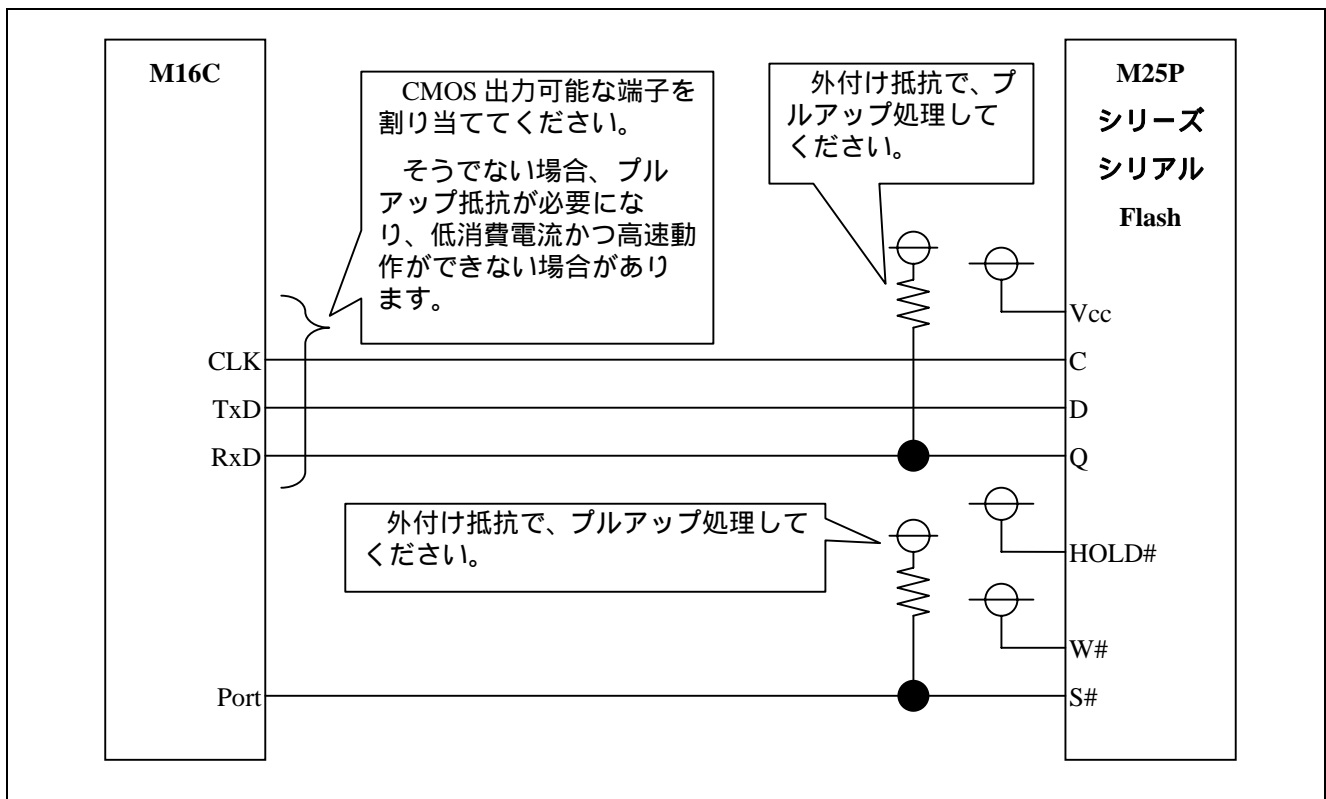


図 1-1 シリアル Flash の接続例

1.2 クロック同期形シリアル I/O で発生させるタイミング

シリアル Flash のタイミングを満足させるために、以下のようなタイミングを発生させます。

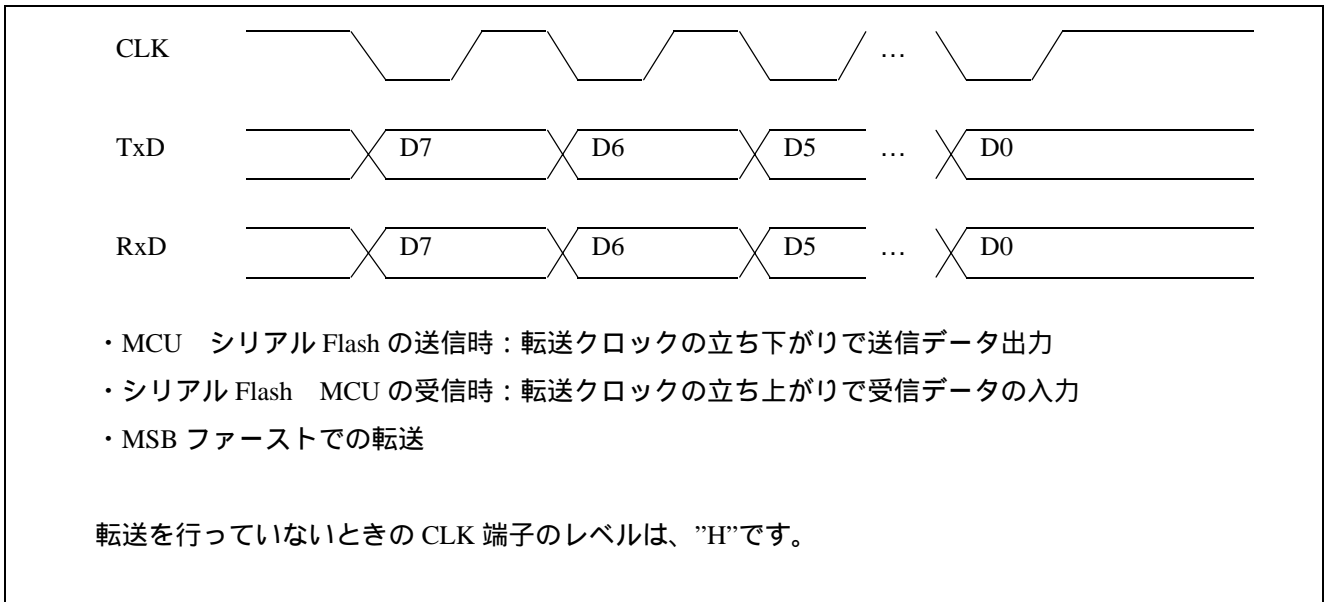


図 1-2 M16C クロック同期形シリアル I/O タイミング設定

使用可能な最大クロック周波数は、MCU およびシリアル Flash のデータシートで、確認してください。

1.3 シリアル Flash の S#端子制御

シリアル Flash の S#端子を M16C の Port に接続し、M16C 汎用ポート出力で、制御させています。

シリアル Flash の S#端子 (M16C の Port) の立ち下がりから、C 端子 (M16C の CLK) の立ち下がりまでの時間は、ソフトウェア・ウェイトで制御しています。

C 端子 (M16C の CLK) の立ち上がりから、S#端子 (M16C の Port) の立ち上がりまでの時間は、ソフトウェア・ウェイトで制御しています。

シリアル Flash のデータシートを確認して、システムに応じたソフトウェア・ウェイト時間を設定してください。

1.4 関数処理終了後の処理

関数の処理が開始されると、ポート機能設定で、まず、シリアル Flash の S#端子 (M16C の Port) に”H”、引き続き、C 端子 (M16C の CLK) に”H”を出力します。次に、シリアル I/O 機能を有効にし、使用するクロック同期形シリアル I/O モードを設定します。そして、シリアル Flash の S#端子 (M16C の Port) を”L”にし、シリアル I/O 機能を使って、命令コード等が出力されます。

関数の処理が終わると、シリアル Flash の S#端子 (M16C の Port) を”H”にした後、M16C のシリアル I/O 機能を無効にし、汎用ポート機能に切り替え、使用している Port/CLK/TxD を”H”にします。

1.5 使用される MCU のハードウェアリソース

使用されるハードウェア・リソースは、以下のとおりです。

送信制御は、送信バッファの空きを検出して、割り込みを使用せずに、送信割り込み要求ビットを利用しています。したがって、割り込み関連を以下のように設定しています。

- 割り込み優先レベルを”000b”のレベル 0 (割り込み禁止) に設定。
- 送信割り込み要因選択ビットを”0:送信バッファ空”に設定。(DMA 使用時には、DMA 要因を”UART 送信割り込み要求 “に設定。)

表 1-1 使用するハードウェア・リソース

使用するリソース	使用数
クロック同期形シリアル I/O	1ck (必須)
Port (シリアル Flash S#端子制御用)	1 ポート (必須)
DMAC	1ch (オプション)

オプションとして、DMAC を RAM UART 送受信バッファレジスタへのアクセスに使用するモードを準備しています。

1.6 M16C の SFR (周辺装置制御レジスタ) の設定 - クロック同期形シリアル I/O と割り込み制御レジスタ

シリアル Flash の仕様 / タイミングを満足させるために、クロック同期形シリアル I/O を以下のように設定します。

送信制御は、送信バッファの空きを検出して、割り込みを使用せずに、送信割り込み要求ビットを利用しています。したがって、割り込み関連を以下のように設定しています。

- 割り込み優先レベルを”000b”のレベル 0 (割り込み禁止) に設定。
- 送信割り込み要因選択ビットを”0:送信バッファ空”に設定。(DMA 使用時には、DMA 要因を”UART 送信割り込み要求 “ に設定。)

1.6.1 M32C/87 の場合

M32C/87 グループのハードウェアマニュアル Rev.1.00 の「表 17.2 クロック同期形シリアル I/O モード時の使用レジスタと設定値」を元に設定例を示します。

N チャンネル・オープンドレイン出力である UART2 以外を使用することを推奨します。

連続受信モードは禁止です。詳細はテクニカルアップデート TN-16C-A162A/J を参照願います。

表 1-2 使用するクロック同期形シリアル I/O モードの設定値

レジスタ	ビット	機能、設定値
UiTB	0~7	送信データを設定してください。
UiRB	0~7	受信データが読めます。
	OER	オーバラン・エラーフラグ
UiBRG	0~7	転送速度を設定してください。 MCU によって、転送可能なクロック周波数が異なります。
UiMR	SMD2~SMD0	“001b” (クロック同期形シリアル I/O モード) にしてください。
	CKDIR	“0” (内部クロック) にしてください。 UiBRG で、クロック周波数を設定してください。
	IOPOL	“0” (反転なし) にしてください。
UiC0	CLK1~CLK0	UiBRG レジスタのカウントソースを選択してください。
	CRS	とりあえず “0” (CTS#機能) を選択してください。 (CRD=1 (無効) のため、本機能は無効になります。)
	TXEPT	送信レジスタ空フラグ (Read Only)
	CRD	“1” (CTS#/RTS#機能の禁止) にしてください。
	NCH	“0” (CMOS 出力) にしてください。
	CKPOL	“0” (で送信データ出力 / で受信データ入力) にしてください。
	UFORM	“1” (MSB ファースト) にしてください。
UiC1	TE	初期化時には、“0” (送信禁止) にしています。 送信を許可する場合、“1” (送信許可) にしてください。
	TI	送信バッファ空フラグ (Read Only)
	RE	初期化時には、“0” (受信禁止) にしています。 受信を許可する場合、“1” (受信許可) にしてください。
	RI	受信完了フラグ (Read Only)
	UiIRS	初期化時には、“0” (UiTB レジスタ空: TI=1) にしてください。
	UiRRM	“0” (連続受信モード禁止) にしてください。
	UiLCH	“0” (データ論理反転なし) にしてください。
	SCLKSTPB	“0” にしてください。
UiSMR	0~7 (注 1)	“0x00” にしてください。
UiSMR2	0~7 (注 1)	“0x00” にしてください。
UiSMR3	0~7 (注 1)	“0x00” にしてください。
UiSMR4	0~7 (注 1)	“0x00” にしてください。

【注】 1.リセット後の値が “0x00” であるため、サンプルプログラム上では、設定しておりません。

以下は、使用する割り込み制御レジスタの設定例です。

送信制御は、送信バッファの空きを検出して、割り込み使用せずに、送信割り込み要求ビットを利用して
 います。

表 1-3 使用する割り込み制御レジスタの設定値

レジスタ	ビット	機能、設定値
SiTIC	ILVL2 ~ ILVL0	“000b”（レベル0：割り込み禁止）にしてください。
	IR	“1”の場合、割り込み要求あり。 必要に応じて、“0”を書き込んでください。

1.6.2 M16C/62P の場合

M16C/62P グループのハードウェアマニュアル Rev.2.41 の「表 17.2 クロック同期形シリアル I/O モード時の使用レジスタと設定値」を元に設定例を示します。

N チャンネル・オープンドレイン出力である UART2 以外を使用することを推奨します。

SI/O3,SI/O4 は、使用しないでください。

表 1-4 使用するクロック同期形シリアル I/O モードの設定値

レジスタ	ビット	機能、設定値
UiTB	0~7	送信データを設定してください。
UiRB	0~7	受信データが読めます。
	OER	オーバラン・エラーフラグ
UiBRG	0~7	転送速度を設定してください。 MCU によって、転送可能なクロック周波数が異なります。
UiMR	SMD2~SMD0	“001b” (クロック同期形シリアル I/O モード) にしてください。
	CKDIR	“0” (内部クロック) にしてください。 UiBRG で、クロック周波数を設定してください。
	IOPOL	“0” (反転なし) にしてください。
UiC0	CLK1~CLK0	UiBRG レジスタのカウントソースを選択してください。
	CRS	とりあえず “0” (CTS#機能) を選択してください。 (CRD=1 (無効) のため、本機能は無効になります。)
	TXEPT	送信レジスタ空フラグ (Read Only)
	CRD	“1” (CTS#/RTS#機能の禁止) にしてください。
	NCH	“0” (CMOS 出力) にしてください。
	CKPOL	“0” (で送信データ出力 / で受信データ入力) にしてください。
	UFORM	“1” (MSB ファースト) にしてください。
U0C1~ U1C1	TE	初期化時には、“0” (送信禁止) にしています。 送信を許可する場合、“1” (送信許可) にしてください。
	TI	送信バッファ空フラグ (Read Only)
	RE	初期化時には、“0” (受信禁止) にしています。 受信を許可する場合、“1” (受信許可) にしてください。
	RI	受信完了フラグ (Read Only)
	4~5	“0” を書き込んでください。読み出した場合、不定です。
	U0LCH/U1LCH	“0” (データ論理反転なし) にしてください。
	U0ERE/U1ERE	“0” (エラー信号出力しない) にしてください。
U2C1	TE	初期化時には、“0” (送信禁止) にしています。 送信を許可する場合、“1” (送信許可) にしてください。
	TI	送信バッファ空フラグ (Read Only)
	RE	初期化時には、“0” (受信禁止) にしています。 受信を許可する場合、“1” (受信許可) にしてください。
	RI	受信完了フラグ (Read Only)
	U2IRS (注 1)	UART2 を使用する場合、以下のように設定してください。 初期化時には、“0” (送信バッファ空: TI=1) にしてください。
	U2RRM (注 1)	UART2 を使用する場合、以下のように設定してください。 初期化時には、“0” (連続受信モード禁止) にしてください。 用途に応じて、UART2 連続受信モードを選択してください。

	U2LCH	“ 0 ” (データ論理反転なし) にしてください。
	U2ERE	“ 0 ” (エラー信号出力しない) にしてください。
UiSMR	0~7 (注3)	“ 0x00 ” にしてください。
UiSMR2	0~7 (注3)	“ 0x00 ” にしてください。
UiSMR3	0~7 (注3)	“ 0x00 ” にしてください。
UiSMR4	0~7 (注3)	“ 0x00 ” にしてください。
UCON	U0IRS	UART0 を使用する場合、以下のように設定してください。 初期化時には、“ 0 ” (送信バッファ空: TI=1) にしてください。
	U1IRS	UART1 を使用する場合、以下のように設定してください。 初期化時には、“ 0 ” (送信バッファ空: TI=1) にしてください。
	U0RRM (注2)	UART0 を使用する場合、以下のように設定してください。 初期化時には、“ 0 ” (連続受信モード禁止) にしてください。 用途に応じて、UART0 連続受信モードを選択してください。
	U1RRM (注2)	UART1 を使用する場合、以下のように設定してください。 初期化時には、“ 0 ” (連続受信モード禁止) にしてください。 用途に応じて、UART1 連続受信モードを選択してください。
	CLKMD0	とりあえず“ 0 ” (CLK1 からクロックを出力) を選択してください。 (CLKMD1=0 のため、本機能は無効になります。)
	CLKMD1	“ 0 ” (CLK 出力は CLK1 のみ) にしてください。
	RCSP	とりあえず“ 0 ” (共通端子) を選択してください。 (CRD は、無効のため、本機能は無効になります。)
	7	“ 0 ” を書き込んでください。読み出した場合、不定です。

- 【注】 1.UART0/UART1 の場合の設定は、UCON (UART 送受信制御レジスタ 2) で同様に設定してください。
 2.UART2 場合の設定は、U2C1 (UART 送受信制御レジスタ 1) で同様に設定してください。
 3.リセット後の値で、問題ないため、サンプルプログラム上では、設定しておりません。

以下は、使用する割り込み制御レジスタの設定例です。

送信制御は、送信バッファの空きを検出して、割り込み使用せずに、送信割り込み要求ビットを利用して
 います。

表 1-5 使用する割り込み制御レジスタの設定値

レジスタ	ビット	機能、設定値
SiTIC	ILVL2~ILVL0	“ 000b ” (レベル 0: 割り込み禁止) にしてください。
	IR	“ 1 ” の場合、割り込み要求あり。 必要に応じて、“ 0 ” を書き込んでください。

1.6.3 M16C/30P の場合

M16C/30P グループのハードウェアマニュアル Rev.1.11 の「表 15.2 クロック同期形シリアル I/O モード時の使用レジスタと設定値」を元に設定例を示します。

N チャネル・オープンドレイン出力である UART2 以外を使用することを推奨します。

表 1-6 使用するクロック同期形シリアル I/O モードの設定値

レジスタ	ビット	機能、設定値
UiTB	0~7	送信データを設定してください。
UiRB	0~7	受信データが読めます。
	OER	オーバラン・エラーフラグ
UiBRG	0~7	転送速度を設定してください。 MCU によって、転送可能なクロック周波数が異なります。
UiMR	SMD2~SMD0	“001b” (クロック同期形シリアル I/O モード) にしてください。
	CKDIR	“0” (内部クロック) にしてください。 UiBRG で、クロック周波数を設定してください。
	IOPOL	“0” (反転なし) にしてください。
UiC0	CLK1~CLK0	UiBRG レジスタのカウントソースを選択してください。
	CRS	とりあえず “0” (CTS#機能) を選択してください。 (CRD=1 (無効) のため、本機能は無効になります。)
	TXEPT	送信レジスタ空フラグ (Read Only)
	CRD	“1” (CTS#/RTS#機能の禁止) にしてください。
	NCH	“0” (CMOS 出力) にしてください。
	CKPOL	“0” (で送信データ出力 / で受信データ入力) にしてください。
	UFORM	“1” (MSB ファースト) にしてください。
U0C1 ~ U1C1	TE	初期化時には、“0” (送信禁止) にしています。 送信を許可する場合、“1” (送信許可) にしてください。
	TI	送信バッファ空フラグ (Read Only)
	RE	初期化時には、“0” (受信禁止) にしています。 受信を許可する場合、“1” (受信許可) にしてください。
	RI	受信完了フラグ (Read Only)
	4~5	“0” を書き込んでください。読み出した場合、不定です。
	U0LCH/U1LCH	“0” (データ論理反転なし) にしてください。
	U0ERE/U1ERE	“0” (エラー信号出力しない) にしてください。
U2C1	TE	初期化時には、“0” (送信禁止) にしています。 送信を許可する場合、“1” (送信許可) にしてください。
	TI	送信バッファ空フラグ (Read Only)
	RE	初期化時には、“0” (受信禁止) にしています。 受信を許可する場合、“1” (受信許可) にしてください。
	RI	受信完了フラグ (Read Only)
	U2IRS (注 1)	UART2 を使用する場合、以下のように設定してください。 初期化時には、“0” (送信バッファ空: TI=1) にしてください。
	U2RRM (注 1)	UART2 を使用する場合、以下のように設定してください。 初期化時には、“0” (連続受信モード禁止) にしてください。 用途に応じて、UART2 連続受信モードを選択してください。
	U2LCH	“0” (データ論理反転なし) にしてください。
U2ERE	“0” (エラー信号出力しない) にしてください。	

UiSMR	0~7 (注3)	“0x00” にしてください。
UiSMR2	0~7 (注3)	“0x00” にしてください。
UiSMR3	0~7 (注3)	“0x00” にしてください。
UiSMR4	0~7 (注3)	“0x00” にしてください。
UCON	U0IRS	UART0 を使用する場合、以下のように設定してください。 初期化時には、“0” (送信バッファ空: TI=1) にしてください。
	U1IRS	UART1 を使用する場合、以下のように設定してください。 初期化時には、“0” (送信バッファ空: TI=1) にしてください。
	U0RRM (注2)	UART0 を使用する場合、以下のように設定してください。 初期化時には、“0” (連続受信モード禁止) にしてください。 用途に応じて、UART0 連続受信モードを選択してください。
	U1RRM (注2)	UART1 を使用する場合、以下のように設定してください。 初期化時には、“0” (連続受信モード禁止) にしてください。 用途に応じて、UART1 連続受信モードを選択してください。
	CLKMD0	とりあえず“0” (CLK1 からクロックを出力) を選択してください。 (CLKMD1=0 のため、本機能は無効になります。)
	CLKMD1	“0” (CLK 出力は CLK1 のみ) にしてください。
	RCSP	とりあえず“0” (共通端子) を選択してください。 (CRD は、無効のため、本機能は無効になります。)
	7	“0” を書き込んでください。読み出した場合、不定です。

【注】 1. UART0/UART1 の場合の設定は、UCON (UART 送受信制御レジスタ 2) で同様に設定してください。

2. UART2 場合の設定は、U2C1 (UART 送受信制御レジスタ 1) で同様に設定してください。

3. リセット後の値で、問題ないため、サンプルプログラム上では、設定しておりません。

以下は、使用する割り込み制御レジスタの設定例です。

送信制御は、送信バッファの空きを検出して、割り込み使用せずに、送信割り込み要求ビットを利用して
います。

表 1-7 使用する割り込み制御レジスタの設定値

レジスタ	ビット	機能、設定値
SiTIC	ILVL2~ILVL0	“000b” (レベル0: 割り込み禁止) にしてください。
	IR	“1” の場合、割り込み要求あり。 必要に応じて、“0” を書き込んでください。

1.6.4 M16C/29 の場合

M16C/29 グループのデータシート Rev.1.00 の「表 14.2 クロック同期形シリアル I/O モード時の使用レジスタと設定値」を元に設定例を示します。

SI/O3,SI/O4 は、使用しないでください。

表 1-8 使用するクロック同期形シリアル I/O モードの設定値

レジスタ	ビット	機能、設定値
UiTB	0~7	送信データを設定してください。
UiRB	0~7	受信データが読めます。
	OER	オーバラン・エラーフラグ
UiBRG	0~7	転送速度を設定してください。 MCU によって、転送可能なクロック周波数が異なります。
UiMR	SMD2~SMD0	“001b” (クロック同期形シリアル I/O モード) にしてください。
	CKDIR	“0” (内部クロック) にしてください。 UiBRG で、クロック周波数を設定してください。
	7	“0” にしてください。
UiC0	CLK1~CLK0	UiBRG レジスタのカウントソースを選択してください。
	CRS	とりあえず“0” (CTS#機能) を選択してください。 (CRD=1 (無効) のため、本機能は無効になります。)
	TXEPT	送信レジスタ空フラグ (Read Only)
	CRD	“1” (CTS#/RTS#機能の禁止) にしてください。
	NCH	“0” (CMOS 出力) にしてください。
	CKPOL	“0” (で送信データ出力 / で受信データ入力) にしてください。
	UFORM	“1” (MSB ファースト) にしてください。
U0C1 ~ U1C1	TE	初期化時には、“0” (送信禁止) にしています。 送信を許可する場合、“1” (送信許可) にしてください。
	TI	送信バッファ空フラグ (Read Only)
	RE	初期化時には、“0” (受信禁止) にしています。 受信を許可する場合、“1” (受信許可) にしてください。
	RI	受信完了フラグ (Read Only)
	4~7	“0” を書き込んでください。読み出した場合、“0” です。
U2C1	TE	初期化時には、“0” (送信禁止) にしています。 送信を許可する場合、“1” (送信許可) にしてください。
	TI	送信バッファ空フラグ (Read Only)
	RE	初期化時には、“0” (受信禁止) にしています。 受信を許可する場合、“1” (受信許可) にしてください。
	RI	受信完了フラグ (Read Only)
	U2IRS (注 1)	UART2 を使用する場合、以下のように設定してください。 初期化時には、“0” (送信バッファ空: TI=1) にしてください。
	U2RRM (注 1)	UART2 を使用する場合、以下のように設定してください。 初期化時には、“0” (連続受信モード禁止) にしてください。 用途に応じて、UART2 連続受信モードを選択してください。
	U2LCH	“0” (データ論理反転なし) にしてください。
	U2ERE	“0” (エラー信号出力しない) にしてください。
U2SMR	0~7 (注 3)	“0x00” にしてください。
U2SMR2	0~7 (注 3)	“0x00” にしてください。

U2SMR3	0~7 (注3)	“0x00” にしてください。
U2SMR4	0~7 (注3)	“0x00” にしてください。
UCON	U0IRS	UART0 を使用する場合、以下のように設定してください。 初期化時には、“0” (送信バッファ空: TI=1) にしてください。
	U1IRS	UART1 を使用する場合、以下のように設定してください。 初期化時には、“0” (送信バッファ空: TI=1) にしてください。
	U0RRM (注2)	UART0 を使用する場合、以下のように設定してください。 初期化時には、“0” (連続受信モード禁止) にしてください。 用途に応じて、UART0 連続受信モードを選択してください。
	U1RRM (注2)	UART1 を使用する場合、以下のように設定してください。 初期化時には、“0” (連続受信モード禁止) にしてください。 用途に応じて、UART1 連続受信モードを選択してください。
	CLKMD0	とりあえず“0” (CTS#機能) を選択してください。 (CLKMD1=0 のため、本機能は無効になります。)
	CLKMD1	“0” (CLK 出力は CLK1 のみ) にしてください。
	RCSP	とりあえず“0” (共通端子) を選択してください。 (CRD は、無効のため、本機能は無効になります。)
	7	“0” を書き込んでください。読み出した場合、不定です。

- 【注】 1. UART0/UART1 の場合の設定は、UCON (UART 送受信制御レジスタ 2) で同様に設定してください。
 2. UART2 場合の設定は、U2C1 (UART 送受信制御レジスタ 1) で同様に設定してください。
 3. リセット後の値で、問題ないため、サンプルプログラム上では、設定しておりません。

以下は、使用する割り込み制御レジスタの設定例です。

送信制御は、送信バッファの空きを検出して、割り込み使用せずに、送信割り込み要求ビットを利用して
 います。

表 1-9 使用する割り込み制御レジスタの設定値

レジスタ	ビット	機能、設定値
SiTIC	ILVL2~ILVL0	“000b” (レベル0: 割り込み禁止) にしてください。
	IR	“1” の場合、割り込み要求あり。 必要に応じて、“0” を書き込んでください。

1.6.5 R8C/25 の場合

R8C/25 グループのハードウェアマニュアル Rev.2.00 の「表 15.2 クロック同期形シリアル I/O モード時の使用レジスタと設定値」を元に設定例を示します。

表 1-10 使用するクロック同期形シリアル I/O モードの設定値

レジスタ	ビット	機能、設定値
UiTB	0~7	送信データを設定してください。
UiRB	0~7	受信データが読めます。
	OER	オーバラン・エラーフラグ
UiBRG	0~7	転送速度を設定してください。 MCU によって、転送可能なクロック周波数が異なります。
UiMR	SMD2~SMD0	“001b” (クロック同期形シリアル I/O モード) にしてください。
	CKDIR	“0” (内部クロック) にしてください。 UiBRG で、クロック周波数を設定してください。
	7	“0” にしてください。
UiC0	CLK1~CLK0	UiBRG レジスタのカウントソースを選択してください。
	2	“0” にしてください。
	TXEPT	送信レジスタ空フラグ (Read Only)
	4	何も配置されていません。書く場合、“0” を書き込んでください。読み出した場合、“0” です。
	NCH	“0” (CMOS 出力) にしてください。
	CKPOL	“0” (で送信データ出力 / で受信データ入力) にしてください。
	UFORM	“1” (MSB ファースト) にしてください。
UiC1	TE	初期化時には、“0” (送信禁止) にしています。 送信を許可する場合、“1” (送信許可) にしてください。
	TI	送信バッファ空フラグ (Read Only)
	RE	初期化時には、“0” (受信禁止) にしています。 受信を許可する場合、“1” (受信許可) にしてください。
	RI	受信完了フラグ (Read Only)
	UiIRS	初期化時には、“0” (送信バッファ空: TI=1) にしてください。
	UiRRM	初期化時には、“0” (連続受信モード禁止) にしてください。 用途に応じて、UARTi 連続受信モードを選択してください。
	6~7	何も配置されていません。書く場合、“0” を書き込んでください。読み出した場合、“0” です。

以下は、使用する割り込み制御レジスタの設定例です。

送信制御は、送信バッファの空きを検出して、割り込み使用せずに、送信割り込み要求ビットを利用して
います。

表 1-11 使用する割り込み制御レジスタの設定値

レジスタ	ビット	機能、設定値
SiTIC	ILVL2~ILVL0	“000b” (レベル 0: 割り込み禁止) にしてください。
	IR	“1” の場合、割り込み要求あり。 必要に応じて、“0” を書き込んでください。

1.7 M16C の SFR (周辺装置制御レジスタ) の設定 - DMAC と割り込み制御レジスタ

DMAC を使用すると、データ転送の高速化が可能です。オプションとして、DMAC を RAM UART 送受信バッファレジスタへのアクセスに使用するモードを準備しています。

1.7.1 M32C/87 の場合

連続受信モード禁止のため、DMA 対応していません。詳細はテクニカルアップデート TN-16C-A162A/J を参照願います。

1.7.2 M16C/62P の場合

M16C/62P グループのハードウェアマニュアル Rev.2.41 を元に設定例を示します。

表 1-12 DMAC の設定値

レジスタ	ビット	機能、設定値
DMiSL	DSEL3 ~ DSEL0	DMA 要因は転送モードに応じて UART 送信又は UART 受信を選択してください。 UART の送受信を要因とするため、DMS を “0” (基本要因) にしてください
	4 ~ 5	“0” を書き込んでください。読み出した場合、“0” です。
	DMS	UART 送受信が要因ですので、“0” (基本要因) にしてください。
	DSR	ソフトウェアトリガを使用しませんので、“0” にしてください。
DMiCOM	DMBIT	“1” (8 ビット) にしてください。
	DMASL	“0” (単転送) にしてください。
	DMAS	DMA 要求 bit。 初期化時には、“0” (要求無し / 要求クリア) にしてください。
	DMAE	初期化時には、“0” (DMA 禁止) にしてください。 DMA を許可する場合、“1” (DMA 許可) にしてください。
	DSD	初期化時には、“0” (固定) にしています。 転送元に応じて、設定してください。
	DAD	初期化時には、“0” (固定) にしています。 転送先に応じて、設定してください。
	6 ~ 7	“0” を書き込んでください。読み出した場合、“0” です。
SARi	0 ~ 19	転送元アドレスを設定してください。
	20 ~ 23	“0” を書き込んでください。読み出した場合、“0” です。
DARi	0 ~ 19	転送元アドレスを設定してください。
	20 ~ 23	“0” を書き込んでください。読み出した場合、“0” です。
TCRi	0 ~ 15	転送回数 - 1 を設定してください。

以下は、使用する割り込み制御レジスタの設定例です。

表 1-13 使用する割り込み制御レジスタの設定値

レジスタ	ビット	機能、設定値
DMiIC	ILVL2 ~ ILVL0	“000b” (レベル 0 : 割り込み禁止) にしてください。
	IR	“1” の場合、割り込み要求あり。 必要に応じて、“0” を書き込んでください。

注 : DMA 転送を使用する場合、UART1 を使用しないことを推奨します。

UART1 の DMA 要因選択レジスタが、UART1 送信は DMA0 に、UART1 受信は DMA1 に割り付けられているため、DMA 転送を実現する場合、DMA0 と DMA1 の 2 つを使用することになり、かつ、プログラムの見直しが必要です。

1.7.3 M16C/30P の場合

M16C/30P グループのハードウェアマニュアル Rev.1.11 を元に設定例を示します。

表 1-14 DMAC の設定値

レジスタ	ビット	機能、設定値
DMiSL	DSEL3 ~ DSEL0	DMA 要因は転送モードに応じて UART 送信又は UART 受信を選択してください。 UART の送受信を要因とするため、DMS を “0” (基本要因) にしてください
	4 ~ 5	“0” を書き込んでください。読み出した場合、“0” です。
	DMS	UART 送受信が要因ですので、“0” (基本要因) にしてください。
	DSR	ソフトウェアトリガを使用しませんので、“0” にしてください。
DMiCOM	DMBIT	“1” (8 ビット) にしてください。
	DMASL	“0” (単転送) にしてください。
	DMAS	DMA 要求 bit。 初期化時には、“0” (要求無し / 要求クリア) にしてください。
	DMAE	初期化時には、“0” (DMA 禁止) にしてください。 DMA を許可する場合、“1” (DMA 許可) にしてください。
	DSD	初期化時には、“0” (固定) にしています。 転送元に応じて、設定してください。
	DAD	初期化時には、“0” (固定) にしています。 転送先に応じて、設定してください。
	6 ~ 7	“0” を書き込んでください。読み出した場合、“0” です。
SARi	0 ~ 19	転送元アドレスを設定してください。
	20 ~ 23	“0” を書き込んでください。読み出した場合、“0” です。
DARi	0 ~ 19	転送元アドレスを設定してください。
	20 ~ 23	“0” を書き込んでください。読み出した場合、“0” です。
TCRi	0 ~ 15	転送回数 - 1 を設定してください。

以下は、使用する割り込み制御レジスタの設定例です。

表 1-15 使用する割り込み制御レジスタの設定値

レジスタ	ビット	機能、設定値
DMiIC	ILVL2 ~ ILVL0	“000b” (レベル 0 : 割り込み禁止) にしてください。
	IR	“1” の場合、割り込み要求あり。 必要に応じて、“0” を書き込んでください。

注 : DMA 転送を使用する場合、UART1 を使用しないことを推奨します。

UART1 の DMA 要因選択レジスタが、UART1 送信は DMA0 に、UART1 受信は DMA1 に割り付けられているため、DMA 転送を実現する場合、DMA0 と DMA1 の 2 つを使用することになり、かつ、プログラムの見直しが必要です。

1.7.4 M16C/29 の場合

M16C/29 グループのデータシート Rev.1.00 を元に設定例を示します。

表 1-16 DMAC の設定値

レジスタ	ビット	機能、設定値
DMiSL	DSEL3 ~ DSEL0	DMA 要因は転送モードに応じて UART 送信又は UART 受信を選択してください。 UART の送受信を要因とするため、DMS を “0” (基本要因) にしてください。
	4 ~ 5	“0” を書き込んでください。読み出した場合、“0” です。
	DMS	UART 送受信を要因とするため、“0” (基本要因) にしてください。
	DSR	ソフトウェアトリガを使用しませんので、“0” にしてください。
DMiCOM	DMBIT	“1” (8ビット) にしてください。
	DMASL	“0” (単転送) にしてください。
	DMAS	DMA 要求 bit。 初期化時には、“0” (要求無し/要求クリア) にしてください。
	DMAE	初期化時には、“0” (DMA 禁止) にしてください。 DMA を許可する場合、“1” (DMA 許可) にしてください。
	DSD	初期化時には、“0” (固定) にしています。 転送元に応じて、設定してください。
	DAD	初期化時には、“0” (固定) にしています。 転送先に応じて、設定してください。
	6 ~ 7	“0” を書き込んでください。読み出した場合、“0” です。
SARi	0 ~ 19	転送元アドレスを設定してください。
	20 ~ 23	“0” を書き込んでください。読み出した場合、“0” です。
DARi	0 ~ 19	転送先アドレスを設定してください。
	20 ~ 23	“0” を書き込んでください。読み出した場合、“0” です。
TCRi	0 ~ 15	転送回数 - 1 を設定してください。

以下は、使用する割り込み制御レジスタの設定例です。

表 1-17 使用する割り込み制御レジスタの設定値

レジスタ	ビット	機能、設定値
DMiIC	ILVL2 ~ ILVL0	“000b” (レベル 0 : 割り込み禁止) にしてください。
	IR	“1” の場合、割り込み要求あり。 必要に応じて、“0” を書き込んでください。

注：DMA 転送を使用する場合、UART1 を使用しないことを推奨します。

UART1 の DMA 要因選択レジスタが、UART1 送信は DMA0 に、UART1 受信は DMA1 に割り付けられているため、DMA 転送を実現する場合、DMA0 と DMA1 の 2 つを使用することになり、かつ、プログラムの見直しが必要です。

1.7.5 R8C/25 の場合

DMAC 機能はありません。

2. サンプルプログラム

シリアル・バス上に、同一デバイスを複数個接続し、制御することが可能です。

本サンプルプログラムで、

- ・データの読出し処理
- ・データの書き込み処理
- ・ソフトウェアプロテクトによるライトプロテクト処理
- ・セクタ消去処理
- ・バルク消去処理
- ・ステータス読み出し処理
- ・ディープパワーダウン処理
- ・ディープパワーダウン解除処理
- ・ID 読み出し処理

を実現しています。

2.1 ソフトウェア動作概要

大まかに、以下の動作を実現しています

(1) ドライバ初期化処理により、ドライバで使用するリソースを取得し、初期化を行います。

この時点で、シリアル Flash に接続される制御信号(Port/CLK/TxD)を、全て”H”にします。

(2) 関数のコールにより、以下の動作を行います。

シリアル Flash の接続端子に対して、非活性状態になるように出力します。

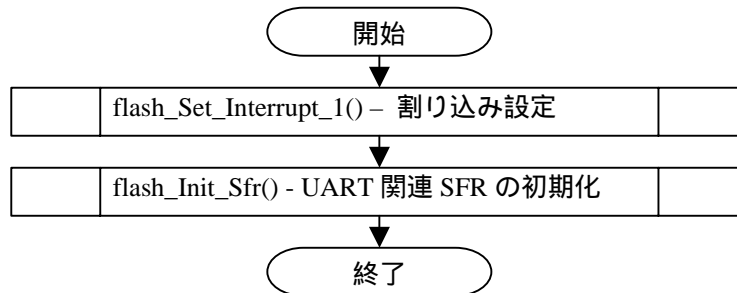
関数毎の処理を行います。

シリアル Flash に接続される制御信号(Port/CLK/TxD)を、全て”H”にします。

2.2 関数の詳細

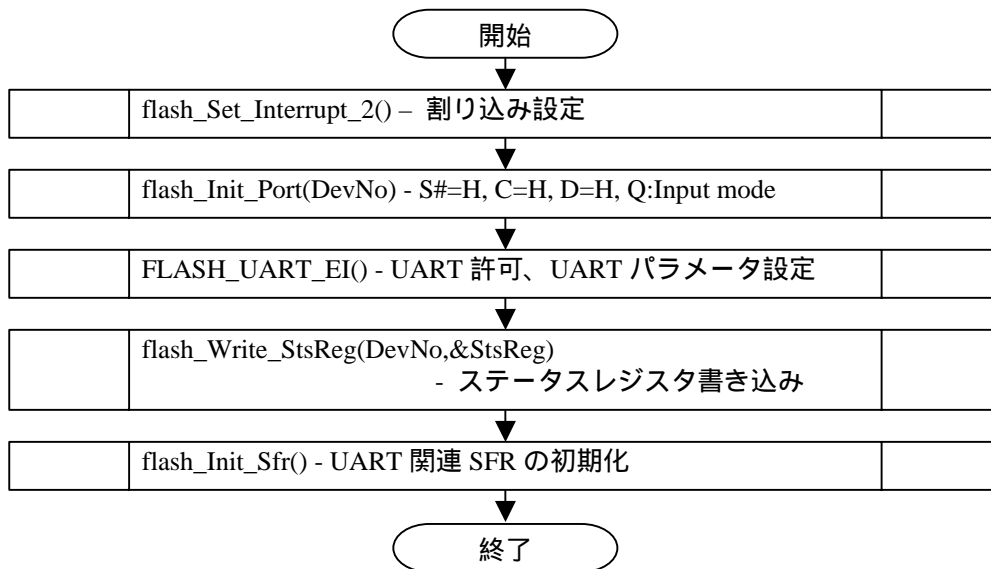
2.2.1 ドライバ初期化処理

関 数 名
Flash ドライバ初期化処理 void flash_Init_Driver(void)
引 数
なし
戻 り 値
なし
機 能
<ul style="list-style-type: none"> ・ Flash ドライバの初期化を行います。 ・ Flash 制御用 SFR の初期化を行います。 ・ システム起動時に一度だけ呼び出してください。
備 考
なし



2.2.2 ライトプロテクト設定処理

関 数 名	
ライトプロテクト設定処理	
signed short flash_Write_Protect(unsigned char DevNo, unsigned char WpSts)	
引 数	
unsigned char	DevNo ; デバイス番号
unsigned char	WpSts ; ライトプロテクト設定データ
戻 り 値	
ライトプロテクト設定結果を返します。	
FLASH_OK	; Successful operation
FLASH_ERR_PARAM	; Parameter error
FLASH_ERR_OTHER	; Other error
機 能	
<ul style="list-style-type: none"> ・ライトプロテクトの設定を行います。 ・ライトプロテクト設定データ(WpSts)でステータスレジスタの BP0,BP1,BP2 ビットは下記の通りに設定されます。 <ul style="list-style-type: none"> WpSts=0 : BP0=0, BP1=0, BP2=0 WpSts=1 : BP0=1, BP1=0, BP2=0 WpSts=2 : BP0=0, BP1=1, BP2=0 WpSts=3 : BP0=1, BP1=1, BP2=0 WpSts=4 : BP0=0, BP1=0, BP2=1 WpSts=5 : BP0=1, BP1=0, BP2=1 WpSts=6 : BP0=0, BP1=1, BP2=1 WpSts=7 : BP0=1, BP1=1, BP2=1 	
備 考	
<ul style="list-style-type: none"> ・SRWD は"0"固定です。 ・BP2 ビットの無い Flash は WpSts を 0 から 3 の範囲で設定してください。 	



2.2.3 データ読出し処理

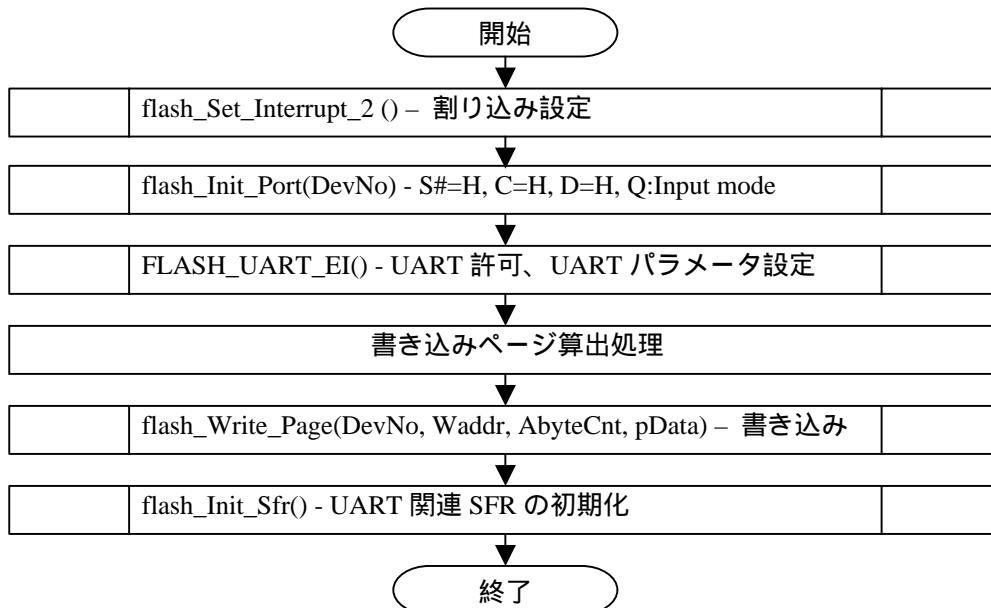
関 数 名			
データ読出し処理			
signed short flash_Read_Data(unsigned char DevNo, unsigned long RAddr, unsigned long RCnt, unsigned char * pData)			
引 数			
unsigned char	DevNo	;	デバイス番号
unsigned long	RAddr	;	読出し開始アドレス
unsigned long	RCnt	;	読出しバイト数
unsigned char FAR*	pData	;	読出しデータ格納バッファポインタ
戻 り 値			
読出し結果を返します。			
FLASH_OK	;		Successful operation
FLASH_ERR_PARAM	;		Parameter error
FLASH_ERR_HARD	;		Hardware error
FLASH_ERR_OTHER	;		Other error
機 能			
<ul style="list-style-type: none"> ・ Flash からバイト単位でデータの読出しを行います。 ・ 指定アドレスから指定バイト数分、データを読出します。 			
備 考			
<ul style="list-style-type: none"> ・ 最大読出し書き込みアドレスは、Flash 容量-1 です。 			



2.2.4 データ書き込み処理

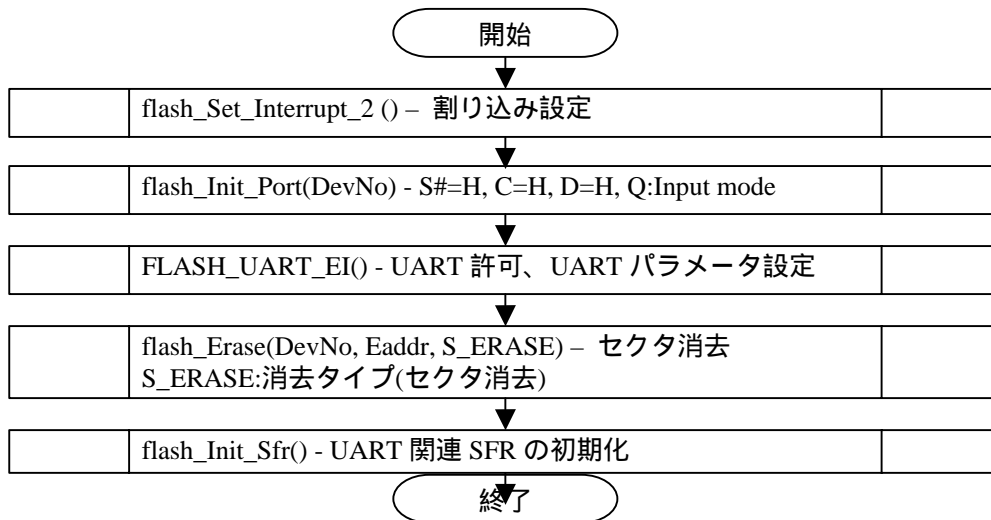
関 数 名			
データ書き込み処理			
signed short flash_Write_Data(unsigned char DevNo, unsigned long WAddr, unsigned long WCnt, unsigned char FAR* pData)			
引 数			
unsigned char	DevNo	;	デバイス番号
unsigned long	WAddr	;	書き込み開始アドレス
unsigned long	WCnt	;	書き込みバイト数
unsigned char FAR*	pData	;	書き込みデータ格納バッファポインタ
戻 り 値			
書き込み結果を返します。			
FLASH_OK	;	Successful operation	
FLASH_ERR_PARAM	;	Parameter error	
FLASH_ERR_HARD	;	Hardware error	
FLASH_ERR_OTHER	;	Other error	
機 能			
<ul style="list-style-type: none"> Flash へバイト単位でデータの書き込みを行います。 指定アドレスから指定バイト数分、データを書き込みます。 			
備 考			
<ul style="list-style-type: none"> Flash への書き込みは、ライトプロテクト解除状態の場合のみ可能です。 書き込みデータはライトプロテクトされたページに書込めず、エラーも返しません。 最大書き込みアドレスは、Flash 容量-1 です。 			

シリアル Flash への書き込みはページ書換えを使用しております。データはページ単位のデータに分割され書込まれます。



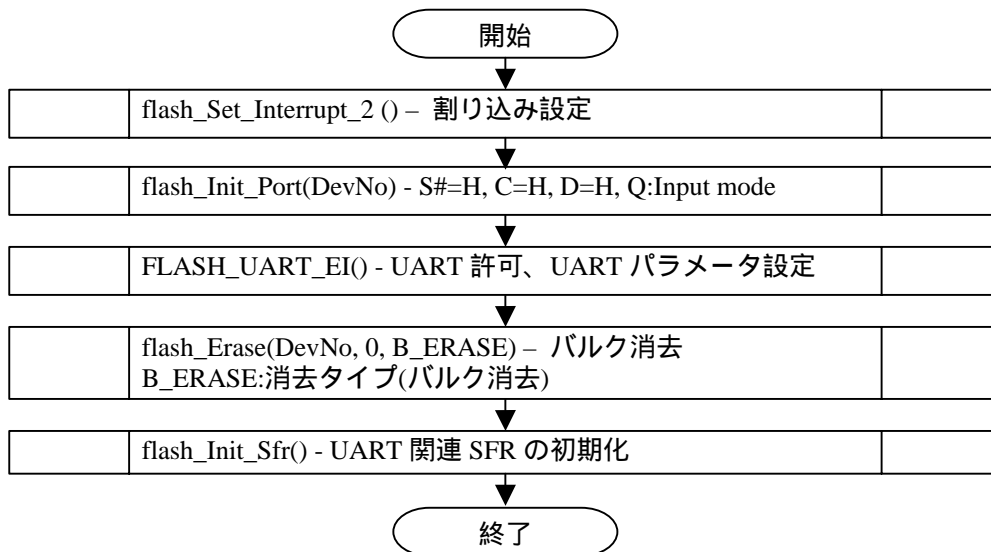
2.2.5 セクタ消去処理

関 数 名	
セクタ消去処理	
signed short flash_SectorErase(unsigned char DevNo, unsigned long EAddr)	
引 数	
unsigned char	DevNo ; デバイス番号
unsigned long	EAddr ; 消去アドレス
戻 り 値	
消去結果を返します。	
FLASH_OK	; Successful operation
FLASH_ERR_PARAM	; Parameter error
FLASH_ERR_HARD	; Hardware error
FLASH_ERR_OTHER	; Other error
機 能	
消去アドレスのセクタを消去します。	
備 考	
<ul style="list-style-type: none"> ・ Flash の消去は、ライトプロテクト解除状態の場合のみ可能です。 ・ ライトプロテクトされているセクタは、消去できず、エラーも返しません。 	



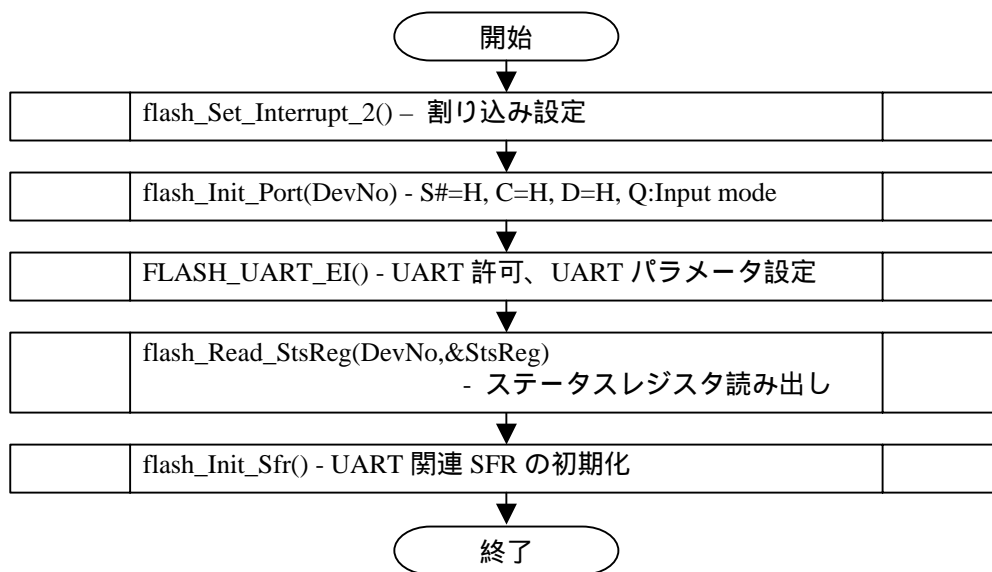
2.2.6 バルク消去処理

関 数 名		
バルク消去処理		
signed short flash_BulkErase(unsigned char DevNo)		
引 数		
unsigned char	DevNo	; デバイス番号
戻 り 値		
消去結果を返します。		
FLASH_OK		; Successful operation
FLASH_ERR_PARAM		; Parameter error
FLASH_ERR_HARD		; Hardware error
FLASH_ERR_OTHER		; Other error
機 能		
メモリの全データを消去します。		
備 考		
<ul style="list-style-type: none"> Flash の消去は、ライトプロテクト解除状態の場合のみ可能です。 ライトプロテクトされている場合、消去できず、エラーも返しません。 		



2.2.7 ステータス読出し処理

関 数 名	
ステータス読出し処理	
signed short flash_Read_Status(unsigned char DevNo, unsigned char * pStatus)	
引 数	
unsigned char DevNo	; デバイス番号
unsigned char FAR* pStatus	; 読出しステータス格納バッファ
戻 り 値	
ステータスレジスタ取得結果を返します。	
FLASH_OK	; Successful operation
FLASH_ERR_PARAM	; Parameter error
FLASH_ERR_HARD	; Hardware error
FLASH_ERR_OTHER	; Other error
機 能	
<ul style="list-style-type: none"> ・ステータスの読出しを行います。 ステータスレジスタの読出しを行います。 ・読出しステータス格納バッファ(pStatus)には下記情報が格納されます。 <ul style="list-style-type: none"> bit7 : SRWD <ul style="list-style-type: none"> 0 : ステータスレジスタ変更可 1 : ステータスレジスタ変更不可 bit6-5 : Reserved (All "0") bit4-2 : BP2/BP1/BP0 bit1 : WEL <ul style="list-style-type: none"> 0 : Write Disable 1 : Write Enable bit0 : WIP <ul style="list-style-type: none"> 1 : 書き込み動作中 	
備 考	
<ul style="list-style-type: none"> ・BP2/BP1/BP0 とプロテクトエリアの関係は使用 Flash の仕様にて確認願います。 ・BP2 ビットの無い Flash では BP2 ビットの情報出力されません。 	



2.2.8 ディープパワーダウン処理

関数名		
ディープパワーダウン処理		
signed short flash_DeepPDown(unsigned char DevNo)		
引数		
unsigned char	DevNo	; デバイス番号
戻り値		
ディープパワーダウン処理結果を返します。		
FLASH_OK	;	Successful operation
FLASH_ERR_PARAM	;	Parameter error
FLASH_ERR_HARD	;	Hardware error
FLASH_ERR_OTHER	;	Other error
機能		
・ Deep Power Down Mode の設定を行います。		
備考		
なし		



2.2.9 ディープパワーダウン解除処理

関 数 名		
ディープパワーダウン解除処理		
signed short flash_ReleaseDeepPDown(unsigned char DevNo, unsigned char FAR* pData)		
引 数		
unsigned char	DevNo	; デバイス番号
unsigned char FAR*	pData	; Electronic Signature 格納バッファ
戻 り 値		
ディープパワーダウン解除処理結果を返します。		
FLASH_OK		; Successful operation
FLASH_ERR_PARAM		; Parameter error
FLASH_ERR_HARD		; Hardware error
FLASH_ERR_OTHER		; Other error
機 能		
<ul style="list-style-type: none"> ・ Deep Power Down mode から Standby mode に遷移します。 ・ Electronic Signature を読出します。 		
備 考		
なし		



2.2.10 ID 読出し処理

関 数 名	
ID 読出し処理	
signed short flash_ReadID(unsigned char DevNo, unsigned char FAR* pData)	
引 数	
unsigned char	DevNo ; デバイス番号
unsigned char FAR*	pData ; ID データ格納バッファ
	ID データは下記順番に 3 byte 格納されます。
	(1) Manufacture ID
	(2) Memory Type
	(3) Memory Capacity
戻 り 値	
ID 読出し結果を返します。	
FLASH_OK	; Successful operation
FLASH_ERR_PARAM	; Parameter error
FLASH_ERR_HARD	; Hardware error
FLASH_ERR_OTHER	; Other error
機 能	
・ Manufacture ID 及び Device ID を出力します。	
備 考	
なし	



2.3 戻り値定義

```
#define FLASH_OK          (short)( 0)    /* Successful operation    */
#define FLASH_ERR_PARAM  (short)(-1)    /* Parameter error        */
#define FLASH_ERR_HARD   (short)(-2)    /* Hardware error         */
#define FLASH_ERR_OTHER  (short)(-3)    /* Other error            */
```

2.4 ユーザー設定例

以下にルネサステクノロジ MCU M16C/62P を使った場合の設定例を示します。
 設定箇所は、各ファイル中の「/** SET **/」というコメントの部分です。

2.4.1 flash.h

(1)使用するデバイスの個数とデバイス番号の定義

使用するデバイスの個数を指定し、各デバイスに番号を割り当ててください。
 下記は、デバイスを 1 個使用し、デバイス番号を 0 に割り当てる場合の例です。
 3 個以上使用したい場合は、flash_io.h も含め、修正する必要があります。

```

/*-----*/
/*   Flash の個数を定義してください。(1~N 個)                               */
/*   デバイス番号は、接続する Flash の個数に合わせて定義してください。   */
/*   Define number of required serial Flash devices.(1~N devices)          */
/*   Define the device number in accordance with the number of serial Flash devices
    */
/*   to be connected.                                                       */
/*-----*/
/* Define no. of devices */
#define FLASH_DEV_NUM      1          /* 1devices          */

/* Define no. of slots */
#define FLASH_DEV0        0          /* Device 0          */
#define FLASH_DEV1        1          /* Device 1          */
  
```

(2)使用するデバイスの定義

使用するデバイスを指定してください。
 下記は、M25P10A デバイスを使用する場合の例です。

```

/*-----*/
/*   使用する Flash を定義してください。                                   */
/*   Define the serial Flash device.                                       */
/*-----*/
// #define M25P05A          /* 512kbit   ( 64kByte) */ /** SET **/
#define M25P10A           /* 1Mbit     ( 128kByte) */ /** SET **/
// #define M25P20          /* 2Mbit     ( 256kByte) */ /** SET **/
// #define M25P40          /* 4Mbit     ( 512kByte) */ /** SET **/
// #define M25P16          /* 16Mbit    ( 2MByte)    */ /** SET **/
// #define M25P32          /* 32Mbit    ( 4MByte)    */ /** SET **/
// #define M25P64          /* 64Mbit    ( 8MByte)    */ /** SET **/
  
```

(3)UART 送信/DMA 割り込みの設定方法の定義

UART 送信割り込み設定 / DMA 割り込みの設定方法を定義してください。

本ソフトウェアでは、UART の送信処理として、割り込み禁止状態にさせ、送信割り込み要求ビット(IR)を使用し、送信制御しております。

この割り込み禁止の設定方法は、3つの方法が選択可能です。

システムに応じた設定をしてください。

上位システムで設定し、ドライバでは設定しない場合

#define FLASH_IC_SETTING0 を有効にしてください。

ドライバ初期化時 (flash_Init_Driver()実行時) に設定する場合

#define FLASH_IC_SETTING1 を有効にしてください。

UART 送信時 (flash_Read_Data()、 flash_Write_Data()実行時) に設定する場合

#define FLASH_IC_SETTING2 を有効にしてください。

なお、 と を同時に有効にさせることも可能です。

注意事項

上記 、 選択時の割り込み設定方法は下記のとおりです。

割り込み禁止(DI) UART(もしくはDMA)送信割り込み設定(割り込み禁止と割り込み要求クリア)

割り込み許可(EI)

上位システムにて割り込み許可フラグ (I フラグ) の管理を行う場合は注意してください。

```

/*-----*/
/*  UART 送信/DMA 割り込み設定の設定方法を定義してください。 */
/*  Define the setting method of the UART transmission interrupt and the DMA interrupt. */
/*
/*  "FLASH_IC_SETTING1"、"FLASH_IC_SETTING2"を選択時の割り込み設定方法は下記のとおりで
す。 */
/*  割り込み禁止(DI) 割り込み設定 割り込み許可(EI) */
/*  上位システムにて割り込み許可フラグ ( I フラグ ) の管理を行う場合は注意してください。 */
/*  上位システムにて割り込み管理を行う場合は"FLASH_IC_SETTING0"を選択してください。 */
/*  The setting method of the interrupt when "FLASH_IC_SETTING1" and
FLASH_IC_SETTING2" are */
/*  selected is as follows. */
/*  Interrupt disable (DI) interrupt setting interrupt enable (EI) */
/*  When manage an interrupt enable flag (I flag) by a higher system, please be
careful. */
/*  When interrupt it by a higher system and manage it, please choose
"FLASH_IC_SETTING0". */
/*-----*/
#define FLASH_IC_SETTING0 /* Doesn't set in this driver */
//#define FLASH_IC_SETTING1 /* When the driver is initialized, it sets */
//#define FLASH_IC_SETTING2 /* When the resource is used, it sets */

```

2.4.2 flash_sfr.h

使用する MCU に対応するヘッダ (flash_sfr.h.xxx) を flash_sfr.h にリネームして使用してください。

以下は、M16C/62P を使用する例です。

サンプルプログラムでは、クロック同期形シリアル I/O として、UART0 を使用する場合での記述例を示して

います。また、DMAC を使用する場合、DMA0 を使用する場合での記述例を示しています。

上記のリソースで、使用する場合には、設定変更箇所はありません。

UART リソース変更の場合 / DMAC リソース変更の場合 / 他 MCU を使用する場合には、システムに応じた

設定が必要です。

(1)UART リソース

```

/*----- UART definitions -----*/
#define FLASH_UART_STIC      s0tic /* UART TX interrupt control register      */

#define FLASH_UART_TXBUF    u0tb /* UART transmit buffer register      */
#define FLASH_UART_TXBUFL  u0tbl /* UART transmit buffer register(lower 8bit)*/
#define FLASH_UART_RXBUF    u0rb /* UART receive buffer register      */
#define FLASH_UART_BRG      u0brg /* UART bit rate generator          */
#define FLASH_UART_MR       u0mr /* UART transmit/receive mode register*/
#define FLASH_UART_C0       u0c0 /* UART transmit/receive control register 0*/
#define FLASH_UART_C1       u0c1 /* UART transmit/receive control register 1*/

#define FLASH_UART_TXEND    txept_u0c0 /* UART TX Reg. empty flag          */
#define FLASH_UART_TXNEXT  ir_s0tic /* UART TX complete flag            */
#define FLASH_UART_TI      ti_u0c1 /* UART TX complete flag            */
#define FLASH_UART_RXNEXT  ri_u0c1 /* UART RX complete flag            */
#define FLASH_UART_IRS     u0irs /* UART transmit interrupt cause select flag*/
#define FLASH_UART_RRM     u0rrm /* UART continuous receive mode enable flag*/
    
```

を追加変更し、それに伴う、/* UART setting */の定義も、「1.6 M16C の SFR (周辺装置制御レジスタ) の設定 - クロック同期形シリアル I/O」を参照し、追加変更してください。

(2)DMAC リソース

```

/*----- DMAC definitions -----*/
#ifdef FLASH_DMA_ON
#define FLASH_DMA_DMIC      dm0ic      /* DMA interrupt control register      */

#define FLASH_DMA_SL        dm0sl      /* DMA request cause select register   */
#define FLASH_DMA_CON       dm0con     /* DMA control register                */
#define FLASH_DMA_SAR       sar0       /* DMA source pointer                  */
#define FLASH_DMA_DAR       dar0       /* DMA destination pointer             */
#define FLASH_DMA_TCR       tcr0       /* DMA transfer counter                */
#define FLASH_DMA_END       ir_dm0ic  /* DMA interrupt request flag          */

```

を追加変更し、それに伴う、/* DMA setting */の定義も、「1.7 M16C の SFR (周辺装置制御レジスタ) の設定 - DMAC」を参照し、追加変更してください。

2.4.3 flash_io.h

使用する MCU に対応するヘッダ(flash_io.h.xxx)を flash_io.h にリネームして使用してください。

以下は、M16C/62P を使用する例です。

(1)使用するマイコンの UART,DMAC 等の使用リソースの定義

使用するマイコンのリソースを指定してください。

下記は、クロック同期形シリアル I/O と DMAC を使用する場合の例です。

```

/*-----*/
/*   使用するマイコンリソースの組合せを定義してください。           */
/*   Define the combination of the MCU's resources.                 */
/*-----*/
// #define FLASH_OPTION_1      /* Low speed      */ /* UART      */
#define FLASH_OPTION_2      /* High speed */ /* UART + DMAC */

```

(2)使用するマイコンの制御ポートの定義

使用するマイコンのリソースを指定してください。

下記は、クロック同期形シリアル I/O の Rx/D、Tx/D、CLK と CS#を割り当てている例です。

デバイスを 2 個接続する場合は、CS1 に関する定義をしてください。

3 個以上使用したい場合は、flash.h も含め、修正する必要があります。

```

/*-----*/
/*   使用する制御ポートを定義してください。                         */
/*   Define the control port.                                       */
/*-----*/
#define FLASH_P_DATAO    p6_3      /* FLASH DataOut      */
#define FLASH_P_DATAI    p6_2      /* FLASH DataIn       */
#define FLASH_P_CLK      p6_1      /* FLASH CLK          */
#define FLASH_D_DATAO    pd6_3     /* FLASH DataOut      */
#define FLASH_D_DATAI    pd6_2     /* FLASH DataIn       */
#define FLASH_D_CLK      pd6_1     /* FLASH CLK          */

#define FLASH_P_CS0      p10_5     /* FLASH CS0 (Negative-true logic) */
#define FLASH_D_CS0      pd10_5    /* FLASH CS0 (Negative-true logic) */
#if (FLASH_DEV_NUM > 1)
#define FLASH_P_CS1      p10_1     /* FLASH CS1 (Negative-true logic) */
#define FLASH_D_CS1      pd10_1    /* FLASH CS1 (Negative-true logic) */
#endif /* #if (FLASH_DEV_NUM > 1) */

```

(3) 消去 busy wait のソフトウェアタイマ値を定義

使用している Flash に応じた消去 busy wait のソフトウェアタイマ値を定義してください

```

/*-----*/
/* 消去 busy wait のソフトウェアタイマ値を定義してください。 */
/* タイムアウトせず、Ready 状態まで待ち続けたい場合は FLASH_EBUSY_WAIT_TIME の定義をコメント */
/* にしてください。 */
/* Define the software timer value of erase busy waiting. */
/* If you want to wait till the flash comes to ready status without time out, */
/* comment the definition FLASH_EBUSY_WAIT_TIME. */
/*-----*/
//#define FLASH_EBUSY_WAIT_TIME

#ifndef FLASH_EBUSY_WAIT_TIME
#define FLASH_EBUSY_WAIT(ushort)40000 /* Erase busy waiting time 40000* 1Ms =
4s */
#else
#define FLASH_EBUSY_WAIT(ushort)0 /* Waiting without time out */
#endif

```

2.4.4 mtl_com.h (共通ヘッダファイル)

使用する MCU に対応するヘッダ (mtl_com.h.xxx) を mtl_com.h にリネームして使用してください。

以下は、M16C/62P を使用する例です。

(1) OS のヘッダファイル定義

本ソフトウェアは、OS 非依存です。

下記の例は、OS を使用しない (MR30 のシステムコールを使用しない) 場合の例です。

```

/* システムコールを使用するため、 */
/* プロトタイプ宣言のある OS のヘッダファイルをインクルードしてください。 */
/* OS を使用しない場合は、下記デファインとインクルードをコメントにしてください。 */
/* In order to use wai_sem/sig_sem/dly_tsk for microITRON (Real-Time OS)-compatible, */
*/
/* include the OS header file that contains the prototype declaration. */
/* When not using the OS, put the following 'define' and 'include' as comments. */
*/
// #define MTL_OS_USE /* Use OS */
// #include <RTOS.h> /* OS header file */
// #include "mtl_os.h"

```

(2) 共通アクセス領域を定義したヘッダファイル定義

MCU の機能レジスタの定義がされているヘッダファイルをインクルードします。

主にデバイスドライバがポート制御等に使用するため、インクルードする必要があります。

下記の例は、M16C/62P のヘッダファイルをインクルードする場合の例です。MCU に合わせて、ヘッダファイルをインクルードしてください。

```

/* MCU の SFR 領域のデファイン値を使用しているため、 */
/* I/O 周りのデファイン定義のあるヘッダファイルをインクルードしてください。 */
/* In order to use definitions of MCU SFR area, */
/* include the header file of MCU SFR definition. */
#include "sfr62p.h" /* definition of MCU SFR */

```

(3) ループタイマの定義

ソフトウェア・ループタイマを使用する場合、以下のヘッダをインクルードします。

主にデバイスドライバが待ち時間を確保するために、使用します。

ソフトウェア・ループタイマを使用しない場合は、下記インクルードをコメントにしてください

下記の例は、ソフトウェア・ループタイマを使用する場合の例です。

```

/* ループタイマを使用しない場合は、下記インクルードをコメントにしてください。 */
/* When not using the loop timer, put the following 'include' as comments. */
#include "mtl_tim.h"

```


(4) エンディアンタイプ定義

M16C ファミリ用 FAT ファイルシステム・ライブラリ用の設定です。

M16C ファミリの場合、リトルエンディアンを指定してください。

```
/* 使用する MCU のエンディアンタイプを指定してください。 */
/* ビックエンディアンを指定する時はリトルエンディアンの定義をコメントにしてください。 */
/* When using M16C or SuperH for Little Endian setting, define it. */
/* When using other MCUs, put 'define' as a comment. */
#define MTL_MCU_LITTLE /* Little Endian */
```

(5) mtl_endi.c の処理の高速化

mtl_end.c の処理の高速化指定が可能です。M16C を使用する場合、高速になります。

下記の例は、M16C ファミリを使用する場合の例です。

```
/* M16C を使用する場合、定義を有効にしてください。 */
/* When using M16C, define it. */
/* It performs the fast processes of 'mtl_endi.c'. */
/* mtl_endi.c の処理の高速化が可能です。 */
#define MTL_ENDI_HISPEED /* Uses the high-speed function. */
```

(6) 使用する標準ライブラリのタイプの定義

使用する標準ライブラリのタイプを定義してください。

下記に示す処理をコンパイラ添付のライブラリで使用する場合は、下記デファイン定義をコメントにしてください。有効にした場合、ユーザ側で、準備する必要があります。

下記の例は、コンパイラ添付のライブラリを使用する場合例です。

```
/* 使用する標準ライブラリのタイプを指定してください。 */
/* 下記に示す処理をコンパイラ添付のライブラリで使用する場合は、 */
/* 下記デファイン定義をコメントにしてください。 */
/* Specify the type of user standard library. */
/* When using the compiler-bundled library for the following processes, */
/* put the following 'define' as comments. */
/* memcmp() / memmove() / memcpy() / memset() / strcat() / strcmp() / strcpy() / */
/* strlen() */
// #define MTL_USER_LIB /* use optimized library */
```

(7) アクセスする RAM 領域の定義

使用する RAM 領域を定義してください。

標準関数や一部の処理に効率の良い処理を適用します。

どちらの定義もしない場合、コンパイルエラーを出力するようにチェックを行っています。

通常は、MTL_MEM_NEAR を定義してください。

ただし、M16C/60、M16C/30P、M16C/20 の FAR 領域の RAM にアクセスする場合、MTL_MEM_FAR を定義してください。

下記の例は、M16C/60、M16C/20、R8C の場合の MTL_MEM_NEAR を定義する設定です。

```

/* 使用する処理群がアクセスする RAM 領域を定義してください。          */
/* 標準関数や一部の処理に効率の良い処理を適用します。                */
/* Define the RAM area to be accessed by the user process.             */
/* Efficient operations for standard functions and processes are applied. */
//#define MTL_MEM_FAR          /* Supports Far RAM area of M16C/60      */
#define MTL_MEM_NEAR          /* Supports Near RAM area. (Others) */

```

上記の define のみを設定し、他は変更しないでください。

2.4.5 mtl_tim.h

(1) ソフトウェア・タイマの定義

内部で使用するソフトウェア・タイマを設定してください。

下記は、24MHz ノーウェイトで動作させた場合の参考値です。

システムに応じて、設定してください。

```

/* タイマのカウント値を定義してください。                              */
/* MCU 及びクロック、ウェイトに応じて設定してください。              */
/* Setting for 24MHz no wait                                           */
#define MTL_T_1US              1          /* 1-us loop count */
#define MTL_T_2US              2          /* 2-us loop count */
#define MTL_T_4US              5          /* 4-us loop count */
#define MTL_T_5US              6          /* 5-us loop count */
#define MTL_T_10US             13         /* 10-us loop count */
#define MTL_T_20US             28         /* 20-us loop count */
#define MTL_T_30US             43         /* 30-us loop count */
#define MTL_T_50US             72         /* 50-us loop count */
#define MTL_T_100US            145        /* 100-us loop count */
#define MTL_T_200US            293        /* 200-us loop count */
#define MTL_T_300US            439        /* 300-us loop count */
#define MTL_T_400US            ( MTL_T_200US * 2 ) /* 400-us loop count */
#define MTL_T_1MS              1471      /* 1-ms loop count */

```

2.5 使用上の注意事項

サンプルプログラムでは、クロック同期形シリアル I/O として、UART0 を使用する場合での記述例を示しています。

また、DMAC を使用する場合、DMA0 を使用する場合での記述例を示しています。

他のリソースを使用する場合、使用にあたっては、ハードウェアに合わせて、ソフトウェアを設定してください。

2.6 組み込み時の注意事項

本サンプルプログラムを組み込む場合は、flash.h をインクルードしてください。

2.7 他 M16C ファミリ MCU を使用する場合

他 M16C ファミリ MCU を使用する場合、容易に対応が可能です。

準備するファイルは、

(1)flash_io.h.xxx に相当する I/O モジュール共通定義

使用 MCU の SFR ヘッダを参照し、使用する I/O ピンを定義してください。

(2)flash_sfr.h.xxx に相当する SFR 共通定義

使用 MCU の SFR ヘッダを参照し、使用する UART,DMAC を定義してください。

(3)mtl_com.h.xxx に相当するヘッダ定義

使用 MCU 用にヘッダを作成、定義してください。

です。添付のものを参考に、作成してください。

また、flash_io.h と flash_sfr.h と mtl_com.h で、作成したヘッダを指定してください。

2.8 ファイルの構成

¥com	<DIR>	共通関数のディレクトリ	
	mtl_com.c	mtl_com.h.common	共通関数の各種定義
	mtl_com.h.m16c29		共通ヘッダファイル M16C/29
	mtl_com.h.m16c30p		共通ヘッダファイル M16C/30P
	mtl_com.h.m16c62p		共通ヘッダファイル M16C/62P
	mtl_com.h.m32c87		共通ヘッダファイル M32C/87
	mtl_com.h.r8c25		共通ヘッダファイル R8C/25
	mtl_mem.c		共通ファイル
	mtl_os.c	mtl_os.h	共通ファイル
	mtl_str.c		共通ファイル
	mtl_tim.c	mtl_tim.h	共通ファイル
	mtl_tim.h.sample	共通ファイル (参考)	
¥sflash_spi	<DIR>	シリアル Flash ドライバのディレクトリ	
	flash.h		ドライバ 共通定義
	flash_io.c		I/O モジュール
	flash_io.h.m16c29		I/O モジュール共通定義 M16C/29
	flash_io.h.m16c30p		I/O モジュール共通定義 M16C/30P
	flash_io.h.m16c62p		I/O モジュール共通定義 M16C/62P
	flash_io.h.m32c87		I/O モジュール共通定義 M32C/87
	flash_io.h.r8c25		I/O モジュール共通定義 R8C/25
	flash_sfr.h.m16c29		SFR 共通定義 M16C/29
	flash_sfr.h.m16c30p		SFR 共通定義 M16C/30P
	flash_sfr.h.m16c62p		SFR 共通定義 M16C/62P
	flash_sfr.h.m32c87		SFR 共通定義 M32C/87
	flash_sfr.h.r8c25		SFR 共通定義 R8C/25
	flash_usr.c		ドライバ ユーザ I/F モジュール
¥sample	<DIR>	サンプル・プログラムのディレクトリ	
	testmain.c		動作検証用のサンプルプログラム 動作検証用に使用してください。
	common.c	common.h	共通関数の各種定義

ホームページとサポート窓口<website and support,ws>

ルネサス テクノロジホームページ

<http://japan.renesas.com/>

お問合せ先

<http://japan.renesas.com/inquiry>

csc@renesas.com

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	07/02/20	—	初版発行
1.01	07/11/09	P4	1.4 項 関数処理終了後の High-z 処理を関数処理終了後の処理に変更。 処理内容変更。
		P15	M32C/87 の DMA 設定例削除
		P19	3 箇所「制御信号(Port/CLK/TxD)を全て High にします。」 に変更
		P20-P30	関数フローから「flash_Open_Port(DevNo) - 使用ポートの Hi-z」 を削除
1.02	08/02/17	1	動作確認デバイスにソフトウェア・バージョンを追加
		6	2. サンプルプログラム にて、以下を削除した。 「シリアル・バス上に、シリアル・バスを使用するリムーバブル・メディア（マルチメディアカード等）が接続された場合のリムーバブル・メディアの挿抜を想定し、関数の完了時点で、シリアル EEPROM に接続される制御信号(Port/CLK/SOUT)を全て High にします。」

すべての商標および登録商標は、それぞれの所有者に帰属します。

本資料ご利用に際しての留意事項

1. 本資料は、お客様に用途に応じた適切な弊社製品をご購入いただくための参考資料であり、本資料中に記載の技術情報について弊社または第三者の知的財産権その他の権利の実施、使用を許諾または保証するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例など全ての情報の使用に起因する損害、第三者の知的財産権その他の権利に対する侵害に関し、弊社は責任を負いません。
3. 本資料に記載の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、あるいはその他軍事用途の目的で使用しないでください。また、輸出に際しては、「外国為替および外国貿易法」その他輸出関連法令を遵守し、それらの定めるところにより必要な手続を行ってください。
4. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例などの全ての情報は本資料発行時点のものであり、弊社は本資料に記載した製品または仕様等を予告なしに変更することがあります。弊社の半導体製品のご購入およびご使用に当たりましては、事前に弊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、弊社ホームページ (<http://www.renesas.com>) などを通じて公開される情報に常にご注意ください。
5. 本資料に記載した情報は、正確を期すため慎重に制作したのですが、万一本資料の記述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、弊社はその責任を負いません。
6. 本資料に記載の製品データ、図、表などに示す技術的な内容、プログラム、アルゴリズムその他応用回路例などの情報を流用する場合は、流用する情報を単独で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。弊社は、適用可否に対する責任を負いません。
7. 本資料に記載された製品は、各種安全装置や運輸・交通用、医療用、燃焼制御用、航空宇宙用、原子力、海底中継用の機器・システムなど、その故障や誤動作が直接人命を脅かしあるいは人体に危害を及ぼすおそれのあるような機器・システムや特に高度な品質・信頼性が要求される機器・システムでの使用を意図して設計、製造されたものではありません（弊社が自動車用と指定する製品を自動車に使用する場合を除きます）。これらの用途に利用されることをご検討の際には、必ず事前に弊社営業窓口へご照会ください。なお、上記用途に使用されたことにより発生した損害等について弊社はその責任を負いかねますのでご了承願います。
8. 第7項にかかわらず、本資料に記載された製品は、下記の用途には使用しないでください。これらの用途に使用されたことにより発生した損害等につきましては、弊社は一切の責任を負いません。
 - 1) 生命維持装置。
 - 2) 人体に埋め込み使用するもの。
 - 3) 治療行為（患部切り出し、薬剤投与等）を行うもの。
 - 4) その他、直接人命に影響を与えるもの。
9. 本資料に記載された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、実装条件およびその他諸条件につきましては、弊社保証範囲内でご使用ください。弊社保証値を越えて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、弊社はその責任を負いません。
10. 弊社は製品の品質および信頼性の向上に努めておりますが、特に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。弊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、誤動作防止設計などの安全設計（含むハードウェアおよびソフトウェア）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。
11. 本資料に記載の製品は、これを搭載した製品から剥がれた場合、幼児が口に入れて誤飲する等の事故の危険性があります。お客様の製品への実装後に容易に本製品が剥がれることがなくよう、お客様の責任において十分な安全設計をお願いします。お客様の製品から剥がれた場合の事故につきましては、弊社はその責任を負いません。
12. 本資料の全部または一部を弊社の文書による事前の承諾なしに転載または複製することを固くお断りいたします。
13. 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弊社営業窓口までご照会ください。

D039444