
RZ/T1、EC-1 グループ

ETG5003.1, 2 機能追加対応編

R01AN3781JJ0200
Rev.2.00
2020/9/30

要旨

RZ/T1、EC-1 の Beckhoff Automation 社が提供する EtherCAT® Slave Stack Code (以下 SSC)環境にて FoE サービスを利用するための ファームウェアの更新機能と SEMI プロファイルのオブジェクトディクショナリを簡単に追加できるサンプルプログラムについて解説します。

対象デバイス

RZ/T1 グループ

EC-1

目次

1. 概要	3
2. ファームウェア更新方法	4
3. サンプルプログラム構成	5
4. ファームウェア更新手順	6
5. サンプルプログラムのハードウェア構成	8
5.1 シリアル・フラッシュ ROM ブート	8
5.2 シリアル・フラッシュ ROM メモリマップ	8
5.3 BANK0 ブート動作概要	10
5.4 BANK1 ファームウェア更新動作概要	12
5.5 BANK1 リブート動作概要	14
5.6 BANK1 ブート動作概要	15
5.7 ローダ用パラメータ	17
6. サンプルプログラムのセクション配置	19
7. サンプルプログラムのビルド構成	20
8. 定数一覧	21
9. 関数一覧	22
10. サンプルプログラム・ソースファイルの作成	23
10.1 SSC Tool のインストール	23
10.2 サンプルプログラムの解凍	23
10.3 SSC ソースファイル作成	23
10.4 bat ファイルの実行	25
11. 動作確認	26
11.1 デバッガ起動(IAR EWARM)	26
11.1.1 BANK0 ビルドとデバッグ	27
11.1.2 BANK1 ビルドとデバッグ	27
11.1.3 BANK1 ダウンロードファイル作成	28
11.2 TwinCAT®起動	29
11.2.1 ESI ファイルの準備	29
11.2.2 TwinCAT®起動	30
11.2.3 ESI ファイルの書込み	32
11.3 TwinCAT®によるファームウェア更新	35
11.4 TwinCAT®による更新ファームウェア読み出し	39
12. Common Device Profile (ETG5003.1)	42
13. ホームページとサポート窓口	43

1. 概要

RZ/T1 および EC-1 の Beckhoff Automation 社が提供する EtherCAT Slave Stack Code(以下 SSC)環境に FoE サービスを利用するための ファームウェアの更新機能 (ETG.5003.2 Ver0.9.13)と Common Device Profile のオブジェクトディクショナリ (ETG.5003.1 Ver1.1.0)を簡単に追加できるサンプルプログラムを提供します。

※本サンプルプログラムはファームウェア更新機能の参考用です。コンフォーマンステストの合格を保証するものではありません。本サンプルプログラムをご参考の上、お客様にて処理の検討・実装をお願い致します。

表 1-1 サポート対象

サポート対象	説明
対象とする RZ/T1、EC-1 ボード	- RZ/T1 評価ボード (RTK7910022C00000BR) - EC-1 リモート I/O ボード(TS-EC-1)
サポートする FoE サービス	ファイル書き込み及びファイル読み出し
サポートするフラッシュメモリ	シリアル・フラッシュ ROM
更新可能なプログラム容量	ATCM 512KB / BTCM 32KB
動作確認済みマスター	TwinCAT®
動作確認済み開発環境	IAR Embedded Workbench® for Arm V7.7 以後

機能概要

ファームウェアの更新機能は ETG.5003.2 の仕様に準拠しており、次の機能を含みます。

- ① FoE プロトコル対応
- ② EtherCAT 動作中のシリアル・フラッシュ ROM の書き換え
- ③ スレーブをセルフブート
- ④ EEPROM の書き換え

ETG5003.1 Common Device Profile につきましては「12. Semiconductor Device Profile」をご参照ください。

2. ファームウェア更新方法

サンプルプログラムを応用して以下のようにスレーブのファームウェア更新が可能となります。

スレーブベンダーはユーザーに対して更新ファームウェア・ファイルとパスワードを提供、ユーザーはTwinCAT 等のマスターから FoE サービスを使い、スレーブにファームウェアをダウンロードすることができます。

更新ファームウェア・ファイルにはチェックサムが付いているので受信データの有効性を確認できます。

更新ファームウェアはシリアル・フラッシュ ROM の出荷時ファームウェアとは別の領域に書き込まれます。更新後はブートシーケンスにより更新ファームウェアのユーザアプリケーションプログラムを ATCM にロードして実行します。

更新に失敗した場合は予め書き込まれた出荷時ファームウェアで復旧が可能です。

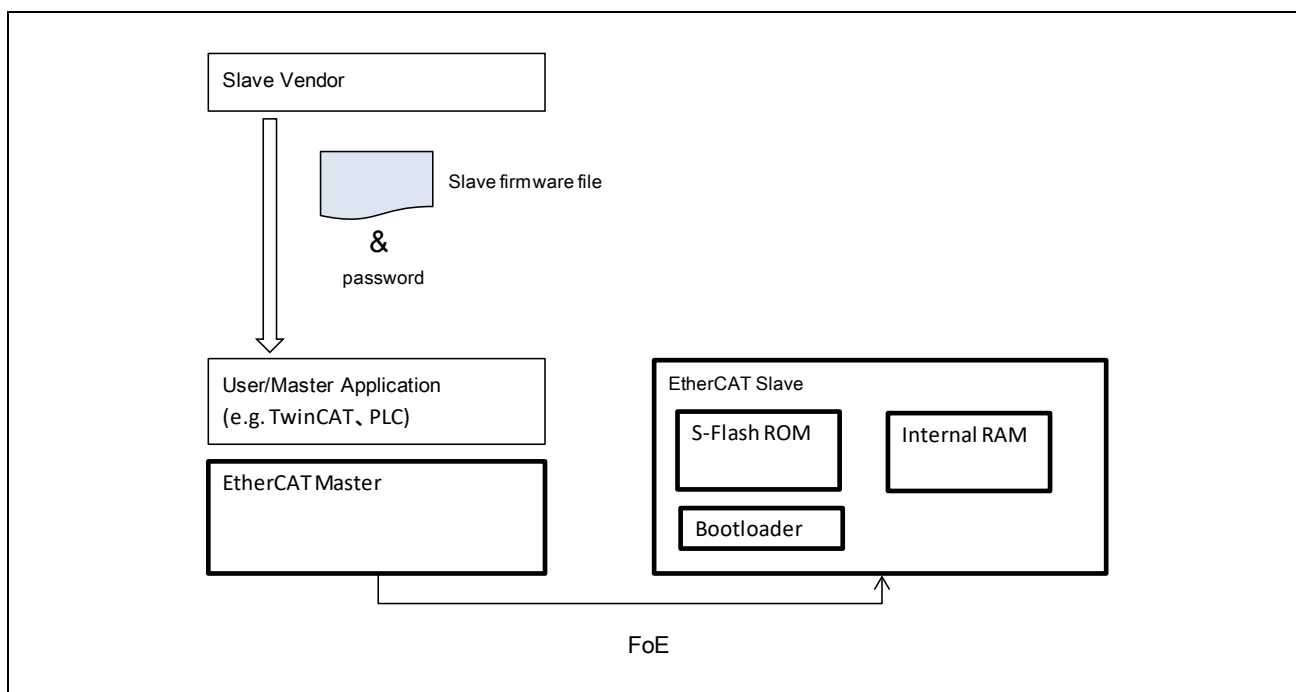


図 2-1 システム構成例

3. サンプルプログラム構成

本書に関連する文書を以下に示します。併せて参照してください。

- RZ/T1 グループ ユーザーズマニュアル ハードウェア編 (R01UH0483JJ)
- RZ/T1 グループ 初期設定 アプリケーションノート (R01AN2554JJ)
- RZ/T1 グループ シリアルフラッシュサンプルプログラム (SPIBSC) (R01AN3010JJ)

サンプルプログラムは RZ/T1 および EC-1 用 SSC にブートローダ機能と FoE サービスのアプリケーションとしてファームウェア更新機能と SEMI プロファイルのオブジェクトディクショナリを追加します。

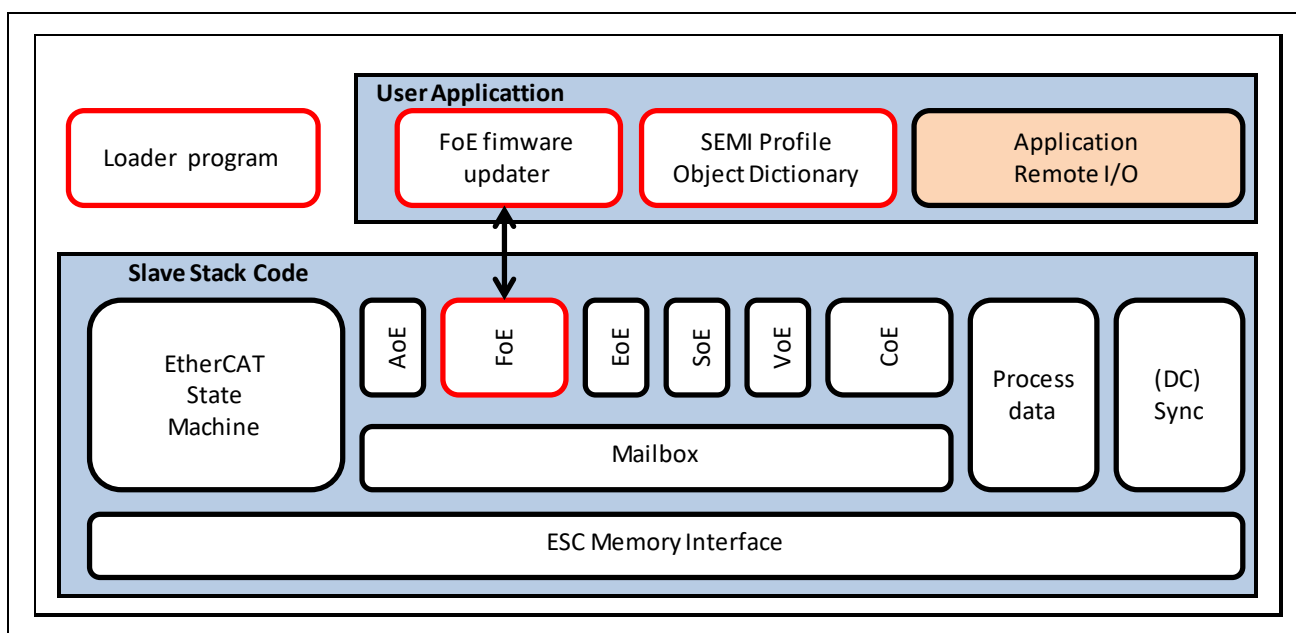


図 3-1 サンプルプログラム構成

4. ファームウェア更新手順

スレーブのファームウェアを更新する際の手順とそのときの EtherCAT マスターの動作とスレーブの動作について説明します。表中の Function は該当するスレーブの動作を、どのプログラムで実現するかを示しています。

表 4-1 ファームウェア更新手順

No	Master/User	Slave	Function		
			SSC	FW boot loader	FW updater
1	request BOOT	confirm BOOT	○		
2	download new slave FW	download new slave FW			
		(1)check filename			○
		(2)check password			○
		(3)write file data to S-Flash			○
		(4)check checksum of S-Flash			○
3		update SII			○
4	request INIT	reboot			
		(1) download new firmware to Internal RAM		○	
		(2)start new FW		○	
5	request PREOP	check if SII and firmware match	○		
6		confirm PREOP	○		
7	user:Check firmware version				
8	request SAFEOP	confirm SAFEOP	○		
9	request OP	confirm OP	○		

1. request BOOT

FoE サービスを実行するために BOOT に遷移します。

2. download new slave FW

マスターから新しい更新ファームウェアをダウンロードします。

スレーブは(1)ファイル名、(2)パスワードが正しいか確認します。正しい場合(3)データをシリアル・フラッシュ ROM に書き込みます。全データ受信後に(4)チェックサムが正しいか確認します。

3. update EEPROM

新しいファームウェアの Revision Number を SII に書き込みます。

4. request INIT

BOOT->INIT に遷移するとスレーブは再起動を行い、(1)シリアル・フラッシュ ROM から内蔵 RAM にプログラムコードをダウンロードし、(2)新しいファームウェアで動作します。

5. request PREOP

SII とファームウェアの Revision Number が一致しているか確認します。

6. confirm:PREOP

PREOP に遷移したことを確認します。

7. user:Check firmware version

ユーザーは CoE オブジェクトの 0x100A の値で更新されたファームウェアバージョンになっているか確認できます。また 0x1018:03 では Revision を確認できます。

8. request SAFEOP

SAFEOP に遷移します。

9. request OP

OP に遷移します。

5. サンプルプログラムのハードウェア構成

5.1 シリアル・フラッシュ ROM ブート

シリアル・フラッシュ ROM に格納されたブートローダが起動するためには、ブート選択を SPI ブートモード (シリアル・フラッシュ) に設定する必要があります。

5.2 シリアル・フラッシュ ROM メモリマップ

シリアル・フラッシュ ROM を 3 つの領域で使い分けます。

表 5-1 シリアル・フラッシュ ROM の領域区分

アドレス範囲	名称(サイズ)	内容
3000_0000H ~3000_FFFFH	ブートローダ用パラメータ領域 (64KB)	RZ/T1およびEC-1のブート機能が参照するブートローダ用パラメータ領域 ※出荷時はBANK0、ファームウェア更新時はBANK1を参照
3001_0000H ~3009_FFFFH	BANK0領域 (64KB+512KB=576KB)	シリアル・フラッシュROMライターやICE等で書き込まれる出荷時ファームウェア用領域
3010_0000H ~3018_FFFFH	BANK1領域 (64KB+512KB=576KB)	FoEサービスにより書き込まれる更新ファームウェア用領域

シリアル・フラッシュ ROM のメモリマップを「図 5.1 メモリマップ」に示します。

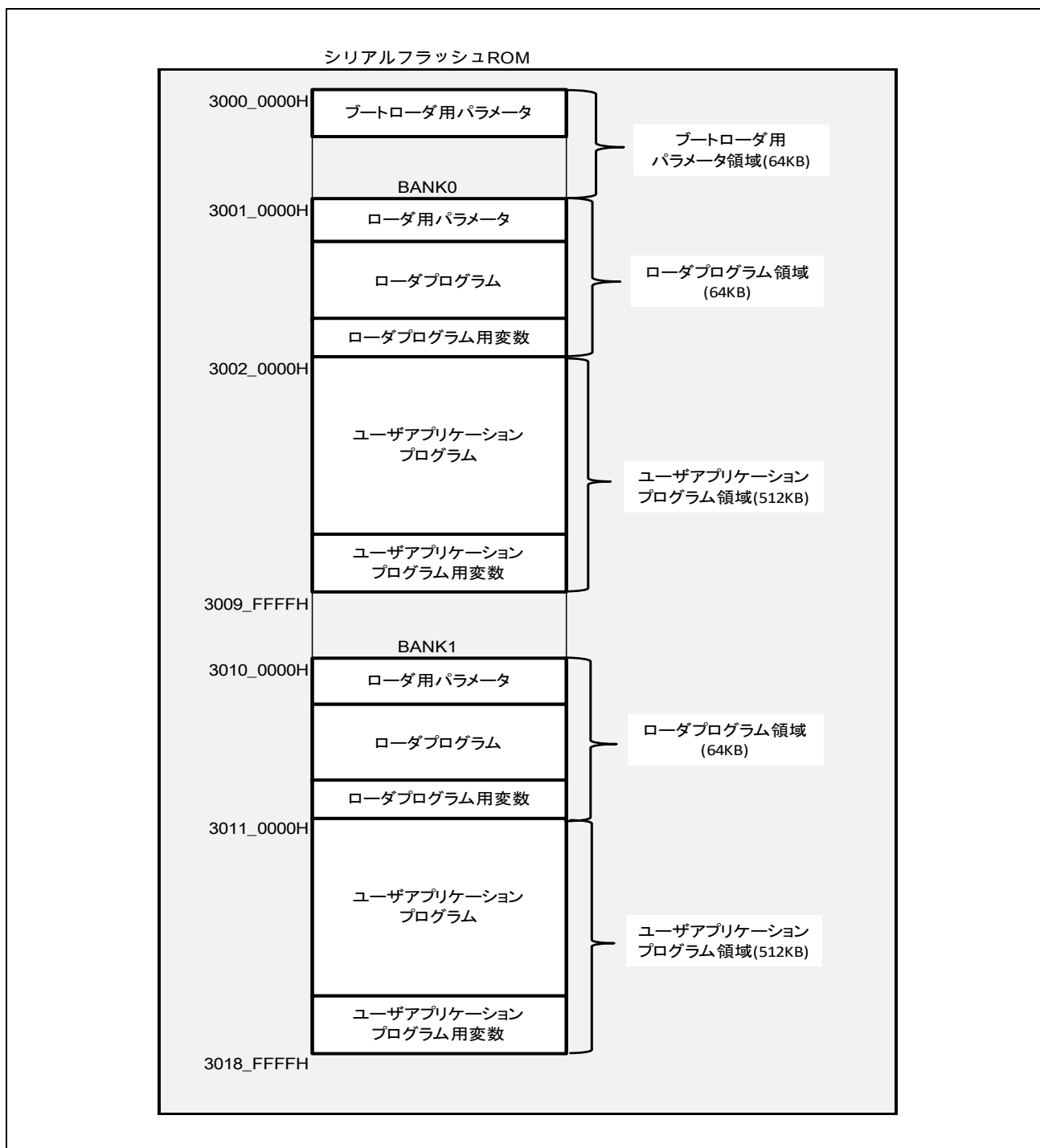


図 5-1 メモリマップ

5.3 BANK0 ブート動作概要

BANK0 に書かれた出荷時ファームウェアのブート動作を「図 5.2 BANK0 ブート動作」を使って説明します。

RZ/T1 および EC-1 のブート機能は

- ① ブートローダ用ローダパラメータ領域の値を参照して
- ② BANK0 のローダプログラムを BTCM に転送した後
- ③ ローダプログラムに処理を移行します。

ローダプログラムは各種スタックポインタの初期化後

- ④ ローダプログラム用変数を BTCM に転送し周辺モジュールの設定等を行います。

またブートローダ用ローダパラメータ領域の値を参照して

- ⑤ ATCM にユーザアプリケーションプログラムを転送した後
ユーザアプリケーションプログラムに処理を移行します。

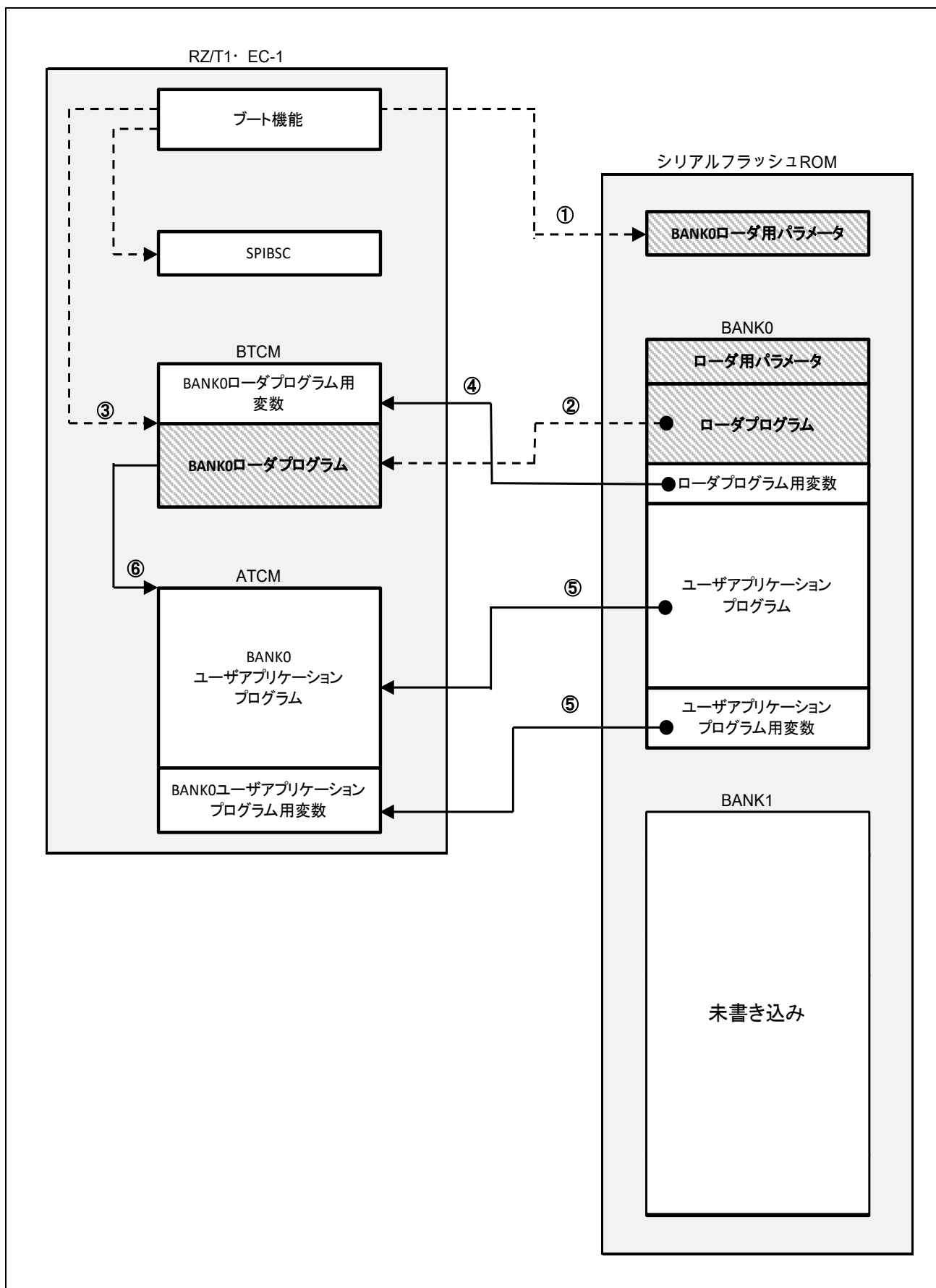


図 5-2 BANK0 ブート動作

5.4 BANK1 ファームウェア更新動作概要

ATCM でユーザアプリケーションプログラムを実行している際に FoE サービスを使いファームウェアを更新する動作を「図 5.3 BANK1 ファームウェア更新動作」を使って説明します。

1. FoE サービスの開始時にマスターから更新ファームウェアの BIN ファイルのファイル名とパスワードが送られますので、ファイル接頭辞とパスワードが正しいかチェックします。正しい場合、BIN データの受信を開始します。
2. シリアル・フラッシュ ROM 先頭のブートローダ用パラメータ領域をイレースします。
3. ファームウェア更新が何等かの理由により中断された場合、出荷時ファームウェアで起動できるよう、BANK0 ローダ用パラメータをブートローダ用パラメータ領域にコピーします。
4. BANK 1 の先頭アドレスから 1 セクタ(64KB)をイレースします。イレース中はマスターがタイムアウトしないように BUSY ステータスを返します。
5. イレースが終わったらデータを受信します。データは ATCM のユーザアプリケーションプログラム用変数に確保された受信バッファに格納されます。受信バッファのデータが 2 ページ分(512 バイト)溜まる毎にシリアル・フラッシュ ROM に書き込みます。マスターには ACK を返します。1 セクタ分の書き込みが終了したら、次のセクタをイレースします。(図中①)
6. BANK1 の最終アドレスを書き終えるまで 3 を繰り返します。
7. BANK1 の全領域を書き終えたらチェックサムが合っているか確認します。
8. 合っている場合、シリアル・フラッシュ ROM 先頭のブートローダ用パラメータ領域を再びイレースします。
9. BANK1 ローダ用パラメータをブートローダ用パラメータ領域にコピーします。(図中②)
10. 更新ファームウェアのリビジョンに SII の Revision Number を更新します。

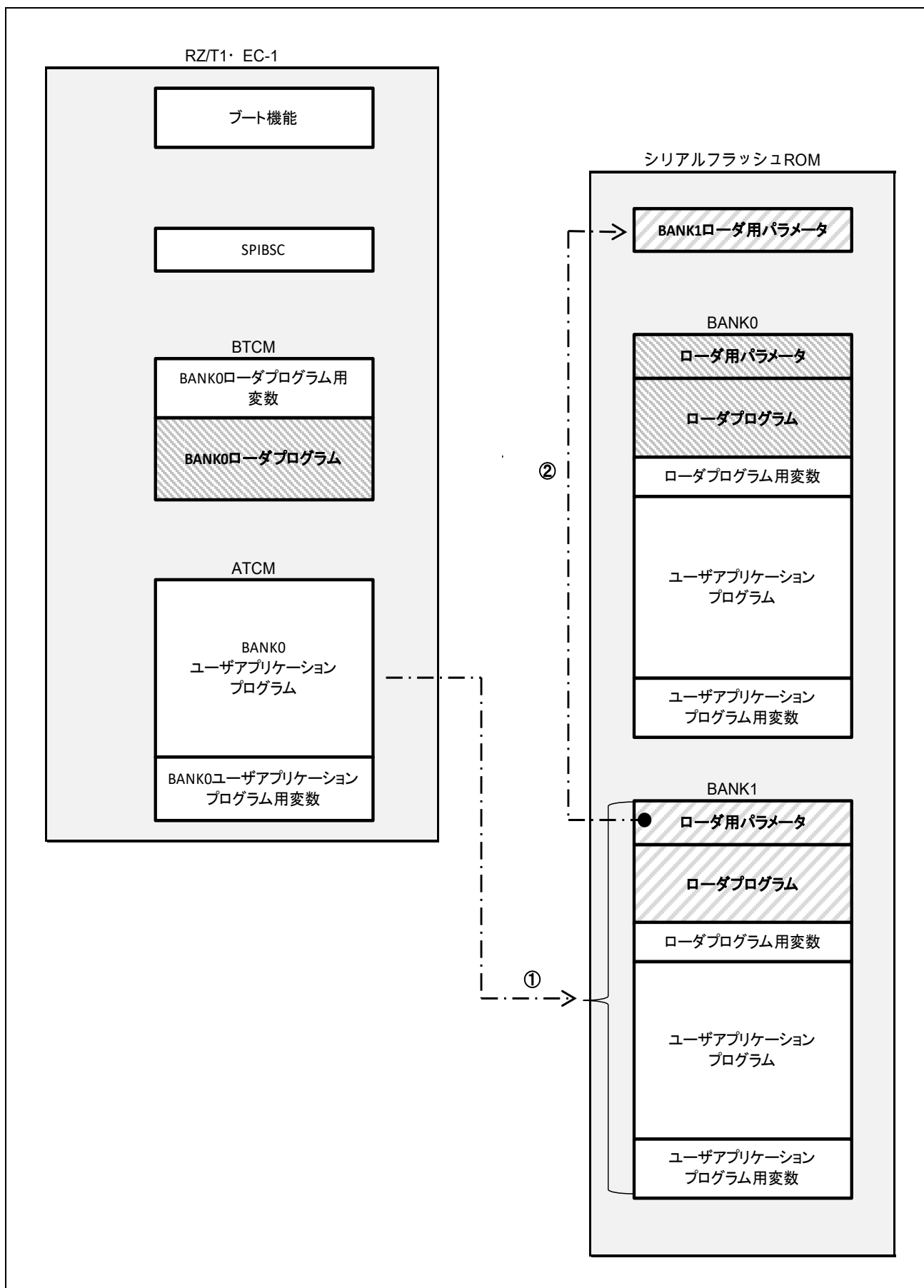


図 5-3 BANK1 ファームウェア更新動作

5.5 BANK1 リブート動作概要

BANK1に更新ファームウェアが書き込まれた後、リブートするまでの動作を「図 5.4 BANK1 リブート動作」を使って説明します。

BANK1のファームウェア更新が正常に終了した後、

- ① 再ローダプログラムがブートローダ用パラメータ領域のBANK1ローダ用パラメータを参照してローダプログラムをBTCMにコピーした後
- ② BANK1ローダプログラムの開始アドレスにジャンプすることでローダプログラムを実行します。

その後、アプリケーションプログラムを実行するまでは「5.6 BANK1 ブート動作概要」の④以後と同様の処理になります。

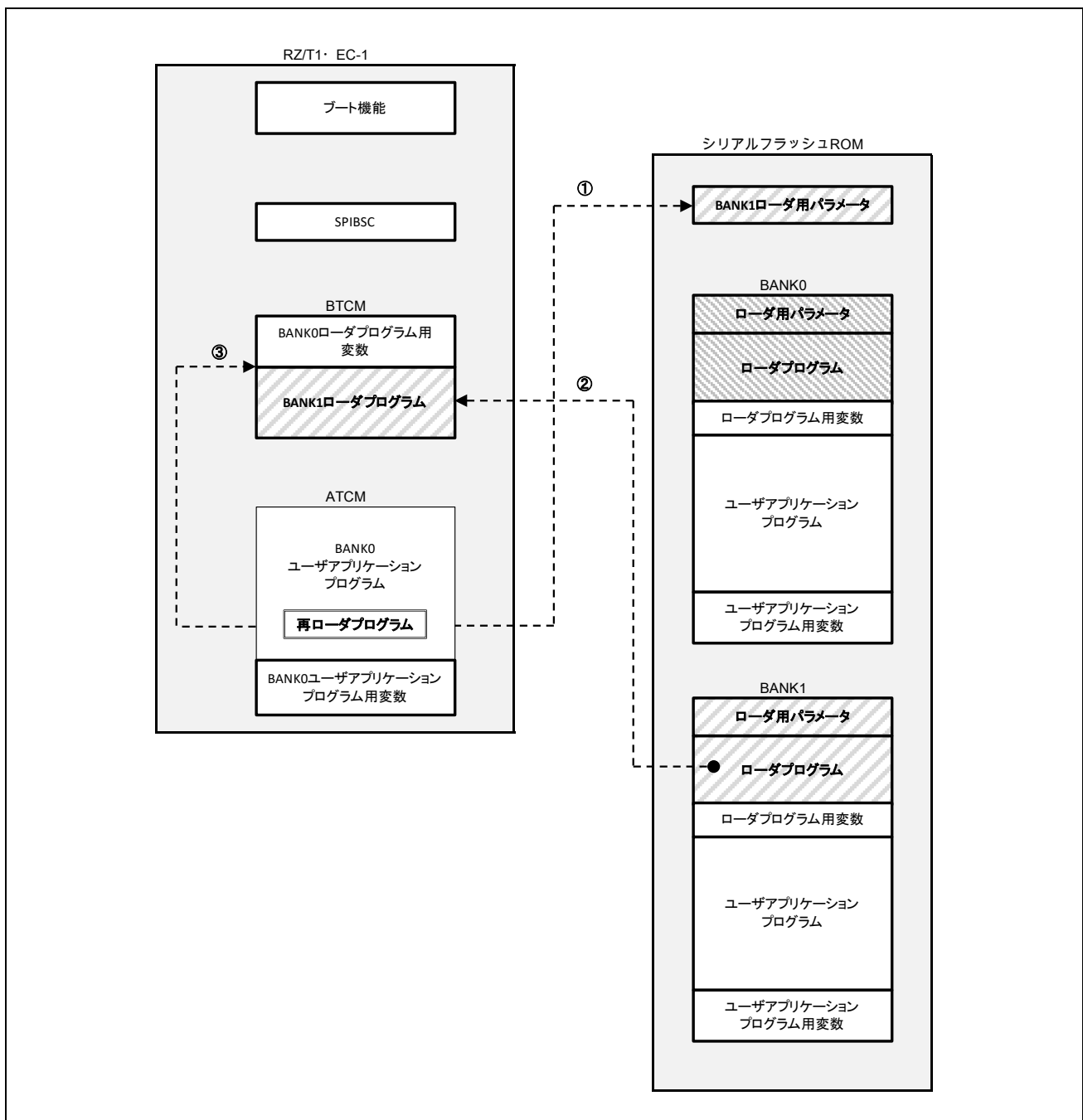


図 5-4 BANK1 リブート動作

5.6 BANK1 ブート動作概要

ファームウェア更新後に電源投入された場合の BANK1 のブート動作を「図 5.5 BANK1 ブート動作」を使って説明します。

- ① ブートローダ用ローダパラメータ領域の値を参照して
- ② BANK1 のローダプログラムを BTCM に転送した後
- ③ ローダプログラムに処理を移行します。

ローダプログラムは各種スタックポインタの初期化後

- ④ ローダ用変数を BTCM に転送し周辺モジュールの設定等を行います。

またブートローダ用ローダパラメータ領域の値を参照して

- ⑤ ATCM にユーザアプリケーションプログラムを転送した後
- ⑥ ユーザアプリケーションプログラムに処理を移行します。

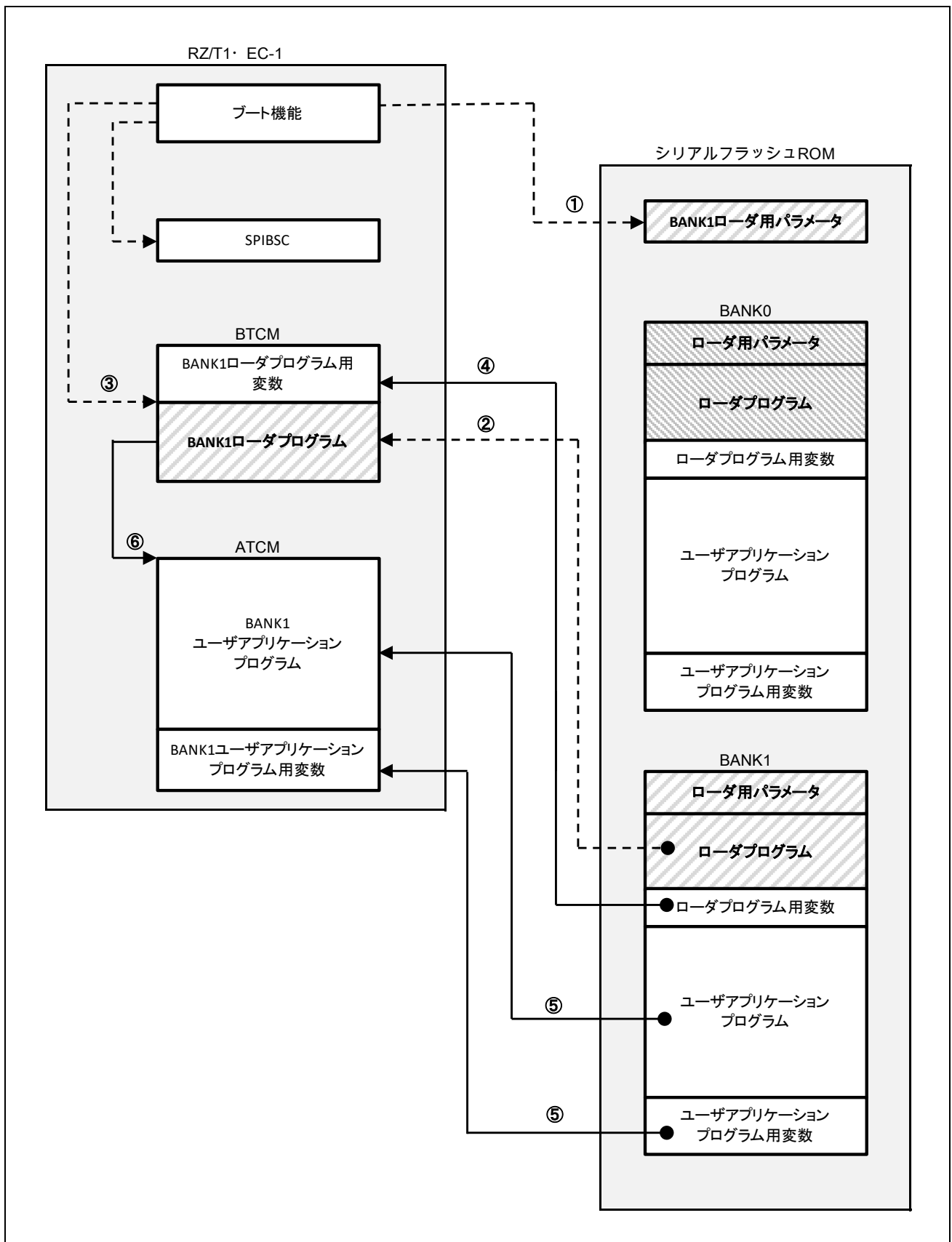


図 5-5 BANK1 ブート動作

5.7 ローダ用パラメータ

サンプルプログラムではユーザアプリケーションプログラム用を ATCM にダウンロードするためのパラメータをデフォルトでは未使用の DUMMY1~3 に追加しています。サンプルプログラムのローダ用パラメータ情報を「」に示す。

また、各パラメータとシリアル・フラッシュ ROM 上のアドレスとの関係を「図 5.6 ローダ用パラメータの参照先」に示す。

表 5-2 ローダ用パラメータ情報

パラメータ名	オフセットアドレス	内容
CACHE_FLG	0000_0000H	ブート処理時に、Cortex-R4 の I1 キャッシュと D1 キャッシュをイネーブルにするか選択します（高速化）。
SSLDR_V	0000_0004H	SSL 遅延レジスタ（SSLDR）の設定値
SPBCR_V	0000_0008H	ビットレート設定レジスタ（SPBCR）の設定値
DRCR_V	0000_000CH	データリードコントロールレジスタ（DRCR）の設定値
SPIBSC_FLG	0000_0010H	ブート処理終了後に、SPIBSC 設定を初期値に戻すかどうかを選択します
LDR_ADDR_NML	0000_0014H	ローダプログラムの先頭アドレスを設定
LDR_SIZE_NML	0000_0018H	ローダプログラムのプログラムサイズを設定
DEST_ADDR_NML	0000_001CH	ローダプログラムを展開する BTCM の先頭アドレスを設定
VECTOR_RBLK	0000_0020H	ユーザアプリケーションプログラムのベクタテーブル先頭アドレスを設定
USR_P_RBLK	0000_0024H	ユーザアプリケーションプログラムの先頭アドレスを設定
USR_D_RBLK	0000_0028H	ユーザアプリケーションプログラム用変数の先頭アドレスを設定
DUMMY4-10	0000_002CH	未使用
	0000_0030H	未使用
	0000_0034H	未使用
	0000_0038H	未使用
	0000_003CH	未使用
	0000_0040H	未使用
	0000_0044H	未使用
CECJ_SUM	0000_0048H	ローダ用パラメータのチェックサム値

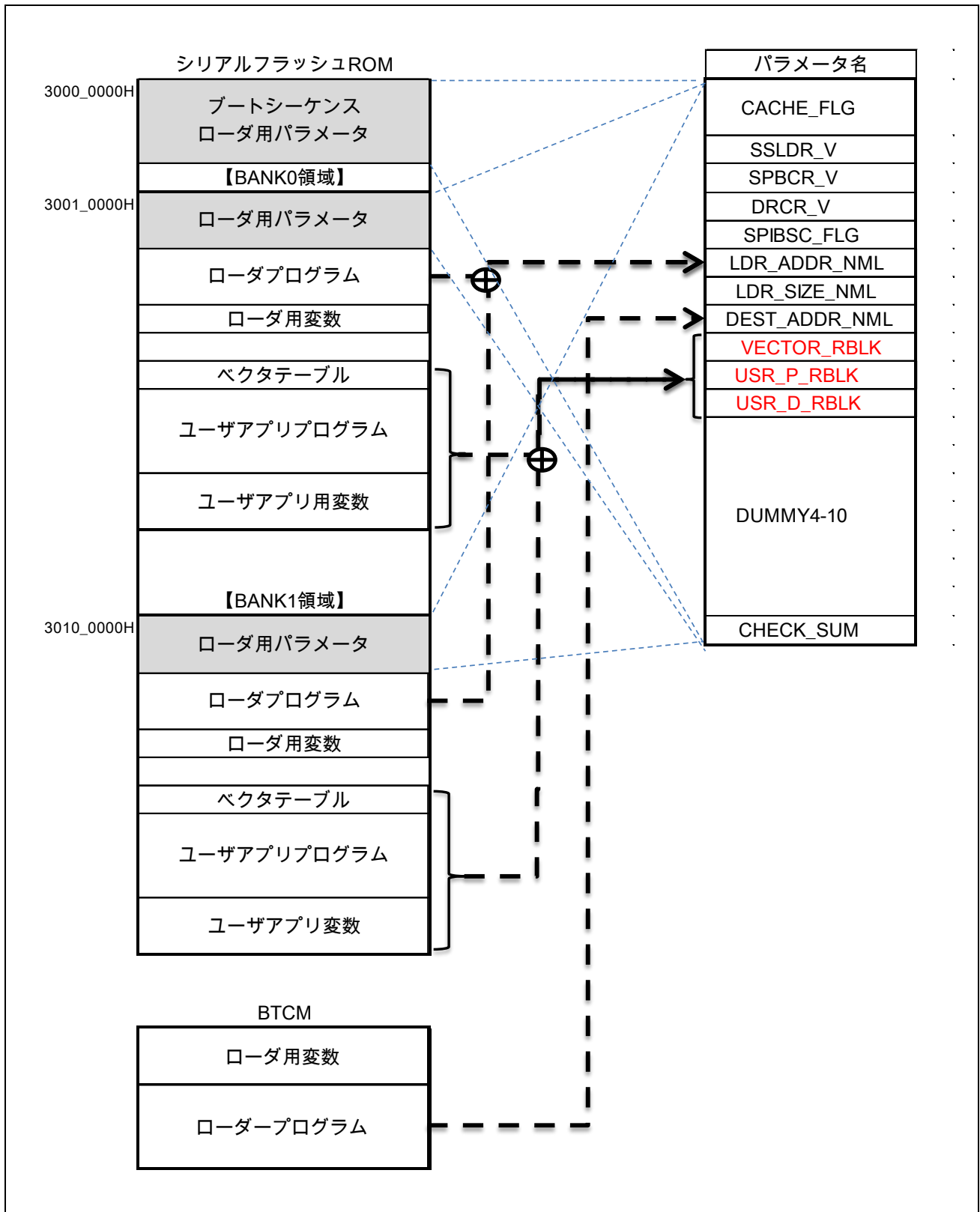


図 5-6 ローダ用パラメータの参照先

6. サンプルプログラムのセクション配置

サンプルプログラムのセクション配置を「図 6.1 サンプルプログラムのセクション配置」に示します。

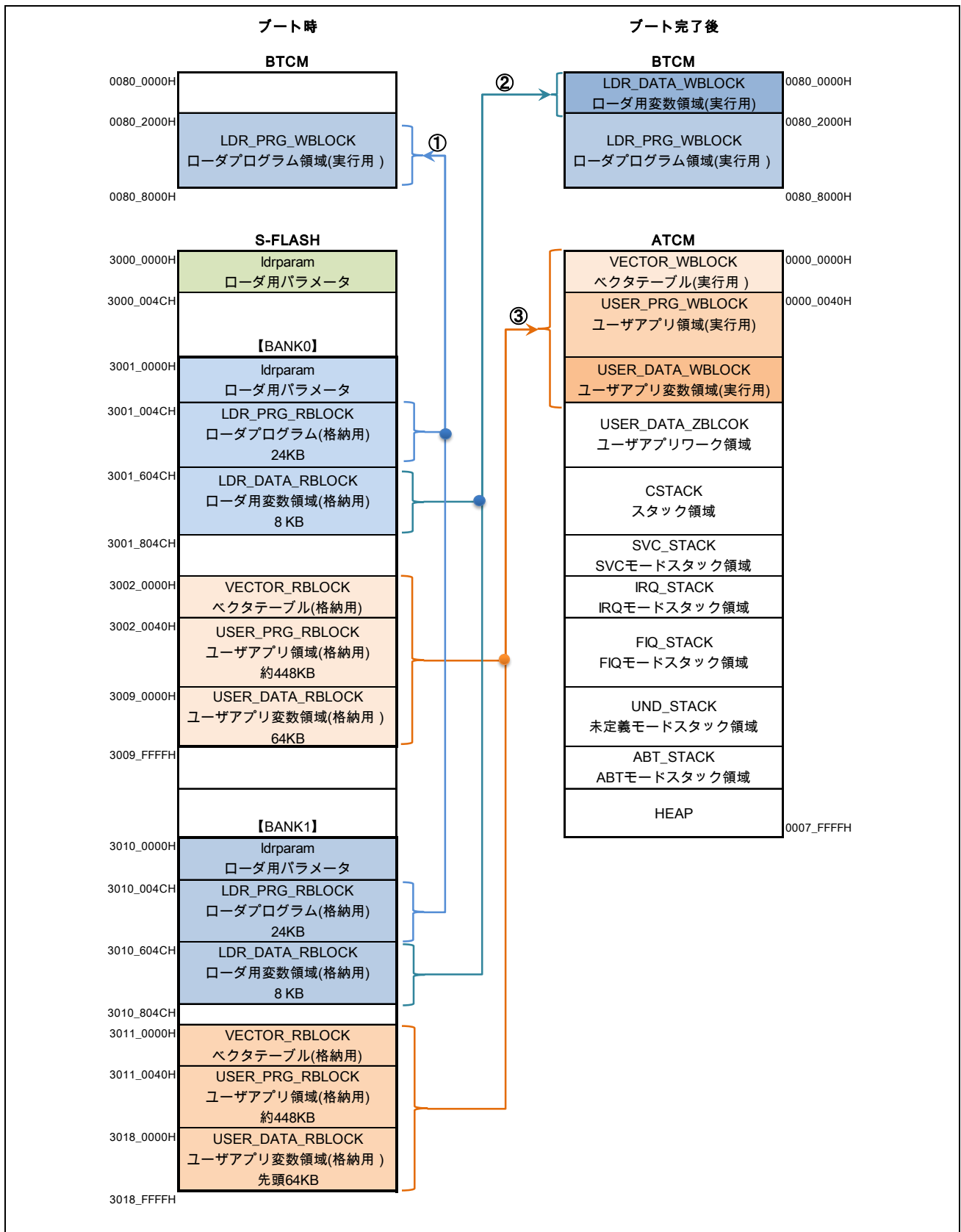


図 6-1 サンプルプログラムのセクション配置

7. サンプルプログラムのビルド構成

出荷時ファームウェアをシリアル・フラッシュ ROM の BANK0 または BANK1 に書き込むためのプロジェクトファイルと更新ファームウェアのダウンロード BIN ファイルを作成するためのプロジェクトファイルを用意しています。

「図 7.1 サンプルプログラムのビルド構成」にプロジェクトファイル、ビルド構成、使用ファイル、リンクするセクションとの関係を示します。

EWARM プロジェクトファイル	RZT1_FoE.eww EC_1_FoE.eww		
プロジェクト	BANK0	BANK1	BANK1
ビルド構成	Debug_BANK0	Debug_BANK1	Release_BANK1
使用icfファイル名	RZ_T1_FoE_serial_boot_BANK0.icf EC_1_FoE_serial_boot_BANK0.icf	RZ_T1_FoE_serial_boot_BANK1.icf EC_1_FoE_serial_boot_BANK1.icf	RZ_T1_FoE_download_BANK1.icf EC_1_FoE_download_BANK1.icf
用途	出荷時ファームウェアデバッグ用	更新ファームウェアデバッグ用	更新ファームウェア ダウンロードBINファイル作成用
3000_0000H	ldrparam BANK0ロード用パラメータ	ldrparam BANK1ロード用パラメータ	
3000_004CH			
3001_0000H	ldrparam ロード用パラメータ		
3001_004CH	LDR_PRG_RBLOCK ロードプログラム(格納用) 24K		
3001_604CH	LDR_DATA_RBLOCK ロード用変数領域(格納用) 8 K		
3001_804CH			
3002_0000H	VECTOR_RBLOCK ベクタテーブル(格納用)		
3002_0040H	USER_PRG_RBLOCK ユーザアプリ領域(格納用) 約448K		
3009_0000H	USER_DATA_RBLOCK ユーザアプリ変数領域(格納用) 64K		
3009_FFFFH			
3010_0000H		ldrparam ロード用パラメータ	ldrparam ロード用パラメータ
3010_004CH		LDR_PRG_RBLOCK ロードプログラム(格納用) 24K	LDR_PRG_RBLOCK ロードプログラム(格納用) 24K
3010_604CH		LDR_DATA_RBLOCK ロード用変数領域(格納用) 8 K	LDR_DATA_RBLOCK ロード用変数領域(格納用) 8 K
3010_804CH			
3011_0000H		VECTOR_RBLOCK ベクタテーブル(格納用)	VECTOR_RBLOCK ベクタテーブル(格納用)
3011_0040H		USER_PRG_RBLOCK ユーザアプリ領域(格納用) 約448K	USER_PRG_RBLOCK ユーザアプリ領域(格納用) 約448K
3018_0000H		USER_DATA_RBLOCK ユーザアプリ変数領域(格納用) 先頭64K	USER_DATA_RBLOCK ユーザアプリ変数領域(格納用) 先頭64K
3018_FFFFH			

図 7-1 サンプルプログラムのビルド構成

8. 定数一覧

表 8-1 サンプルプログラムで使用する定数(1)

定数名	設定値	内容
SPIBSC_LDR_ADDR	(0x10000014)	ローダ用パラメータの”LDR_ADDR_NML”が格納されているアドレス
SPIBSC_LDR_SIZE	(0x10000018)	ローダ用パラメータの”LDR_SIZE_NML”が格納されているアドレス
SPIBSC_DEST_ADDR	(0x1000001C)	ローダ用パラメータの”DEST_ADDR_NML”格納されているアドレス
SPIBSC_VCTR_ADDR	(0x10000020)	ローダ用パラメータの”VECTOR_RBLK”格納されているアドレス
SPIBSC_USRP_ADDR	(0x10000024)	ローダ用パラメータの”USR_P_RBLK”が格納されているアドレス
SPIBSC_USRD_ADDR	(0x10000028)	ローダ用パラメータの”USR_D_RBLK”が格納されているアドレス

表 8-2 サンプルプログラムで使用する定数(2)

定数名	設定値	内容
SF_PAGE_SIZE	(256)	シリアル・フラッシュ ROM のページサイズ
SF_SECTOR_SIZE	(65536)	シリアル・フラッシュ ROM のセクタサイズ(64KB)
SF_NUM_OF_SECTOR	(1024)	シリアル・フラッシュ ROM の総セクタ数
SF_FOE_BANK0_ADDR	(0x10010000)	BANK0 開始アドレス
SF_FOE_BANK1_ADDR	(0x10100000)	BANK1 開始アドレス
SF_FOE_APPLI_SIZE	(0x00090000)	BANK サイズ(576KB)

9. 関数一覧

ブートローダと FoE ファームウェア更新関連の関数について説明します。

表 9-1 ブートローダ関連 関数一覧

関数名	関数概要
copy_to_atcm	ブート時にユーザアプリケーションプログラムをシリアル・フラッシュ ROM から ATCM に展開する処理。BTCM 上で実行される。
copy_to_btcm	リブート時にローダプログラムを BTCM に展開する関数。ATCM 上で実行される。

表 9-2 FoE ファームウェア更新関連 関数一覧

関数名	関数概要
BL_Start	INIT->BOOT 遷移開始処理
BL_StartDownload	FoE ファイルデータ・ダウンロード開始処理
BL_Data	FoE ファイルデータ・受信処理
BL_CheckSum	更新ファームウェア領域チェックサム確認処理
BL_Data_write	ファイルデータ・シリアル・フラッシュ ROM 書き込み処理
BL_SetRebootFlag	再起動フラグ設定
BL_CheckRebootFlag	再起動フラグ確認
BL_Reboot	再起動処理(BOOT->INIT)
BL_Copy_1Page	シリアル・フラッシュ ROM 1 ページ分のデータコピー

10. サンプルプログラム・ソースファイルの作成

10.1 SSC Tool のインストール

ETG からライセンス許諾を得て SSC Tool を入手の上、インストールしてください。

本サンプルプログラムが想定している SSC Tool は Version5.12 です。

10.2 サンプルプログラムの解凍

サンプルプログラムファイルを解凍して下さい。

10.3 SSC ソースファイル作成

(1) サンプルプログラムに同梱されている SSC Tool プロジェクトファイル(*.esp)を実行して、SSC Tool を起動します。

・ RZ/T1

¥workspace¥xxxx¥EtherCAT_SSC_FoE¥src¥sample¥src¥ssc_project¥RZT1-R EtherCAT [FoE] s.esp

※ IAR EWARM : xxxxx = iccarm

Renesas e2Studio : xxxxx = kpitgcc

・ EC-1

¥Source¥Project¥EtherCAT_Comb_FoE¥SSC¥EC-1 [FoE].esp

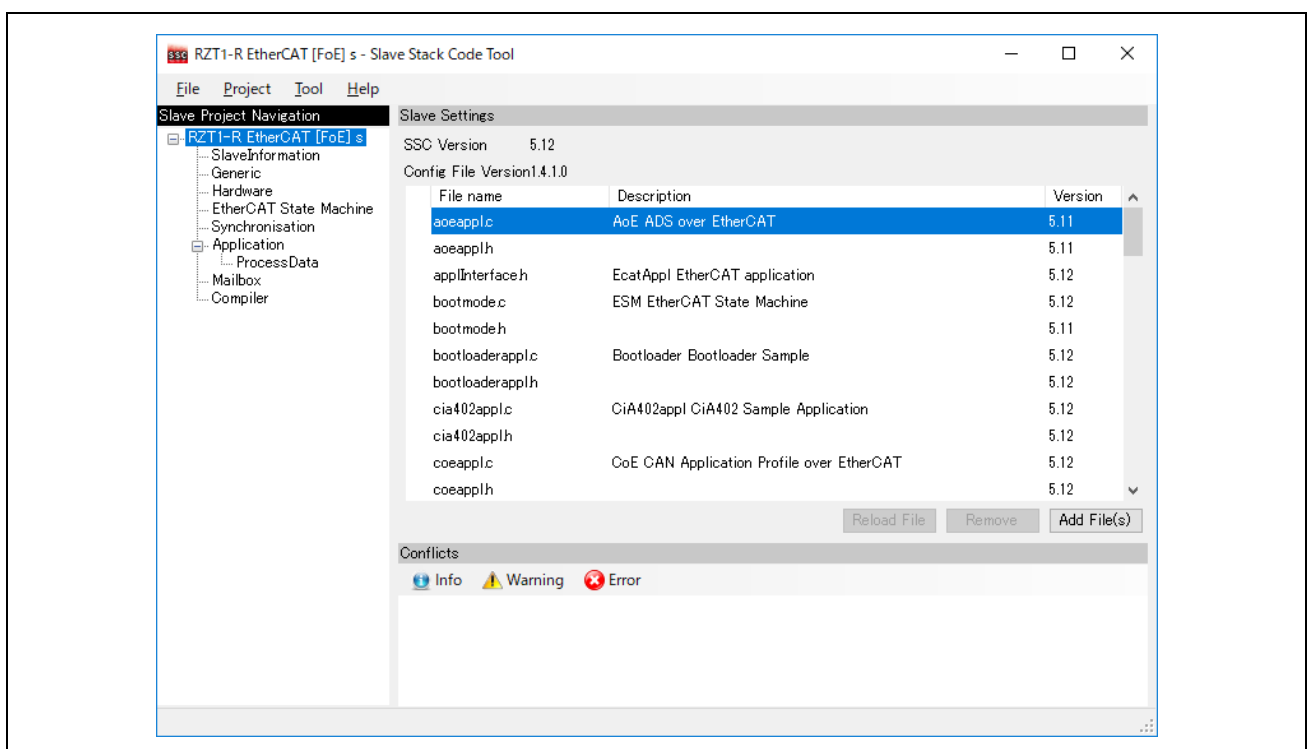


図 10-1 RZ/T1 SSC Tool 起動画面

(2) ソースファイルを作成します。

Menu [Project] -> [Create new Slave Files] -> "start"->"OK"

上記の通り操作を行うことで、“¥Src”ディレクトリとソースファイルが作成されます。

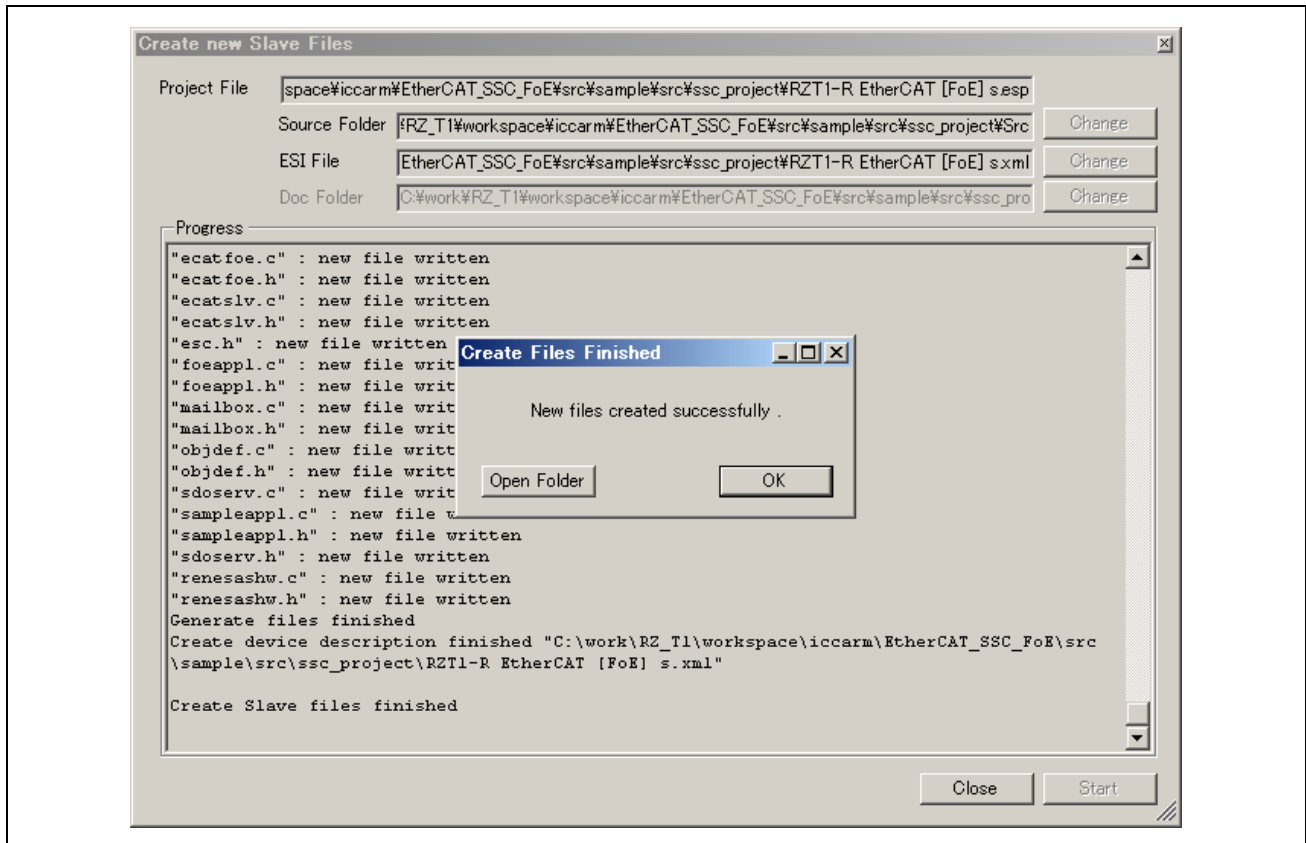


図 10-2 RZ/T1 SSC Tool ソースファイル作成画面

■注意

ソースファイル作成にあたり、次の設定はチェックしないでください。

“Tool”->“Options”->“Create Files” タブの“Add comments if obsolete code was skipped”

patch コマンドのインストール

PC に patch コマンドがインストールされていない場合、インストールしてください。

必要な patch コマンドは GNU Patch の Ver2.5.9 以後のバージョンになります。

下記のリンクから patch コマンド (Ver2.5.9) をダウンロードし、“patch.exe” をディレクトリの通っているパスに格納します。

<http://gnuwin32.sourceforge.net/packages/patch.htm>

10.4 bat ファイルの実行

bat ファイルは、SSC ソースファイルにブートローダ機能や FoE ファームウェア更新機能等の追加を行う、patch ファイルの適用を行います。

サンプルプログラムには patch ファイルの適用を行う”apply_patch.bat”が付属していますので、“apply_patch.bat”を実行して下さい。



```
--- Patching process start ---  
--- Move Src folder ---  
    1 個のディレクトリを移動しました。  
patching file Src/bootmode.c  
patching file Src/bootmode.h  
patching file Src/coeappl.c  
patching file Src/ecat_def.h  
patching file Src/ecatappl.c  
patching file Src/ecatfoe.h  
patching file Src/ecatslv.c  
patching file Src/foeappl.c  
patching file Src/mailbox.h  
patching file Src/objdef.h  
patching file Src/sampleappl.h  
--- Patching process end ---  
続行するには何かキーを押してください . . .
```

図 10-3 “apply_patch.bat” 実行画面

■注意

Windows7 で patch コマンドが実行できない場合

コマンドプロンプトのアイコンまたはショートカットを右クリックして、「管理者として実行」をクリックしてコマンドプロンプトを起動してください。

11. 動作確認

11.1 デバッガ起動(IAR EWARM)

同梱されている IAR プロジェクトファイルをダブルクリックして、IAR Embedded Workbench for Arm を起動します。

- ・ RZ/T1

¥workspace¥icarm¥EtherCAT_SSC_FoE¥RZ_T1_FoE.eww

- ・ EC-1

¥Source¥Project¥EtherCAT_ComB_FoE¥IAR¥EC_1_FoE.eww

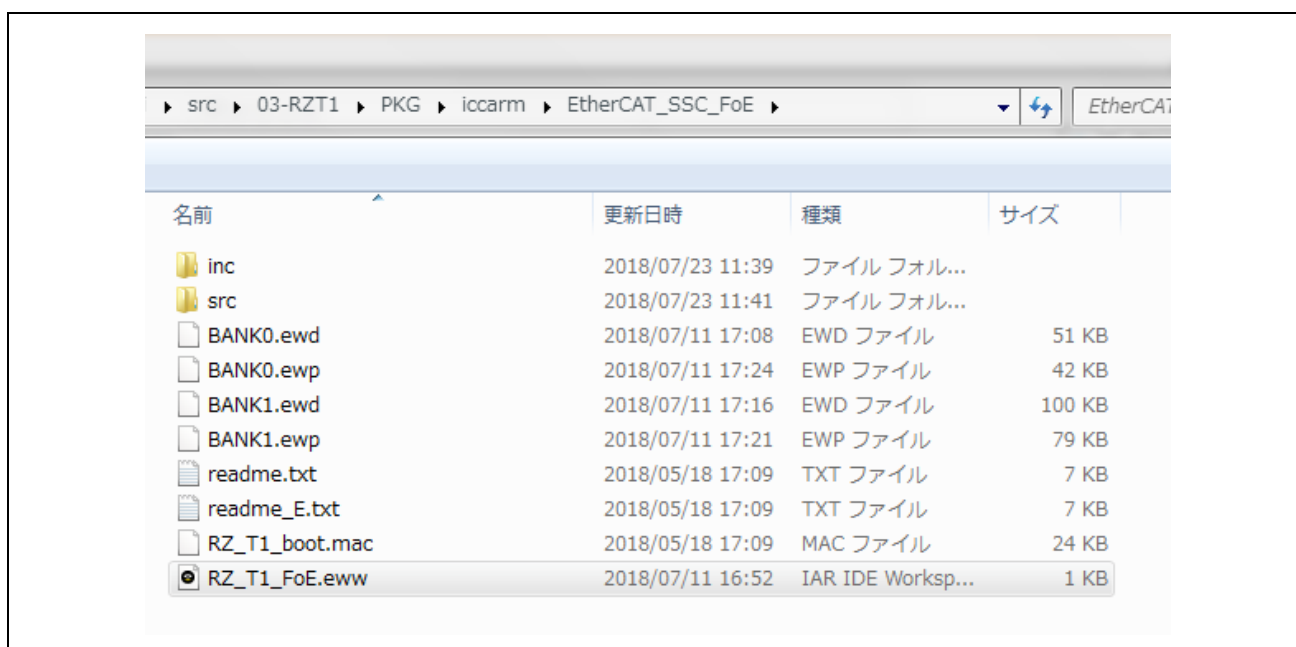


図 11-1 RZ/T1 IAR プロジェクトファイル ディレクトリ画面

ICE を評価ボードに接続してください。

11.1.1 BANK0 ビルドとデバッグ

- (1) 出荷時ファームウェアは” BANK0” プロジェクトを選択します。
- (2) “Debug_BANK0”モードを設定して「プロジェクト」->「すべてを再ビルド」でビルドします。
- (3) 「ダウンロードしてデバッグ」をダブルクリックしてシリアル・フラッシュ ROM の BANK0 に出荷時ファームウェアのコードを書き込みます。

エラー表示がなくデバッガ画面になれば成功です。

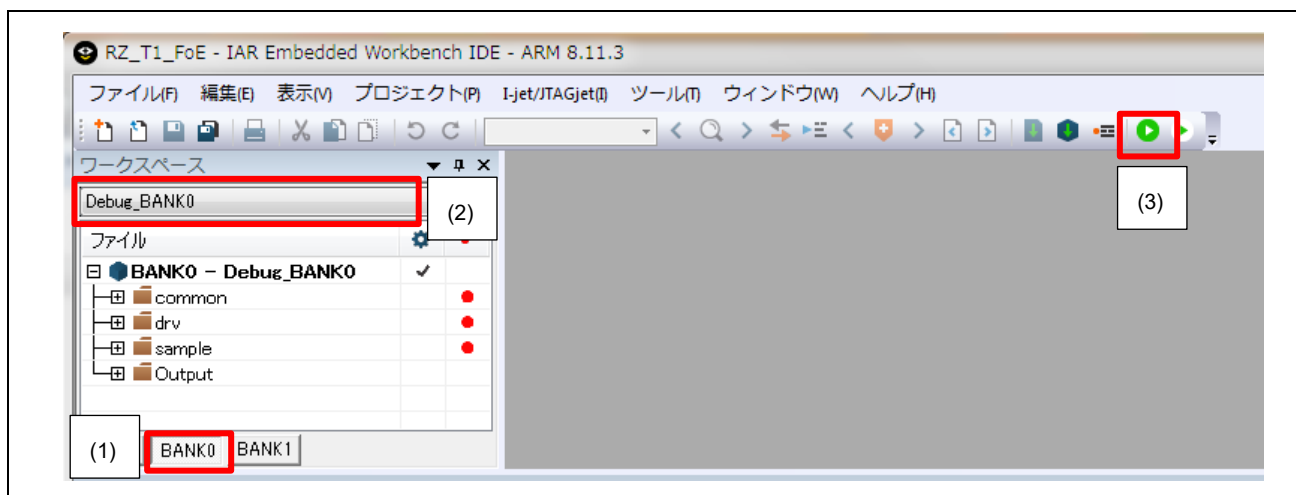


図 11-2 RZ/T1 Debug_BANK0 ビルド画面

11.1.2 BANK1 ビルドとデバッグ

- (1) 更新ファームウェアは” BANK1” プロジェクトを選択します。
- (2) “Debug_BANK1”モードを設定して「プロジェクト」->「すべてを再ビルド」でビルドします。
- (3) 「ダウンロードしてデバッグ」をダブルクリックしてシリアル・フラッシュ ROM の BANK1 に更新ファームウェアのコードを書き込みます。

エラー表示がなくデバッガ画面になれば成功です。

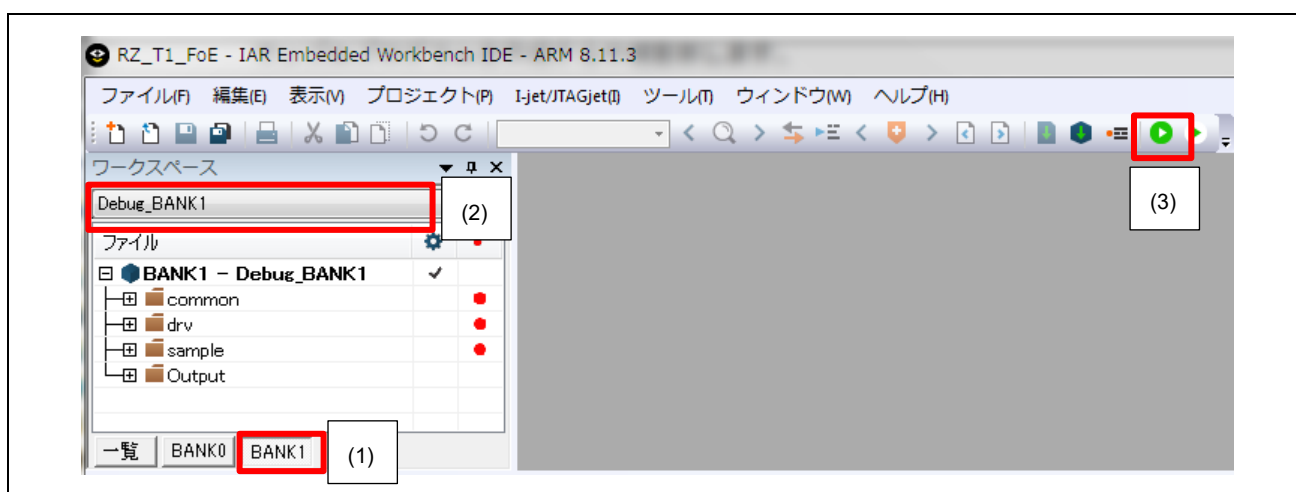


図 11-3 RZ/T1 Debug_BANK1 ビルド画面

11.1.3 BANK1 ダウンロードファイル作成

更新ファームウェアのデバッグが完了したらダウンロードファイルを作成します。

- (1) 更新ファームウェアは” BANK1” プロジェクトを選択します。
- (2) “Release_BANK1”モードを設定して「プロジェクト」->「すべてを再ビルド」でビルドします。

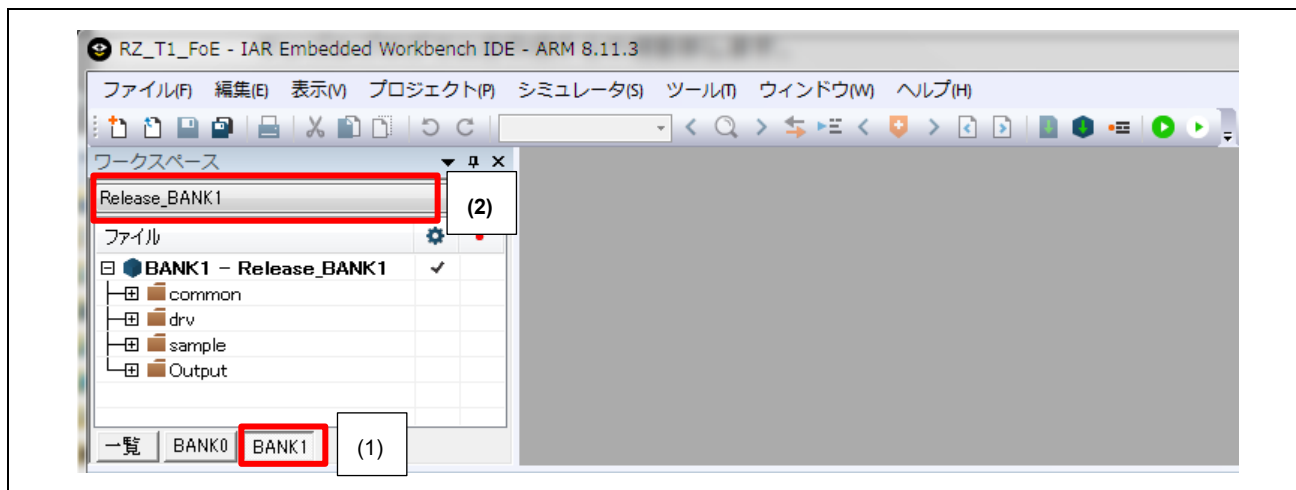


図 11-4 RZ/T1 Release_BANK1 ビルド画面

ビルドが終了すると更新ファームウェアのダウンロードファイルが出来上がります。

・ RZ/T1

¥workspace¥iccar¥EtherCAT_SSC_FoE¥Release_BANK1¥Exe¥ ECATFW__B1_FoE.efw

・ EC-1

¥Source¥Project¥EtherCAT_RemoteIO¥IAR¥ Release_BANK1¥Exe¥ ECATFW__B1_FoE.efw

更新ファームウェア・ファイルに関するパラメータのうち下記の項目はソースファイルにて変更可能です。

表 11-1 更新ファームウェア・ファイル パラメータ

パラメータ	概要	ソースファイル対応箇所
ファイル名接頭辞	文字列 : " ECATFW__B1"	sampleappl.c 内 aFileNameHeader
ファイルパスワード	数字 8 桁 : 00000000	sampleappl.c 内 aFilePassword
ファームウェアバージョン	RZ/T1 文字列 : "5.12" EC-1 文字列 : "1.01"	ecat_def.h 内 DEVICE_SW_VERSION

11.2 TwinCAT®起動

11.2.1 ESI ファイルの準備

同梱されている ESI ファイルを”C:¥TwinCAT¥Io¥EtherCAT”の下へコピーします。

- ・ RZ/T1

¥workspace¥icarm¥EtherCAT_SSC_FoE¥src¥sample¥src¥ESI_File¥RZT1-R EtherCAT [FoE] s.xml

- ・ EC-1

¥Source¥Project¥EtherCAT_Comb_FoE¥SSC¥ESI_File¥EC-1 [FoE].xml

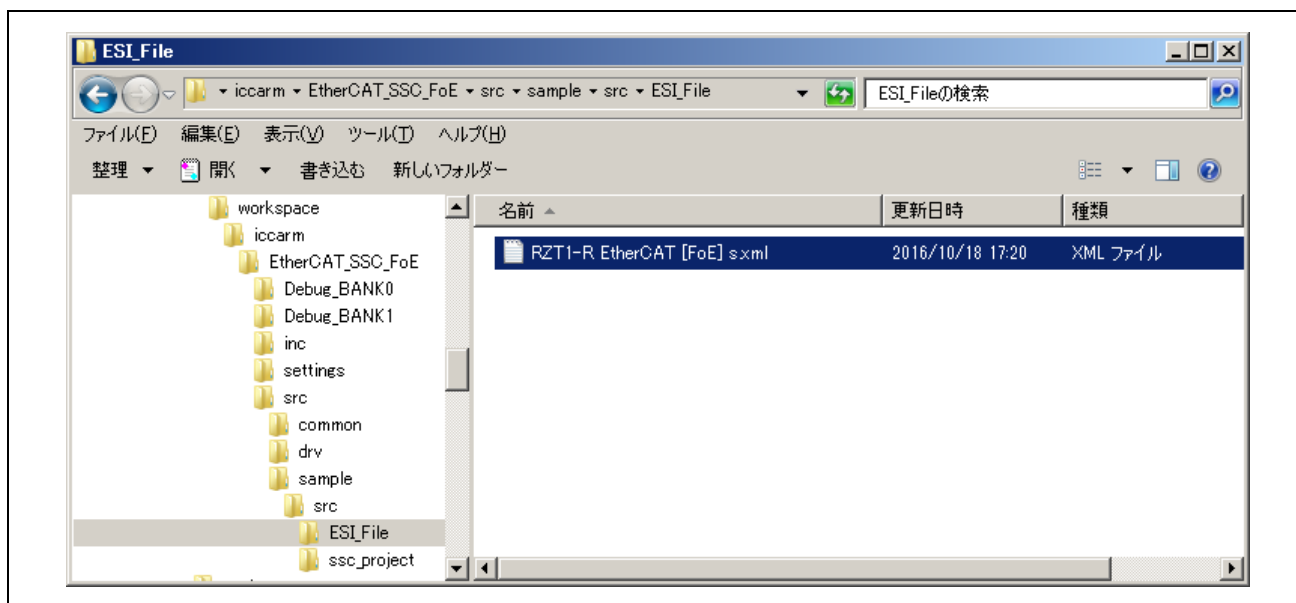


図 11-5 RZ/T1 ESI ファイル

11.2.2 TwinCAT®起動

TwinCAT System Manager を起動します。

“I/O Device”を右クリックし、“Scan Devices...”を選択します。

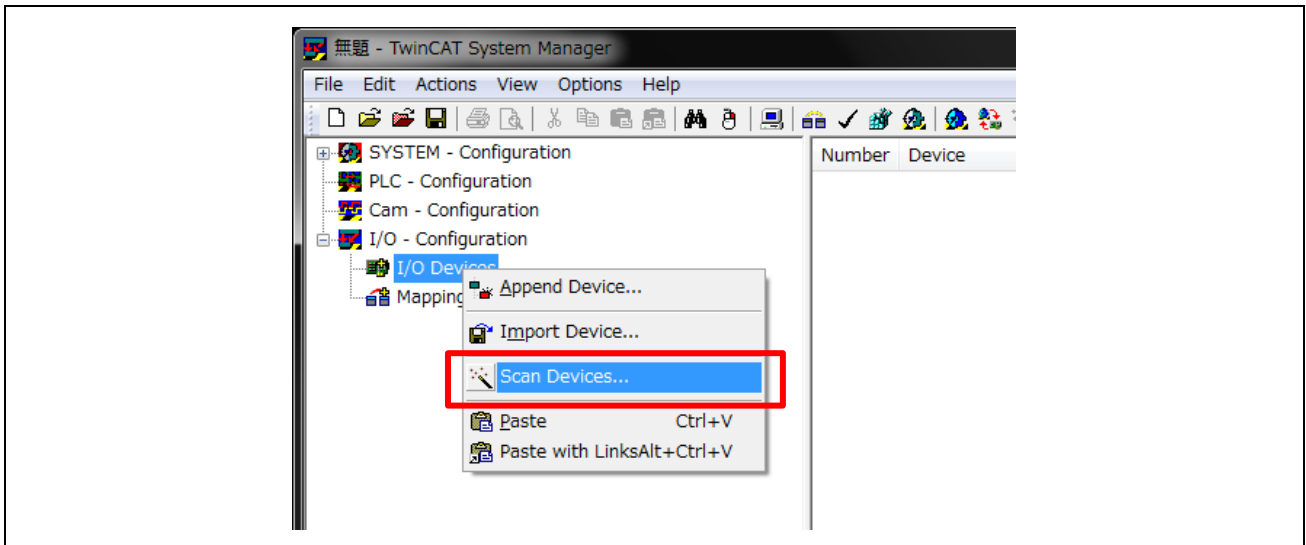


図 11-6 デバイス検索 1

“OK”を選択します。

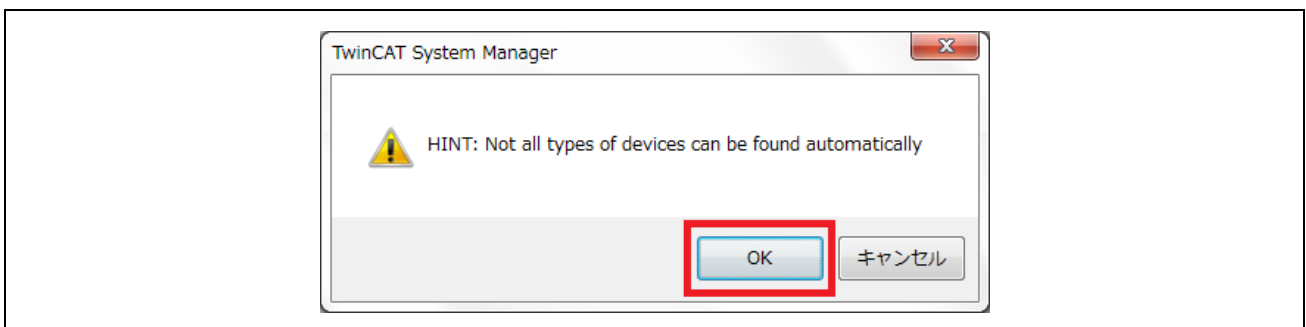


図 11-7 デバイス検索 2

“EtherCAT”のみにチェックを入れて、“OK”を選択します。

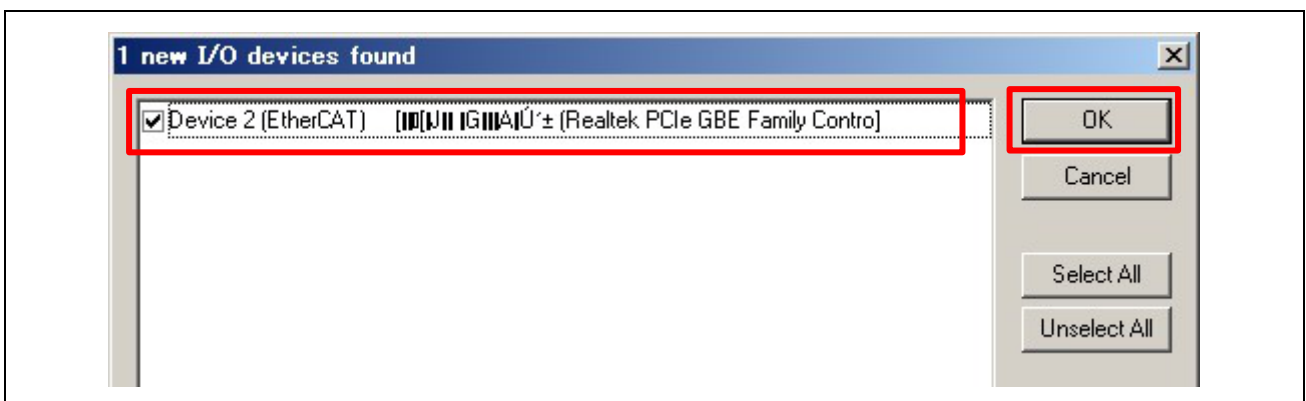


図 11-8 デバイス検索 3

“OK”ボタンをクリックします。



図 11-9 デバイス検索 4

“OK”ボタンをクリックし、Free Run 開始します。

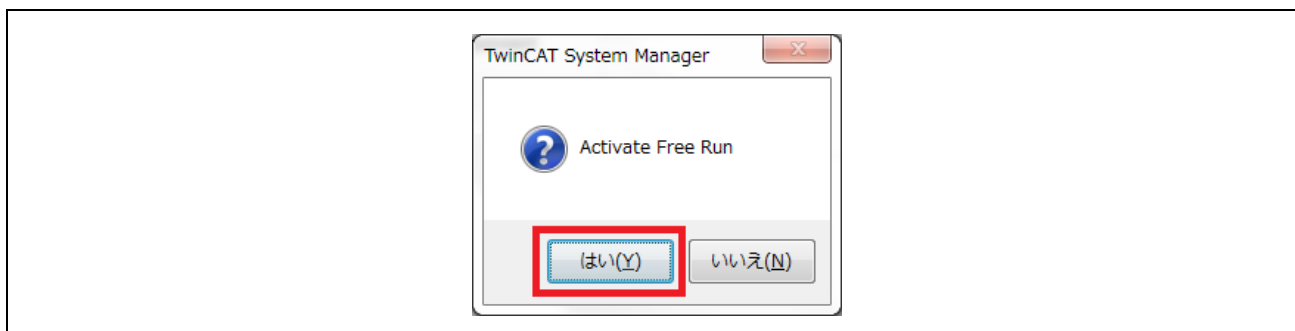


図 11-10 デバイス検索 5

11.2.3 ESI ファイルの書込み

Box 名が下記のデバイス名になっている場合、ESI ファイルを書き込み済みになります。
本項目は実施せず、11.4 を行って下さい。異なるデバイス名の場合は、以下を実施します。

デバイス名

- ・ RZ/T1 : “RZ/T1-R EtherCAT FoE”
- ・ EC-1 : ”EC-1 FoE”

“Box1”を選択し、“EtherCAT”タブを選択、“Advanced Settings...”ボタンをクリックします。

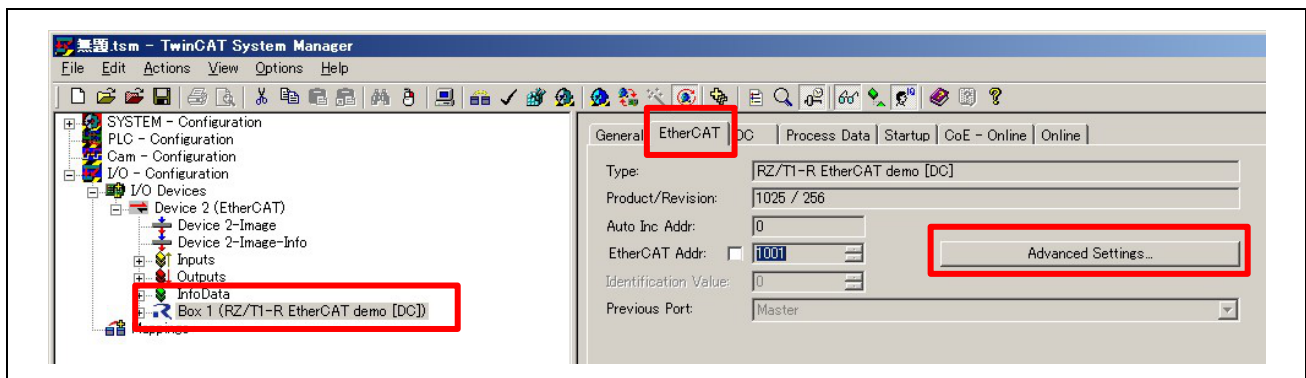


図 11-11 RZ/T1 ESI ファイル書込み 1

“Hex Editor”を選択し、“Download from List...”ボタンをクリックします。

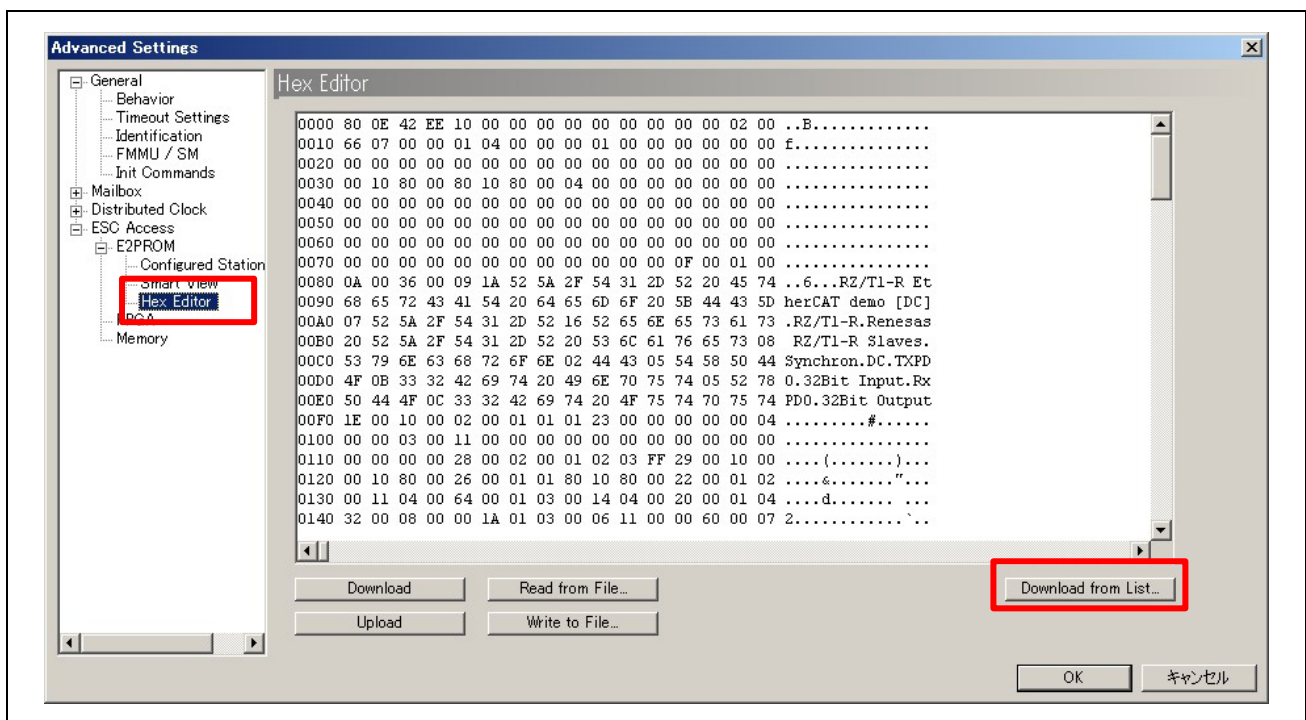


図 11-12 RZ/T1 ESI ファイル書込み 2

書き込む ESI ファイルのデバイス名を選択し、“OK”ボタンをクリックします。

RZ/T1、EC-1 それぞれのデバイス名は下記を参照。

※EEPROM への書き込みが行われるので時間がかかります。

デバイス名

- ・ RZ/T1 : “RZ/T1-R EtherCAT FoE”
- ・ EC-1 : “EC-1 FoE”

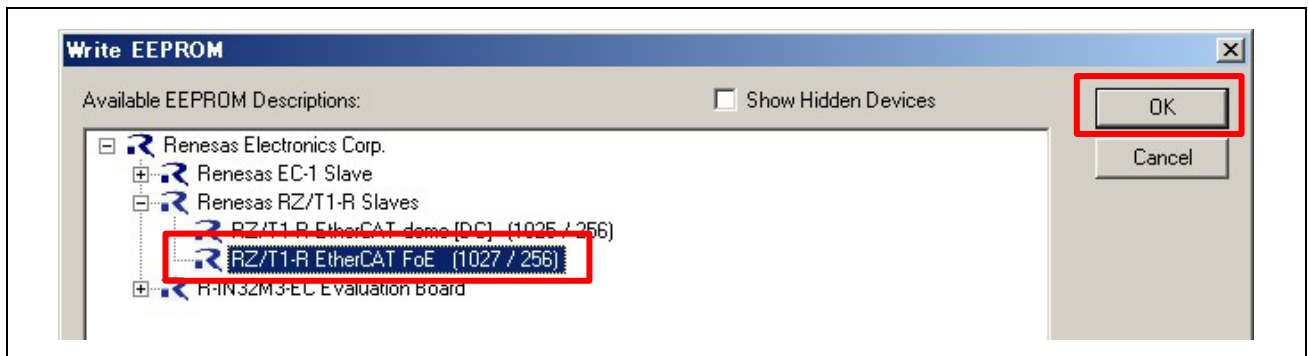


図 11-13 RZ/T1 ESI ファイル選択

“OK”ボタンをクリックします。ESI ファイルの内容が EEPROM への書き込みが完了しました。

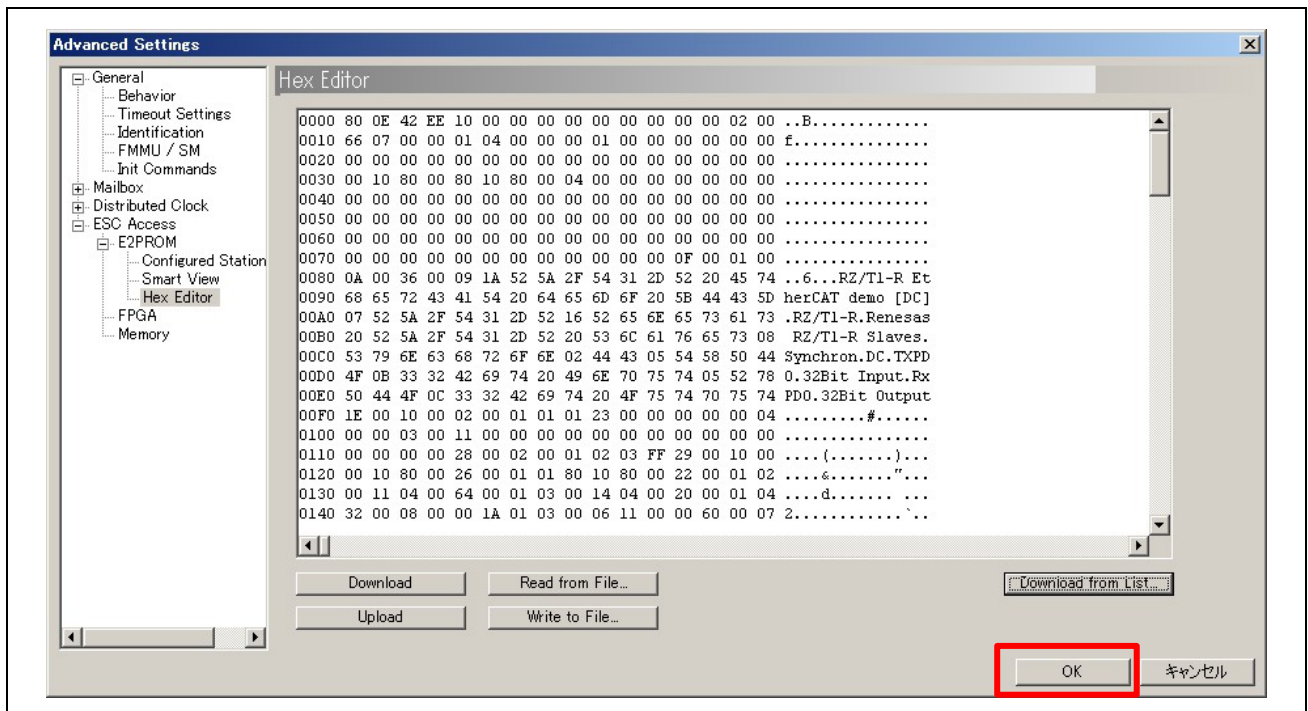


図 11-14 RZ/T1 ESI ファイル書き込み完了

ESI ファイルの書き込み後、デバイスの再検出を行うため、“I/O Devices”の下の Device を右クリックし、“Delete Device”を選択します。

デバイスの削除後、11.3.2 のデバイス検索からやり直してください。

Box 名が書き込んだ ESI のデバイス名となっていれば、11.4 を実施します。

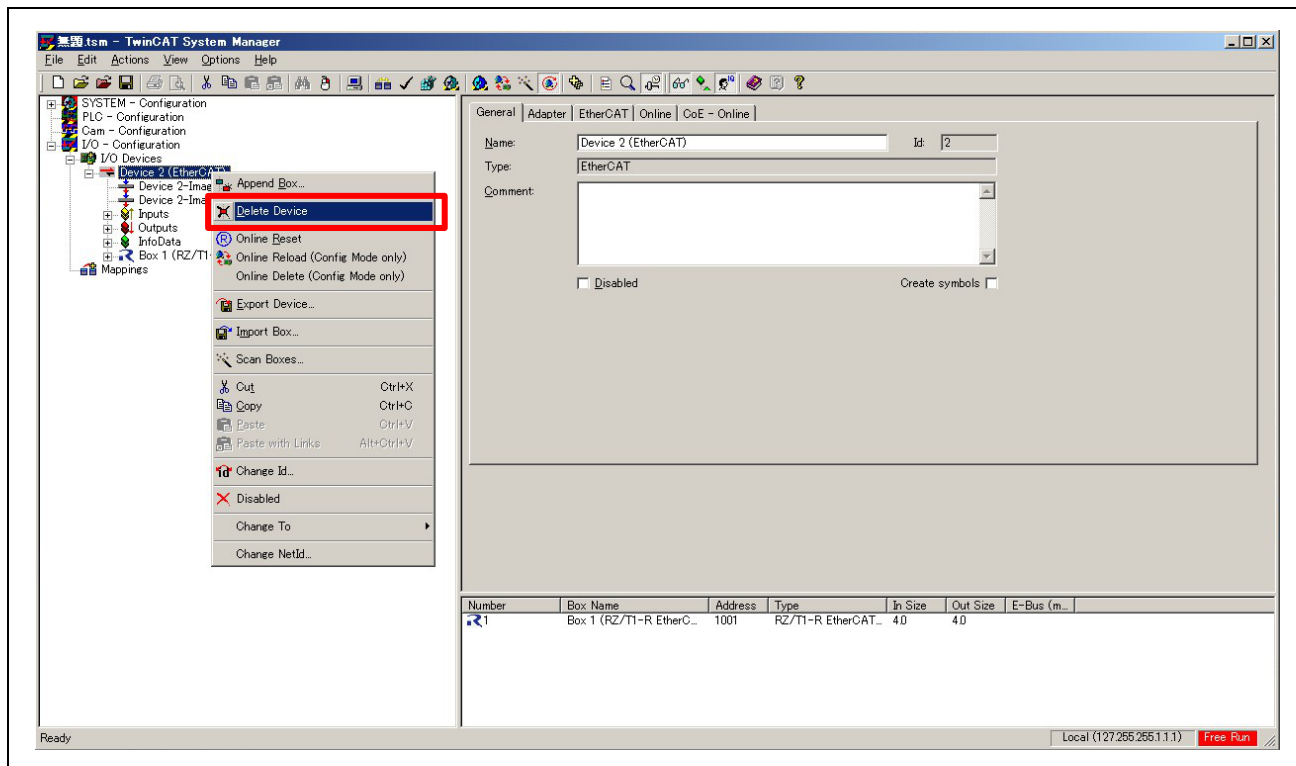


図 11-15 RZ/T1 デバイス削除

11.3 TwinCAT®によるファームウェア更新

“Box 1 (RZ/T1-R EtherCAT FoE または EC-1 FoE)”を選択、“Online”タブをクリックします。

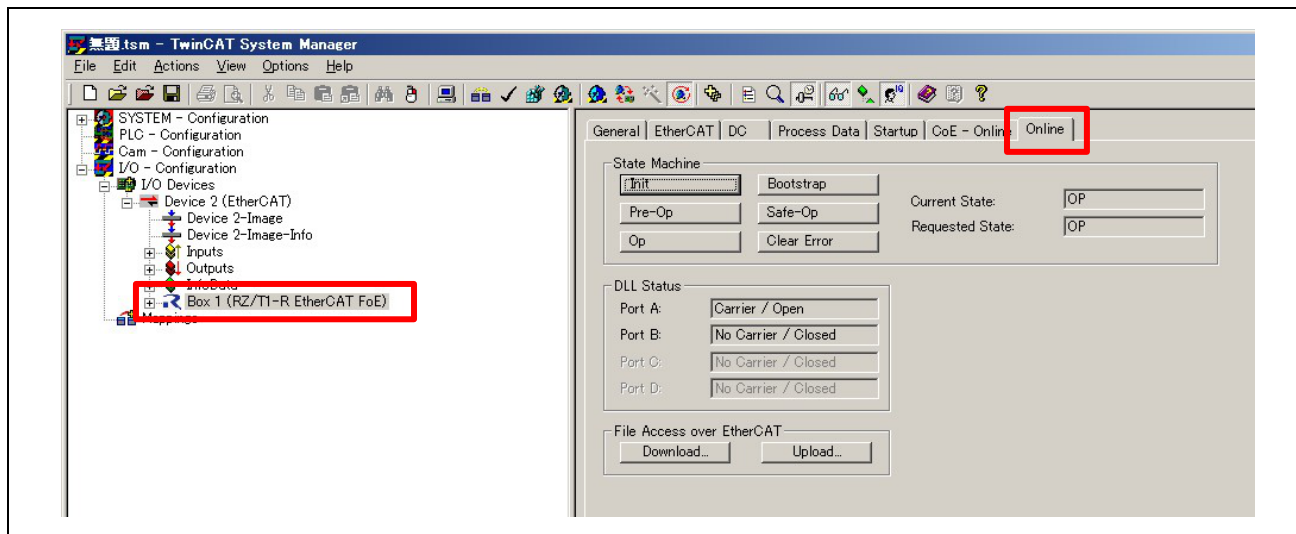


図 11-16 RZ/T1 “Online”タブ画面 1

(1) “Init”ボタン->(2) “Bootstrap”ボタンを順に押し、Current State が(3) “BOOT”に遷移することを確認してください。

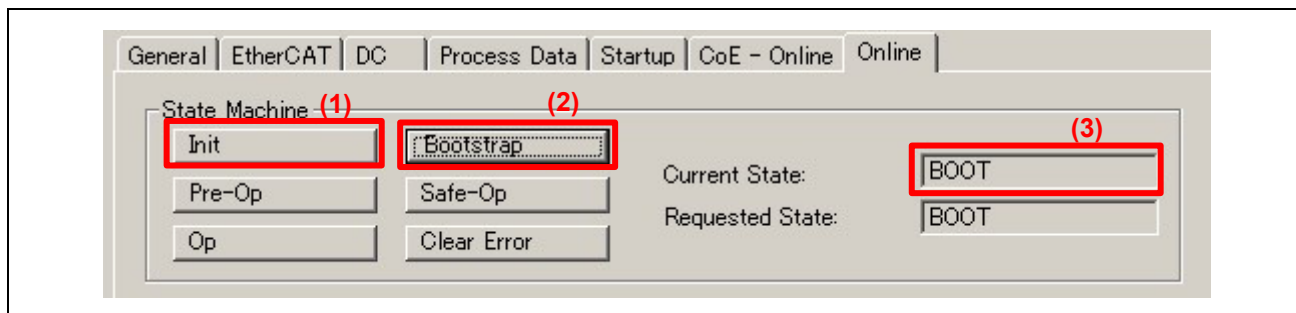


図 11-17 RZ/T1 “Online”タブ画面 2

次に File Access over EtherCAT の”Download”ボタンを押すと、ダウンロードファイルの選択ウインドウが開きます。更新ファームウェア・ファイルを選び、「開く」を押してください。

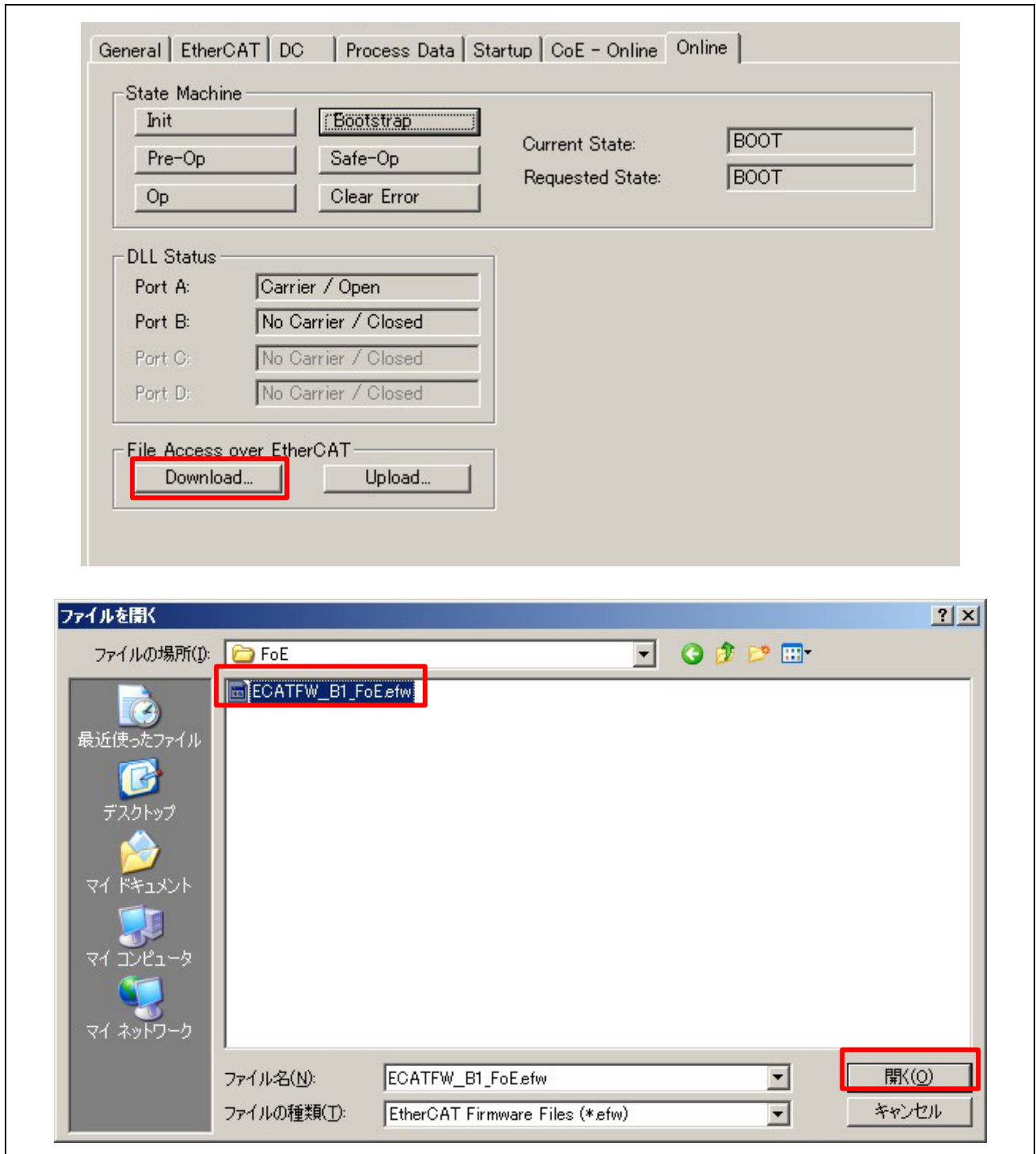


図 11-18 RZ/T1 更新ファームウェア・ファイル選択画面

ファイル名編集ウィンドウが開きます。
パスワードは"00000000"のまま"OK"を押します。

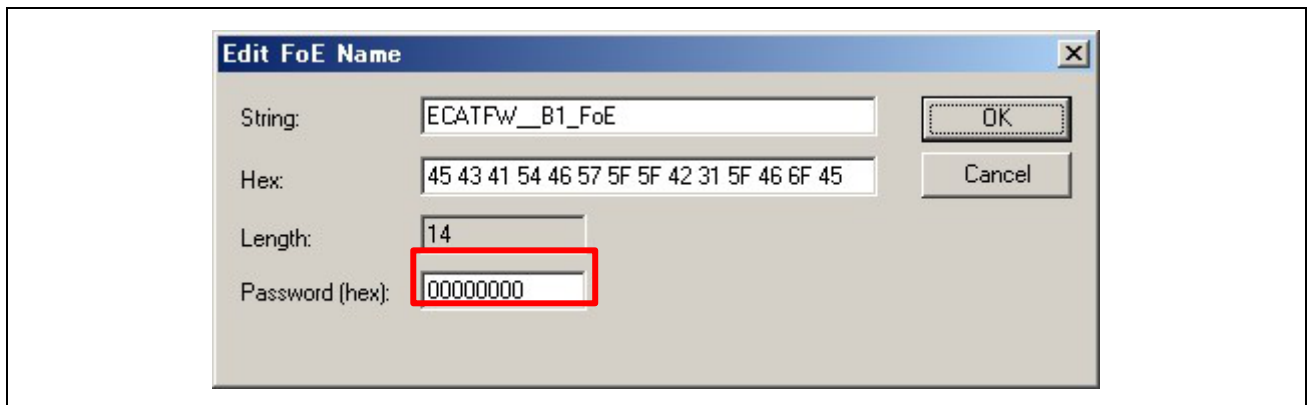


図 11-19 RZ/T1 更新ファームウェア・ファイル名編集画面

TwinCAT System Manager の画面最下部左側に"Downloading"のメッセージと共にダウンロード状況が表示されます。エラーメッセージが表示されず、上のウィンドウ(図 11.21)が消えて"Ready"になれば、ファームウェア更新の成功です。

"Online"タブで(1)"Init"ボタンを押すと更新されたファームウェアで再起動します。

(2)"Preop"ボタン -> (3)"Op"ボタンと押し、遷移させることで Current State が"OP"に遷移し、動作を確認することができます。

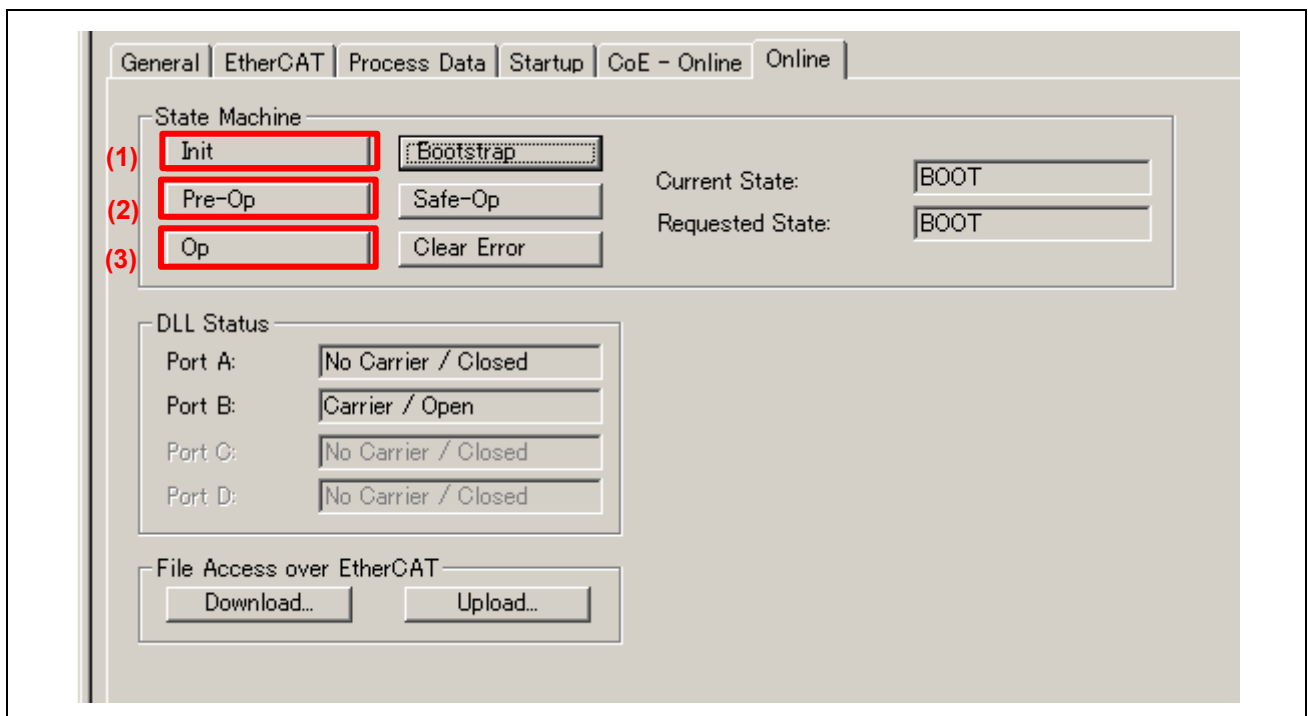


図 11-20 RZ/T1 "Online"タブ画面 3

ファームウェアのバージョンは、“CoE - Online”画面の 0x100A で確認することができます。(1)
また Revision は 0x1018:03 で確認することができます。(2)

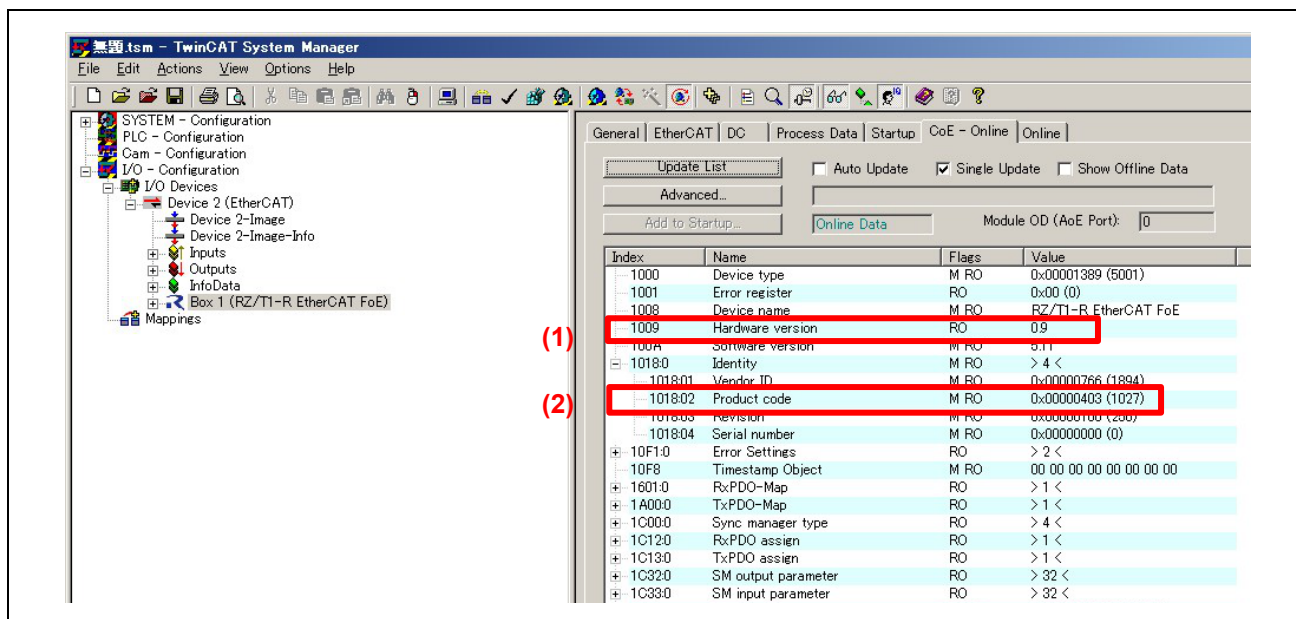


図 11-21 RZ/T1 “CoE”タブ画面

11.4 TwinCAT®による更新ファームウェア読み出し

FoE サービスを利用して BANK1 領域に格納されている更新ファームウェアのバイナリデータを読み出すことができます。

■注意 BANK0 領域に格納されている出荷時ファームウェアは読み出すことはできません。

バイナリデータを格納するアップロードファイルのパラメータを示します。

表 11-2 アップロードファイル パラメータ

パラメータ	概要	ソースファイル対応箇所
ファイル名接頭辞	文字列 : " ECATFW__B1"	sampleappl.c 内 aFileNameHeader
ファイルパスワード	数字 8 桁 : 00000000	sampleappl.c 内 aFilePassword

読み出し手順を示します。

“Box 1 (RZ/T1-R EtherCAT FoE または EC-1 FoE)” を選択、”Online” タブをクリックします。

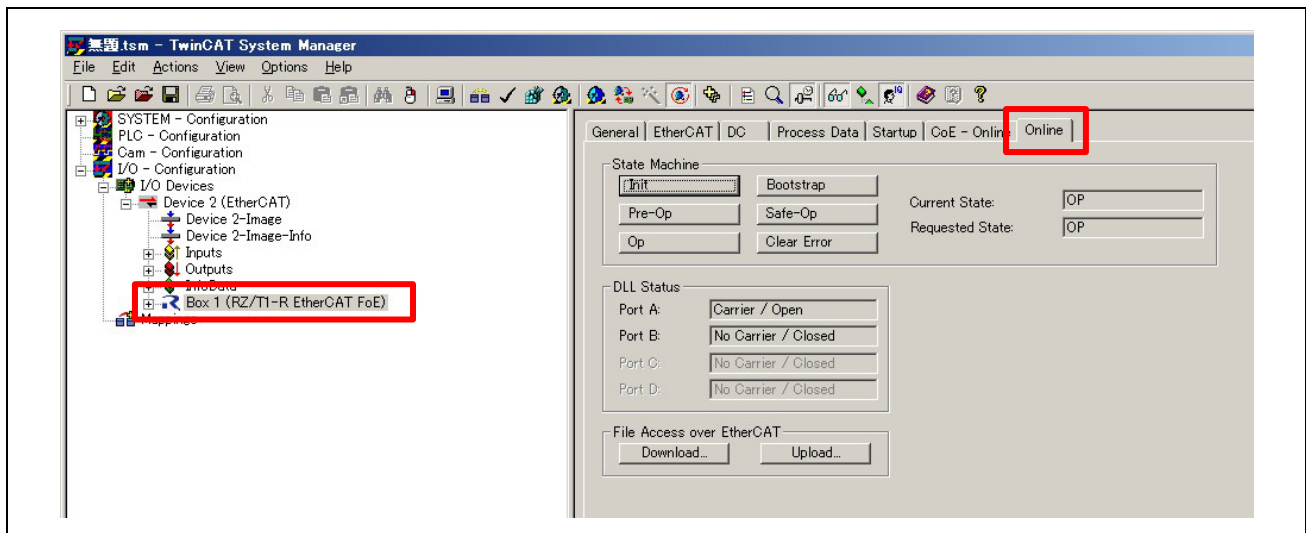


図 11-22 RZ/T1 “Online”タブ画面 1

(1) “Init”ボタン->(2)“Bootstrap”ボタンを順に押し、Current State が(3)“BOOT”に遷移することを確認してください。

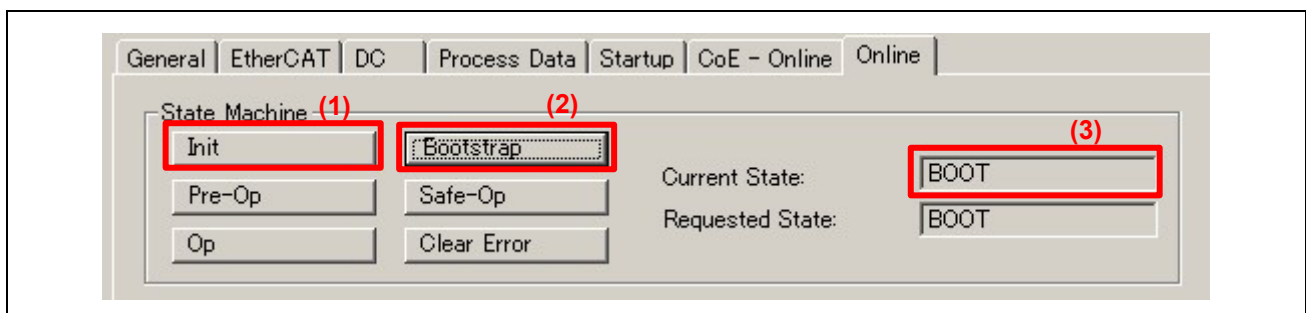


図 11-23 RZ/T1 “Online”タブ画面 2

次に File Access over EtherCAT の"Upload"ボタンを押すと、アップロードファイルの保存ウインドウが開きます。アップロードファイル名を入力し、「保存」を押してください。

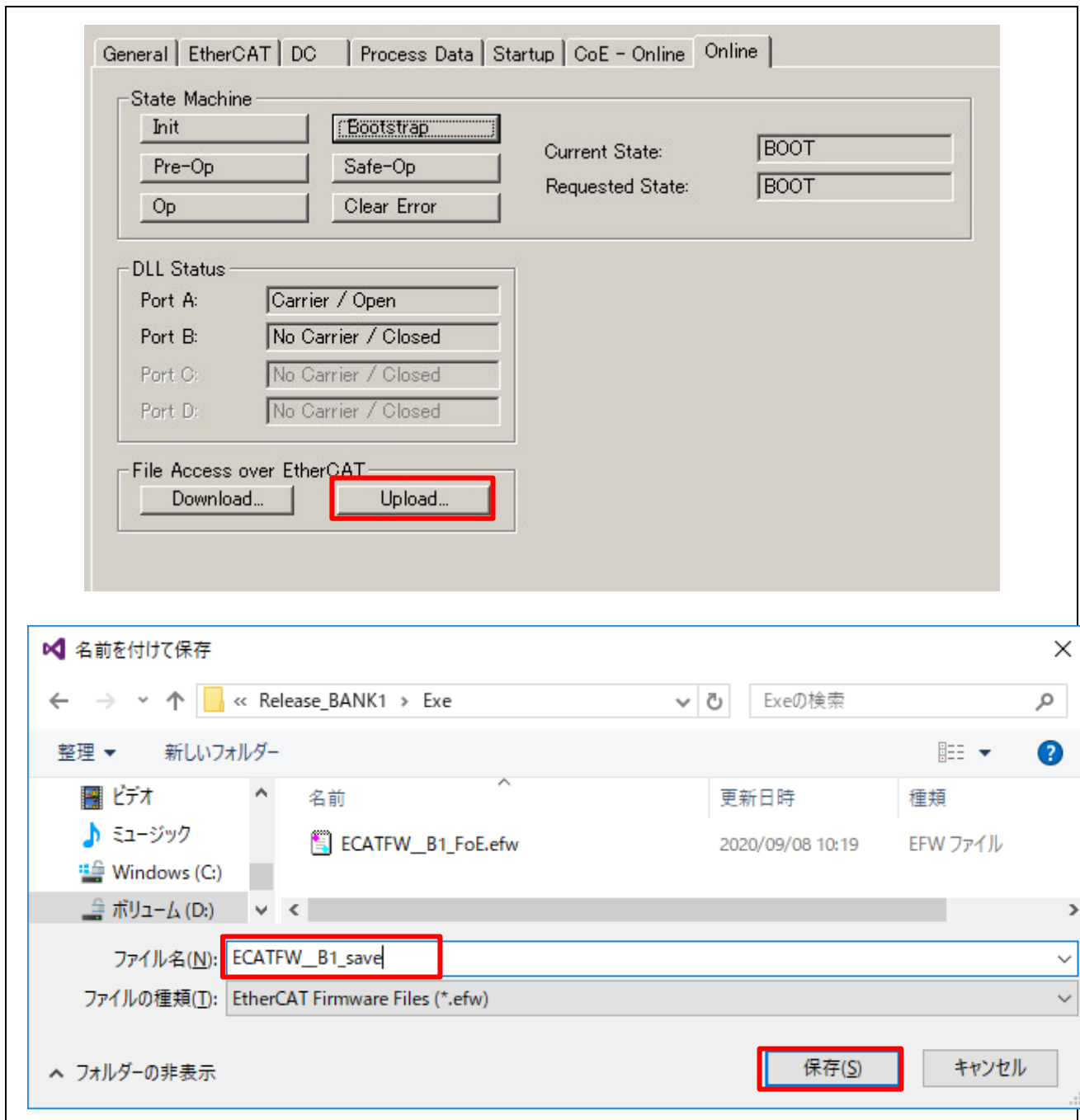


図 11-24 RZ/T1 アップロードファイル選択画面

ファイル名編集ウィンドウが開きます。
パスワードは"00000000"のまま"OK"を押します。

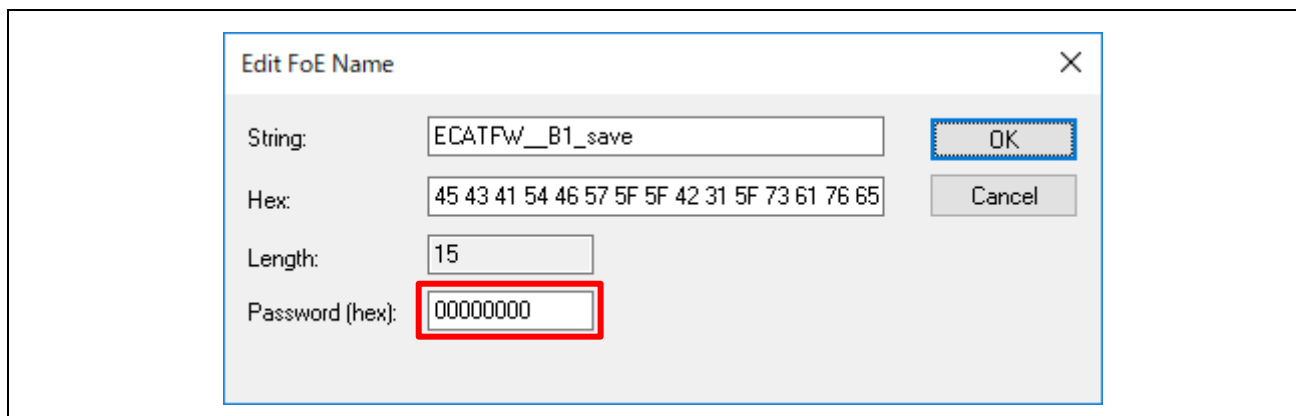


図 11-25 RZ/T1 アップロードファイル名編集画面

TwinCAT System Manager の画面最下部左側に"Uploading"のメッセージと共にアップロード状況が表示されます。エラーメッセージが表示されず、上のウィンドウ(図 11-41)が消えて"Ready"になれば、アップロードの成功です。

アップロードファイル(ECATFW_B1_save.efw)と更新ファームウェア・ファイル(ECATFW_B1_FoE.efw)をバイナリ比較すると一致していることが確認できます。

12. Common Device Profile (ETG5003.1)

EtherCAT にて半導体デバイスを取り扱う場合は、ETG5003 の仕様に規定されたデバイスプロファイルをサポートする必要があります。

ETG.5003 の構成は以下の内容となります。

1. Common Device Profile(CDP) [ETG.5003.1]
2. Firmware update functionality [ETG.5003.2]
3. Specific Device Profile(SDP) [ETG.5003.2xxx]

Common Device Profile (CDP) は、Specific Device Profile (SDP)で説明されているすべてのデバイスに適用される要件を指定します。

サンプルプログラムでは CDP [ETG.5003.1 Ver1.1.0] Appendix A 相当のオブジェクトディクショナリ定義を提供します。CDP 定義の個々のアドレスについては、ご使用になる SDP に応じて要否をご検討ください。

また、サンプルプログラムでの提供はオブジェクトディクショナリ定義の枠組みのみとなります。設定や必要処理に関しては別途検討・実装してください。

CDP 定義は下記に追加されています。

表 12-1 Common Device Profile での変更ファイル一覧

ファイル名	追加/変更箇所
coeappl.c	asEntryDesc0x1C3x[] (0xD, 0xE)へ型定義を追加
sampleappl.h	ApplicationObjDic[] へCDP定義を追加 CDPの各種アドレス定義、設定値を追加
objdef.h	TSYNCMANPAR定義を変更
<ul style="list-style-type: none"> ・ RZ/T1 RZT1-R EtherCAT [FoE]s.xml ・ EC-1 EC-1 [FoE].xml 	CDPの各種Datatype定義、Object定義を追加

Common Device Profile Ver1.1.0 については、下記の ETG.5003.1 規格書を参照ください。

また、CDP に関するご質問は、ETG 協会へお問い合わせください。

ETG5003.1 規格書 :

ETG5003-1 S (R) V1.1.0

EtherCAT Semiconductor Device Profile

Part1 Common Device Profile

13. 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://www.renesas.com/>

お問い合わせ先

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

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。

改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2017.05.22	-	初版発行
2.00	2020.09.30	13	5.4 BANK0 ローダ用パラメータをコピーする手順を追加 USER_DATA_WBLOCK/USER_DATA_RBLOCK のアドレスを削除 ソースファイル対応箇所のファイル名とシンボル名を修正
		20	
		30,41	
		4	表 1-1 サポートする FoE サービスにファイル読み出しを追加
		52-54	11.5 TwinCAT による更新ファームウェア読み出しを追加
		27-30	11.1 デバッガ起動をデバッガ起動(IAR EWARM)に変更
		55	12. Common Device Profile (ETG5003.1)の記載を変更
		13	5.4 BANK0 ローダ用パラメータをコピーする手順を追加

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、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

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. 電源オフ時における入力信号

当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れしないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

未使用端子は、「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。

5. クロックについて

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、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。

6. 入力端子の印加波形

入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、 $V_{IL}(\text{Max.})$ から $V_{IH}(\text{Min.})$ までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

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、将来の拡張機能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。

8. 製品間の相違について

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っていると、フラッシュメモリ、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。

○Arm® および Cortex® は、Arm Limited（またはその子会社）の EU またはその他の国における登録商標です。All rights reserved.

○Ethernet およびイーサネットは、富士ゼロックス株式会社の登録商標です。

○IEEE は、the Institute of Electrical and Electronics Engineers, Inc. の登録商標です。

○TRON は” The Real-time Operation system Nucleus” の略称です。

○ITRON は” Industrial TRON” の略称です。

○ μ ITRON は” Micro Industrial TRON” の略称です。

○TRON、ITRON、および μ ITRON は、特定の商品ないし商品群を指す名称ではありません。

○EtherCAT®は、ドイツ Beckhoff Automation GmbH によりライセンスされた特許取得済み技術であり登録商標です。

○その他、本資料中の製品名やサービス名は全てそれぞれの所有者に属する商標または登録商標です。

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生じた損害（お客様または第三者いずれに生じた損害も含まれます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品、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またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 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、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング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）、交通管制（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、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（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。

6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合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、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他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。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 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。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものとなります。
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。
12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

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が直接的、間接的に支配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。

(Rev.4.0-1 2017.11)

本社所在地

〒135-0061 東京都江東区豊洲3-2-24（豊洲フォレシア）

www.renesas.com

お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。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。