

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。

# HN58X25xxx シリーズ・シリアル EEPROM

## ルネサス SH クロック同期式 SCI を使った制御

### 要旨

本資料は、ルネサス SuperH ファミリのクロック同期式シリアルコミュニケーションインタフェース(以下、SCI と略します。)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。

SuperH ファミリには、クロック同期式 SCI が搭載されています。クロック同期式 SCI 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、クロック同期式 SCI 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            SuperH ファミリ
- ・ 条件            クロック同期式 SCI を使用

SCI を持つ SuperH 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に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.....	7

## 1. HN58X25xxx シリーズのシリアル EEPROM 制御方法

### 1.1 動作概要

SuperH 内蔵のクロック同期式 SCI を使って、HN58X25xxx シリーズのシリアル EEPROM 制御を実現します。

以下に接続方法を示します。

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 SuperH の Port に接続し、SuperH 汎用ポート出力で、制御する。
- ・データの入出力を、クロック同期式 SCI (内部クロックを使用) で、制御する。

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。

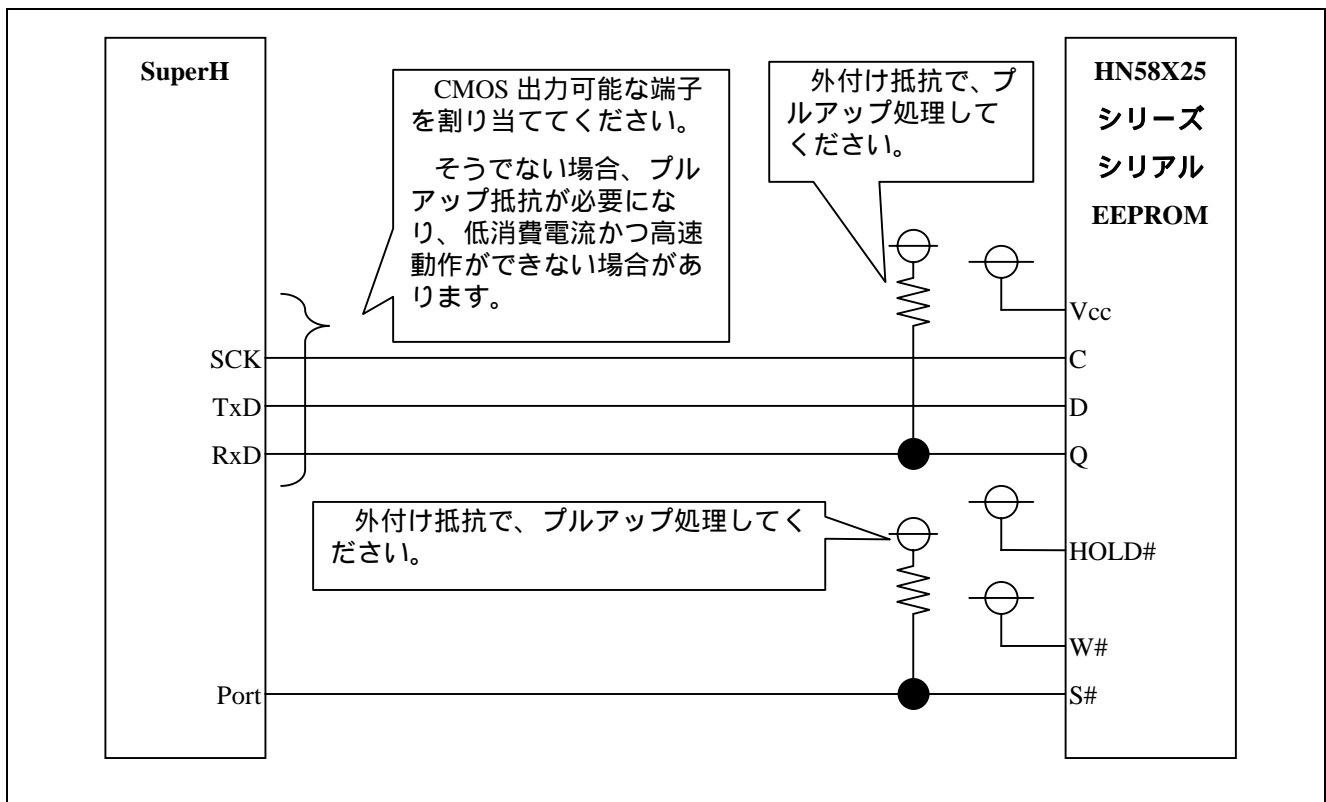


図 1-1 シリアル EEPROM の接続例

## 1.2 クロック同期式 SCI 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。

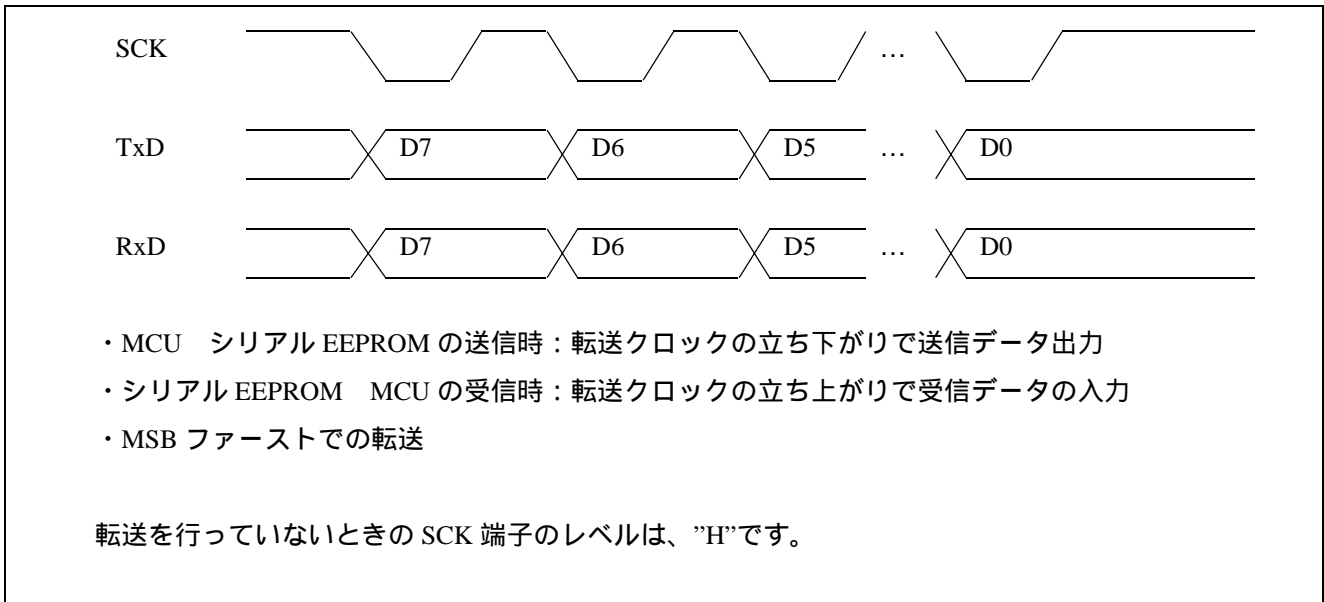


図 1-2 SH クロック同期式 SCI タイミング設定

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 SuperH の Port に接続し、SuperH 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 ( SuperH の Port ) の立ち下がりから、C 端子 ( SuperH の SCK ) 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 ( SuperH の SCK ) の立ち上がりから、S#端子 ( SuperH 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 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
クロック同期式 SCI	1ch ( 必須 )
Port ( シリアル EEPROM S#端子制御用 )	1 ポート ( 必須 )

MSB ファースト設定が可能な MCU は、MSB ファースト設定にしています。

LSB ファーストのみサポートしている MCU は、ソフトウェアで変換しています。

## 1.5 SuperH のレジスタの設定 - クロック同期式 SCI

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

FIFO を使用しない設定での設定例です。

### 1.5.1 SH7206 の場合

SH7206 グループのハードウェアマニュアル Rev.1.00 のレジスタの説明を元に、SCIF の設定例を示します。

表 1-2 使用するクロック同期式 SCI モードの設定値

レジスタ	ビット	機能、設定値
SCFRDR	7-0	受信データが読めます。
SCFTDR	7-0	送信データを設定してください。
SCSMR	15-8	リザーブビット。読み出すと、“0”が読み出されます。書き込む場合、“0”を書き込んでください。
	C/A#	“1”（クロック同期式モード）を書き込んでください。
	CHR	“0”を書き込んでください。 クロック同期式モードなので、8bit データ固定になります。
	PE	“0”を書き込んでください。 クロック同期式モードなので、パリティビットの付加およびチェックは行いません。
	O/E#	“0”を書き込んでください。 クロック同期式モードなので、無効です。
	STOP	“0”を書き込んでください。 クロック同期式モードなので、無効です。
	2	リザーブビット。読み出すと、“0”が読み出されます。書き込む場合、“0”を書き込んでください。
	CKS1-CKS0	クロックソースを選択してください。
SCSCR	15-8	リザーブビット。読み出すと、“0”が読み出されます。書き込む場合、“0”を書き込んでください。
	TIE	“0”を書き込んでいます。
	RIE	“0”を書き込んでいます。
	TE	送受信時に“1”を書き込んでください。
	RE	受信時に“1”を書き込んでください。
	REIE	“0”を書き込んでいます。
	2	リザーブビット。読み出すと、“0”が読み出されます。書き込む場合、“0”を書き込んでください。
	CKE1-CKE0	“00”（内部クロック / SCK 端子は入力端子（入力信号は無視））を書き込んでください。
SCFSR	PER3-PER0	ステータスが読めます。
	FER3-FER0	ステータスが読めます。
	ER	ステータスが読めます。初期化時に“0”を書き込んでいます。
	TEND	ステータスが読めます。初期化時に“0”を書き込んでいます。
	TDFE	ステータスが読めます。初期化時に“1”を書き込んでいます。クリア目的ではありません。
	BRK	ステータスが読めます。初期化時に“0”を書き込んでいます。

	FER	ステータスが読めます。
	PER	ステータスが読めます。
	RDF	ステータスが読めます。初期化時に "0" を書き込んでいます。
	DR	ステータスが読めます。初期化時に "0" を書き込んでいます。
SCBRR	7-0	転送速度を設定してください。
SCFCR	15-11	リザーブビット。読み出すと、"0" が読み出されます。書き込む場合、 "0" を書き込んでください。
	RSTRG2-RSTGRO	"000b" を書き込んでください。
	RTRG1-RTRGO	"00b" を書き込んでください。
	TTRG1-TTRGO	"11b" を書き込んでください。
	MCE	"0" を書き込んでください。
	TFRST	初期化時に "1" を書き込んでいます。 送受信時に "0" を書き込んでください。
	RFRST	初期化時に "1" を書き込んでいます。 受信時に "0" を書き込んでください。
	LOOP	"0" を書き込んでください。
SCFDR	15-13	リザーブビット。読み出すと、"0" が読み出されます。書き込む場合、 "0" を書き込んでください。
	T4-T0	未送信データ数を示します。
	7-5	リザーブビット。読み出すと、"0" が読み出されます。書き込む場合、 "0" を書き込んでください。
	R4-R0	受信データ数を示します。
SCSPTR	15-8	リザーブビット。読み出すと、"0" が読み出されます。書き込む場合、 "0" を書き込んでください。
	RTSIO	初期化時に "0" を書き込んでいます。
	RTSDT	初期化時に "0" を書き込んでいます。
	CTSIO	初期化時に "0" を書き込んでいます。
	CTSDT	初期化時に "0" を書き込んでいます。
	SCKIO	初期化時に "0" を書き込んでいます。
	SCKDT	初期化時に "0" を書き込んでいます。
	SPB2IO	初期化時に "0" を書き込んでいます。
SPB2DT	初期化時に "0" を書き込んでいます。	



## 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 に接続される端子は、以下のようになります。

S 端子：MCU は、入力モード。

SCK 端子：MCU は、入力モード。

TxD 端子：MCU は、入力モード。

RxD 端子：MCU は、入力モード。外部プルアップ抵抗により、"H"入力。

(2)関数のコールにより、以下の動作を行います。

各関数毎の処理を行います。

シリアル EEPROM に接続される端子は、以下のようになります。

S 端子：MCU は、入力モード。

SCK 端子：MCU は、入力モード。

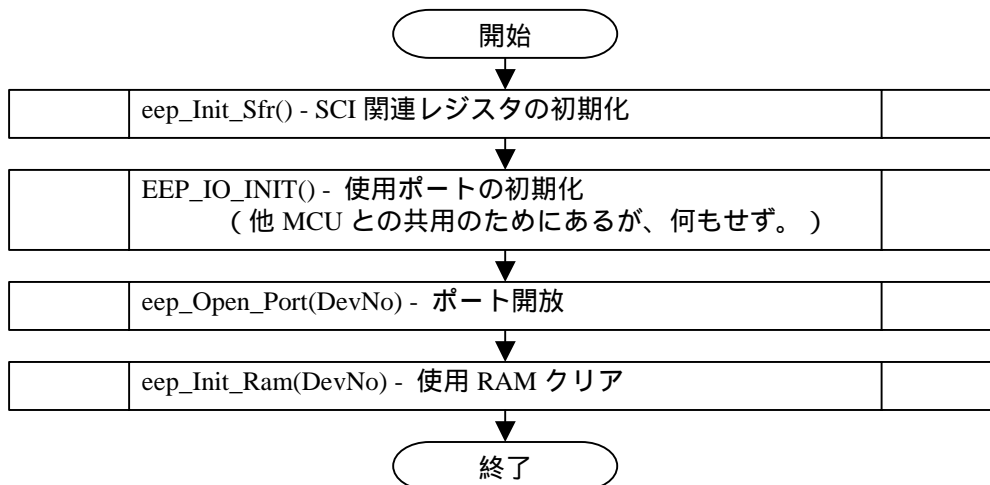
TxD 端子：MCU は、入力モード。

RxD 端子：MCU は、入力モード。外部プルアップ抵抗により、"H"入力。

## 2.2 関数の詳細

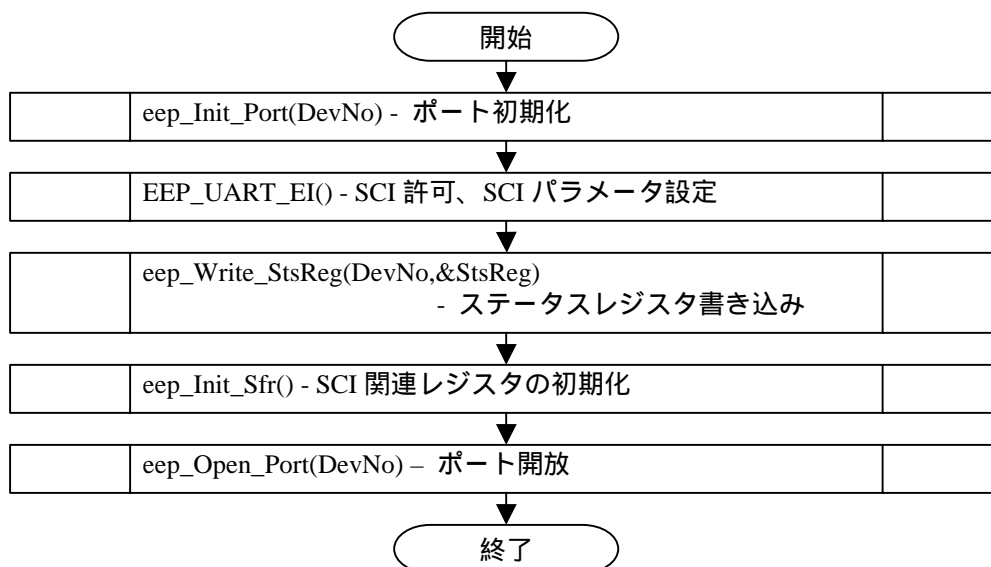
### 2.2.1 ドライバ初期化処理

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"> <li>・ EEPROM ドライバの初期化を行います。</li> <li>・ EEPROM 制御用 SFR の初期化を行います。</li> <li>・ 以下の処理をデバイス毎行います。 <ul style="list-style-type: none"> <li>(1)EEPROM 制御 port の開放を行います。</li> <li>(2)EEPROM 制御用 RAM の初期化を行います。</li> </ul> </li> <li>・ システム起動時に一度だけ呼び出してください。</li> </ul>
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	; 正常終了
EEP_ERR_PARAM	; パラメータエラー
EEP_ERR_OTHER	; その他エラー
機 能	
<ul style="list-style-type: none"> <li>・ライトプロテクトの設定を行います。</li> <li>・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"> <li>EEP_WP_NONE : プロテクト解除</li> <li>EEP_WP_UPPER_QUART : 上位 1/4 プロテクト設定</li> <li>EEP_WP_UPPER_HALF : 上位 1/2 プロテクト設定</li> <li>EEP_WP_WHOLE_MEM : 全領域プロテクト設定</li> </ul> </li> </ul>	
備 考	
なし	



### 2.2.3 データ読出し処理

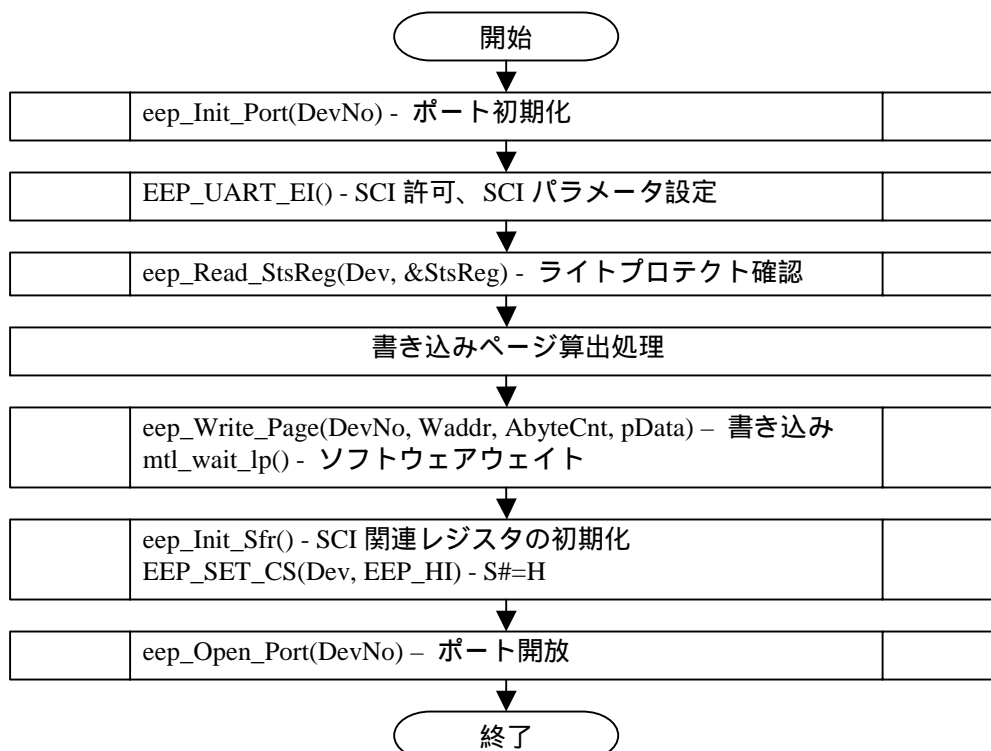
関 数 名	
データ読出し処理	
signed short eep_Read_Data(unsigned char DevNo, unsigned short RAddr, unsigned short RCnt, unsigned char * pData)	
引 数	
unsigned char	DevNo ; デバイス番号
unsigned short	RAddr ; 読出し開始アドレス
unsigned short	RCnt ; 読出しバイト数
unsigned char FAR*	pData ; 読出しデータ格納バッファポインタ
戻 り 値	
読出し結果を返します。	
EEP_OK	; 正常終了
EEP_ERR_PARAM	; パラメータエラー
EEP_ERR_HARD	; ハードエラー
EEP_ERR_OTHER	; その他エラー
機 能	
<ul style="list-style-type: none"> <li>・ EEPROM からバイト単位でデータの読出しを行います。</li> <li>・ 指定アドレスから指定バイト数分、データを読出します。</li> </ul>	
備 考	
<ul style="list-style-type: none"> <li>・ 最大書込みアドレスは、EEPROM 容量-1 です。</li> </ul>	



## 2.2.4 データ書き込み処理

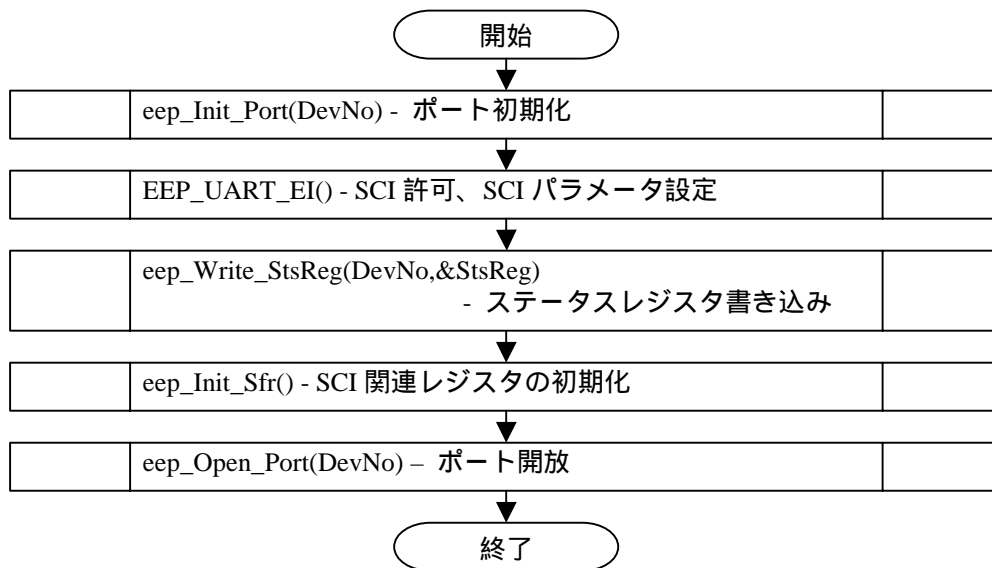
関 数 名			
データ書き込み処理			
signed short eep_Write_Data(unsigned char DevNo, unsigned short WAddr, unsigned short WCnt, unsigned char FAR* pData)			
引 数			
unsigned char	DevNo	;	デバイス番号
unsigned short	WAddr	;	書き込み開始アドレス
unsigned short	WCnt	;	書き込みバイト数
unsigned char FAR*	pData	;	書き込みデータ格納バッファポインタ
戻 り 値			
書き込み結果を返します。			
EEP_OK	;		正常終了
EEP_ERR_PARAM	;		パラメータエラー
EEP_ERR_HARD	;		ハードエラー
EEP_ERR_WP	;		ライトプロテクトエラー
EEP_ERR_OTHER	;		その他エラー
機 能			
<ul style="list-style-type: none"> <li>・EEPROMへバイト単位でデータの書き込みを行います。</li> <li>・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。</li> </ul>			
備 考			
<ul style="list-style-type: none"> <li>・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。</li> <li>・最大書き込みアドレスは、EEPROM容量-1です。</li> </ul>			

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	; 正常終了
EEP_ERR_PARAM	; パラメータエラー
EEP_ERR_HARD	; ハードエラー
EEP_ERR_OTHER	; その他エラー
機 能	
<ul style="list-style-type: none"> <li>・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。</li> <li>・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。</li> </ul>	
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	(signed short)( 0)	/* 正常終了	*/
#define EEP_ERR_PARAM	(signed short)(-1)	/* パラメータエラー	*/
#define EEP_ERR_HARD	(signed short)(-2)	/* ハードエラー	*/
#define EEP_ERR_WP	(signed short)(-3)	/* ライトプロテクトエラー	*/
#define EEP_ERR_OTHER	(signed short)(-4)	/* 上記以外のエラー	*/



## 2.4 ユーザー設定例

以下にルネサステクノロジ MCU SH7206 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、256kbit 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)                     */

```

## 2.4.2 eep\_sfr.h

### (1)ヘッダの定義

使用する MCU に対応するヘッダを指定してください。

以下は、SH7206 を使用する例です。

ヘッダが無い場合、追加し、かつ、添付のものを参照し、MCU 毎の eep\_sfr.h.xxx を作成してください。

```

//#include "Eep_sfr.h.3029"          /* EEPROM driver SFR common definitions */
//#include "Eep_sfr.h.36049"        /* EEPROM driver SFR common definitions */
//#include "Eep_sfr.h.36064"        /* EEPROM driver SFR common definitions */
//#include "Eep_sfr.h.38024"        /* EEPROM driver SFR common definitions */
//#include "Eep_sfr.h.38076"        /* EEPROM driver SFR common definitions */
//#include "Eep_sfr.h.2378"         /* EEPROM driver SFR common definitions */
//#include "Eep_sfr.h.1657"         /* EEPROM driver SFR common definitions */
//#include "Eep_sfr.h.1650"         /* EEPROM driver SFR common definitions */
//#include "Eep_sfr.h.7149"         /* EEPROM driver SFR common definitions */
#include "Eep_sfr.h.7206"           /* EEPROM driver SFR common definitions */

```

## 2.4.3 eep\_sfr.h.xxx (グループ毎に用意されるファイルです。)

サンプルプログラムでは、クロック同期式 SCI として、チャンネル 0 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。

### (1)UART リソース

```

/*----- UART definitions -----*/
#ifdef EEP_UART_USED
#define EEP_UART_MSTP  CPG.STBCR4.BIT.MSTP44  /* UART Module stop control flag*/

#define EEP_UART_SMR   SCIF3.SCSMR.WORD      /* UART Serial mode register */
#define EEP_UART_SCR   SCIF3.SCSCR.WORD      /* UART Serial control register*/
#define EEP_UART_FSR   SCIF3.SCFSR.WORD      /* UART Serial status register */
#define EEP_UART_BRR   SCIF3.SCBRR.BYTE      /* UART Bit rate register */
#define EEP_UART_FCR   SCIF3.SCFCR.WORD      /* UART FIFO control register */
#define EEP_UART_SPTR  SCIF3.SCSPTR.WORD     /* UART Serial port register */
#define EEP_UART_LSR   SCIF3.SCLSR.WORD      /* UART Line status register */
#define EEP_UART_TXBUF SCIF3.SCFSTR.BYTE     /* UART Transmit FIFO data register */
#define EEP_UART_RXBUF SCIF3.SCFRDR.BYTE     /* UART Receive FIFO data register */

#define EEP_UART_ORER  SCIF3.SCLSR.BIT.ORER   /* UART Overrun error flag */

#define EEP_UART_TXEND SCIF3.SCFSTR.BIT.TEND  /* UART Transmit end flag */
#define EEP_UART_TXNEXT SCIF3.SCFSTR.BIT.TDFE /* UART Transmit FIFO data empty*/
#define EEP_UART_RXNEXT SCIF3.SCFSTR.BIT.RDF  /* UART Receive FIFO data full*/

```

を追加変更し、それに伴う、/\* UART setting \*/の定義も、  
「SuperH のレジスタの設定 - クロック同期式 SCI」を参照し、追加変更してください。

## 2.4.4 eep\_io.h

### (1)ヘッダの定義

使用する MCU に対応するヘッダを指定してください。

以下は、SH7206 を使用する例です。

ヘッダが無い場合、追加し、かつ、添付のものを参照し、MCU 毎の eep\_io.h.xxx を作成してください。

```
//#include "Eep_io.h.3029" /* EEPROM driver I/O module common definitions */
//#include "Eep_io.h.36049" /* EEPROM driver I/O module common definitions */
//#include "Eep_io.h.36064" /* EEPROM driver I/O module common definitions */
//#include "Eep_io.h.38024" /* EEPROM driver I/O module common definitions */
//#include "Eep_io.h.38076" /* EEPROM driver I/O module common definitions */
//#include "Eep_io.h.2378" /* EEPROM driver I/O module common definitions */
//#include "Eep_io.h.1657" /* EEPROM driver I/O module common definitions */
//#include "Eep_io.h.1650" /* EEPROM driver I/O module common definitions */
//#include "Eep_io.h.7149" /* EEPROM driver I/O module common definitions */
#include "Eep_io.h.7206" /* EEPROM driver I/O module common definitions */
```

## 2.4.5 eep\_io.h.xxx (グループ毎に用意されるファイルです。)

### (1)使用するマイコンの UART 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、クロック同期式 SCI 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 EEP_OPTION_1           /* Low speed          */ /* UART          */
```

### (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、クロック同期式 SCI の RxD、TxD、SCK 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、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_P_DATAO          PORT.PEDRL.BIT.PE12DR /* EEP DataOut */
#define EEP_P_DATAI          PORT.PEDRL.BIT.PE11DR /* EEP DataIn  */
#define EEP_P_CLK            PORT.PEDRL.BIT.PE9DR   /* EEP CLK      */
#define EEP_D_DATAO          PORT.PEIORL.BIT.PE12IOR /* EEP DataOut */
#define EEP_D_DATAI          PORT.PEIORL.BIT.PE11IOR /* EEP DataIn  */
#define EEP_D_CLK            PORT.PEIORL.BIT.PE9IOR  /* EEP CLK      */
#define EEP_PCR_DATAO        PORT.PECRL4.BIT.PE12MD /* EEP DataOut */
#define EEP_PCR_DATAI        PORT.PECRL3.BIT.PE11MD /* EEP DataIn  */
#define EEP_PCR_CLK          PORT.PECRL3.BIT.PE9MD  /* EEP CLK      */

#define EEP_P_CS0             PORT.PEDRL.BIT.PE15DR /* EEP CS0 (Negative-true logic)*/
#define EEP_D_CS0             PORT.PEIORL.BIT.PE15IOR /* EEP CS0 (Negative-true logic)*/
#define EEP_PCR_CS0          PORT.PECRL4.BIT.PE15MD /* 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)*/
#define EEP_PCR_CS1          /* EEP CS1 (Negative-true logic)*/
#endif /* #if (EEP_DEV_NUM > 1) */
```

## 2.4.6 mtl\_com.h (共通ヘッダファイル)

### (1)ヘッダの定義

使用する MCU に対応するヘッダを指定してください。

以下は、SH7206 を使用する例です。

ヘッダが無い場合、追加し、かつ、添付のものを参照し、MCU 毎の mtl\_com.h.xxx を作成してください。

```
#include "mtl_com.h.7206"
```

## 2.4.7 mtl\_com.h.xxx (グループ毎に用意されるファイルです。)

SH7206 を使用した場合の設定例です。

### (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 を使用しない場合の例です。

```
/* wai_sem/sig_sem/dly_tsk を使用するため、 */ /** SET **/
/* 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。 */ /** SET **/
/* 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。 */ /** SET **/
// #define MTL_OS_USE /* OS 使用 */ /** SET **/
// #include <mr30.h> /* OS のインクルードファイル */ /** SET **/
```

### (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、SH7206 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、 */ /** SET **/
/* 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。 */ /** SET **/
#include "7206.h" /* SH7206 Include File */ /** SET **/
```

### (3)エンディアンタイプ定義

他目的の設定です。変更しないでください。

```
/* 使用する MCU のエンディアンタイプを指定してください。          */    /** SET **/
/* 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。*/    /** SET **/
// #define MTL_MCU_LITTLE          /* リトルエンディアン          */    /** SET **/
```

### (4)使用する標準ライブラリのタイプの指定

他目的の設定です。変更しないでください。

```
/* 使用する標準ライブラリのタイプを指定してください。          */    /** SET **/
/* 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、          */    /** SET **/
/* 下記デファイン定義をコメントにしてください。          */    /** SET **/
/* memcmp() / memcpy() / memset() / strcat() / strcmp() / strcpy() / strlen()
   */    /** SET **/
// #define MTL_USER_LIB          /* 最適化済みライブラリ使用          */    /** SET **/
```

### (5)使用する処理群がアクセスするRAM領域の定義

SH ファミリの場合の MTL\_MEM\_NEAR を定義してください。

```
/* 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。          */    /** SET **/
/* 標準関数や一部の処理に効率の良い処理を適用します。          */    /** SET **/
// #define MTL_MEM_FAR          /* 外部 RAM 領域にも対応          */    /** SET **/
#define MTL_MEM_NEAR          /* 内部 RAM 領域のみ対応          */    /** SET **/
```

上記のdefineのみを設定し、他は変更しないでください。

**(6)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。

システムに応じて、設定してください。

下記は、16.67MHz ノーウェイト、命令キャッシュオフ状態で動作させた場合の参考値です。

命令キャッシュ ON 状態での設定値は、変わりますので、システムに応じた値を設定してください。

```

/*-----*/
/* タイマのカウント値を定義してください。                                     */
/*   16.67MHz で算出しています。                                             */
#define MTL_T_1US          28          /* 1 μs のループ回数 */
#define MTL_T_2US          64          /* 2 μs のループ回数 */
#define MTL_T_4US          138         /* 4 μs のループ回数 */
#define MTL_T_5US          174         /* 5 μs のループ回数 */
#define MTL_T_10US         358         /* 10 μs のループ回数 */
#define MTL_T_20US         724         /* 20 μs のループ回数 */
#define MTL_T_30US         1090        /* 30 μs のループ回数 */
#define MTL_T_50US         1823        /* 50 μs のループ回数 */
#define MTL_T_100US        3655        /* 100 μs のループ回数 */
#define MTL_T_200US        7320        /* 200 μs のループ回数 */
#define MTL_T_300US        10990       /* 300 μs のループ回数 */
#define MTL_T_400US        ( MTL_T_200US * 2 ) /* 400 μs のループ回数 */
#define MTL_T_1MS          36650       /* 1ms のループ回数 */
// #define MTL_T_2MS        ( MTL_T_1MS * 2 ) /* 2ms のループ回数 */
// #define MTL_T_5MS        ( MTL_T_1MS * 5 ) /* 5ms のループ回数

```

## 2.5 使用上の注意事項

サンプルプログラムでは、クロック同期式 SCI 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 SuperH ファミリ MCU を使用する場合

他 SuperH 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)`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 UART 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、`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	共通関数の各種定義	
	mtl_com.h.1650		共通ヘッダファイル	
	mtl_com.h.1657		共通ヘッダファイル	
	mtl_com.h.2378		共通ヘッダファイル	
	mtl_com.h.3029		共通ヘッダファイル	
	mtl_com.h.36049		共通ヘッダファイル	
	mtl_com.h.38024		共通ヘッダファイル	
	mtl_com.h.38076		共通ヘッダファイル	
	mtl_com.h.7149		共通ヘッダファイル	
	mtl_com.h.7206		共通ヘッダファイル	
¥drv	<DIR>	サンプル・デバイスドライバのディレクトリ		
	¥seep_spi <DIR>	シリアル EEPROM のディレクトリ		
	eep.h		ドライバ 共通定義	
	eep_usr.c		ドライバ ユーザ I/F モジュール	
	eep_io.c	eep_io.h	I/O モジュール	
	eep_io.h.1650		I/O モジュール共通定義	
	eep_io.h.1657		I/O モジュール共通定義	
	eep_io.h.2378		I/O モジュール共通定義	
	eep_io.h.3029		I/O モジュール共通定義	
	eep_io.h.36049		I/O モジュール共通定義	
	eep_io.h.38024		I/O モジュール共通定義	
	eep_io.h.38076		I/O モジュール共通定義	
	eep_io.h.7149		I/O モジュール共通定義	
	eep_io.h.7206		I/O モジュール共通定義	
	eep_sfr.h		SFR 共通定義	
	eep_sfr.h.1650		SFR 共通定義	
	eep_sfr.h.1657		SFR 共通定義	
	eep_sfr.h.2378		SFR 共通定義	
	eep_sfr.h.3029		SFR 共通定義	
	eep_sfr.h.36049		SFR 共通定義	
	eep_sfr.h.38024		SFR 共通定義	
	eep_sfr.h.38076		SFR 共通定義	
	eep_sfr.h.7149		SFR 共通定義	
	eep_sfr.h.7206		SFR 共通定義	
	¥sample	<DIR>	サンプル・プログラムのディレクトリ	
		usr_tst tsk.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。
		tsk.c	tsk.h	共通関数の各種定義
	common.c	common.h	共通関数の各種定義	

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2005/10/18	—	初版発行

### 安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

### 本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジ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ルネサス テクノロジが所有する知的財産権その他の権利の実施、使用を許諾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の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジは責任を負いません。
3. 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全ての情報は本資料発行時点のものであり、ルネサス テクノロジは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジ半導体製品のご購入に当たりましては、事前にルネサス テクノロジ、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジホームページ(<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。
4. 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誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位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。ルネサス テクノロジは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジ、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気付きの点がございましたらルネサス テクノロジ、ルネサス販売または特約店までご照会ください。