

## R32C/100 シリーズ

クロック同期型シリアルインタフェースモードを使用したEEPROM制御

R01AN1198JJ0100

Rev.1.00

2012.12.14

## 要旨

本アプリケーションノートでは、ルネサスエレクトロニクス製 R1EX25032ASA00A を例にクロック同期型シリアルインタフェースモードを使用したシリアルEEPROM制御について説明します。

R32C/118 グループではクロック同期型シリアルインタフェースモードが使用できるチャンネル数は9チャンネル(UART0~UART8)あります。本アプリケーションノートではUART2を使用しています。UART2以外のチャンネルを使用する場合は、ユーザズマニュアルハードウェア編を参照して、UARTi(i=0~8)関連レジスタを変更してください。

## 対象デバイス

R32C/116 グループ

R32C/117 グループ

R32C/118 グループ

本アプリケーションノートを他のマイコンへ適用する場合、そのマイコンの仕様にあわせて変更し、十分評価してください。

## 目次

1.	仕様	3
2.	動作確認条件	4
3.	関連アプリケーションノート	4
4.	ハードウェア説明	5
4.1	使用端子一覧	5
5.	ソフトウェア説明	6
5.1	動作概要	8
5.1.1	シリアル送受信の単位	8
5.1.2	コマンドシーケンス	8
5.1.3	各処理の実行手順	8
5.2	定数一覧	10
5.3	定義/列挙型一覧	11
5.4	変数一覧	12
5.5	関数一覧	13
5.6	関数仕様	14
5.7	フローチャート	19
5.7.1	メイン処理	19
5.7.2	EEPROM初期化シーケンス開始	20
5.7.3	EEPROMライトシーケンス開始	21
5.7.4	EEPROMリードシーケンス開始	22
5.7.5	コマンド移行判定用変数クリア	22
5.7.6	コマンド実行開始処理	23
5.7.7	コマンド実行終了処理	24
5.7.8	コマンド移行処理	25
5.7.9	EEPROM S端子制御	27
5.7.10	UART2シリアル送信(UART2送信割り込み)	27
5.7.11	UART2シリアル受信(UART2受信割り込み)	28
5.7.12	5msタイムアウト処理(タイマA0割り込み)	28
5.7.13	EEPROMライト要求受け付け(INT0割り込み)	29
5.7.14	EEPROMリード要求受け付け(INT1割り込み)	29
5.7.15	タイマA0初期化	30
5.7.16	UART2初期化	31
6.	サンプルコード	32
7.	参考ドキュメント	32

## 1. 仕様

クロック同期型シリアルインタフェースモード(UART2)を使用して、EEPROMを制御します。EEPROMのステータスレジスタを初期化(EEPROM初期化処理)した後、要求に応じてEEPROMへのデータのライト(EEPROMライト処理)、またはEEPROMからデータのリード(EEPROMリード処理)を行います。

EEPROMはルネサスエレクトロニクス製のR1EX25xxxシリーズを使用します。EEPROMの詳細はデータシートを参照してください。

<使用条件>

- ・ビットレート：1Mbps
- ・転送データ長：1~32バイト(命令コード、メモリアドレスを含みません。)

表 1.1 に使用する周辺機能と用途を、図 1.1 に接続図を示します。

表 1.1 使用する周辺機能と用途

周辺機能	用途
シリアルインタフェース(UART2)	EEPROMとの通信
INT0割り込み	EEPROMライト処理を実行
INT1割り込み	EEPROMリード処理を実行
タイマA0	タイムアウト検出用(5ms) (EEPROMおよびEEPROMのステータスレジスタへのライトにかかる最大時間)

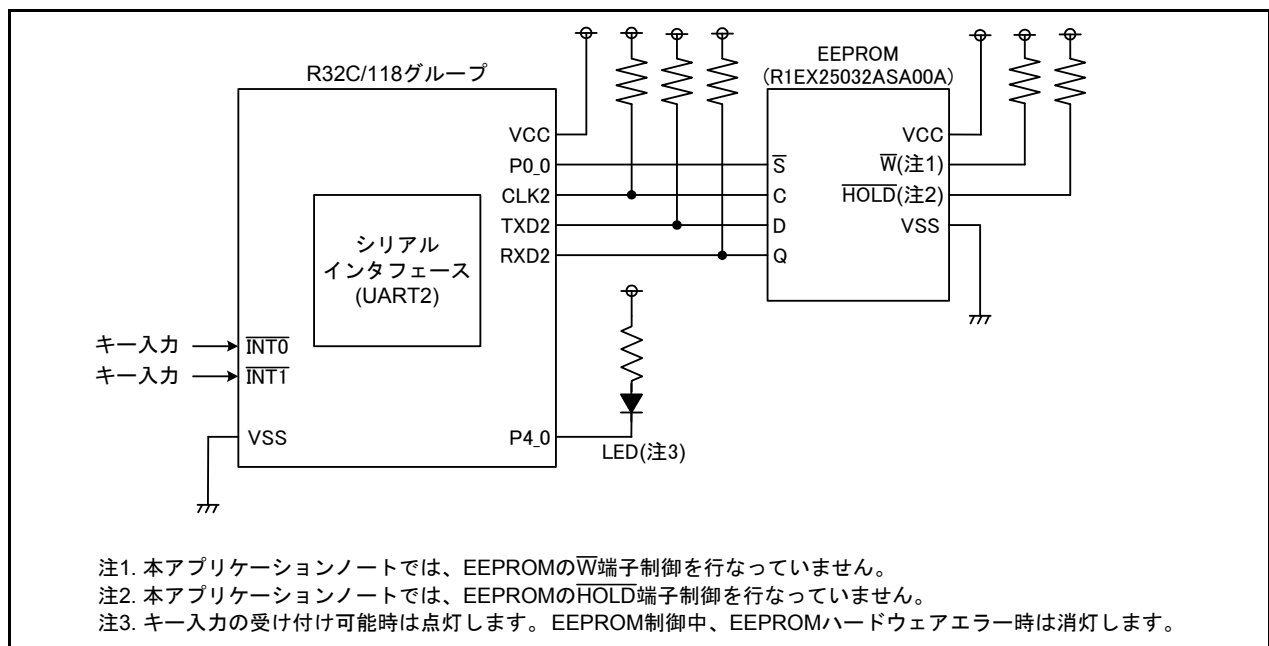


図 1.1 接続図

## 2. 動作確認条件

本アプリケーションノートのサンプルコードは、下記の条件で動作を確認しています。

表 2.1 動作確認条件

項目	内容
使用マイコン	R5F64189DFD(R32C/118グループ)
使用デバイス(EEPROM)	R1EX25032ASA00A
動作周波数	<ul style="list-style-type: none"> <li>・ XINクロック : 16MHz</li> <li>・ PLLクロック : 100MHz</li> <li>・ ベースクロック : 50MHz</li> <li>・ CPUクロック : 50MHz</li> <li>・ 周辺バスクロック : 25MHz</li> <li>・ 周辺機能クロック : 25MHz</li> </ul>
動作電圧	5V
統合開発環境	ルネサスエレクトロニクス製 High-performance Embedded Workshop Version 4.08
Cコンパイラ	ルネサスエレクトロニクス製 R32C/100 Series C Compiler V.1.02 Release 01 コンパイルオプション -D__STACKSIZE__=0X300 -D__ISTACKSIZE__=0X300 -DVECTOR_ADR=0x0FFFFFFBDC -c -finfo -dir "\$(CONFIGDIR)" (統合開発環境のデフォルト設定を使用しています。)
動作モード	シングルチップモード
サンプルコードのバージョン	1.00
使用ボード	Renesas Starter Kit for R32C/118 (製品型名 : R0K564189S000BE)

## 3. 関連アプリケーションノート

本アプリケーションノートに関連するアプリケーションノートを以下に示します。併せて参照してください。

- ・ R32C/100シリーズメインクロック通倍モード設定手順(RJJ05B1259JJ)
- ・ R32C/100シリーズシリアルインタフェース動作(特殊モード2のマスタ送受信)(R01AN0493JJ)
- ・ R32C/100シリーズシリアルインタフェース動作(クロック同期型シリアルインタフェースモードの送信)(RJJ05B1377)
- ・ R32C/100シリーズシリアルインタフェース動作(クロック同期型シリアルインタフェースモードの受信)(R01AN0178JJ)

## 4. ハードウェア説明

### 4.1 使用端子一覧

表 4.1に使用端子と機能を示します。

表 4.1 使用端子と機能

端子名	入出力	内容
P0_0	出力	EEPROMのS端子(チップセレクト端子)制御用出力
P7_0/TXD2	出力	EEPROMへのシリアルデータ出力
P7_1/RXD2	入力	EEPROMからのシリアルデータ入力
P7_2/CLK2	出力	EEPROMとのシリアルデータの入出力タイミングを設定するためのクロック出力
P4_0	出力	EEPROM制御中の確認LED出力
P8_2/INT0	入力	EEPROMライト処理を実行
P8_3/INT1	入力	EEPROMリード処理を実行

## 5. ソフトウェア説明

本アプリケーションノートのサンプルコードは、リセットスタート時にEEPROM初期化処理を行い、 $\overline{\text{INT0}}$  信号の立ち下がりエッジを検出するとEEPROMに32バイトライトを行い(EEPROMライト処理)(注1)、 $\overline{\text{INT1}}$  信号の立ち下がりエッジを検出するとEEPROMから32バイトリードを行います(EEPROMリード処理)(注2)。

- 注1. 本アプリケーションノートのサンプルコードを応用し EEPROM に任意のデータをライトする場合は、ライトする EEPROM アドレスがページの境界を跨がないように EEPROM アドレスとライトするデータのサイズを指定してください。
- 注2. 本アプリケーションノートのサンプルコードを応用し EEPROM から任意のデータをリードする場合は、リードする EEPROM アドレスが最終アドレスより大きくならないように EEPROM アドレスとリードするデータのサイズを指定してください。

表 5.1 に EEPROM 制御のための命令一覧を示します。詳細は R1EX25xxx シリーズのデータシートを参照してください。

表 5.1 EEPROM制御のための命令一覧

命令	内容	命令コード
WREN	ライト許可	0000 0110
WRDI(注1)	ライト禁止	0000 0100
RDSR	EEPROMのステータスレジスタのリード	0000 0101
WRSR	EEPROMのステータスレジスタのライト	0000 0001
READ	データのリード	0000 0011
WRITE	データのライト	0000 0010

注1. 本アプリケーションノートでは使用していません。

図 5.1 に EEPROM のステータスレジスタ構成を示します。表 5.1 に示す RDSR 命令によりリードでき、WRSR 命令によりライトできます。詳細は R1EX25xxx シリーズのデータシートを参照してください。

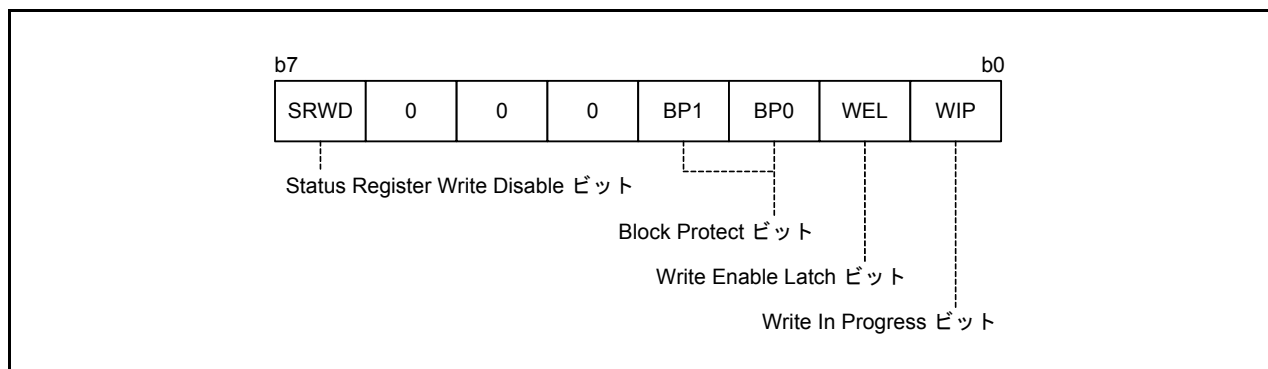


図 5.1 EEPROMのステータスレジスタ構成

表 5.2にEEPROM制御に使用するUART2の設定を示します。

表 5.2 EEPROM制御に使用するUART2の設定

モード	クロック同期型シリアルインタフェースモード
送受信クロック	内部クロック
U2BRGカウンタソース	f1
TXD2、CLK2端子出力形式	プッシュプル出力
ビットオーダ	MSBファースト
CTS機能	禁止
CLK極性	送受信クロックの立ち下がりに同期して送信データ出力、立ち上がりに同期して受信データ入力
データ論理	反転なし
UART2送信割り込み	使用(割り込み要因 : U2TBレジスタ空)
UART2受信割り込み	使用
ビットレート	1Mbps

また、ビットレートの計算式は以下のようになります。

<ビットレートの計算式>

$$\begin{aligned}
 \text{ビットレート} &= \text{U2BRGカウンタソース} \div (2 \times (\text{U2BRGレジスタの設定値} + 1)) \\
 &= 25\text{MHz}(f1) \div (2 \times (11 + 1)) \\
 &\doteq 1\text{Mbps}
 \end{aligned}$$

## 5.1 動作概要

### 5.1.1 シリアル送受信の単位

本アプリケーションノートでは、EEPROM 命令(WREN, WRSR, RDSR, WRITE, READ)単位で行う“ポート操作およびシリアル送受信”をシリアル送受信の最小単位とし、この最小単位を“コマンド”と呼びます。

コマンドでは以下の処理を行います。

- (1)  $\bar{S}$ 端子に接続されたポートP0\_0を“L”にする。
- (2) EEPROM 命令の送信
- (3) EEPROM 命令のパラメータ送受信 (必要時)
- (4) ポートP0\_0を“H”にする。

### 5.1.2 コマンドシーケンス

EEPROM初期化処理、EEPROMライト処理、EEPROMリード処理では、以下のようにコマンドがそれぞれ決められた順序で実行されます。本アプリケーションノートでは、このコマンド実行順序を“コマンドシーケンス”と呼びます。

EEPROM初期化シーケンス：

WREN コマンド → RDSR コマンド → WRSR コマンド → RDSR コマンド

EEPROMライトシーケンス：

WREN コマンド → RDSR コマンド → WRITE コマンド → RDSR コマンド

EEPROMリードシーケンス：

READ コマンド

### 5.1.3 各処理の実行手順

本アプリケーションノートでは、シリアルインタフェースのクロック同期型シリアルインタフェースモードを使用し、EEPROMとの通信を半二重で行います。

#### (1) コマンドシーケンスの開始

リセットスタート後はEEPROMの初期化シーケンス開始、INT0割り込みではEEPROMライトシーケンス開始、INT1割り込みではEEPROMリードシーケンス開始の関数が呼び出されます。

各関数では、以下の処理が実行されます。

- コマンド移行判定に使用される変数のクリア
- コマンドシーケンスの設定
- コマンド実行状態フラグ(proc\_state)を“コマンド実行開始(STATE\_BEGIN)”に設定
- EEPROMライト処理、EEPROMリード処理では送信するデータを送信データ配列(txdata\_rw[])に格納

#### (2) コマンド実行開始処理

コマンド実行状態フラグが“コマンド実行開始(STATE\_BEGIN)”になると以下の処理が実行されます。

- 送信割り込みおよび受信割り込み内で使用する変数の設定
- ポートP0\_0から“L”を出力
- コマンドがRDSR[WIPビットの確認]の場合はタイムアウト検出用タイマを開始
- コマンド実行状態フラグを“コマンド実行中(STATE\_EXECUTE)”に設定
- UART2送受信許可
- EEPROM命令送信



- (3) 送信データの送信  
U2TBレジスタの内容が送信シフトレジスタに転送されると送信割り込みが発生します。  
送信割り込み処理ルーチンでは、送信データ配列(コマンドにより `txdata_rw[]`, `txdata_wren[]`, `txdata_rdsr[]`, `txdata_wrsr[]` のいずれか)に格納された送信データを1バイトU2TBレジスタに書き込みます。これを送信割り込みが発生するたびに送信データが無くなるまで繰り返します。
- (4) 受信データの格納  
データを受信すると受信割り込みが発生します。  
受信割り込み処理ルーチンでは、U2RBレジスタをリードし受信データ格納バッファ(`rxdata_buf[]`)に格納します。  
全データを受信すると送受信を禁止し、コマンド実行状態フラグを“コマンド実行終了 (STATE\_END)”にします。
- (5) コマンド実行終了処理  
コマンド実行状態フラグが“コマンド実行終了 (STATE\_END)”になると以下の処理が実行されます。  
  - ポート P0\_0 から“H”を出力  
<コマンドが RDSR の場合>
  - EEPROMステータス変数(`rdsr_val`)にEEPROMのステータスレジスタの値を格納  
<コマンドが READ の場合>
  - 指定領域にリードしたEEPROMデータを格納
- (6) コマンド移行処理  
(5)の後、以下の処理が実行されます。  
コマンド移行判定を行い、`p_cmd_type`をインクリメントし、コマンドシーケンスを進めます。また、次のコマンドを開始するためにコマンド実行状態フラグを“コマンド実行開始 (STATE\_BEGIN)”にします。

以降、上記(2)～(6)の処理を繰り返し、全てのコマンドが終了したらコマンド実行状態フラグを“コマンド実行なし (STATE\_NON)”にし、コマンドシーケンスを終了します。

図 5.2 にタイミング図を示します。

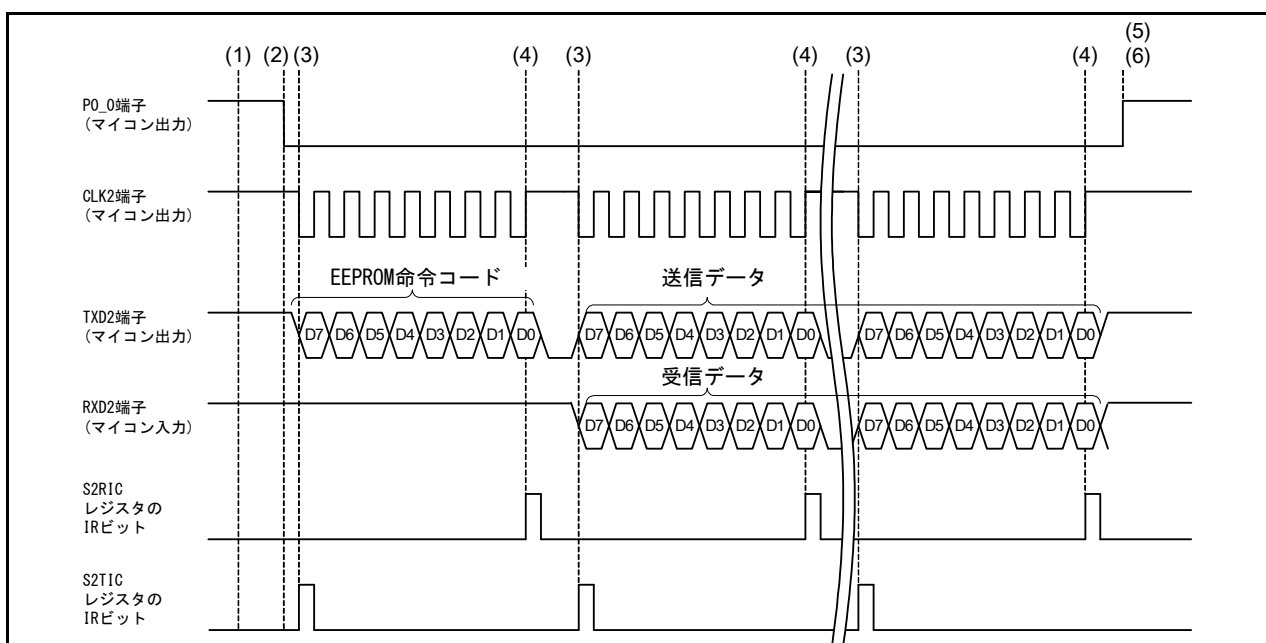


図 5.2 タイミング図

## 5.2 定数一覧

表 5.3 にサンプルコードで使用する定数を示します。

表 5.3 サンプルコードで使用する定数

定数名	設定値	内容
EEP_T_S_HOLD	0x000A	S端子のセットアップ/ホールド待ち時間
EEP_REG_INI	0x00	EEPROMのステータスレジスタの初期設定値
EEP_REG_WEL	0x02	EEPROMのステータスレジスタのWELビット
EEP_REG_WIP	0x01	EEPROMのステータスレジスタのWIPビット
EEPROM_MEM_ADDR	0x0000	WRITEあるいはREAD時のEEPROM上のアドレス
BUFSIZE	32	送受信するデータ本体のサイズ
TXSIZE	(BUFSIZE+3)	最大送信データ数 (+3はEEPROM命令コードとEEPROMアドレス分)
DUMMY	0xff	データ受信処理で送信するダミーデータ
UART_BRG	(12 - 1)	UARTボーレート設定値

### 5.3 定義/列挙型一覧

図 5.3 にサンプルコードで使用する定義/列挙型を示します。

```

/* **** ポートの定義 **** */
#define EEP_P_S      p0_0          /* EEPROM S端子制御用ポートレジスタ */
#define EEP_D_S      pd0_0        /* EEPROM S端子制御用ポート方向レジスタ */
#define EEP_P_LED    p4_0          /* LED用ポートレジスタ */
#define EEP_D_LED    pd4_0        /* LED用ポート方向レジスタ */

/* **** コマンドの定義 **** */
typedef enum
{
    CMD_WREN,                /* WREN */
    CMD_RDSR_WEL,           /* RDSR[WELビットの確認] */
    CMD_RDSR_WIP,           /* RDSR[WIPビットの確認] */
    CMD_WRSR,               /* WRSR */
    CMD_READ,               /* READ */
    CMD_WRITE,              /* WRITE */
    CMD_TERMINATE,          /* 全コマンド終了 */
} eeprom_command_t;

/* **** コマンド実行状態の定義 **** */
typedef enum
{
    STATE_NON,              /* コマンド実行なし */
    STATE_BEGIN,            /* コマンド実行開始 */
    STATE_EXECUTE,          /* コマンド実行中 */
    STATE_END,              /* コマンド実行終了 */
    STATE_ERROR,            /* コマンド実行エラー発生 */
} command_process_state_t;

/* **** 割り込み要求レベルの定義 **** */
typedef enum
{
    INT_LEVEL_DISABLE = 0,   /* 割り込み禁止 */
    INT_LEVEL_1      = 1,   /* レベル1 */
    INT_LEVEL_2      = 2,   /* レベル2 */
    INT_LEVEL_3      = 3,   /* レベル3 */
    INT_LEVEL_4      = 4,   /* レベル4 */
    INT_LEVEL_5      = 5,   /* レベル5 */
    INT_LEVEL_6      = 6,   /* レベル6 */
    INT_LEVEL_7      = 7,   /* レベル7 */
} interrupt_level_t;

/* **** ポート状態の定義 **** */
typedef enum
{
    LEVEL_LOW,              /* "L"レベル */
    LEVEL_HIGH              /* "H"レベル */
} logic_level_t;

```

図 5.3 サンプルコードで使用する定義/列挙型

## 5.4 変数一覧

表 5.4 にグローバル変数を、表 5.5 に const 型変数を示します。

表 5.4 グローバル変数

型	変数名	内容	使用関数
uint8_t	rdsr_val	EEPROMステータス変数	variable_clear, cmd_proc_end, cmd_proc_change
uint8_t	txdata_rw[]	WRITEあるいはREAD時の送信データ配列	R_EEP_StartWriteSeq, R_EEP_StartReadSeq (注1)
uint8_t	rxdata_buf[]	受信データ格納バッファ	cmd_proc_begin, cmd_proc_end
uint8_t	comm_data_size_array[]	各コマンドの送信データサイズ	R_EEP_StartWriteSeq, R_EEP_StartReadSeq, cmd_proc_begin, cmd_proc_end
uint8_t const *	p_cmd_type	コマンドへのポインタ	R_EEP_StartInitSeq, R_EEP_StartWriteSeq, R_EEP_StartReadSeq, cmd_proc_begin, cmd_proc_end, cmd_proc_change
uint8_t	proc_state	コマンド実行状態フラグ	main, R_EEP_StartInitSeq, R_EEP_StartWriteSeq, R_EEP_StartReadSeq, cmd_proc_begin, cmd_proc_change, uart2_rx_interrupt, int0_interrupt, int1_interrupt
bool	f_timeout	タイムアウトフラグ	variable_clear, cmd_proc_begin, cmd_proc_change, timer_a0_interrupt
uint8_t *	p_txdata	送信データへのポインタ	cmd_proc_begin, uart2_tx_interrupt
uint8_t	remaining_tx_size	未送信データサイズ	cmd_proc_begin, uart2_tx_interrupt
uint8_t *	p_rxdata_buf	受信データ格納バッファへのポインタ	cmd_proc_begin, uart2_rx_interrupt
uint8_t	remaining_rx_size	未受信データサイズ	cmd_proc_begin, uart2_rx_interrupt
uint8_t *	p_eep_read_data	リードしたEEPROMデータを格納する領域へのポインタ	R_EEP_StartReadSeq, cmd_proc_end
uint8_t	write_buffer[BUFSIZE]	EEPROMライト用のデータ領域	int0_interrupt, main
uint8_t	read_buffer[BUFSIZE]	リードしたEEPROMデータを格納する領域	int1_interrupt

注1. p\_tx\_data\_array[] を構成する要素として使用される。

表 5.5 const型変数

型	変数名	内容	使用関数
const uint8_t	cmd_seq_init[]	EEPROM初期化処理のコマンドシーケンス格納用配列	R_EEP_StartInitSeq
const uint8_t	cmd_seq_write[]	EEPROMライト処理のコマンドシーケンス格納用配列	R_EEP_StartWriteSeq
const uint8_t	cmd_seq_read[]	EEPROMリード処理のコマンドシーケンス格納用配列	R_EEP_StartReadSeq
const uint8_t	txdata_wren[]	WREN時の送信データ配列	(注1)
const uint8_t	txdata_rdsr[]	RDSR時の送信データ配列	(注1)
const uint8_t	txdata_wrsr[]	WRSR時の送信データ配列	(注1)
const uint8_t * const	p_tx_data_array[]	各コマンドに対応した送信データ配列へのポインタ	cmd_proc_begin

注1. p\_tx\_data\_array[] を構成する要素として使用される。

## 5.5 関数一覧

表 5.6に関数を示します。

表 5.6 関数

関数名	概要
main	メイン処理
R_EEP_StartInitSeq	EEPROM初期化シーケンス開始
R_EEP_StartWriteSeq	EEPROMライトシーケンス開始
R_EEP_StartReadSeq	EEPROMリードシーケンス開始
variable_clear	コマンド移行判定用変数クリア
cmd_proc_begin	コマンド実行開始処理
cmd_proc_end	コマンド実行終了処理
cmd_proc_change	コマンド移行処理
eep_s_control	EEPROM S端子制御
uart2_tx_interrupt	UART2シリアル送信(UART2送信割り込み)
uart2_rx_interrupt	UART2シリアル受信(UART2受信割り込み)
timer_a0_interrupt	5msタイムアウト処理(タイマA0割り込み)
int0_interrupt	EEPROMライト要求受け付け(INT0割り込み)
int1_interrupt	EEPROMリード要求受け付け(INT1割り込み)
timer_a0_init	タイマA0初期化
uart2_init	UART2初期化

## 5.6 関数仕様

サンプルコードの関数仕様を示します。

---

### main

---

概要	メイン処理
ヘッダ	なし
宣言	void main(void)
説明	システムクロックとUART2とタイマA0の初期設定および変数の初期化を行います。 その後、EEPROM初期化シーケンスを開始し、メインループに入ります。 メインループ内では、コマンド実行状態フラグに応じてコマンド実行開始処理、コマンド実行終了処理、コマンド移行処理を行います。
引数	なし
リターン値	なし

---

### R\_EEP\_StartInitSeq

---

概要	EEPROM初期化シーケンス開始
ヘッダ	なし
宣言	void R_EEP_StartInitSeq(void)
説明	コマンドシーケンスを“EEPROM初期化シーケンス”に設定し、コマンド実行状態フラグを“コマンド実行開始 (STATE_BEGIN)”にします。
引数	なし
リターン値	なし

---

### R\_EEP\_StartWriteSeq

---

概要	EEPROMライトシーケンス開始
ヘッダ	なし
宣言	void R_EEP_StartWriteSeq(uint8_t *p_write_data, uint8_t size, uint16_t eeprom_adrs)
説明	WRITEコマンドの送信データサイズの設定と、シリアルデータの生成を行います。 その後、コマンドシーケンスを“EEPROMライトシーケンス”に設定し、コマンド実行状態フラグを“コマンド実行開始 (STATE_BEGIN)”にします。
引数	uint8_t *p_write_data : ライトデータが格納されている領域へのポインタ uint8_t size : ライトするデータのサイズ[バイト] uint16_t eeprom_adrs : EEPROMアドレス
リターン値	なし

---

### R\_EEP\_StartReadSeq

---

概要	EEPROMリードシーケンス開始
ヘッダ	なし
宣言	void R_EEP_StartReadSeq(uint8_t *p_read_data, uint8_t size, uint16_t eeprom_adrs)
説明	READコマンドの送信データサイズの設定と、シリアルデータの生成を行います。 また、リードしたEEPROMデータを格納する領域へのポインタ (p_eeep_read_data) を設定します。 その後、コマンドシーケンスを“EEPROMリードシーケンス”に設定し、コマンド実行状態フラグを“コマンド実行開始 (STATE_BEGIN)”にします。
引数	uint8_t *p_read_data : リードデータを格納する領域へのポインタ uint8_t size : リードするデータのサイズ[バイト] uint16_t eeprom_adrs : EEPROMアドレス
リターン値	なし

---

**variable\_clear**

---

概要	コマンド移行判定用変数クリア
ヘッダ	なし
宣言	void variable_clear(void)
説明	コマンド移行の判定に使用する変数をクリアします。
引数	なし
リターン値	なし

---

**cmd\_proc\_begin**

---

概要	コマンド実行開始処理
ヘッダ	なし
宣言	void cmd_proc_begin(void)
説明	シリアル送受信処理用の変数を設定し、 $\bar{S}$ 端子に接続されたポートP0_0を“L”にします。 また、コマンドがRDSR[WIPビットの確認]の場合は、タイムアウト検出用タイマを開始します。 その後、コマンド実行状態フラグを“コマンド実行中 (STATE_EXECUTE)”に設定し、UART2送受信を許可し、1バイト目のデータを送信します。
引数	なし
リターン値	なし

---

**cmd\_proc\_end**

---

概要	コマンド実行終了処理
ヘッダ	なし
宣言	void cmd_proc_end(void)
説明	$\bar{S}$ 端子に接続されたポートP0_0を“H”にします。 コマンドがRDSRの場合は受信したEEPROMのステータスレジスタの値をEEPROMステータス変数に設定します。コマンドがREADの場合は受信したEEPROMデータを指定領域へ格納します。
引数	なし
リターン値	なし

---

**cmd\_proc\_change**

---

概要	コマンド移行処理
ヘッダ	なし
宣言	void cmd_proc_change(void)
説明	<p>コマンド移行判定とコマンド移行を行います。</p> <ul style="list-style-type: none"> <li>●コマンド移行判定           <p>EEPROMのステータスレジスタおよびタイムアウトの発生を確認し、コマンド移行の有無やコマンド実行エラーの発生を判定します。</p> </li> <li>●コマンド移行           <p>&lt;コマンド実行エラーなし かつ コマンド移行ありの時&gt;                p_cmd_typeをインクリメントし、コマンドシーケンスを進めます。                コマンドが“全コマンド実行終了(CMD_TERMINATE)”となった場合はコマンド実行状態フラグを“コマンド実行なし(STATE_NON)”, それ以外の場合は“コマンド実行開始(STATE_BEGIN)”にします。</p> <p>&lt;コマンド実行エラーなし かつ コマンド移行なしの時&gt;                コマンド実行状態フラグを“コマンド実行開始(STATE_BEGIN)”にします。</p> <p>&lt;コマンド実行エラー発生時&gt;                コマンド実行状態フラグを“コマンド実行エラー発生(STATE_ERROR)”にします。                なお、本アプリケーションノートのサンプルコードではコマンド実行エラー発生状態からの復帰処理は行いません。</p> </li> </ul>
引数	なし
リターン値	なし

---

**eep\_s\_control**

---

概要	EEPROM S端子制御
ヘッダ	なし
宣言	void eep_s_control(logic_level_t level)
説明	ポートP0_0からEEPROM S端子へ“H”または“L”を出力します。
引数	logic_level_t level                   : 出力レベル設定 LEVEL_LOW : “L”出力 LEVEL_HIGH: “H”出力
リターン値	なし

---

**uart2\_tx\_interrupt**

---

概要	UART2シリアル送信(UART2送信割り込み)
ヘッダ	なし
宣言	void uart2_tx_interrupt(void)
説明	<p>U2TBレジスタが空になると発生するUART2送信割り込みで呼び出されます。            未送信データが存在する場合、未送信データを1バイト送信します。</p>
引数	なし
リターン値	なし



---

**uart2\_rx\_interrupt**

---

概要	UART2シリアル受信(UART2受信割り込み)
ヘッダ	なし
宣言	void uart2_rx_interrupt(void)
説明	受信完了したときに発生するUART2受信割り込みで呼び出されます。 UART2受信バッファ(U2RB)をリードし、受信データ格納バッファ(rxdata_buf[])へ格納します。 全データの受信が完了した場合はUART2送受信を禁止し、コマンド実行状態フラグを“コマンド実行終了(STATE_END)”にします。
引数	なし
リターン値	なし

---

**timer\_a0\_interrupt**

---

概要	5msタイムアウト処理(タイマA0割り込み)
ヘッダ	なし
宣言	void timer_a0_interrupt(void)
説明	タイマA0開始後5msを経過すると発生するタイマA0割り込みで呼び出されます。 タイムアウトフラグ(f_timeout)をセットし、タイムアウト検出用タイマを停止します。
引数	なし
リターン値	なし

---

**int0\_interrupt**

---

概要	EEPROMライト要求受け付け(INT0割り込み)
ヘッダ	なし
宣言	void int0_interrupt(void)
説明	ポートP8_2への“L”入力で発生するINT0割り込みで呼び出されます。 “EEPROMライトシーケンス開始(R_EEP_StartWriteSeq)”を実行します。
引数	なし
リターン値	なし

---

**int1\_interrupt**

---

概要	EEPROMリード要求受け付け(INT1割り込み)
ヘッダ	なし
宣言	void int1_interrupt(void)
説明	ポートP8_3への“L”入力で発生するINT1割り込みで呼び出されます。 “EEPROMリードシーケンス開始(R_EEP_StartReadSeq)”を実行します。
引数	なし
リターン値	なし

---

**timer\_a0\_init**

---

概要	タイマA0初期化
ヘッダ	なし
宣言	void timer_a0_init(void)
説明	タイマA0をタイマモードに設定します。
引数	なし
リターン値	なし

---

**uart2\_init**

---

概要	UART2初期化
ヘッダ	なし
宣言	void uart2_init(void)
説明	UART2をクロック同期型シリアルインタフェースモードに設定します。
引数	なし
リターン値	なし

## 5.7 フローチャート

### 5.7.1 メイン処理

図 5.4にメイン処理のフローチャートを示します。

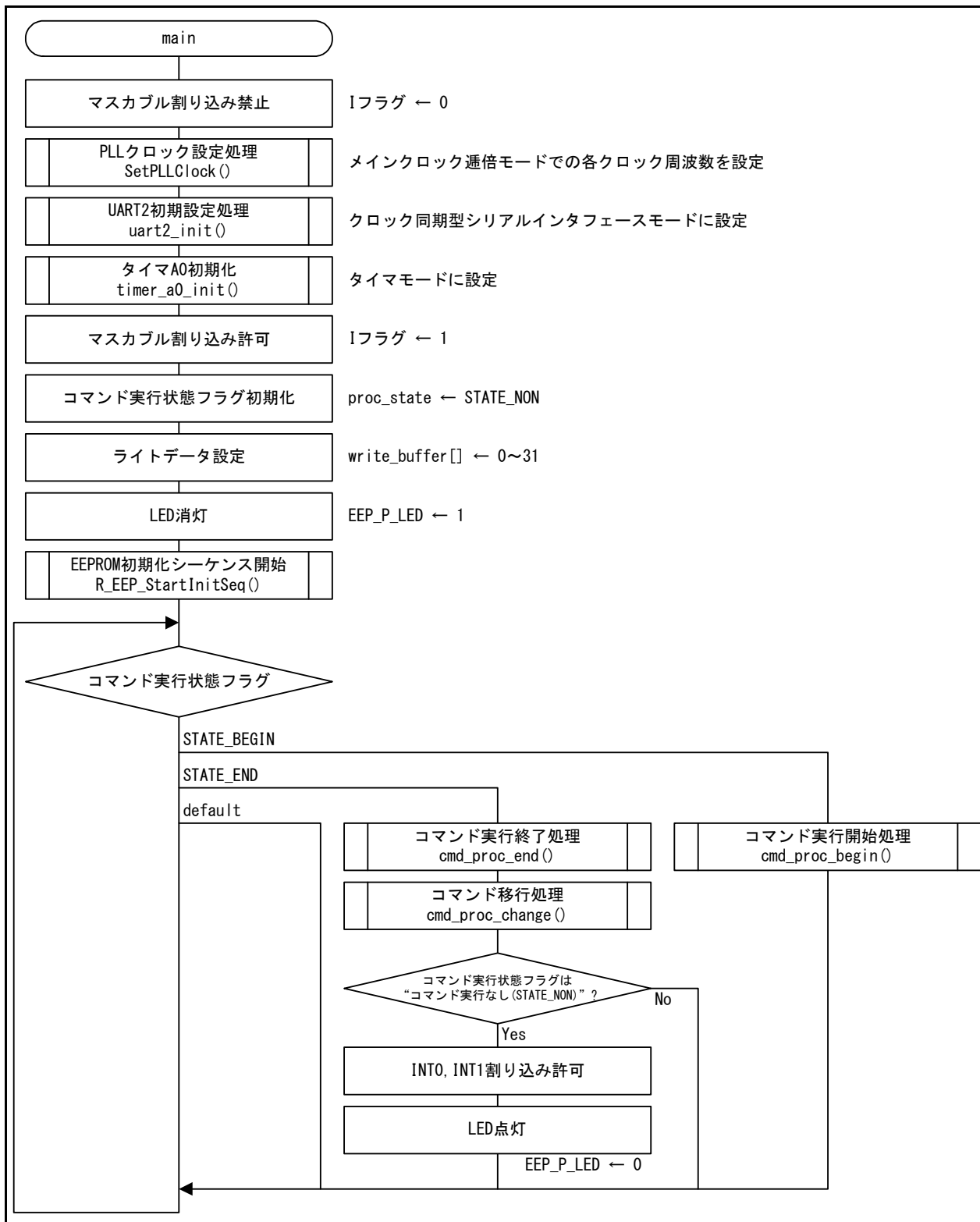


図 5.4 メイン処理

## 5.7.2 EEPROM初期化シーケンス開始

図 5.5にEEPROM初期化シーケンス開始のフローチャートを示します。

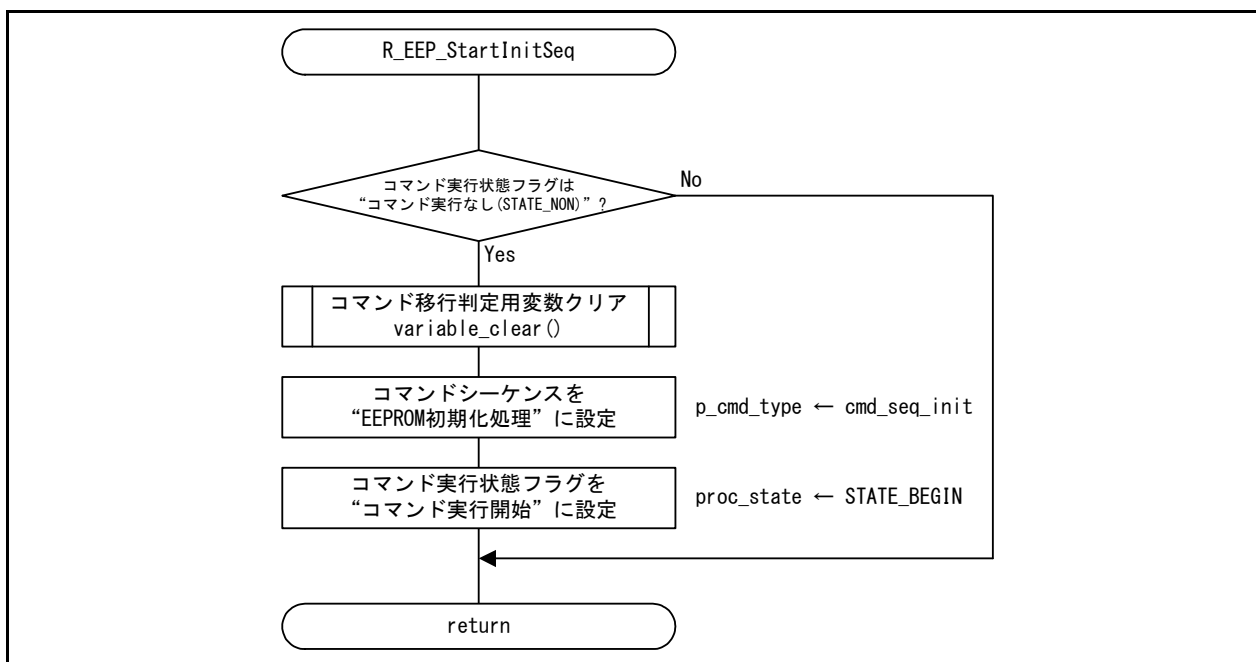


図 5.5 EEPROM初期化シーケンス開始

### 5.7.3 EEPROMライトシーケンス開始

図 5.6にEEPROMライトシーケンス開始のフローチャートを示します。

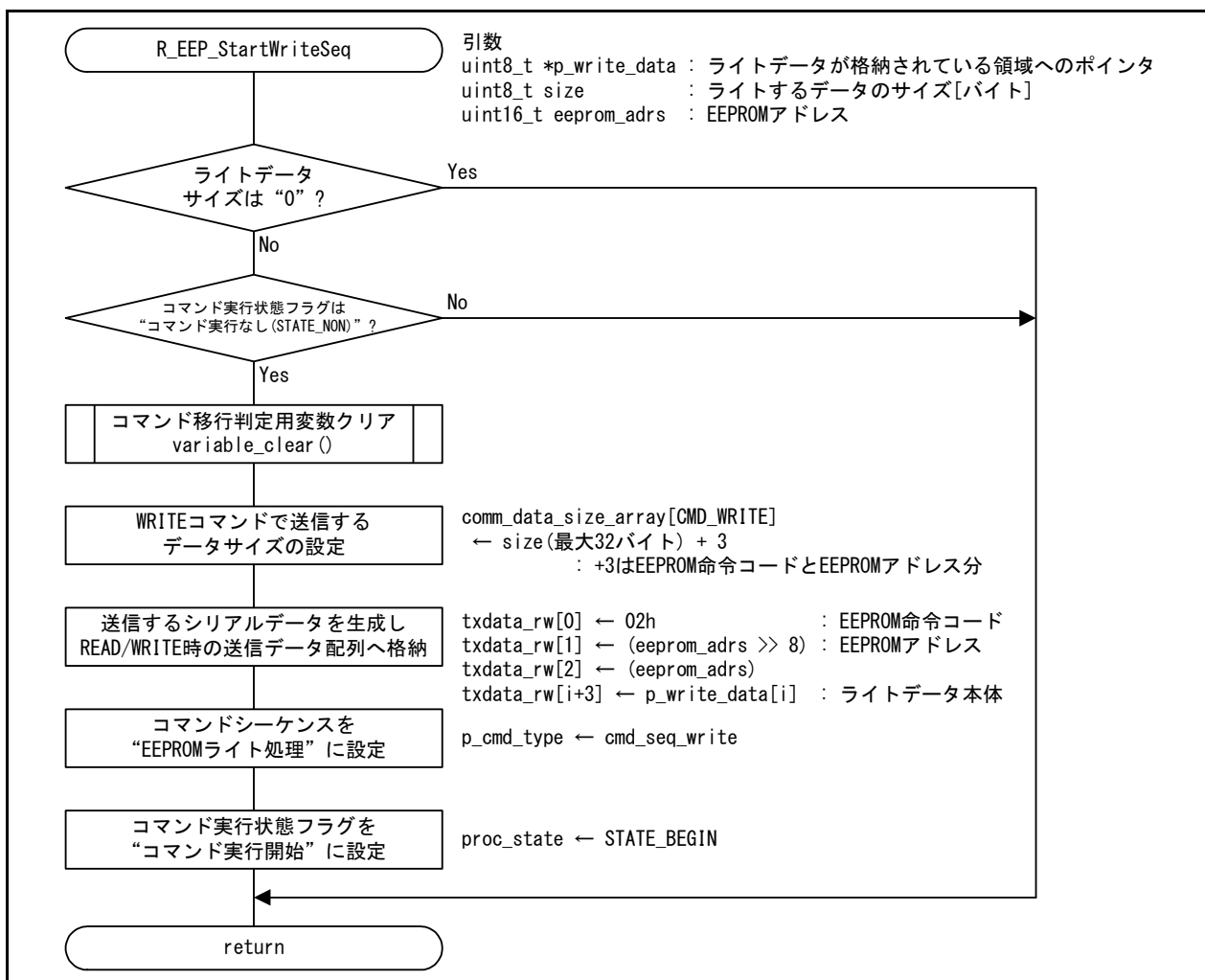


図 5.6 EEPROMライトシーケンス開始

### 5.7.4 EEPROMリードシーケンス開始

図 5.7にEEPROMリードシーケンス開始のフローチャートを示します。

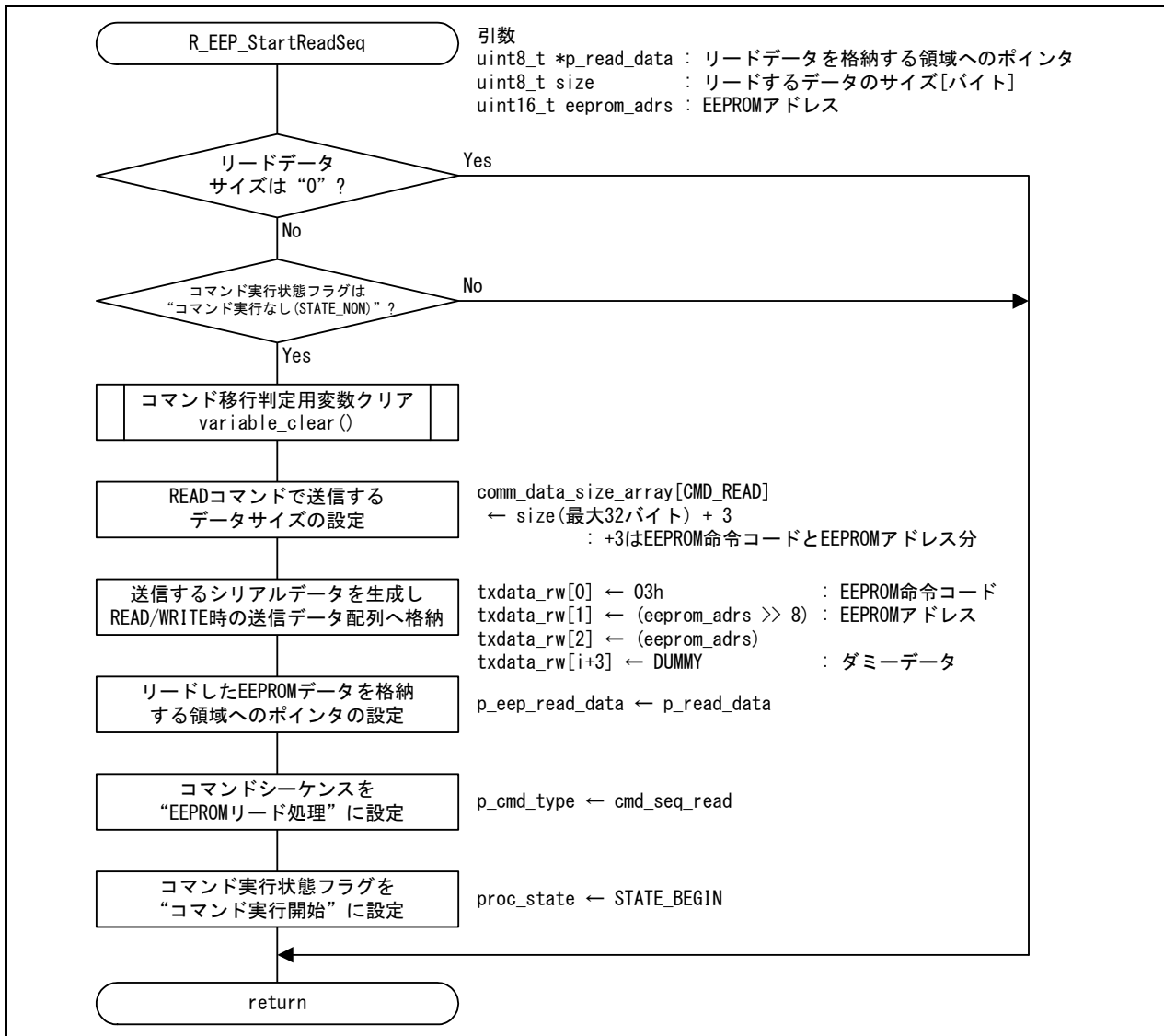


図 5.7 EEPROMリードシーケンス開始

### 5.7.5 コマンド移行判定用変数クリア

図 5.8にコマンド移行判定用変数クリアのフローチャートを示します。

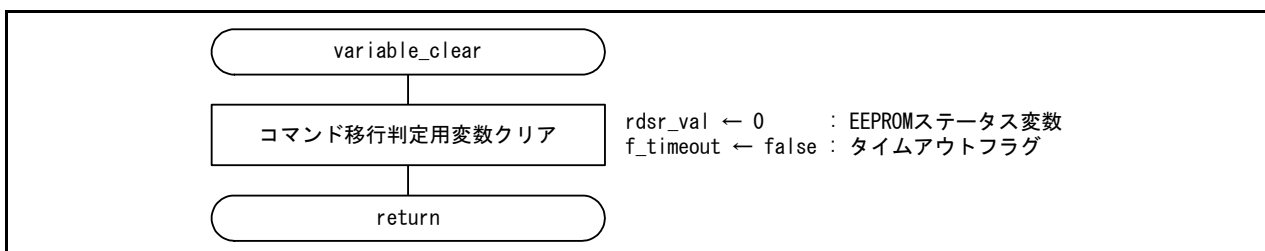


図 5.8 コマンド移行判定用変数クリア

## 5.7.6 コマンド実行開始処理

図 5.9にコマンド実行開始処理のフローチャートを示します。

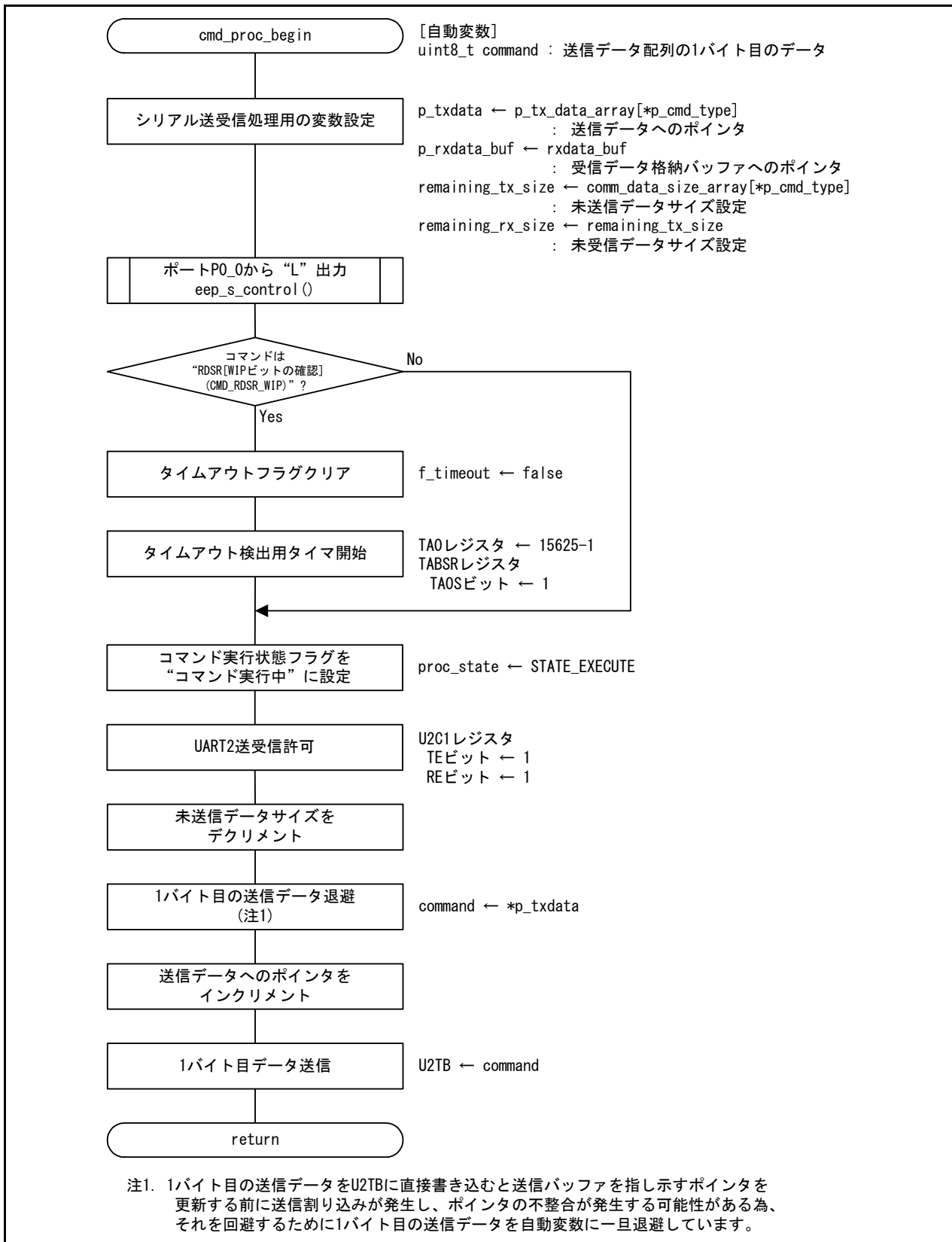


図 5.9 コマンド実行開始処理

### 5.7.7 コマンド実行終了処理

図 5.10にコマンド実行終了処理のフローチャートを示します。

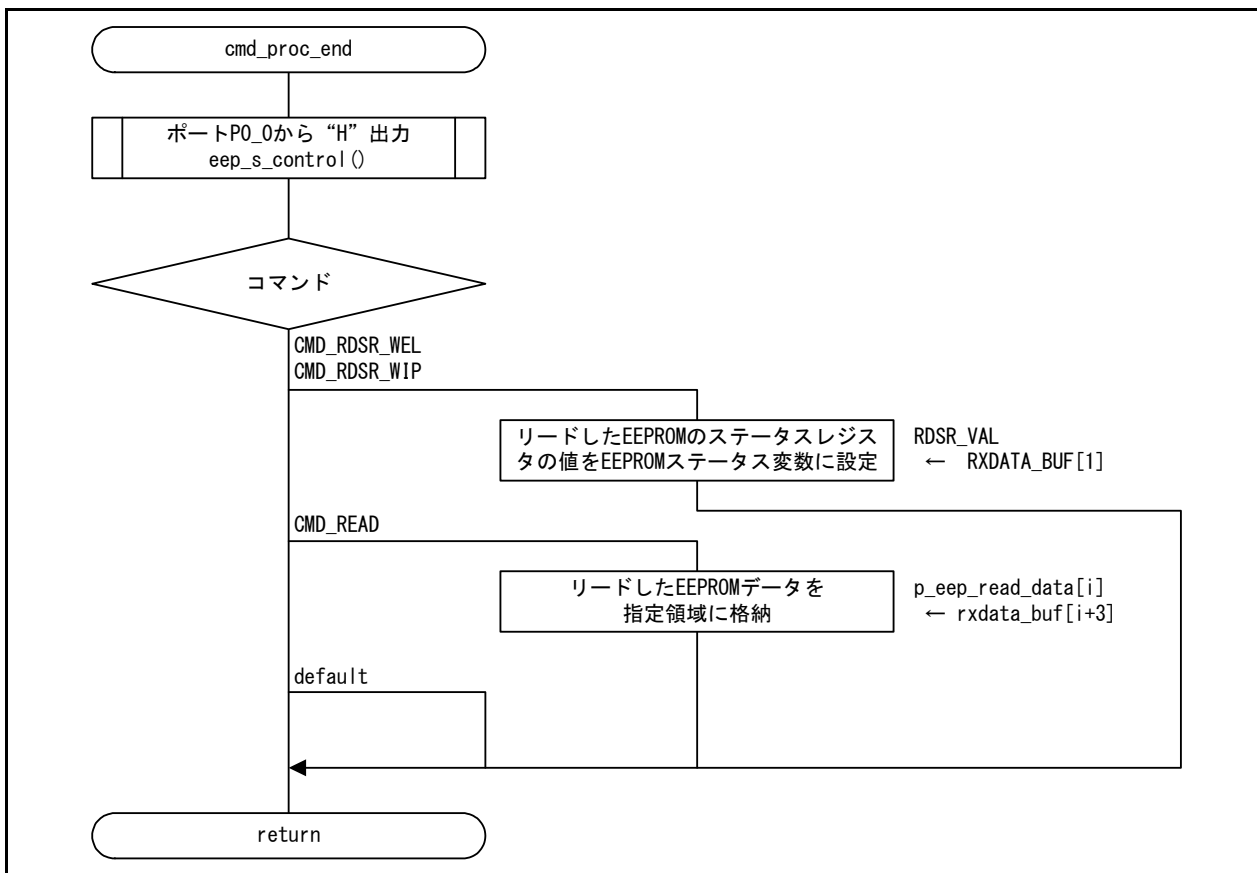


図 5.10 コマンド実行終了処理



### 5.7.8 コマンド移行処理

図 5.11 と図 5.12 にコマンド移行処理のフローチャートを示します。

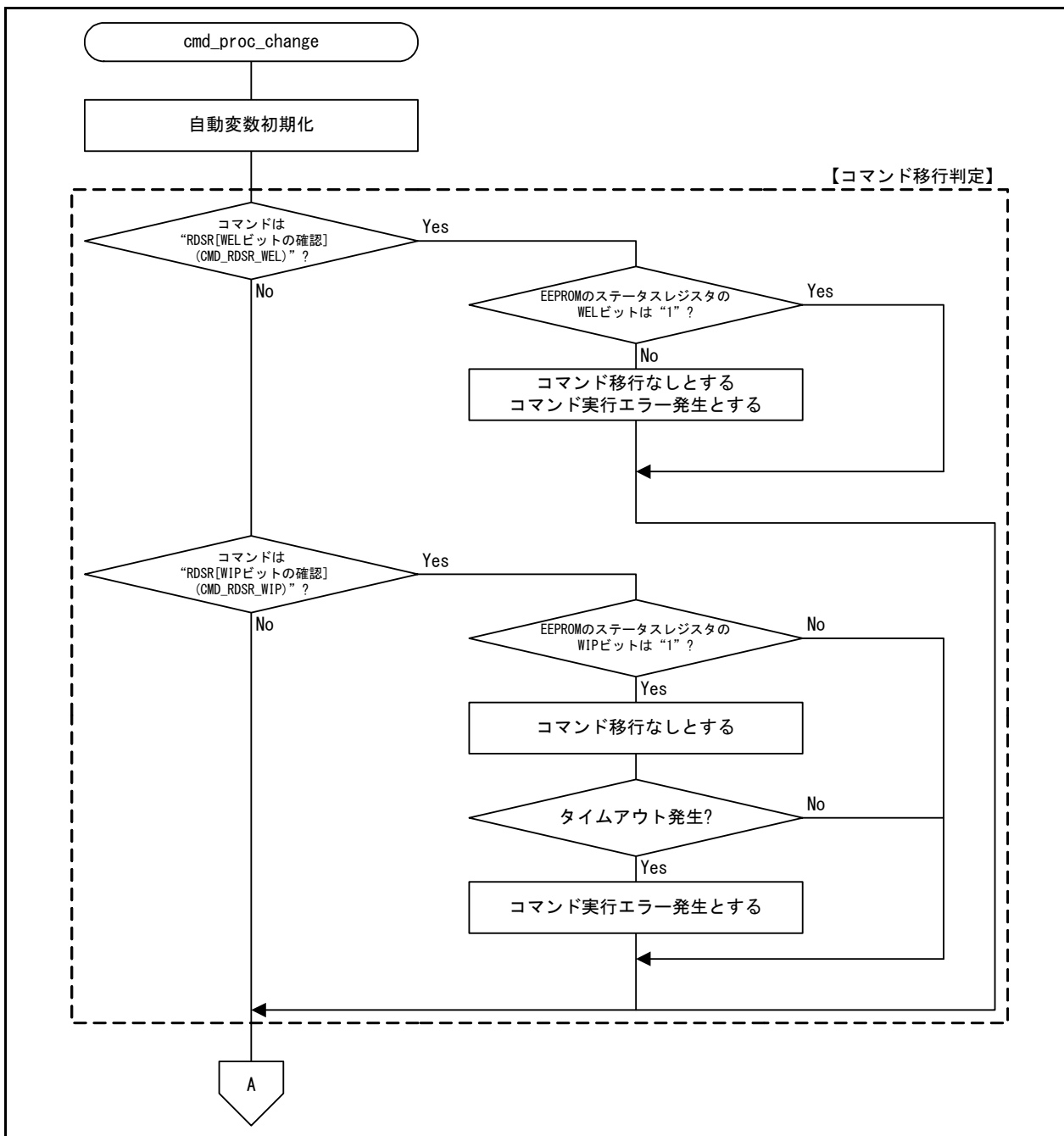


図 5.11 コマンド移行処理(1/2)

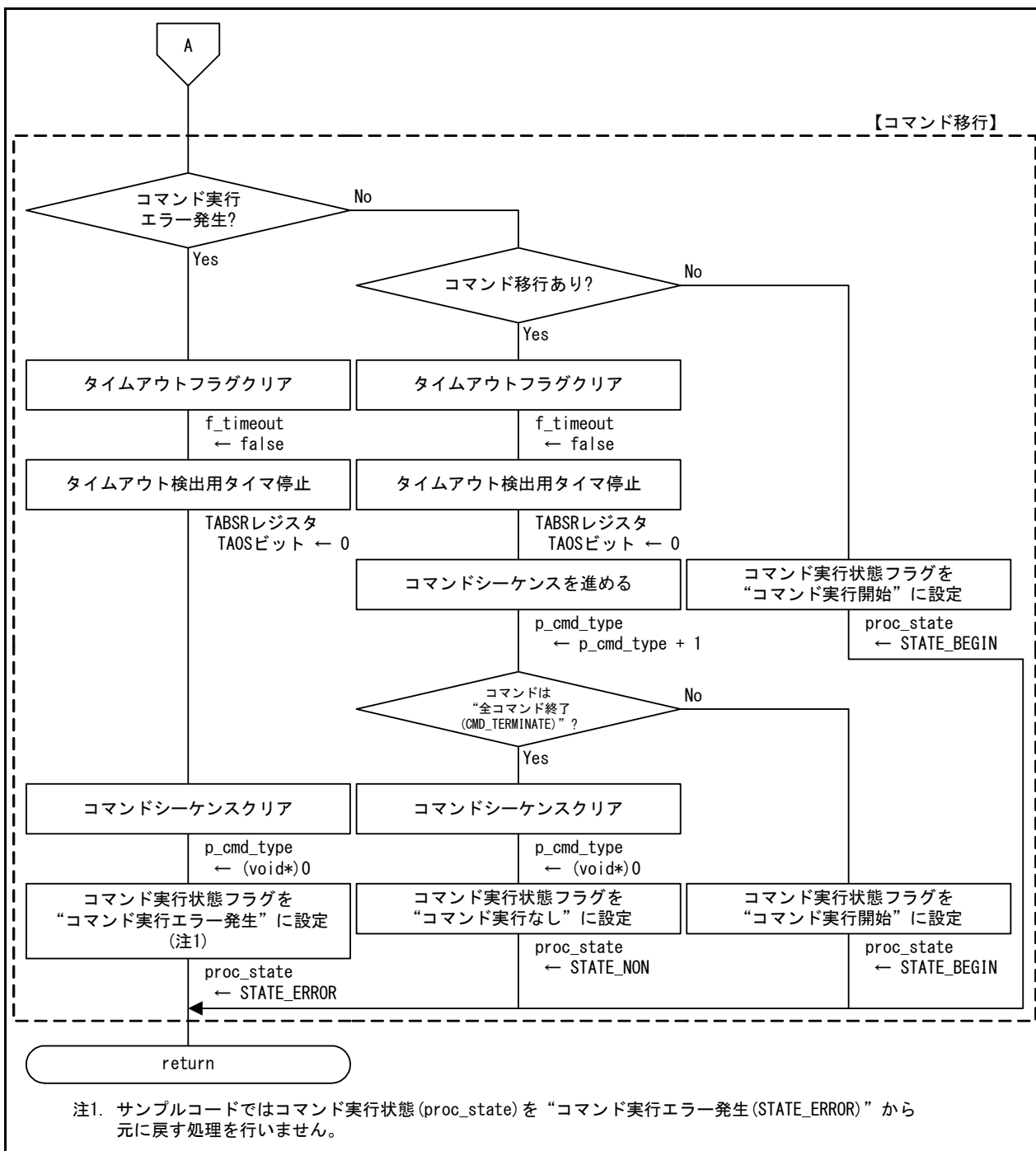


図 5.12 コマンド移行処理(2/2)

### 5.7.9 EEPROM $\bar{S}$ 端子制御

図 5.13 に EEPROM  $\bar{S}$  端子制御のフローチャートを示します。

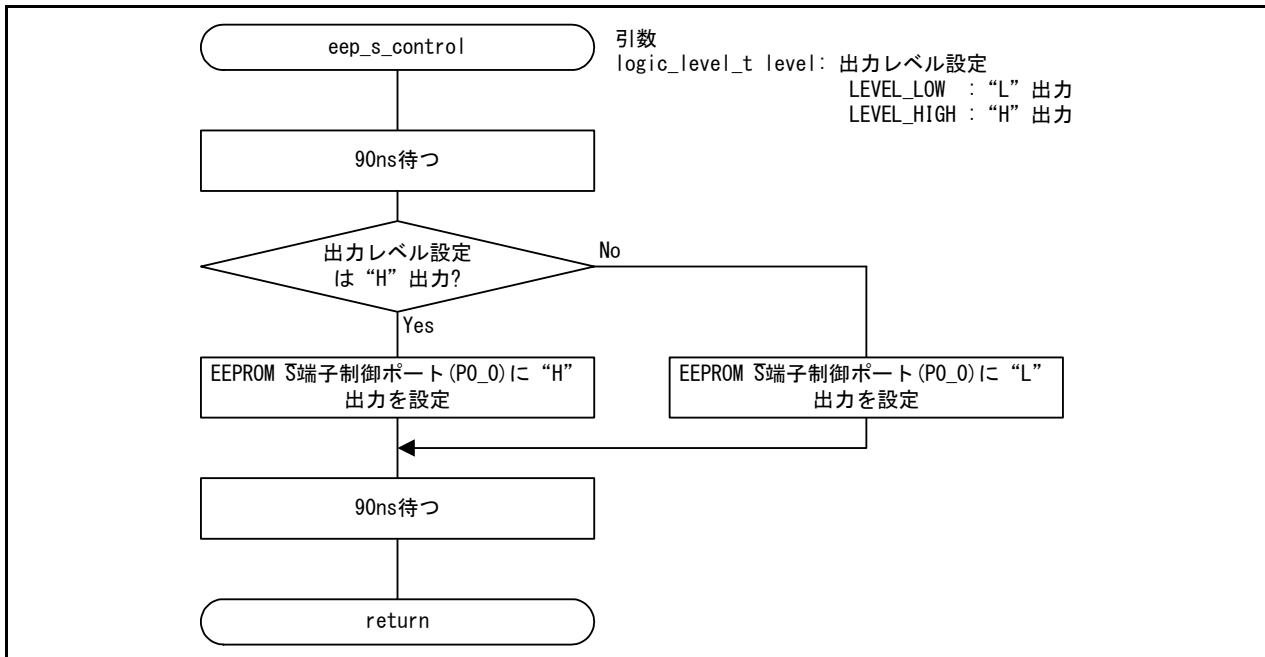


図 5.13 EEPROM  $\bar{S}$  端子制御

### 5.7.10 UART2シリアル送信(UART2送信割り込み)

図 5.14 に UART2シリアル送信(UART2送信割り込み)のフローチャートを示します。

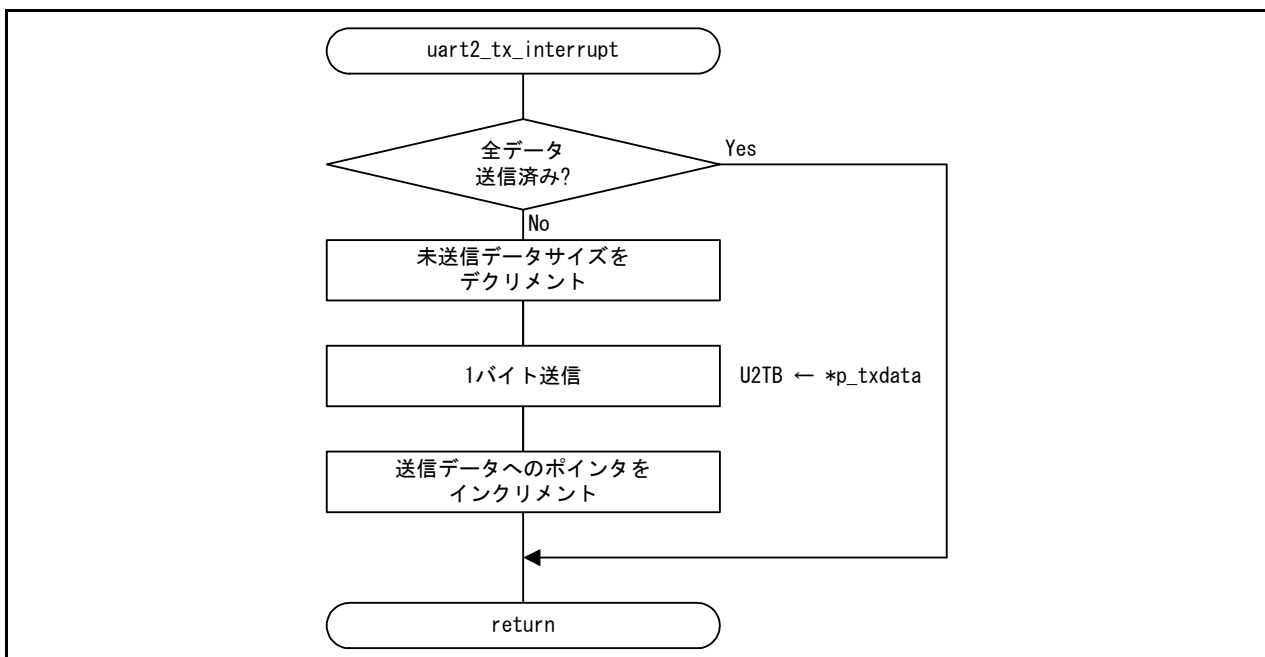


図 5.14 UART2シリアル送信(UART2送信割り込み)

### 5.7.11 UART2シリアル受信(UART2受信割り込み)

図 5.15にUART2シリアル受信(UART2受信割り込み)のフローチャートを示します。

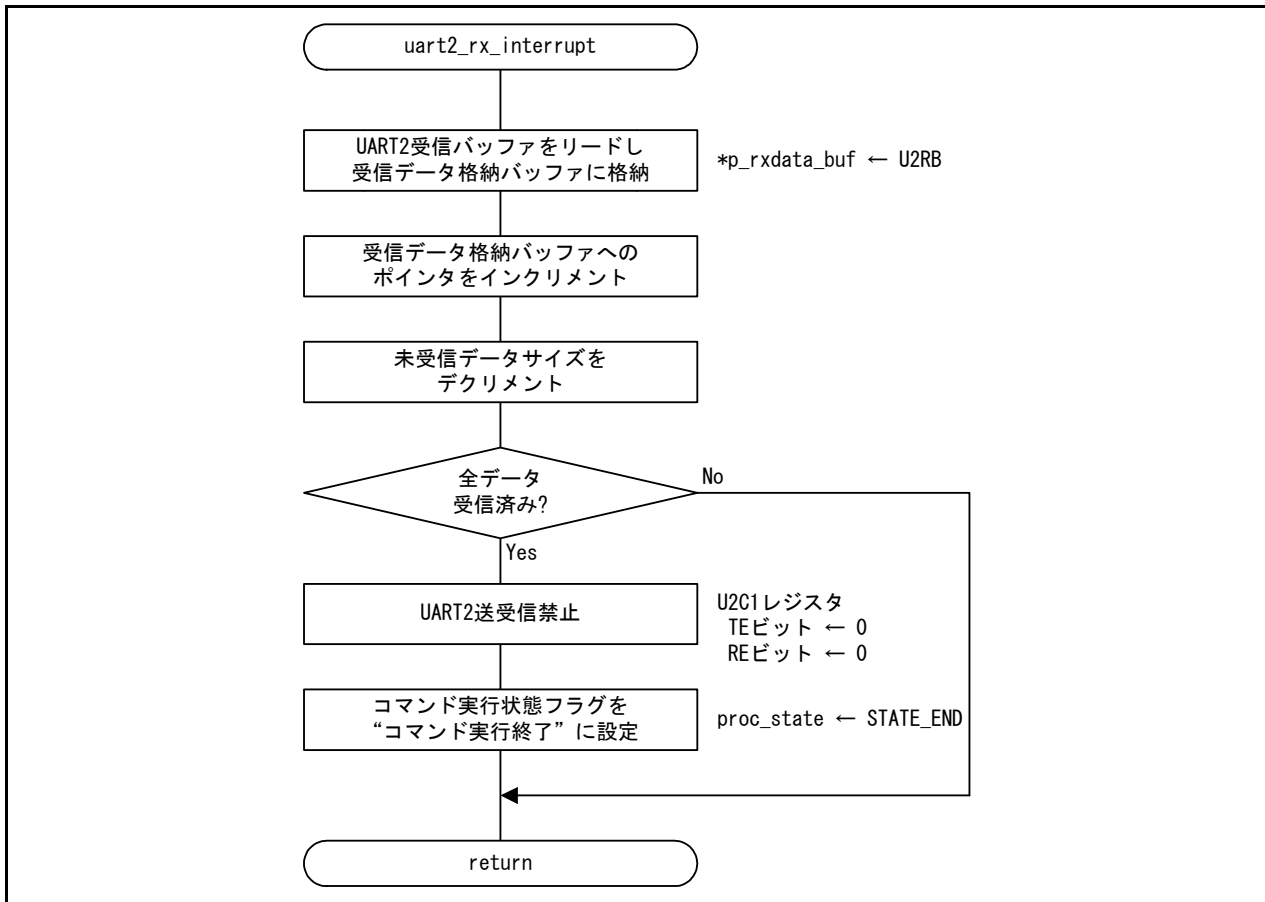


図 5.15 UART2シリアル受信(UART2受信割り込み)

### 5.7.12 5msタイムアウト処理(タイマA0割り込み)

図 5.16に5msタイムアウト処理(タイマA0割り込み)のフローチャートを示します。

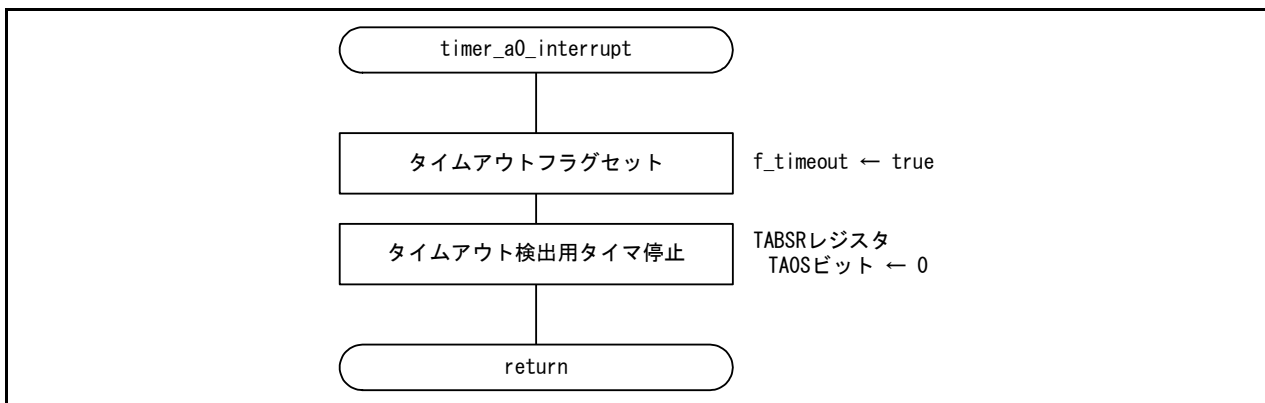


図 5.16 5msタイムアウト処理(タイマA0割り込み)

### 5.7.13 EEPROMライト要求受け付け(INT0割り込み)

図 5.17にEEPROMライト要求受け付け(INT0割り込み)のフローチャートを示します。

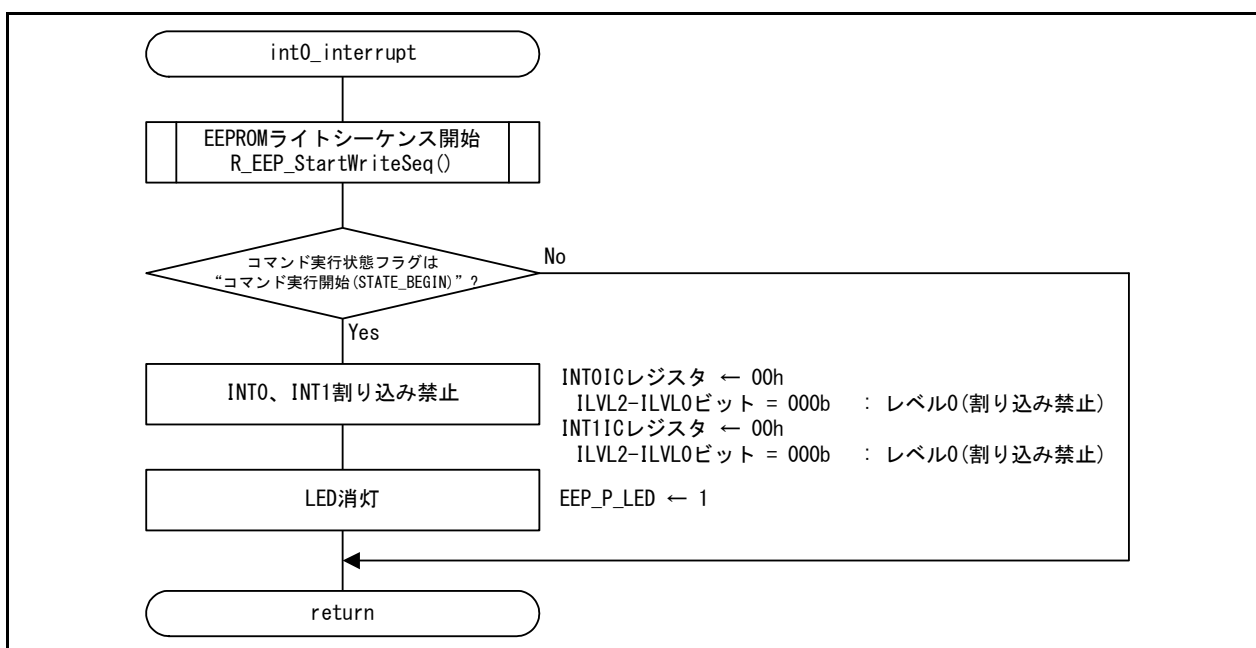


図 5.17 EEPROMライト要求受け付け(INT0割り込み)

### 5.7.14 EEPROMリード要求受け付け(INT1割り込み)

図 5.18にEEPROMリード要求受け付け(INT1割り込み)のフローチャートを示します。

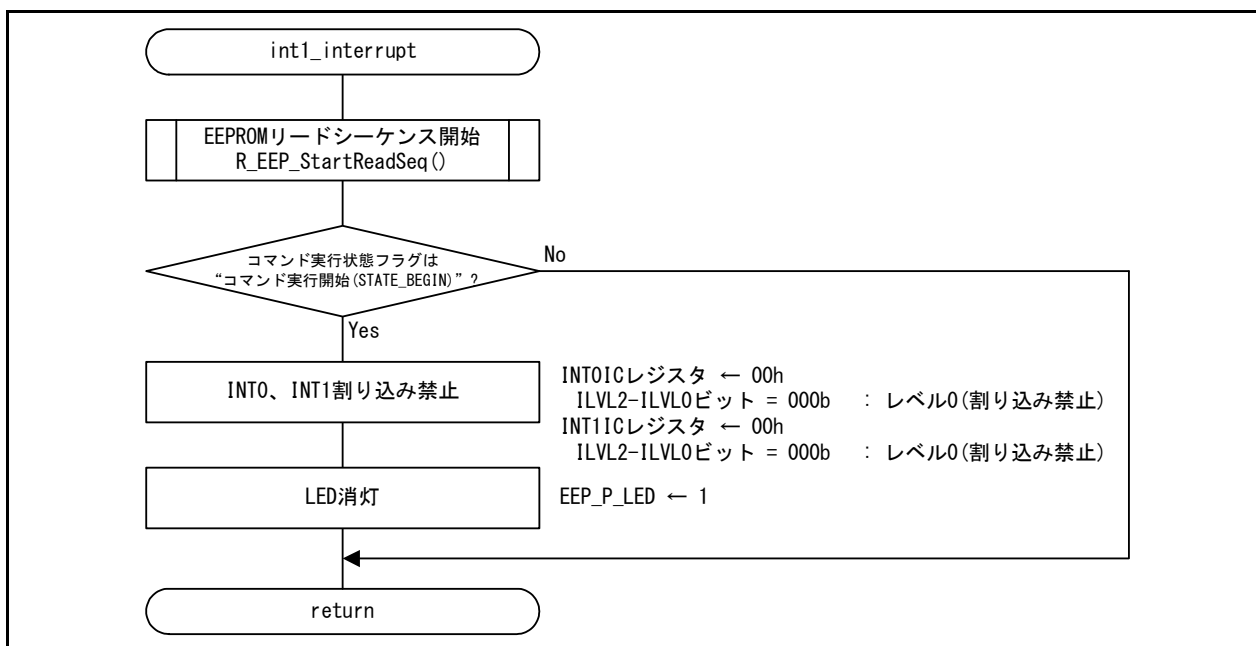


図 5.18 EEPROMリード要求受け付け(INT1割り込み)

## 5.7.15 タイマA0初期化

図 5.19にタイマA0初期化のフローチャートを示します。

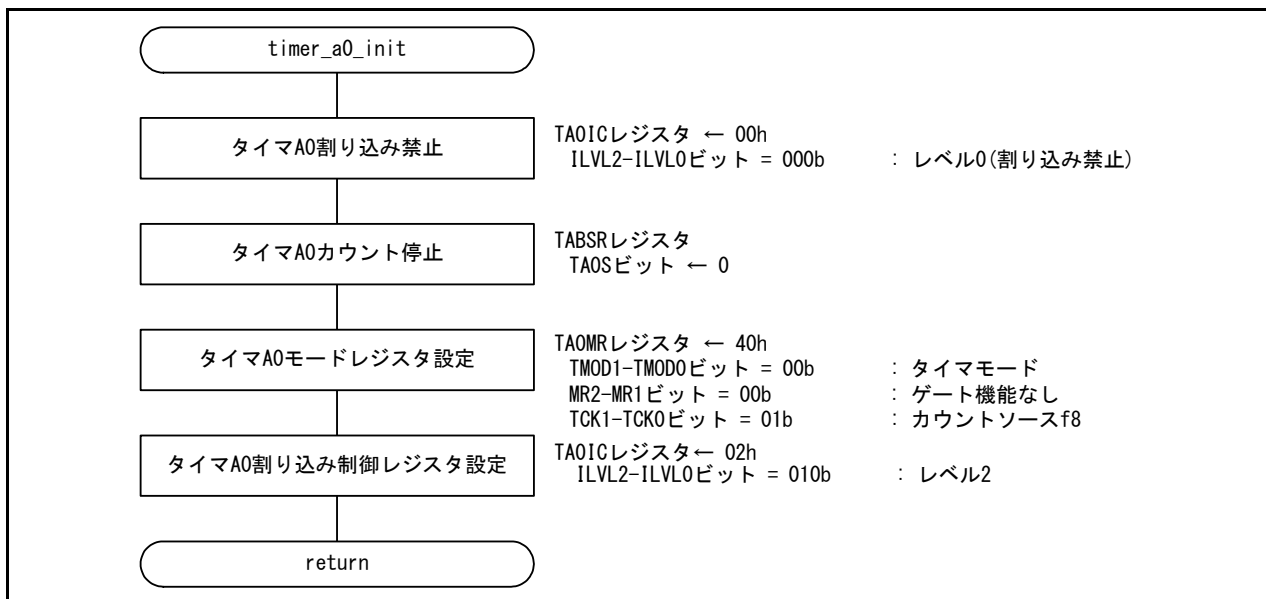


図 5.19 タイマA0初期化

## 5.7.16 UART2初期化

図 5.20 に UART2 初期化のフローチャートを示します。

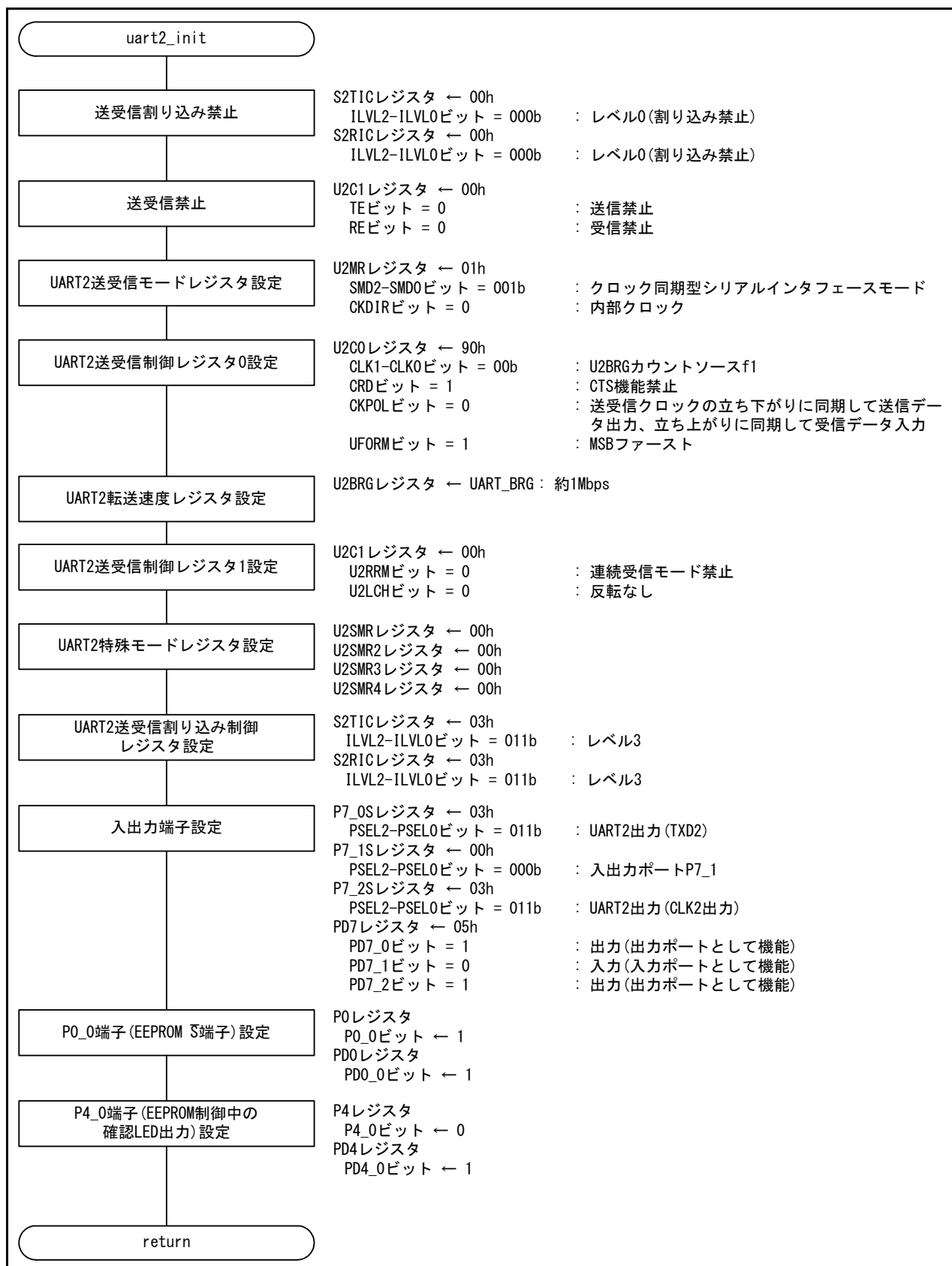


図 5.20 UART2 初期化

## 6. サンプルコード

サンプルコードは、ルネサスエレクトロニクスホームページから入手してください。

## 7. 参考ドキュメント

R32C/116グループ ユーザーズマニュアル ハードウェア編 Rev.1.20

R32C/117グループ ユーザーズマニュアル ハードウェア編 Rev.1.20

R32C/118グループ ユーザーズマニュアル ハードウェア編 Rev.1.20

(最新版をルネサス エレクトロニクスホームページから入手してください。)

テクニカルアップデート/テクニカルニュース

(最新の情報をルネサス エレクトロニクスホームページから入手してください。)

Cコンパイラマニュアル

R32C/100 シリーズ用 Cコンパイラパッケージ V.1.02

Cコンパイラユーザーズマニュアル Rev.2.00

(最新版をルネサス エレクトロニクスホームページから入手してください。)

R1EX25xxx シリーズ EEPROM データシート Rev.0.01

(最新版をルネサスエレクトロニクスホームページから入手してください。)

## ホームページとサポート窓口

ルネサス エレクトロニクスホームページ

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

お問合せ先

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



改訂記録	R32C/100シリーズ クロック同期型シリアルインターフェースモードを使用した EEPROM制御
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2012.12.14	-	初版発行

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。

## 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

### 1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。未使用端子は、本文「未使用端子の処理」で説明する指示に従い処理してください。

### 2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。

外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。

同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

### 3. リザーブアドレス（予約領域）のアクセス禁止

【注意】リザーブアドレス（予約領域）のアクセスを禁止します。

アドレス領域には、将来の機能拡張用に割り付けられているリザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

### 4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。

リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

### 5. 製品間の相違について

【注意】型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。

同じグループのマイコンでも型名が違うと、内部 ROM、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

## ご注意書き

1. 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・システムの設計において、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て、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。
2. 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。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。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。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等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。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
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、産業用ロボット等  
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、各種安全装置等  
当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図し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損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。
6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の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。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、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、社会的損害等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、お客様の機器・システムとしての出荷保証を行ってください。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、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の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い合わせください。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。
9. 本資料に記載されている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・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事情報に使用しないでください。当社製品または技術を輸出する場合は、「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。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
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禁じます。

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。



ルネサス エレクトロニクス株式会社

■営業お問合せ窓口

<http://www.renesas.com>

※営業お問合せ窓口の住所・電話番号は変更になることがあります。最新情報につきましては、弊社ホームページをご覧ください。

ルネサス エレクトロニクス販売株式会社 〒100-0004 千代田区大手町2-6-2（日本ビル）

(03)5201-5307

■技術的なお問合せおよび資料のご請求は下記へどうぞ。

総合お問合せ窓口：<http://japan.renesas.com/contact/>