

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事業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。

HN58X25xxx シリーズ・シリアル EEPROM

M16C クロック同期形専用シリアル I/O(SI/O)を使った制御

要旨

本資料は、ルネサス M16C ファミリのクロック同期形専用シリアル I/O を使用したルネサス HN58X25xxx シリーズのシリアル EEPROM 制御を実現するための参考資料です。

M16C MCU には、クロック同期形専用シリアル I/O が搭載されているものがあります。クロック同期形専用シリアル I/O とソフトウェアを使用することにより、HN58X25xxx シリーズのシリアル EEPROM 制御が可能です。

本資料は、クロック同期形専用シリアル I/O を用いて、HN58X25xxx シリーズのシリアル EEPROM 制御するためのサンプルプログラムを紹介致します。

動作確認デバイス

本資料で説明する応用例は、以下の MCU、条件での利用に適用されます。

- ・ MCU M16C ファミリ
- ・ 条件 クロック同期形専用シリアル I/O を使用
- ・ ソフトウェアバージョン Ver.1.01

使用しております SFR (周辺装置制御レジスタ) を持つ M16C ファミリであれば、本プログラムを使用することができます。ただし、一部の機能を機能追加等で変更している場合がありますので、MCU のマニュアルで確認してください。

本アプリケーションノートをご使用に際しては十分な評価を行ってください。

目次

1. HN58X25xxx シリーズのシリアル EEPROM 制御方法	2
2. サンプルプログラム	6

1. HN58X25xxx シリーズのシリアル EEPROM 制御方法

1.1 動作概要

M16C 内蔵のクロック同期形専用シリアル I/O を使って、HN58X25xxx シリーズのシリアル EEPROM 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 EEPROM 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、クロック同期形専用シリアル I/O (内部クロックを使用) で制御する。
クロック同期形専用シリアル 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。

使用可能なクロック周波数は、MCU とシリアル EEPROM 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。

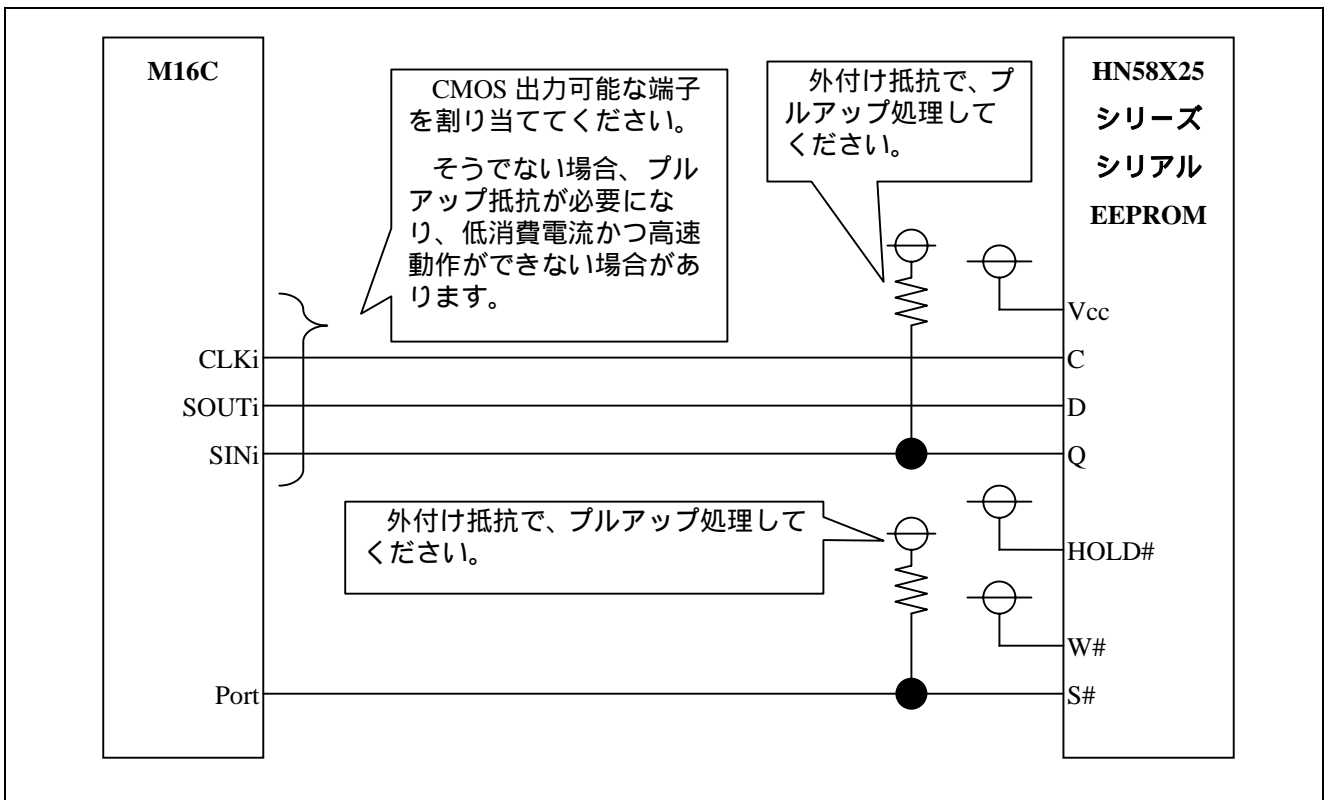


図 1-1 シリアル EEPROM の接続例

1.2 クロック同期形専用シリアル I/O で発生させるタイミング

シリアル EEPROM 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。

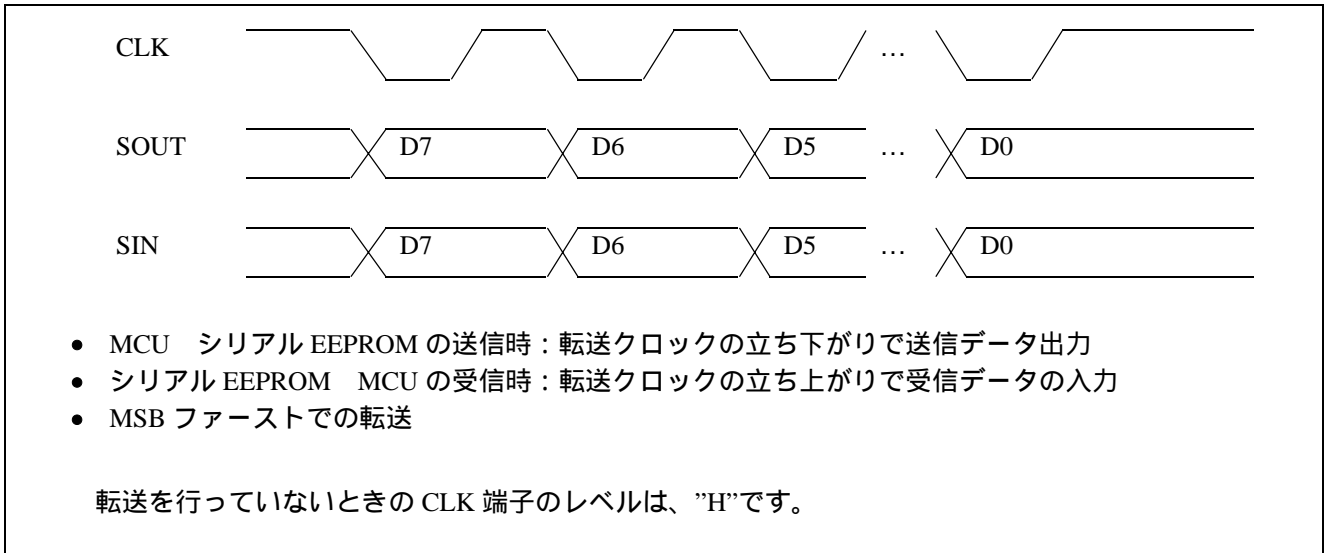


図 1-2 M16C クロック同期形専用シリアル I/O タイミング設定

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

1.3 シリアル EEPROM の S#端子制御

シリアル EEPROM 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 EEPROM の S#端子 (M16C の Port) の立ち下がりから、C 端子 (M16C の CLK) 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。

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 EEPROM 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 EEPROM 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専用シリアル I/O モードを設定します。そして、シリアル EEPROM 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 EEPROM 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/SOUT 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。

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
クロック同期形専用シリアル I/O	1ck (必須)
Port (シリアル EEPROM S#端子制御用)	1ポート (必須)

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

クロック同期形専用シリアル I/O による送受信は次の通り行われます。

- データの送信は SI/Oi 送受信レジスタに送信データを書き込む事で開始されます。
- データの受信は SI/Oi 送受信レジスタにダミーデータを書き込む事で開始されます。
- 送受信制御は、割り込み制御レジスタの割り込み要求ビットを利用しています。割り込み要求ビットの値の変化で転送完了の確認を行っています。設定は下記の通りです。
 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。
 割り込み要求ビットを "0:割り込み要求なし" に設定。

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

1.6.1 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。

SI/O3 は、送信のみのため使用しないでください

SI/O4 の S4C は PRCR レジスタの PRC2 ビットを書き込み許可にした次の命令で書きます。また、SIN4、SOUT4、CLK4 のポート制御も PRCR レジスタの PRC2 ビットを書き込み許可にした次の命令で書きます。

表 1-2 使用するクロック同期形専用シリアル I/O モードの設定値

レジスタ	ビット	機能、設定値
S4TRR	0~7	送信データを設定してください。 受信データが読めます。
S4BRG	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。
S4C	SM41 ~ SM40	S4BRG レジスタのカウントソースを選択してください。
	SM42	"0" SOUT4 出力を選択してください。
	SM43	"1" SOUT4 出力を選択してください。
	SM44	"0" (で送信データ出力 / で受信データ入力) にしてください。
	SM45	"1" (MSB ファースト) にしてください。
	SM46	"1" (内部クロック) にしてください
	SM47	"1" (High) にしてください

以下は、使用する割り込み制御レジスタの設定例です。

表 1-3 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
S4IC	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/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。

SI/O4 使用の場合、S4C は、PRCR レジスタの PRC2 ビットを書き込み許可にした次の命令で書きます。また、SIN4, SOUT4, CLK4 のポート制御も PRCR レジスタの PRC2 ビットを書き込み許可にした次の命令で書きます。

表 1-4 使用するクロック同期形専用シリアル I/O モードの設定値

レジスタ	ビット	機能、設定値
SiTRR	0~7	送信データを設定してください。 受信データが読めます。
SiBRG	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。
SiC	SMi1 ~ SMi0	SiBRG レジスタのカウントソースを選択してください。
	SMi2	“ 0 ” SOUTi 出力を選択してください。
	SMi3	“ 1 ” SOUTi 出力を選択してください。
	SMi4	“ 0 ” (で送信データ出力 / で受信データ入力) にしてください。
	SMi5	“ 1 ” (MSB ファースト) にしてください。
	SMi6	“ 1 ” (内部クロック) にしてください
	SMi7	“ 1 ” (High) にしてください

以下は、使用する割り込み制御レジスタの設定例です。

表 1-5 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
SiIC	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。

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、

- データの読み出し処理
 - 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
 - ステータス読み出し処理
- 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 EEPROM に接続される制御信号(Port/CLK/SOUT)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 EEPROM 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。

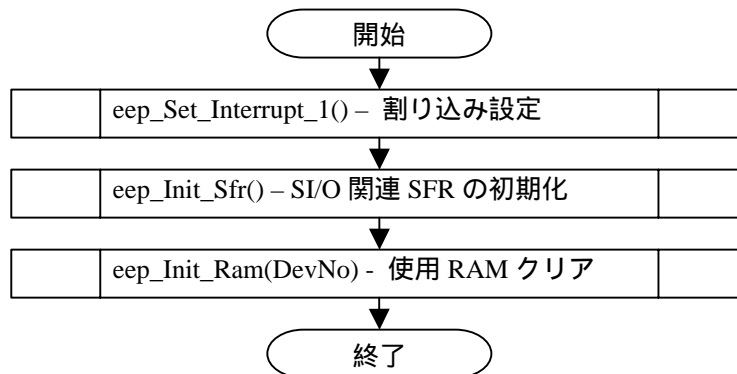
関数毎の処理を行います。

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

2.2 関数の詳細

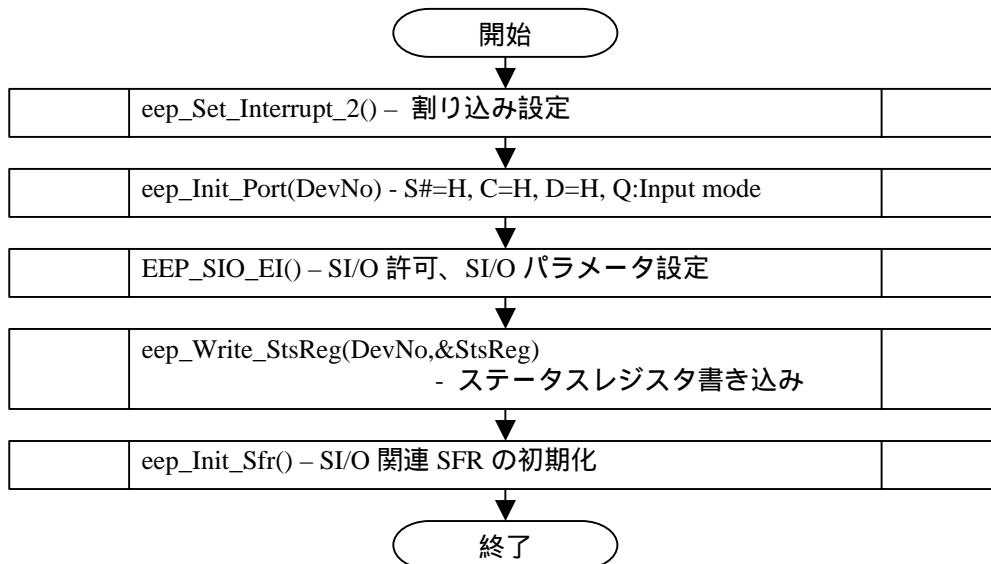
2.2.1 ドライバ初期化処理

関 数 名
EEPROM ドライバ初期化処理 void eep_Init_Driver(void)
引 数
なし
戻 り 値
なし
機 能
<ul style="list-style-type: none"> ・ EEPROM ドライバの初期化を行います。 ・ EEPROM 制御用 SFR の初期化を行います。 ・ 以下の処理をデバイス毎行います。 <ul style="list-style-type: none"> (1)EEPROM 制御用 RAM 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



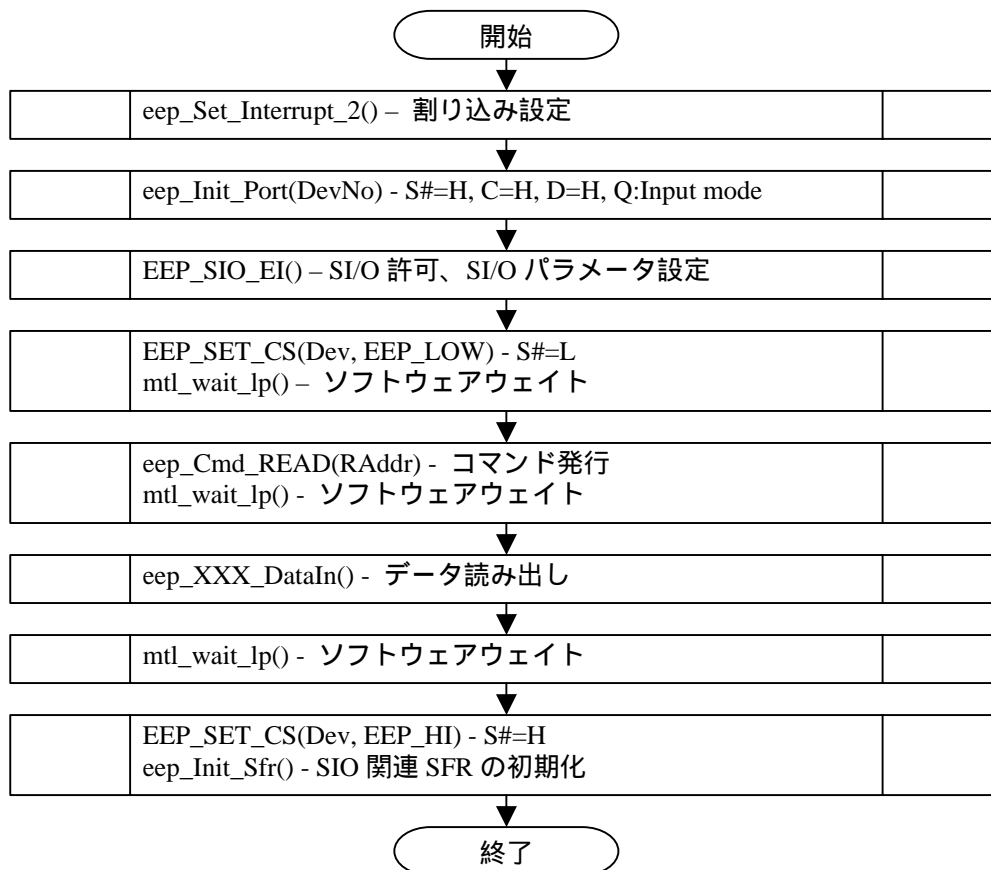
2.2.2 ライトプロテクト設定処理

関 数 名	
ライトプロテクト設定処理	
signed short eep_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。	
Eep_OK	; Successful operation
Eep_ERR_PARAM	; Parameter error
Eep_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)は下記のように設定してください。 <ul style="list-style-type: none"> Eep_WP_NONE : プロテクト解除 Eep_WP_UPPER_QUART : 上位 1/4 プロテクト設定 Eep_WP_UPPER_HALF : 上位 1/2 プロテクト設定 Eep_WP_WHOLE_MEM : 全領域プロテクト設定 	
備 考	
なし	



2.2.3 データ読出し処理

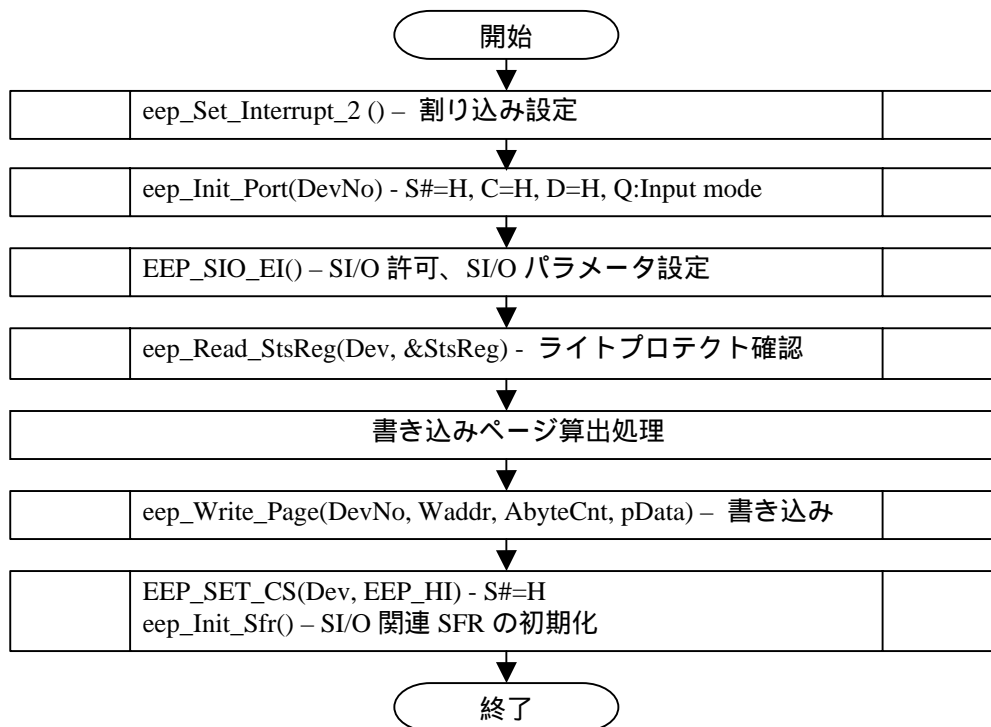
関 数 名			
データ読出し処理			
signed short eep_Read_Data(unsigned char DevNo, unsigned short RAddr, unsigned short RCnt, unsigned char * pData)			
引 数			
unsigned char	DevNo	;	デバイス番号
unsigned short	RAddr	;	読出し開始アドレス
unsigned short	RCnt	;	読出しバイト数
unsigned char FAR*	pData	;	読出しデータ格納バッファポインタ
戻り値			
読出し結果を返します。			
EEP_OK	;		Successful operation
EEP_ERR_PARAM	;		Parameter error
EEP_ERR_HARD	;		Hardware error
EEP_ERR_OTHER	;		Other error
機 能			
<ul style="list-style-type: none"> ・EEPROM 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			
・最大読出し書込みアドレスは、EEPROM 容量-1 です。			



2.2.4 データ書き込み処理

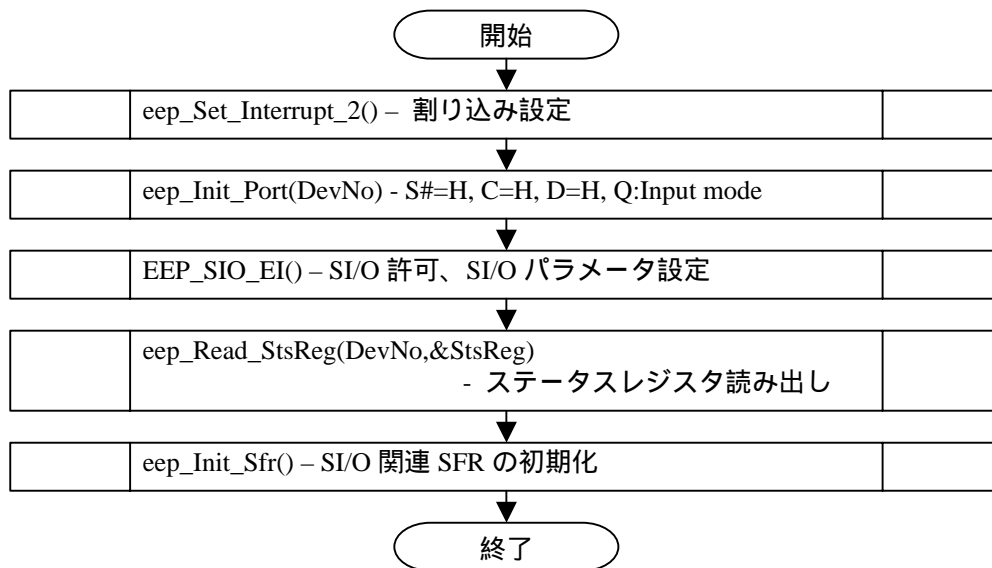
関 数 名	
データ書き込み処理	
signed short eep_Write_Data(unsigned char DevNo, unsigned short WAddr, unsigned short WCnt, unsigned char FAR* pData)	
引 数	
unsigned char	DevNo ; デバイス番号
unsigned short	WAddr ; 書き込み開始アドレス
unsigned short	WCnt ; 書き込みバイト数
unsigned char FAR*	pData ; 書き込みデータ格納バッファポインタ
戻 り 値	
書き込み結果を返します。	
EEP_OK	; Successful operation
EEP_ERR_PARAM	; Parameter error
EEP_ERR_HARD	; Hardware error
EEP_ERR_WP	; Write-protection error
EEP_ERR_OTHER	; Other error
機 能	
<ul style="list-style-type: none"> ・EEPROM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"> ・EEPROM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。 ・最大書き込みアドレスは、EEPROM容量-1です。 	

シリアルEEPROMへの書き込みは、アドレス管理を行い、ページ書き換えを使用しております。



2.2.5 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 eep_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。	
EEP_OK	; Successful operation
EEP_ERR_PARAM	; Parameter error
EEP_ERR_HARD	; Hardware error
EEP_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。 	
Memory size	512Byte
bit7-4	: Reserved (All "1")
bit3-2	: BP1/BP0 00 : None protect 01 : Upper quarter protect 10 : Upper half protect 11 : Whole memory protect
bit1	: WEL 0 : Write Disable 1 : Write Enable
bit0	: WIP 1 : 書込み動作中
Memory size	> 512Byte
bit7	: SRWD 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
bit6-4	: Reserved (All "0")
bit3-2	: BP1/BP0 00 : None protect 01 : Upper quarter protect 10 : Upper half protect 11 : Whole memory protect
bit1	: WEL 0 : Write Disable 1 : Write Enable
bit0	: WIP 1 : 書込み動作中
備 考	
なし	



2.3 戻り値定義

```
#define EEP_OK          (short)( 0)    /* Successful operation    */
#define EEP_ERR_PARAM  (short)(-1)    /* Parameter error        */
#define EEP_ERR_HARD   (short)(-2)    /* Hardware error         */
#define EEP_ERR_WP     (short)(-3)    /* Write-protection error */
#define EEP_ERR_OTHER  (short)(-4)    /* 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/29 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 eep.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、eep_io.h も含め、修正する必要があります。

```

/*-----*/
/*   EEPROM 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 EEPROM の個数に合わせて定義してください。     */
/*   Define number of required serial EEPROM devices. (1~N devices)           */
/*   Define the device number in accordance with the number of serial EEPROM devices
    */
/*   to be connected.                                                         */
/*-----*/
/* Define no. of devices */
#define EEP_DEV_NUM      1      /* 1devices                               */

/* Define no. of slots */
#define EEP_DEV0         0      /* Device 0                               */
#define EEP_DEV1         1      /* Device 1                               */

```

(2)使用するデバイスの容量の定義

使用するデバイスの容量を指定してください。

下記は、4kbit デバイスを使用する場合の例です。

```

/*-----*/
/*   使用する EEPROM を定義してください。                                     */
/*   Define the serial EEPROM device.                                         */
/*-----*/
// #define EEP_SIZE_002K      /* 2kbit (256 Byte)                       */
#define EEP_SIZE_004K      /* 4kbit (512 Byte)                       */
// #define EEP_SIZE_008K      /* 8kbit (1kByte)                         */
// #define EEP_SIZE_016K      /* 16kbit (2kByte)                        */
// #define EEP_SIZE_032K      /* 32kbit (4kByte)                        */
// #define EEP_SIZE_064K      /* 64kbit (8kByte)                        */
// #define EEP_SIZE_128K      /* 128kbit (16kByte)                      */
// #define EEP_SIZE_256K      /* 256kbit (32kByte)                      */

```


(3)SI/O 送信割り込みの設定方法の定義

SI/O 送信割り込み設定方法を定義してください。

本ソフトウェアでは、SI/O の送受信処理として、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 EEP_IC_SETTING0 を有効にしてください。

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

#define EEP_IC_SETTING1 を有効にしてください。

SI/O 送信時 (eep_Read_Data()、 eep_Write_Data()実行時) に設定する場合

#define EEP_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) SI/O 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。

```

/*-----*/
/*  SIO 送信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 SIO transmission interrupt. */
/*  "EEP_IC_SETTING1"、"EEP_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"EEP_IC_SETTING0"を選択してください。 */
/*  The setting method of the interrupt when "EEP_IC_SETTING1" and "EEP_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 "EEP_IC_SETTING0".
*/
/*-----*/
#define EEP_IC_SETTING0 /* Doesn't set in this driver */
//#define EEP_IC_SETTING1 /* When the driver is initialized, it sets */
//#define EEP_IC_SETTING2 /* When the resource is used, it sets */

```

2.4.2 eep_sfr.h

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

以下は、M16C/29 を使用する例です。

M16C/29 サンプルプログラムでは、クロック同期形専用シリアル I/O として、SI/O3 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。

SIO 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)SIO リソース

```

/*----- SIO definitions -----*/
#define EEP_SIO_STIC s3ic      /* SIO interrupt control register      */

#define EEP_SIO_BUF  s3trr     /* SIO transmit/receive buffer register */
#define EEP_SIO_BRG  s3brg     /* SIO bit rate register                */
#define EEP_SIO_SIC  s3c       /* SIO transmit/receive control register 0 */

#define EEP_SIO_NEXT ir_s3ic   /* SIO complete flag                    */

```

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

2.4.3 eep_io.h

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

以下は、M16C/29 を使用する例です。

(1)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、クロック同期形専用シリアル I/O の SIN、SOUT、CLK と S#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、eep.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 EEP_PRC2      prc2      /* Port9 write-protection register */

#define EEP_P_DATAO     p3_1      /* EEP DataOut */
#define EEP_P_DATAI     p3_2      /* EEP DataIn */
#define EEP_P_CLK       p3_0      /* EEP CLK */
#define EEP_D_DATAO     pd3_1     /* EEP DataOut */
#define EEP_D_DATAI     pd3_2     /* EEP DataIn */
#define EEP_D_CLK       pd3_0     /* EEP CLK */

#define EEP_P_CS0       p3_3      /* EEP CS0 (Negative-true logic) */
#define EEP_D_CS0       pd3_3     /* EEP CS0 (Negative-true logic) */
#if (EEP_DEV_NUM > 1)
#define EEP_P_CS1       /* EEP CS1 (Negative-true logic) */
#define EEP_D_CS1       /* EEP CS1 (Negative-true logic) */
#endif /* #if (EEP_DEV_NUM > 1) */

```

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/29 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/29 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 "sfr29.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。

```
/* MCU が、(1)SuperH でリトルエンディアン設定、(2)M16C の場合、定義を有効にしてください。*/
/* その他の MCU 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_endi.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。          */
/* mtl_endi.c の処理の高速化が可能です。                    */
/* When using M16C, define it.                                */
/* It performs the fast processes of 'mtl_endi.c'.            */
#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指定

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。

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() / memcpy() / memset() / strcat() / strcmp() / strcpy() / strlen()*/
// #define MTL_USER_LIB          /* use optimized library                    */
```

(7)使用する処理群がアクセスする RAM 領域の定義

M16C ファミリー用 FAT ファイルシステム・ライブラリ用の設定です。M16C ファミリー用 FAT ファイルシステム・ライブラリに応じて設定してください。

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。

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。

M16C/62P、M16C/29 の場合は MTL_MEM_FAR が MTL_MEM_NEAR の定義が可能です。

他のソフトウェアやシステムに応じて設定してください。

下記の例は 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、20MHz 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。

```

/* タイマのカウンタ値を定義してください。 */
/* MCU 及びクロック、ウエイトに応じて設定してください。 */
#define MTL_T_1US 1 /* loop Number of 1us */
#define MTL_T_2US 2 /* loop Number of 2us */
#define MTL_T_4US 5 /* loop Number of 4us */
#define MTL_T_5US 6 /* loop Number of 5us */
#define MTL_T_10US 13 /* loop Number of 10us */
#define MTL_T_20US 27 /* loop Number of 20us */
#define MTL_T_30US 40 /* loop Number of 30us */
#define MTL_T_50US 68 /* loop Number of 50us */
#define MTL_T_100US 137 /* loop Number of 100us */
#define MTL_T_200US 275 /* loop Number of 200us */
#define MTL_T_300US 413 /* loop Number of 300us */
#define MTL_T_400US ( MTL_T_200US * 2 ) /* loop Number of 400us */
#define MTL_T_1MS 1381 /* loop Number of 1ms */

```

2.5 使用上の注意事項

サンプルプログラムは、クロック同期形専用シリアル I/O として、

- (1)M16C/29 では SI/03 を使用する場合での記述例を示しています。
- (2)M16C/62P では SI/04 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、eep.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、以下です。

(1)eep_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)eep_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 SIO 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。

添付のものを参考に、作成してください。

また、eep_io.h と eep_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.m16c62p	共通ヘッダファイル M16C/62P
	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)
¥seep_spi	<DIR>	シリアル EEPROM ドライバのディレクトリ
	eep.h	ドライバ 共通定義
	eep_io.c	I/O モジュール
	eep_io.h.m16c29	I/O モジュール共通定義 M16C/29
	eep_io.h.m16c62p	I/O モジュール共通定義 M16C/62P
	eep_sfr.h.m16c29	SFR 共通定義 M16C/29
	eep_sfr.h.m16c62p	SFR 共通定義 M16C/62P
	eep_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

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	2007/02/26	—	初版発行
1.01	2008/02/08	-	テンプレートの変更
		1	動作確認デバイス に、ソフトウェアバージョンを追記。
		4	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。
		7	3 箇所「制御信号(Port/CLK/SOUT)を全て"H"にします。」 に変更
		8-13	関数フローから「eep_Open_Port(DevNo) - 使用ポートの Hi-z」 を削除
1.02	2008/02/13	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"H"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きよう、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