

## RZ/T2M, RZ/T2L グループ

Safety Motor Control Reference Kit スタートアップマニュアル(EtherCAT/FSoE)

### 要旨

本書は、ルネサスエレクトロニクス製 RZ/T2M, RZ/T2L グループの MPU を搭載した RZ/T2M&RZ/T2L Safety Motor Control Reference Kit のスタートアップマニュアルです。

### 動作確認デバイス

RZ/T2M, RZ/T2L 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. 概要	6
1.1 RZ/T2M&RZ/T2L Safety Motor Control Reference Kit 概要	6
1.2 接続構成	6
1.3 接続機器	7
1.3.1 セーフティモータ制御ボードと Inverter Board との接続	7
1.3.2 モータ/エンコーダと Inverter Board との接続	8
1.3.3 PC との接続	8
1.3.4 ICE	8
1.3.5 RZ/T2M / RZ/T2L_A/ RZ/T2L_B	8
1.3.6 電源	8
2. 動作環境	9
2.1 RZ/T2M	9
2.2 RZ/T2L_A, RZ/T2L_B	10
2.3 フォルダ、ファイル構成	10
3. 関連ドキュメント	11
4. ボードの設定	12
4.1 セーフティモータ制御ボードの設定	12
4.1.1 Switches	12
4.1.2 Jumpers	13
4.2 Inverter Board	13
4.2.1 Jumpers	13
5. プログラム書き込み	14
5.1 RZ/T2M	14
5.1.1 EtherCAT SubDevice Stack の生成	14
5.1.2 プロジェクトのビルド	16
5.1.3 プロジェクトの書き込み	19
5.2 RZ/T2L	20
5.2.1 開発ツール	20
5.2.2 ワークスペースフォルダのコピー	21
5.2.2.1 SIL3 システムソフトウェアキットのコピー	22
5.2.2.2 FSoE アプリケーションソフトウェアキットのコピー	23
5.2.2.3 セルフテストソフトウェアキットのコピー	25
5.2.2.4 セーフティモータ制御ボード用のパッチプログラムのコピー	25
5.2.3 ワークスペース起動	26
5.2.4 ビルド	27
5.2.5 ROM への書き込み	28
6. 実行方法	32
6.1 TwinCAT との接続	32

6.1.1	ESI ファイルの準備 (初回のみ)	32
6.1.2	TwinCAT ドライバのインストール (初回のみ)	32
6.1.3	デバイスのスキャン・EEPROM への書き込み	33
6.2	FSoE マスタの準備	38
6.2.1	Safety プロジェクトの追加	38
6.2.2	PLC プロジェクトの追加	39
6.2.3	ErrAck 信号の選択	40
6.2.4	FSoE マスタ機器の確認	41
6.3	FSoE マスタプロジェクトのダウンロード	42
6.4	TwinCAT プロジェクトの実行方法	44
6.5	動作確認方法	50
6.5.1	Safety Drive 機能の選択	50
6.5.2	CiA402 動作モードの選択	51
6.5.3	FSoE 通信の確立・モータ停止要求の解除	52
6.5.4	CiA402 State Machine 制御方法	56
6.5.4.1	Operation enable への遷移方法	56
6.5.4.2	Fault からの復帰方法	59
6.5.5	モータ動作方法	61
6.5.5.1	csp mode	61
6.5.5.2	pp mode	62
6.5.5.3	csv mode	64
6.5.5.4	hm mode	65
6.5.6	Safety Drive 機能の確認	68
6.5.6.1	STO	68
6.5.6.2	SS1-t	69
6.5.6.3	RZ/T2L 異常による安全制御	70
6.5.6.4	FSoE マスタ異常による安全制御	70
7.	ソフトウェア仕様	71
7.1	ソフトウェア構成図	71
7.2	Motor Control(RZ/T2M CPU0)	72
7.2.1	ソフトウェア全体構成	72
7.2.2	ベースプロジェクトからの変更点	73
7.2.3	Network CPU I/F モジュール	76
7.2.3.1	モジュール構成	76
7.2.3.2	Shared Memory 割り当て	77
7.2.3.3	API	78
7.2.3.4	プライベート関数	84
7.2.3.5	Callback 関数	87
7.2.3.6	マクロ定義	88
7.2.3.7	列挙型	88
7.2.3.8	構造体・変数情報	89
7.2.3.9	グローバル変数情報	92
7.3	EtherCAT(CiA402) (RZ/T2M CPU1)	93
7.3.1	基本仕様	93
7.3.2	ソフトウェア全体構成	94
7.3.3	src フォルダ構成	95

7.3.4	hal_entry・メイン処理	96
7.3.5	CiA402 Drive Profile	99
7.3.5.1	CiA402 動作モード	99
7.3.5.2	Cyclic synchronous position mode	100
7.3.5.3	Cyclic synchronous velocity mode	101
7.3.5.4	Profile position mode	102
7.3.5.5	Homing mode	104
7.3.5.6	CiA402 状態遷移	106
7.3.5.7	CiA402 Error Code	107
7.3.5.8	CiA402 Drive Profile 向け関数	108
7.3.5.9	マクロ定義	123
7.3.5.10	SSC tool 生成後ファイルからの変更点	123
7.3.5.11	Object Dictionary	126
7.3.6	Safety Drive	131
7.3.6.1	Safety Drive 向け関数	133
7.3.6.2	マクロ定義	133
7.3.7	Fusa CPU Communication	134
7.3.7.1	FSoE PDO 設定	134
7.3.7.2	機能安全 CPU との送受信に使用されるオブジェクト	135
7.3.7.3	関数	136
7.3.7.4	構造体・グローバル変数情報	137
7.3.8	Motor CPU I/F	138
7.3.8.1	モジュール構成	138
7.3.8.2	Shared Memory 割り当て	138
7.3.8.3	API	139
7.3.8.4	プライベート関数	144
7.3.8.5	Callback 関数	147
7.3.8.6	マクロ定義	148
7.3.8.7	列挙型	148
7.3.8.8	構造体・変数情報	150
7.3.8.9	グローバル変数情報	152
7.3.9	Fusa I/F	154
7.3.9.1	モジュール構成	154
7.3.9.2	送受信パケット	154
7.3.9.3	API	155
7.3.9.4	プライベート関数	160
7.3.9.5	Callback 関数	160
7.3.9.6	マクロ定義	161
7.3.9.7	列挙型	162
7.3.9.8	構造体・変数情報	163
7.3.9.9	グローバル変数情報	164
7.4	機能安全制御 (RZ/T2L-A, RZ/T2L-B)	165
7.4.1	概要	165
7.4.2	機能概要	166
7.4.3	ソフトウェア構成	167
7.4.4	モータ停止モード選択	170
7.4.5	安全状態解除	171

7.4.6	安全制御の実行 .....	172
7.4.7	SafeData 仕様 .....	173
7.4.8	ステータス LED 仕様 .....	174
8.	Appendix .....	175
8.1	統合開発環境インストール .....	175
8.1.1	EWARM .....	175
8.1.2	e <sup>2</sup> studio .....	175
8.2	付録 : SSC Tool 使用時の注意点 .....	176
8.3	付録 : patch のインストール方法 .....	178
8.3.1	Git for Windows (64bit) を使用する方法 .....	178
8.3.2	MinGW Installation Manager を使用する方法 .....	179
8.4	付録 : FSoE マスタプログラムの作成手順 .....	181
8.5	付録 : RZ/T2M のみの動作方法 .....	194
8.5.1	接続環境 .....	194
8.5.2	ボードの設定 .....	195
8.5.2.1	セーフティモータ制御ボードの設定 .....	195
8.5.3	RZ/T2M プログラム書き込み .....	196
	改訂記録 .....	197

## 1. 概要

### 1.1 RZ/T2M&RZ/T2L Safety Motor Control Reference Kit 概要

RZ/T2M&RZ/T2L Safety Motor Control Reference Kit(以下、セーフティモータ制御キット)は次のボードやソフトウェア等から構成されています。

- ・ RZ/T2M と RZ/T2Lx2 を搭載したボード(以下、セーフティモータ制御ボード)
- ・ モータ制御、EtherCAT 通信、FSoE 対応のセーフティドライブ機能を搭載したファームウェア
- ・ セーフティモータ制御ボードのユーザズマニュアル&回路図

Renesas 製の RZ/T series Inverter Board Kit(Inverter Board +BLDC)を接続することにより、1 軸のモータ制御ができます。また、本キットを用いることで EtherCAT(CiA402)、FSoE 通信や IEC61800-5-2 : STO, SS1-t に対応したモータ停止を行うことができます。

ソフトウェア機能を以下に示します。

- ・ シリアルエンコーダを使用した永久磁石同期モータのベクトル制御
- ・ EtherCAT(CiA402)、FSoE 通信
- ・ IEC61800-5-2 : STO, SS1-t に対応したモータ停止

### 1.2 接続構成

図 1-1 にセーフティモータ制御キットの接続構成を示します。

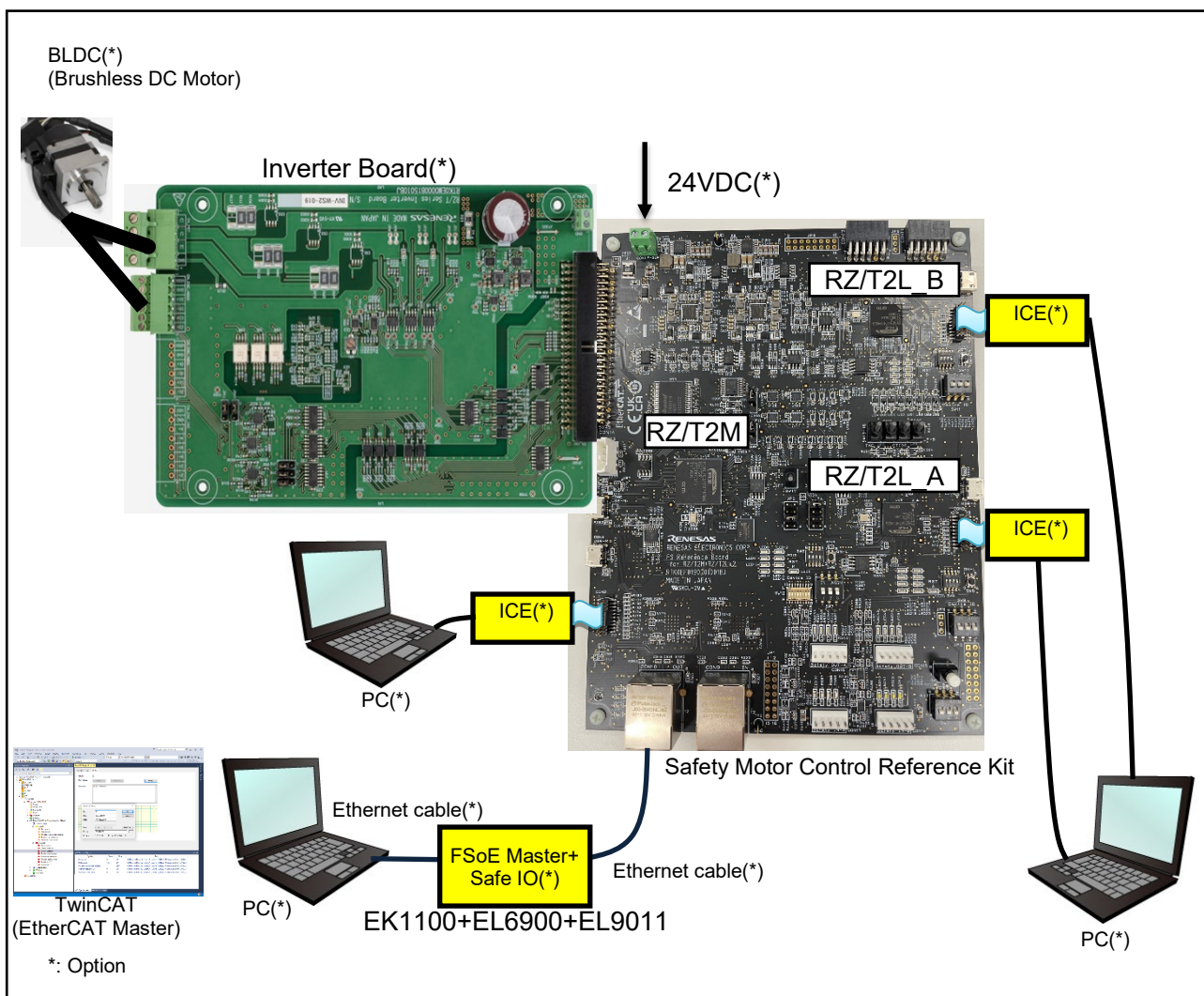


図 1-1 セーフティモータ制御キット接続構成

### 1.3 接続機器

表 1.1 に接続機器一覧を示します。

表 1.1 接続機器一覧

Item	型名	備考
RZ/T Series Inverter Board Kit	Renesas 社製 RTK0EM0000S05010BJ(※) <ul style="list-style-type: none"> <li>・ Inverter Board</li> <li>・ 多摩川精機社製:TSM3101N2001E020</li> </ul>	-
ICE	<ul style="list-style-type: none"> <li>・ IAR システムズ社製 I-jet</li> <li>・ SEGGER 社製 J-Link Base Ver.11.0 以上</li> </ul>	-
FSoE マスタ	Beckhoff Automation 社製 <ul style="list-style-type: none"> <li>・ EK1100</li> <li>・ EL6900</li> <li>・ EL9011</li> </ul>	-

※Renesas 社製 RZ/T Series Inverter Board Kit は下記から購入することができます。

<https://www.renesas.com/invb-lv-rzt-b>

#### 1.3.1 セーフティモータ制御ボードと Inverter Board との接続

セーフティモータ制御ボードと Inverter Board は、直接接続します。

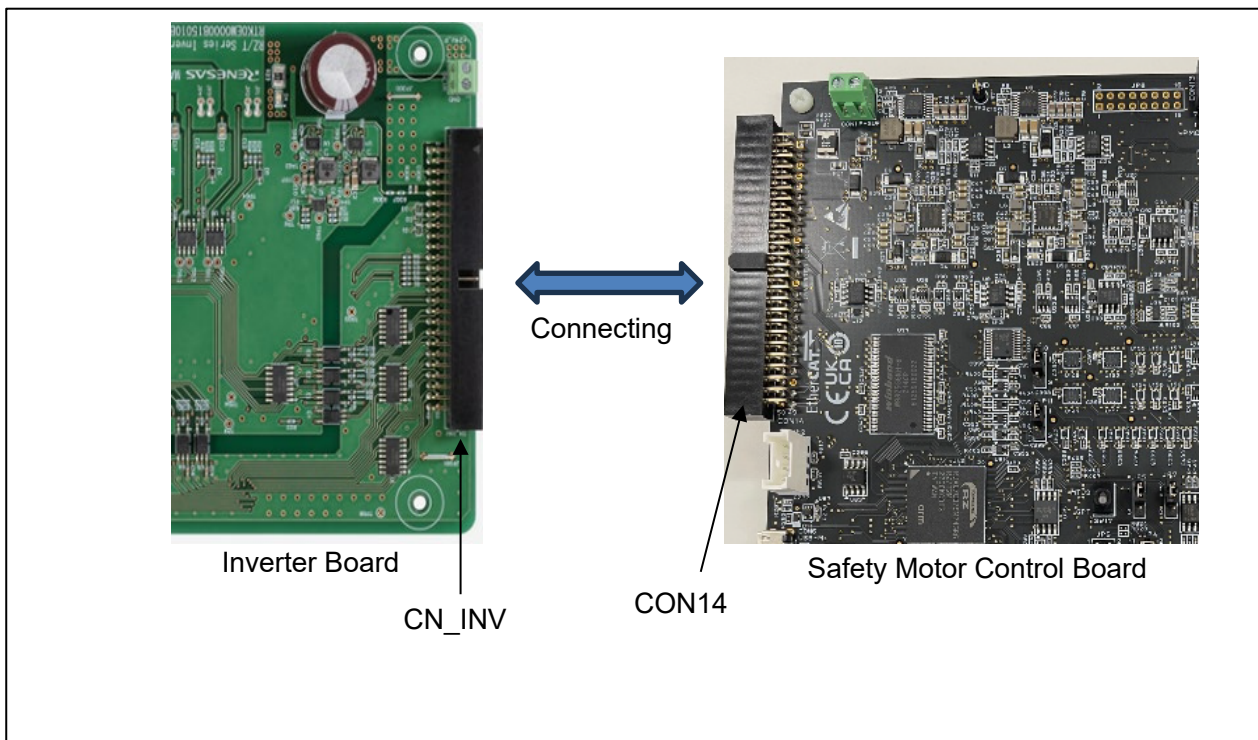


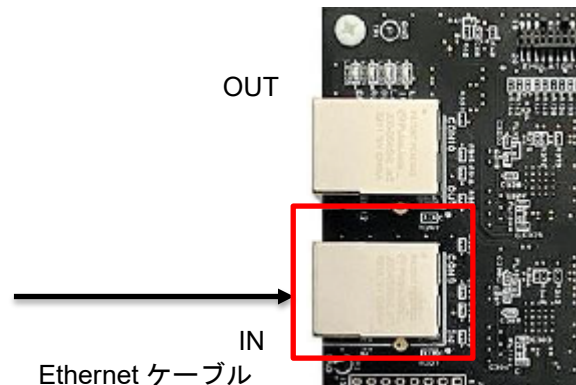
図 1-2 セーフティモータ制御ボードと Inverter Board との接続

### 1.3.2 モータ/エンコーダと Inverter Board との接続

多摩川精機社製:TSM3101N2001E020 (モータ/エンコーダ(FA-CODER))と Inverter Board との接続に関しては、「RZ/T Series Inverter Board/Kit ユーザーズマニュアル(R12UZ0155)」を参照してください。

### 1.3.3 PC との接続

FSoE マスタの OUT 側とセーフティモータ制御ボードの RJ45 コネクタの IN 側に Ethernet ケーブルを接続します。



### 1.3.4 ICE

#### 1.3.5 RZ/T2M / RZ/T2L\_A/ RZ/T2L\_B

RZ/T2M、RZ/T2L\_A、RZ/T2L\_B の ICE の接続方法については、RTK0EF0190D01001BJ ユーザーズマニュアル(R30UZ0214)を参照してください。

### 1.3.6 電源

セーフティモータ制御ボードに 24V DC を印加します。

電源の印加方法については、RTK0EF0190D01001BJ ユーザーズマニュアル(R30UZ0214)を参照してください。

Inverter Board を接続している場合は、セーフティモータ制御ボードから DC24V を供給することが可能です。その場合、Inverter Board に DC24V を印加しないでください。


万が一、Inverter Board の電流が不足している場合は、Inverter Board とセーフティモータ制御ボードへの印加を分けることができます。

別々に電源を供給する場合は、RTK0EF0190D01001BJ ユーザーズマニュアル(R30UZ0214)を参照してください。

## 2. 動作環境

## 2.1 RZ/T2M

表 2.1 動作環境(RZ/T2M)

項目	内容
使用ボード	セーフティモータ制御ボード
MPU	RZ/T2M グループ R9A07G075M28GBG: 320pinFBGA
動作周波数	CPU Core0 : 800MHz (Arm <sup>®</sup> Cortex <sup>®</sup> -R52) CPU Core1 : 800MHz (Arm <sup>®</sup> Cortex <sup>®</sup> -R52)
動作電圧	3.3V/1.8V/1.1V
動作モード	xSPI0 boot mode(x1 boot serial flash)
使用デバイス	Serial Flash ROM (64Mbyte) RENESAS 製 AT25SF128A
通信プロトコル	EtherCAT <sup>®</sup>  
統合開発環境	RENESAS 製 e <sup>2</sup> studio 2025-04-1 (25.4.1)
エミュレータ	SEGGER 製 J-Link Base Ver.11.0 以上
SSC Tool	Provided by EtherCAT Technology Group (ETG) Slave Stack Code (SSC) Tool Version 5.13
Software PLC	Beckhoff Automation TwinCAT <sup>®</sup> 3
Flexible Software Package (FSP)	Version.2.3.0

## 2.2 RZ/T2L\_A, RZT2L\_B

表 2.2 動作環境(RZ/T2L\_A, RZT2L\_B)

項目	内容
使用ボード	セーフティモータ制御ボード
MPU	RZ/T2L Group R9A07G074M04GBG: 196pin LFBGA
動作周波数	800MHz (Arm® Cortex®-R52)
動作電圧	3.3V/1.8V/1.1V
メモリ	Serial Flash ROM (64Mbyte) RENESAS AT25SF128A
統合開発環境	IAR Systems Embedded Workbench® for Arm Version 9.20.3 Functional Safety
エミュレータ	IAR Systems I-jet

RZ/T2L のソフトウェアについては、下記から 3 つのダウンロードしてください。

<https://info.renesas.com/rel-ia-fusa-software-info-request>

- ① RZ(CR52 core)ファミリ セルフテストソフトウェアキット  
ファイル名: RTK0EF0102F01001SJ
- ② RZ/T2L グループ SIL3 システムソフトウェアキット  
ファイル名: RTK0EF0154F01001SJ
- ③ RZ ファミリ FSoE アプリケーションソフトウェアキット  
ファイル名: RTK0EF0129F01001SJ

上記ソフトウェアを使用した RZ/T2L\_A/B へのプログラム書き込み方法については、「5 プログラム書き込み」を参照してください。

## 2.3 フォルダ、ファイル構成

```

r01an8173ej0100-rzt2m-rzt2l-smcr-package
|- r01an8003jj0100-motor-encoder-foc-rzt2m.pdf // Encoder Vector Control of Permanent Magnet
|                                             Synchronous Motor - Serial Encoder (Japanese)
|- r01an8003ej0100-motor-encoder-foc-rzt2m.pdf // Encoder Vector Control of Permanent Magnet
|                                             Synchronous Motor - Serial Encoder (English)
|- r01an8173jj0100-rzt2m-rzt2l-smcr-startup.pdf // This document (Japanese)
|- r01an8173ej0100-rzt2m-rzt2l-smcr-startup.pdf // This document (English)
|
|- Common
|   |- ESI
|   |   +- Renesas EtherCAT RZT2 SafetyMotorSolution.xml // ESI file
|   |
|   |- Patch
|   |   |- apply_patch_e2studio.bat // Bat file
|   |   +- RZT2_Safety_Motor_Solution.patch // Patch file
|   |
|   |- SSCconfig
|   |   +- Renesas_EtherCAT_RZT2_Safety_Motor_Solution.esp // esp file
|   |
|- RZT2M
|   +- RZT2M_SMCRK.zip // RZ/T2M software
|
+- RZT2L
   +- RZT2L.zip // RZ/T2L software

```

### 3. 関連ドキュメント

- RTK0EF0190D01001BJ ユーザーズマニュアル(R30UZ0214)
- RZ/T Series Inverter Board/Kit ユーザーズマニュアル(R12UZ0155)
- RZ/T2M 永久磁石同期モータのエンコーダベクトル制御 - シリアルエンコーダ アプリケーションノート(R01AN8003)

- CiA402 規格書 :

- IEC 61800-7-201 Edition 1.0

Adjustable speed electrical power drive systems Part 7-201: Generic interface and use of profiles for power drive systems Profile type 1 specification

- IEC 61800-7-301 Edition 1.0

Adjustable speed electrical power drive systems Part 7-301: Generic interface and use of profiles for power drive systems Mapping of profile type 1 to network technologies

## 4. ボードの設定

### 4.1 セーフティモータ制御ボードの設定

#### 4.1.1 Switches

##### SW1

Pin	1	2	3
	OFF	OFF	OFF

##### SW3 - 5

Pin	1	2	3	4
	ON	ON	ON	OFF

##### SW7

Pin	1	2	3
	OFF	OFF	OFF

##### SW9

Pin	1	2	3
	OFF	OFF	OFF




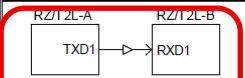
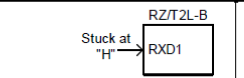
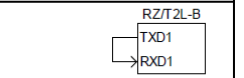
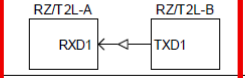
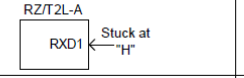
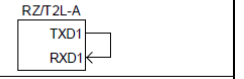
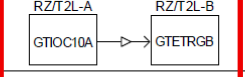
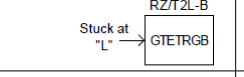
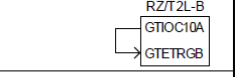
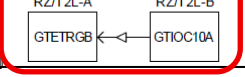
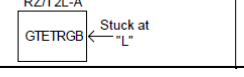
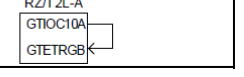
##### SW11

Pin	1	2	3
	OFF	OFF	OFF




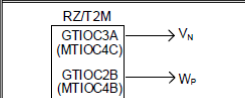
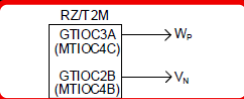
##### SW12

Pin	1	2	3	4	5	6	7	8
	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

#### SW13 -16(赤枠設定)

Switch	 Up (Silkscreen: "-") Normal connection	 Center (Silkscreen: "F") Pseudo stuck-at fault	 Down (Silkscreen: "LP-B") Loop-back
SW13			
SW14			
SW15			
SW16			

#### SW17(赤枠設定)

Switch	 Up (Silkscreen "MTU3")	 Center Setting not allowed (Do not set to center)	 Down (Silkscreen "GPT")
SW17			

セーフティモータ制御ボードのスイッチの詳細は、「RTK0EF0190D01001BJ ユーザーズマニュアル (R30UZ0214)」を参照してください。

## 4.1.2 Jumpers

No	JP	設定
1	JP4	1-2 short
2	JP5	1-2 short
3	JP6	5-6 short
4	JP7	1-2 short
5	JP10	1-2 short
6	JP11	1-2 short

セーフティモータ制御ボードのジャンパの詳細は、「RTK0EF0190D01001BJ ユーザーズマニュアル (R30UZ0214)」を参照してください。

## 4.2 Inverter Board

### 4.2.1 Jumpers

No	JP	設定
1	JP7,8,9	1-2 short
2	JP10	2-3 short
3	JP11	2-3 short

Inverter Board のジャンパの詳細は、「RZ/T Series Inverter Board/Kit ユーザーズマニュアル (R12UZ0155)」を参照してください。

## 5. プログラム書き込み

### 5.1 RZ/T2M

#### 5.1.1 EtherCAT SubDevice Stack の生成

※ 本サンプル・プロジェクトには EtherCAT SubDevice スタックコードは同梱されていません。

EtherCAT スレーブスタックコードの生成には”EtherCAT SubDevice Stack Code(SSC) Tool”が必要です。

SSC Tool は ETG 協会から入手可能です。

※ SSC Tool 起動前に”付録：SSC Tool 使用時の注意点”を確認してください。

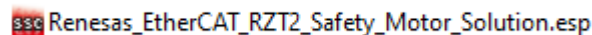
※ 本ガイドを実行するためには、PC に patch.exe がインストールされている必要があります。

“付録：patch のインストール方法”を参照して patch.exe をインストールして下さい。

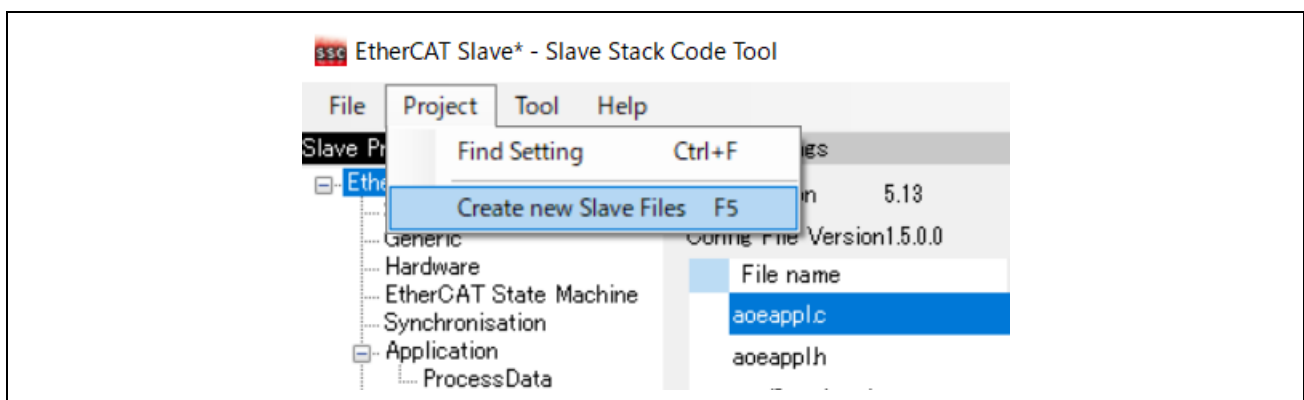
(1) 下記のフォルダ構成になるように、RZT2M¥RZT2M\_SMCRK.zip を解凍します。

```
r01an8173ej0100-rzt2m-rzt2l-smcr-package
├── Common
├── RZT2M
│   └── RZT2M_SMCRK
│       ├── RZT2M_SMCRK_ESC_E2S_V100
│       └── RZT2M_SMCRK_INVBLB_SPM_ENCD_FOC_E2S_V100
```

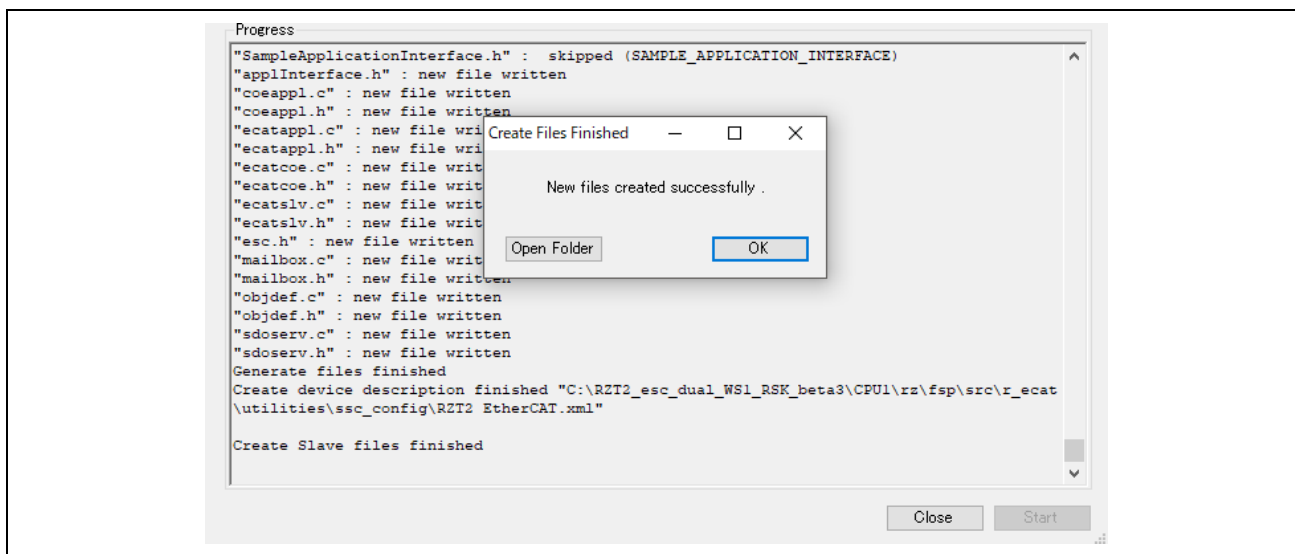
(2) “Common¥SSCConfig”フォルダに格納されている esp ファイルをダブルクリックし、SSC Tool を起動して下さい。



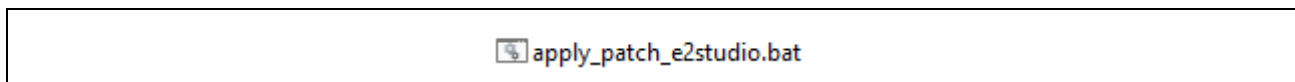
(3) [Project]→[Create New Slave Files]をクリック [Current new Slave Files] →[Start]をクリックします。



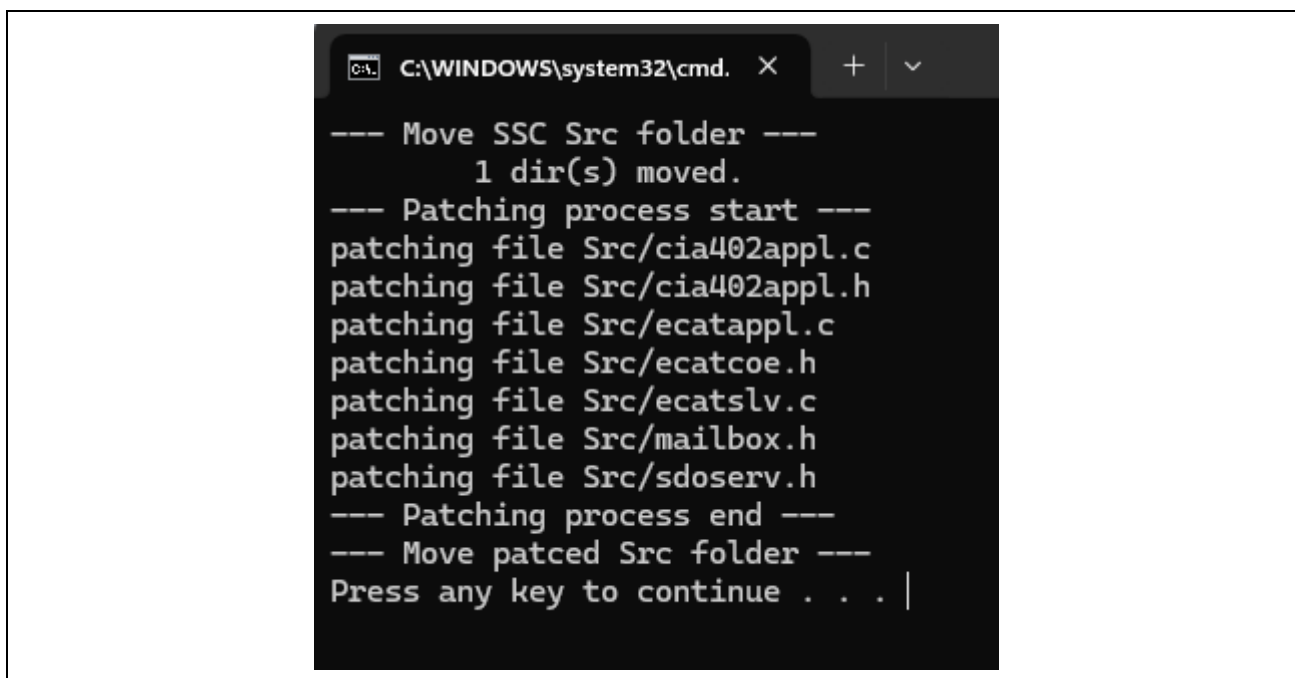
- (4) ソースコードが生成され、成功すると” New Files created successfully” と表示されるので[OK]をクリックします。



- (5) “Common¥Patch”フォルダに格納されている[apply\_patch\_e2studio.bat]をダブルクリックして bat ファイルを実行します。

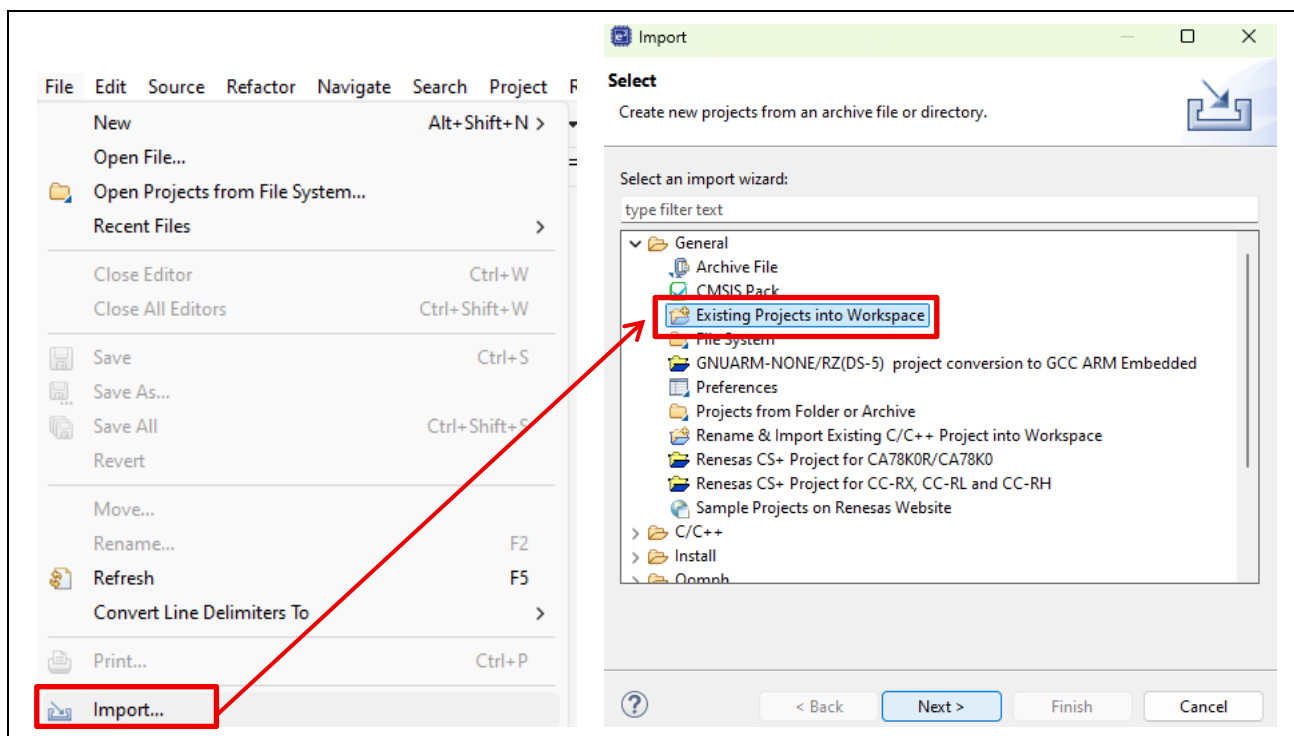


- (6) Bat ファイル実行後、RZT2M¥RZT2M\_SMCRK¥RZT2M\_SMCRK\_ESC\_E2S\_V100¥src¥ethercat¥beckhoff フォルダに Src フォルダが移動されます。

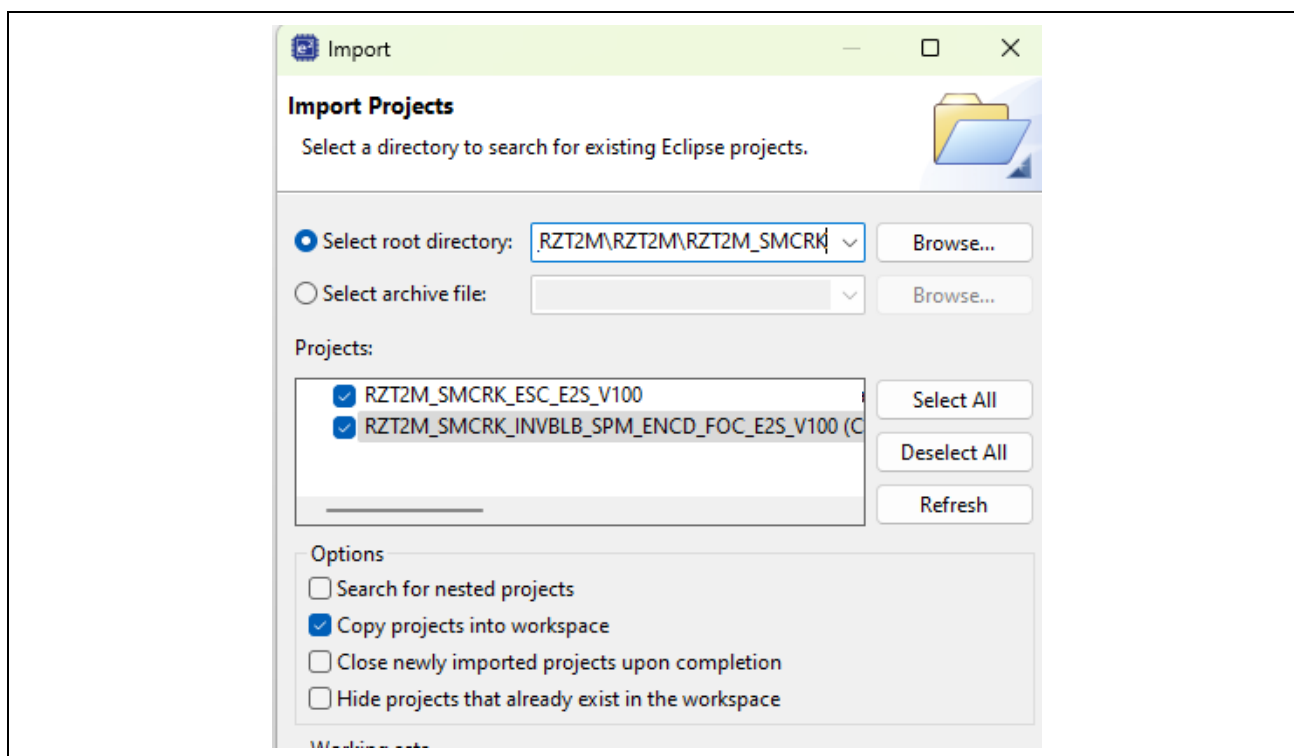


## 5.1.2 プロジェクトのビルド

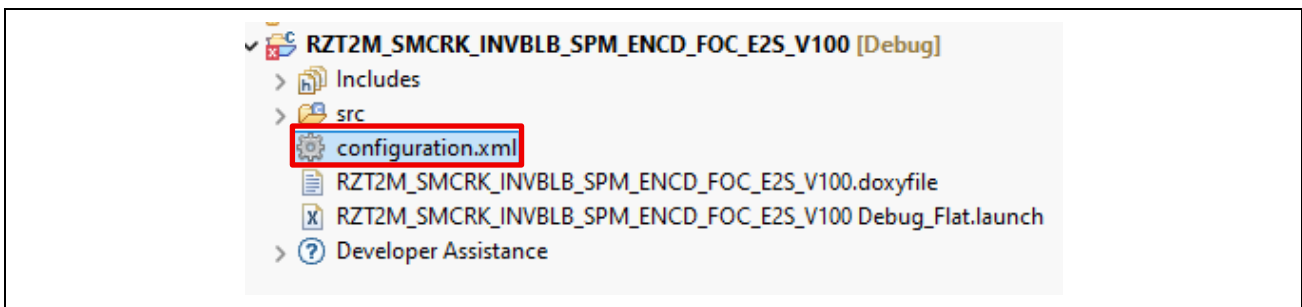
- (1) e<sup>2</sup> studio を起動し、上のメニューバーから"File">"Import">"Existing Projects into Workspace"を選択して"Next"を押します。



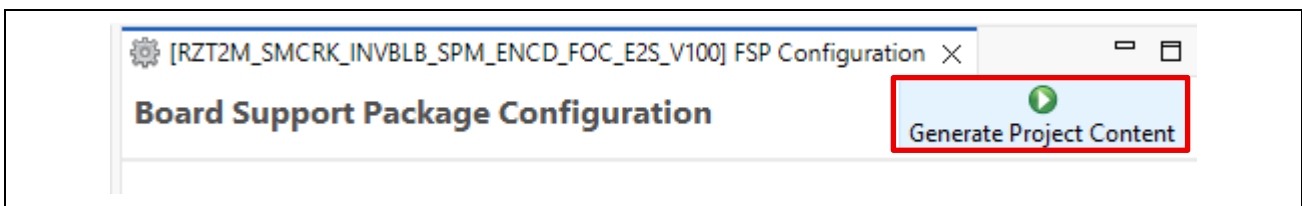
- (2) "Select root directory" を選択し、"RZT2M\RZT2M\_SMCRK"フォルダを選択します。
- (3) "RZT2M\_SMCRK\_INVBLB\_SPM\_ENCD\_FOC\_E2S\_V100"および"RZT2M\_SMCRK\_ESC\_E2S\_V100"にチェックマークを付けます。Options の中で、"Copy projects into workspace"にチェックマークを付け、ワークスペース内にコピーしながらインポートします。



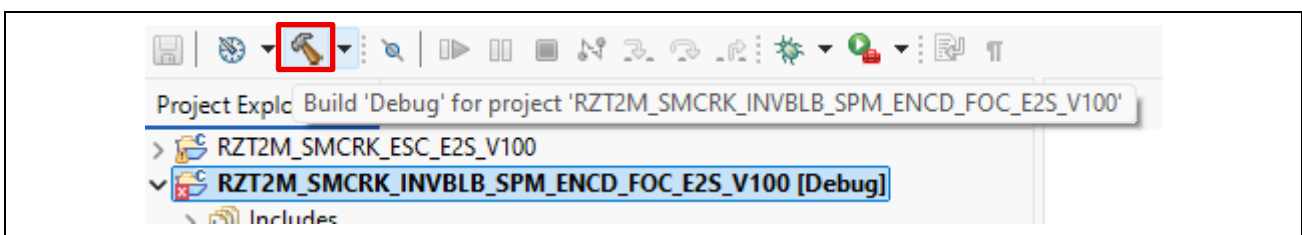
(4) RZT2M\_SMCRK\_INVBLB\_SPM\_ENCD\_FOC\_E2S\_V100 の configuration.xml を開きます。



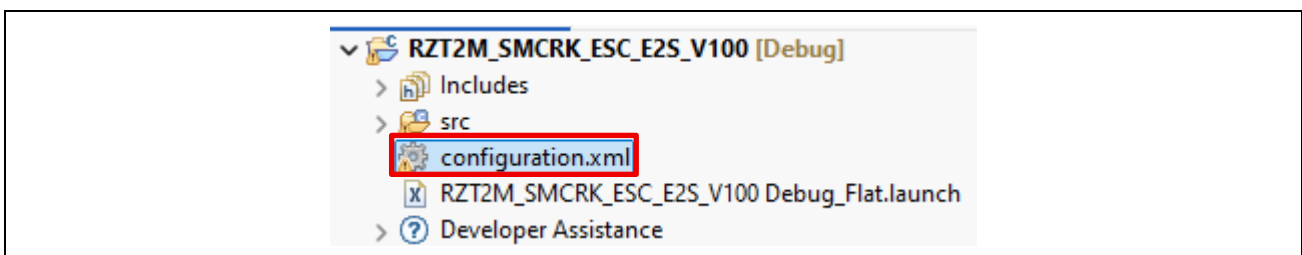
(5) コード生成を行います。



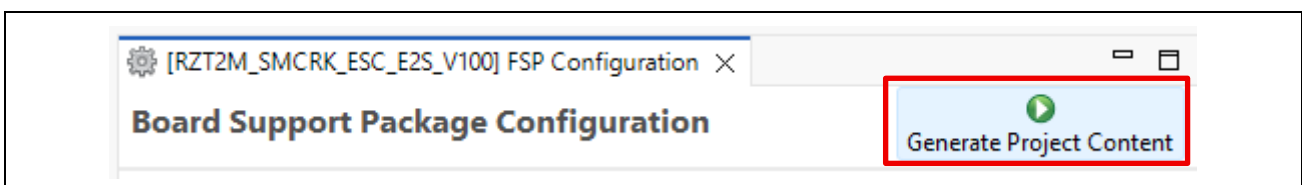
(6) RZT2M\_SMCRK\_INVBLB\_SPM\_ENCD\_FOC\_E2S\_V100 のビルドを実行します。



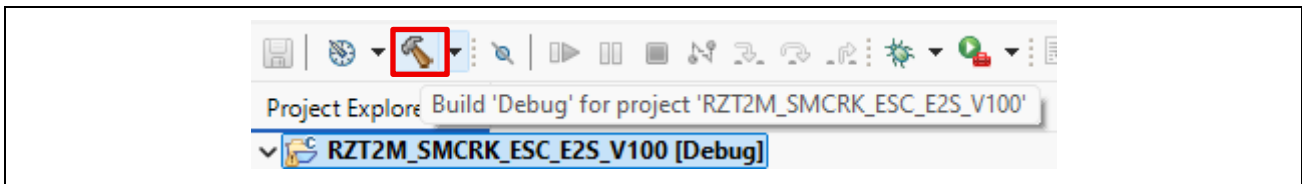
(7) RZT2M\_SMCRK\_ESC\_E2S\_V100 の configuration.xml を開きます。



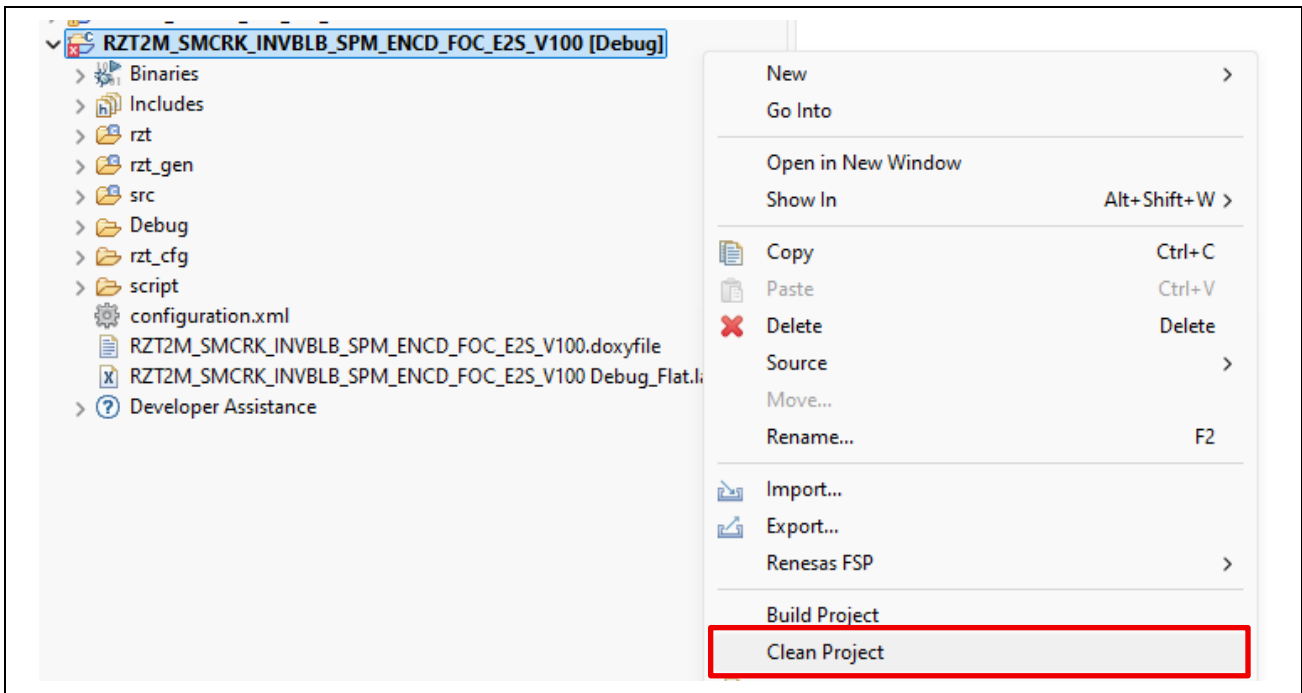
(8) コード生成を行います。



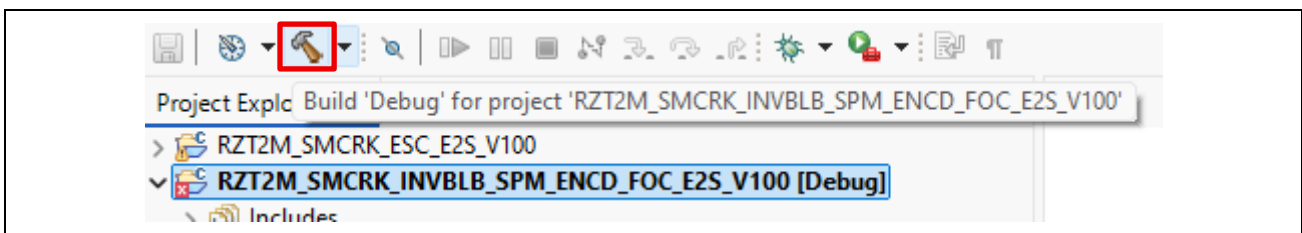
(9) RZT2M\_SMCRK\_ESC\_E2S\_V100 のビルドを実行します。



(10) RZT2M\_SMCRK\_INVBLB\_SPM\_ENCD\_FOC\_E2S\_V100 プロジェクトでクリーンを行います。

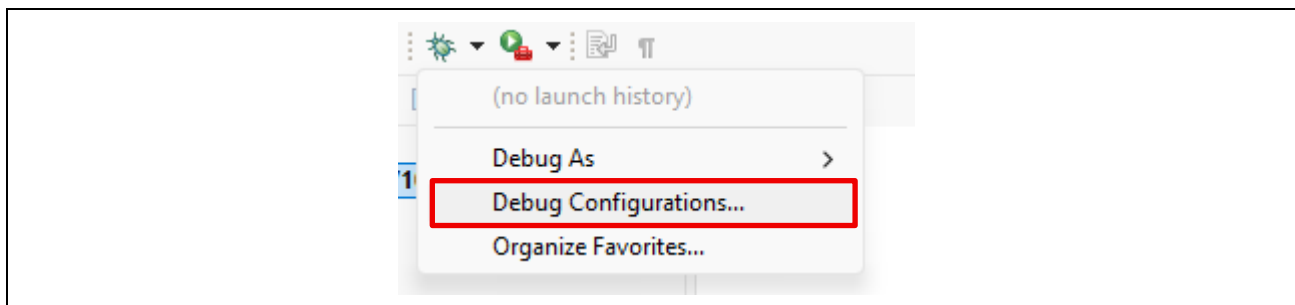


(11) クリーン完了後、再度RZT2M\_SMCRK\_INVBLB\_SPM\_ENCD\_FOC\_E2S\_V100のビルドを実行します。

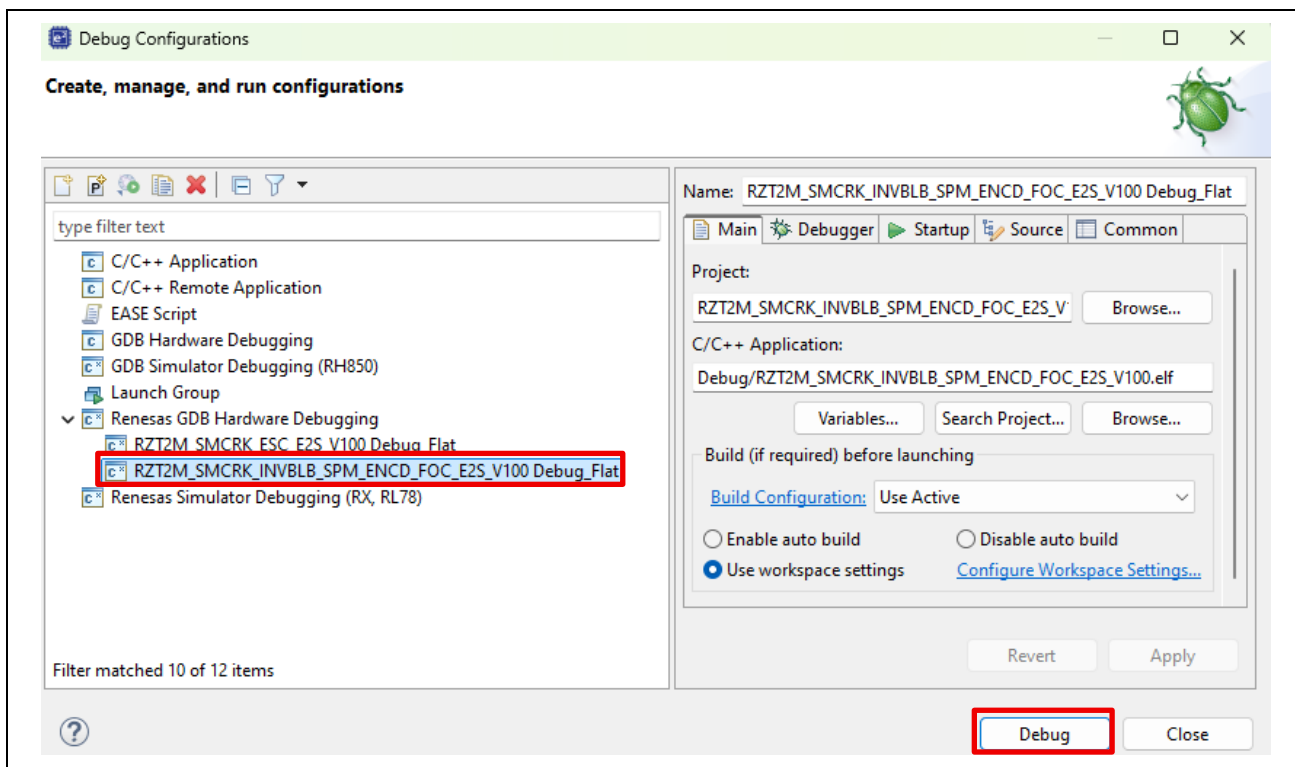


## 5.1.3 プロジェクトの書き込み

- (1) セーフティモータ制御ボードに 24V DC を印加し、電源を入れます。
- (2) J-Link デバッガを RZ/T2M に接続した後、デバッグ構成を開きます。



- (3) 『RZT2M\_SMCRCR\_INVBLB\_SPM\_ENCD\_FOC\_E2S\_V100 Debug\_Flat』を選択してデバッグを行います。



- (4) 書き込みが完了し、デバッグが開始したらデバッグを終了し、ボード上のリセットボタン SW2 を押します。

本サンプルはスタンドアロンで動作させます。

## 5.2 RZ/T2L

本章では、RZ/T2L機能安全制御用ソフトウェアをリファレンスボード "RTK0EF0190D01001BJ" のフラッシュROMに書き込むまでの手順を示します。本章の図はRZ/T2L-A側での例となりますが、以降の作業についてはRZ/T2L-AおよびRZ/T2L-Bのそれぞれに対して個別に実施してください。

### 5.2.1 開発ツール

RZ/T2L機能安全制御用ソフトウェア（表7.66, #4 および#8）の使用には下記ツールが必要です。

- IARシステムズ社製 機能安全対応 Embedded Workbench for ARM (EWARM-FS) v9.20.3（注1, 注2）

RZ/T2L機能安全制御用ソフトウェアを リファレンスボード "RTK0EF0190D01001BJ" のフラッシュROMに書き込むプロジェクト（表7.66, #5 および#9）の使用には下記ツールが必要です。

- IARシステムズ社製 Embedded Workbench for ARM v9.20.3 以降の任意のバージョン（注3）

EWARM-FS、I-jetエミュレータ、リファレンスボードの仕様や使用方法については、それぞれのマニュアルを参照してください。

注1：EWARM-FS v9.20.3 SP1は使用できません。

注2：EWARM-FS v9.20.3でRZ/T2Lを使用する際には、あらかじめEWARM-FS v9.20.3に対して専用のデバイスパッチを適用してください。デバイスパッチはIARシステムズ社より提供されています。

注3：EWARM-FS v9.20.3 以外で使用する場合は、プロジェクト起動時に、使用するツールバージョンへの変換処理を行ってください。

## 5.2.2 ワークスペースフォルダのコピー

RZ/T2L のソフトウェアについては、下記から 3 つのダウンロードしてください。

<https://info.renesas.com/rel-ia-fusa-software-info-request>

- ① RZ(CR52 core)ファミリ セルフテストソフトウェアキット  
ファイル名: RTK0EF0102F01001SJ
- ② RZ/T2L グループ SIL3 システムソフトウェアキット  
ファイル名: RTK0EF0154F01001SJ
- ③ RZ ファミリ FSoE アプリケーションソフトウェアキット  
ファイル名: RTK0EF0129F01001SJ

本章では、①～③を使用して、プロジェクトを作成する方法を示します。

以降に、ワークスペースフォルダのコピー方法を示します。

- 1) PCの任意の場所に、2重化構成システム用の2つのRZ/T2L用ワークスペースフォルダ（名称：  
RZ/T2L-A用：FSoE\_SW\_M, RZ/T2L-B用：FSoE\_SW\_S）をそのままコピーしてください。

(例) RZ/T2L-A用 : C:\¥FSoE\_SW\_M ← (表7.66, #2 をコピー)  
RZ/T2L-B用 : C:\¥FSoE\_SW\_S ← (表7.66, #6 をコピー)

本章では下記の前提で説明を行います。データパスや名前については、必要に応じて置き換えてお読みください。

### [ RZ/T2L-A ]

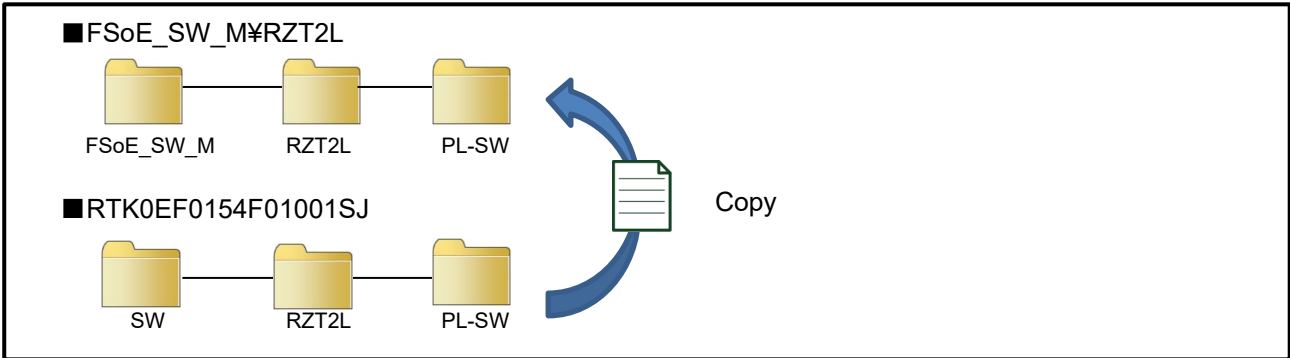
- ・ワークスペースパス : "C:\¥FSoE\_SW\_M"
- ・ワークスペース名 : "RZT2L\_main"
- ・プロジェクトパス : "C:\¥FSoE\_SW\_M¥RZT2L¥PL-SW"
- ・プロジェクト名 : "RZT2L"
- ・ヘキサ・ファイルの出力先 : "C:\¥FSoE\_SW\_M¥RZT2L¥PL-SW¥Debug¥Exe"

### [ RZ/T2L-B ]

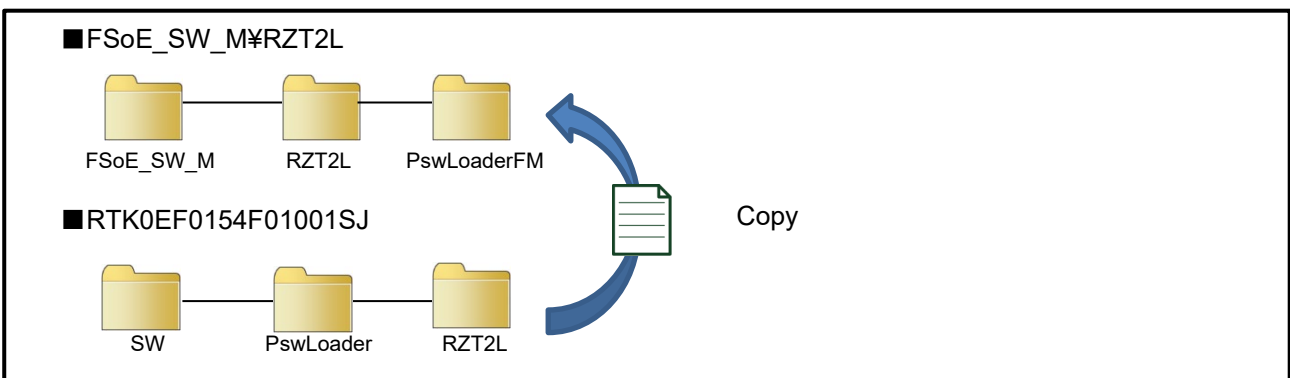
- ・ワークスペースパス : "C:\¥FSoE\_SW\_S"
- ・ワークスペース名 : "RZT2L\_sub"
- ・プロジェクトパス : "C:\¥FSoE\_SW\_S¥RZT2L¥PL-SW"
- ・プロジェクト名 : "RZT2L"
- ・ヘキサ・ファイルの出力先 : "C:\¥FSoE\_SW\_S¥RZT2L¥PL-SW¥Debug¥Exe"

## 5.2.2.1 SIL3 システムソフトウェアキットのコピー

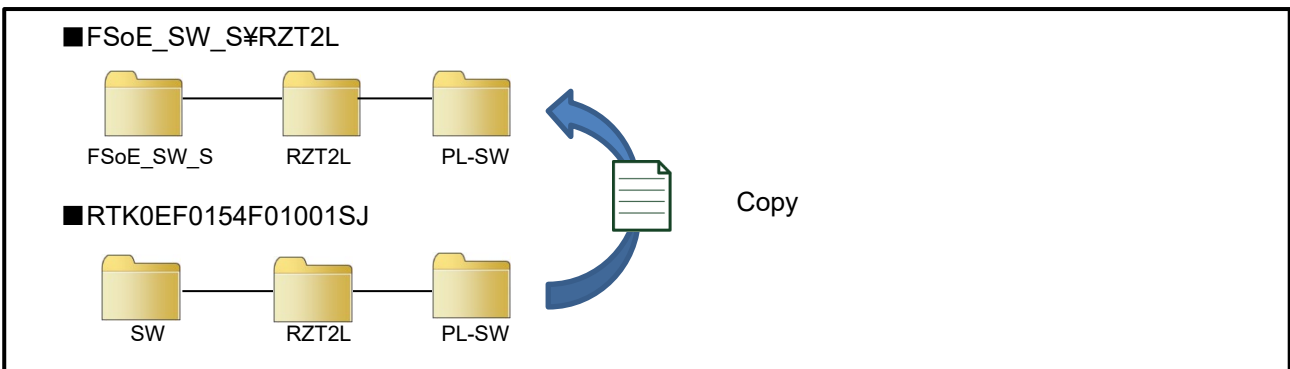
## (1) RZ/T2L\_A の PL-SW



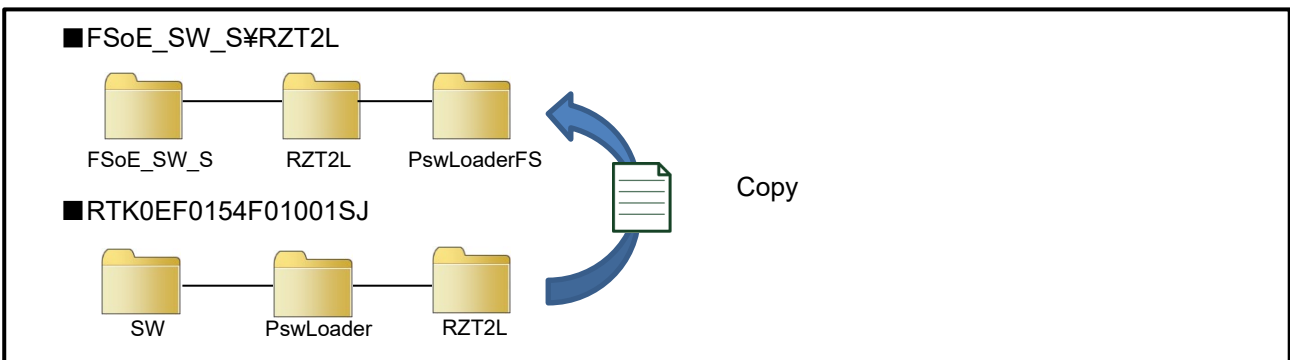
## (2) RZ/T2L\_A の PswLoaderFM



## (3) RZ/T2L\_B の PL-SW

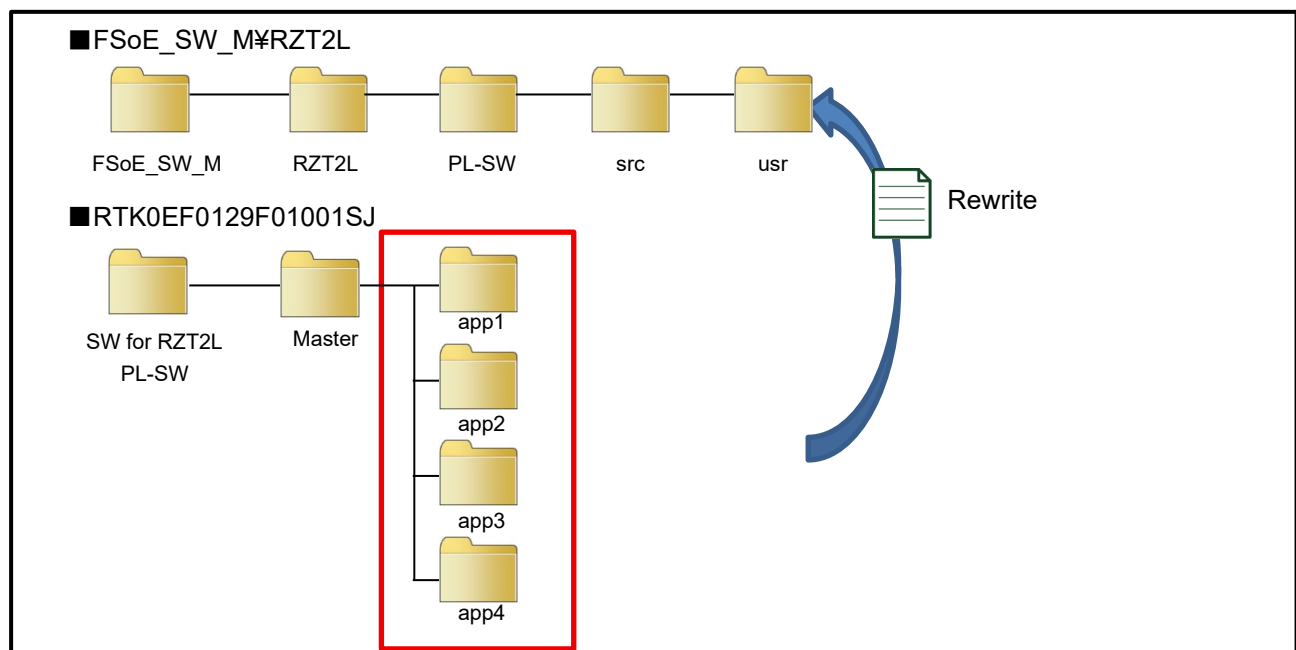
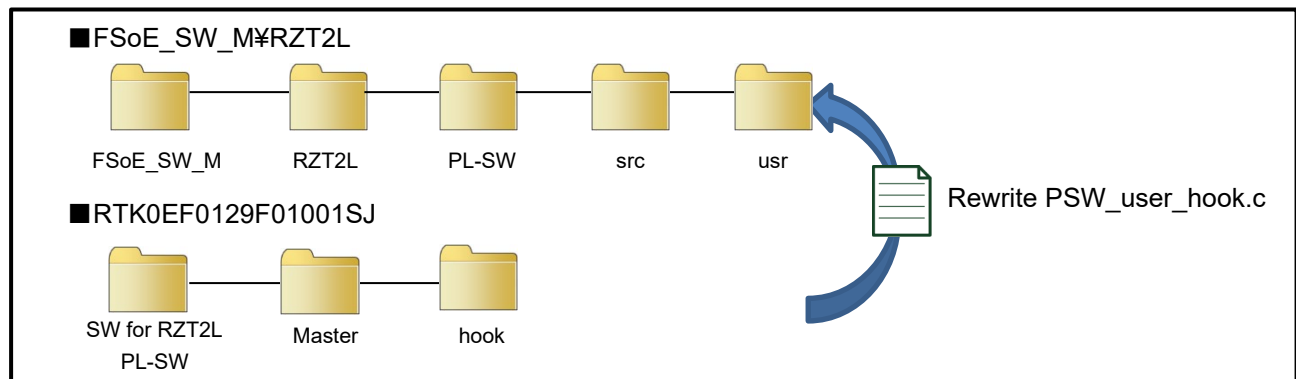
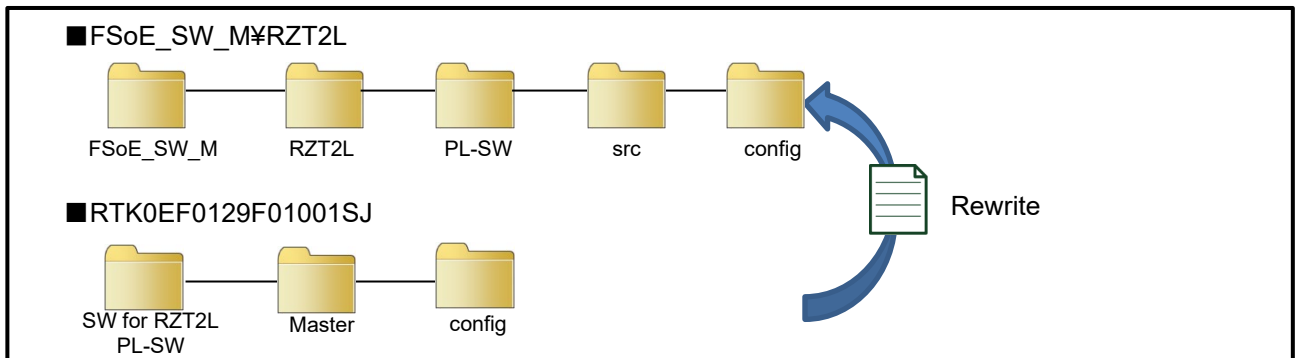
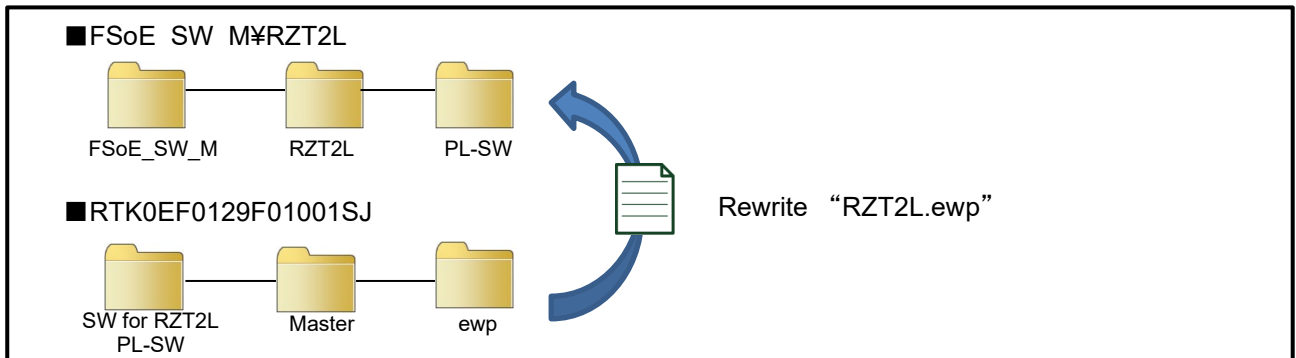


## (4) RZ/T2L\_B の PswLoaderFS

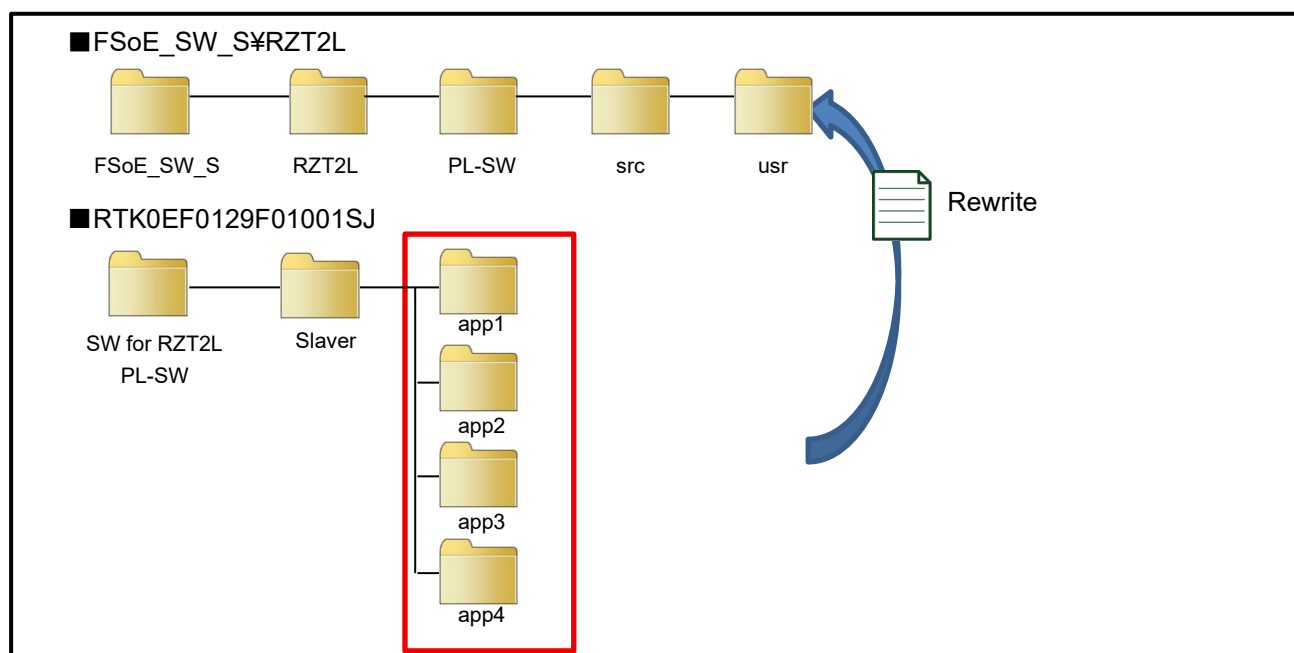
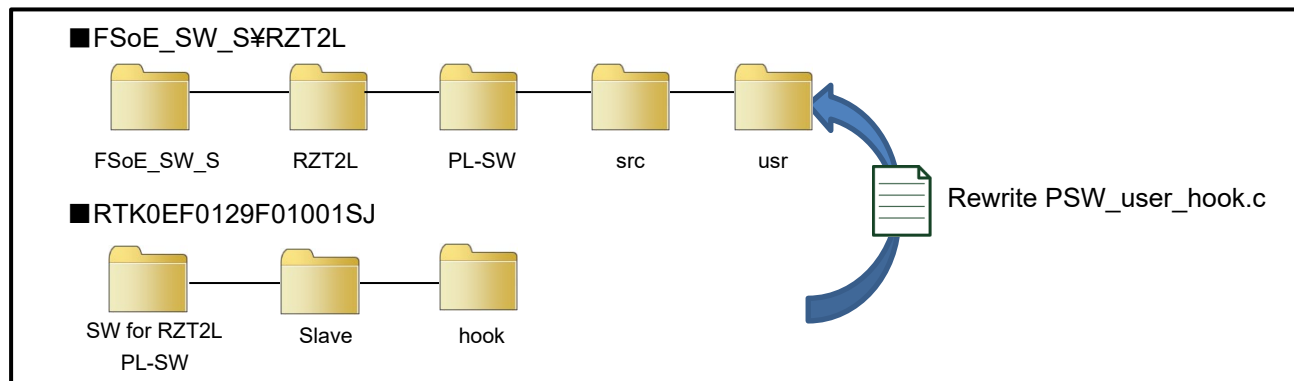
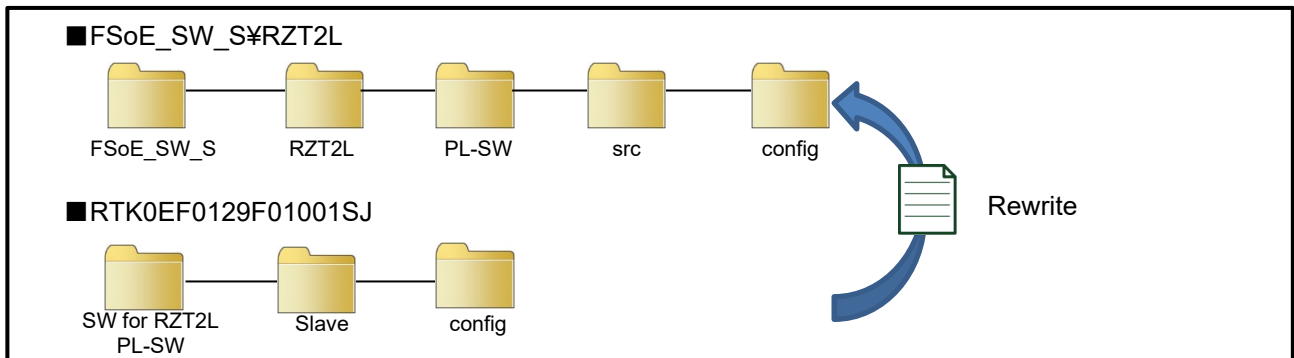
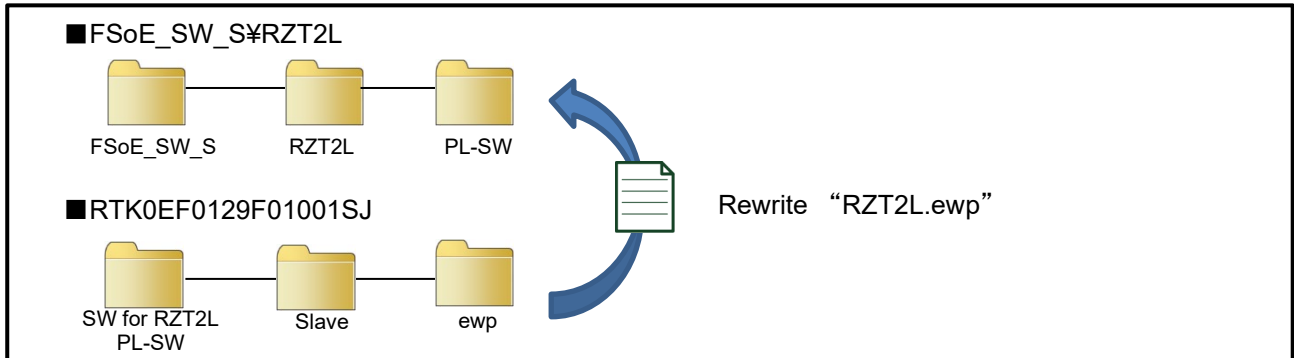


5.2.2.2 FSoE アプリケーションソフトウェアキットのコピー

(1) RZ/T2L\_A

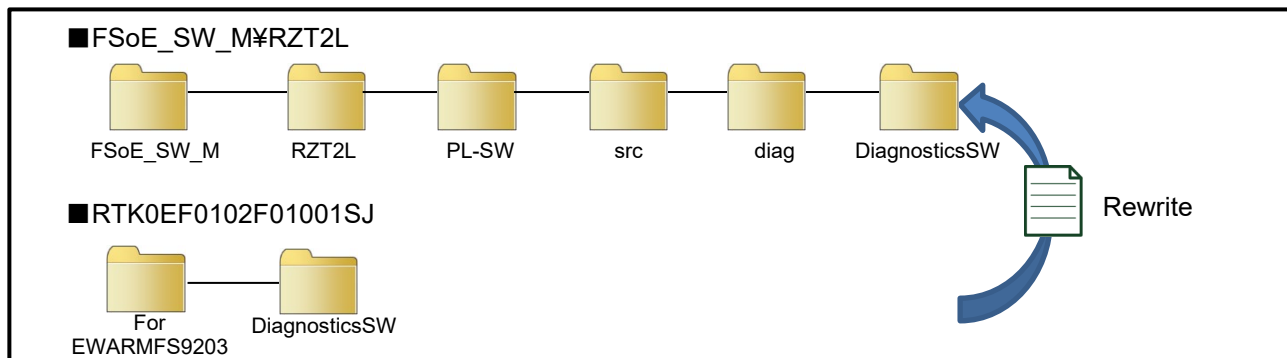


(2) RZ/T2L\_B

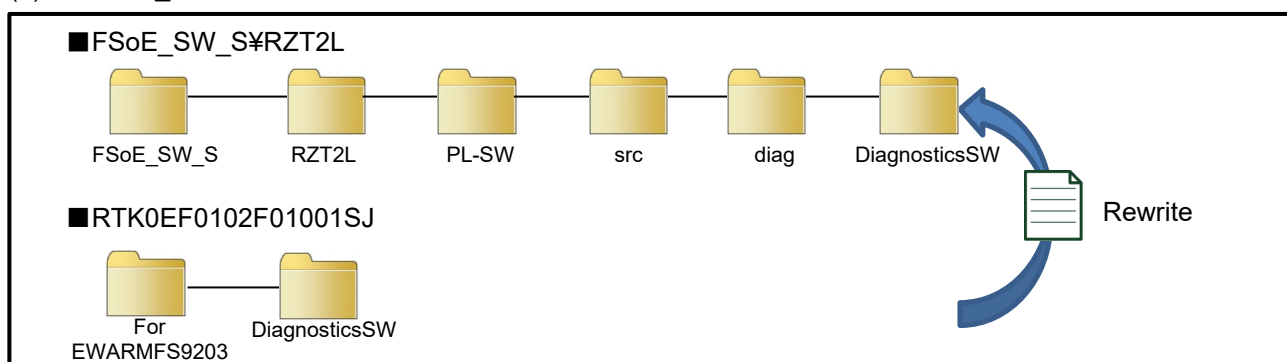


5.2.2.3 セルフテストソフトウェアキットのコピー

(1) RZ/T2L\_A

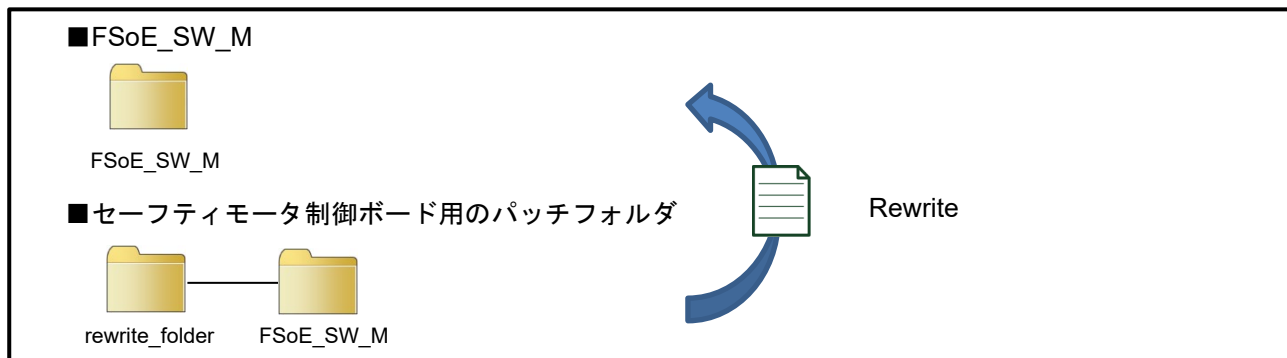


(2) RZ/T2L\_B

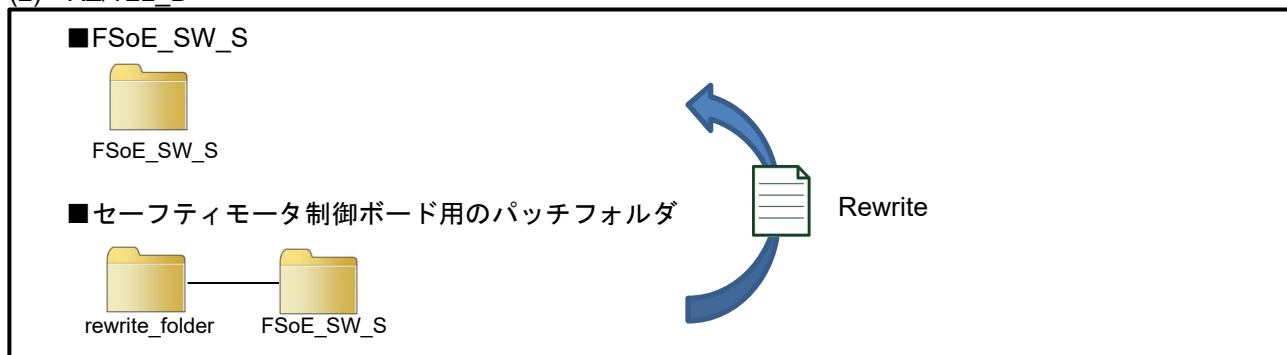


5.2.2.4 セーフティモータ制御ボード用のパッチプログラムのコピー

(1) RZ/T2L\_A



(2) RZ/T2L\_B



### 5.2.3 ワークスペース起動

以降に、プロジェクトのワークスペース起動手順について示します。

- 1) Windowsの[スタート]メニューからEWARM-FS 9.20.3を起動します。
- 2) [ファイル]→[ワークスペースを開く]を選択します。(図5-1, #1 #2)
- 3) PL-SWフォルダ内にあるワークスペースファイル (RZT2L\_main.eww / RZT2L\_sub.eww) を開きます。(図5-1, #3)

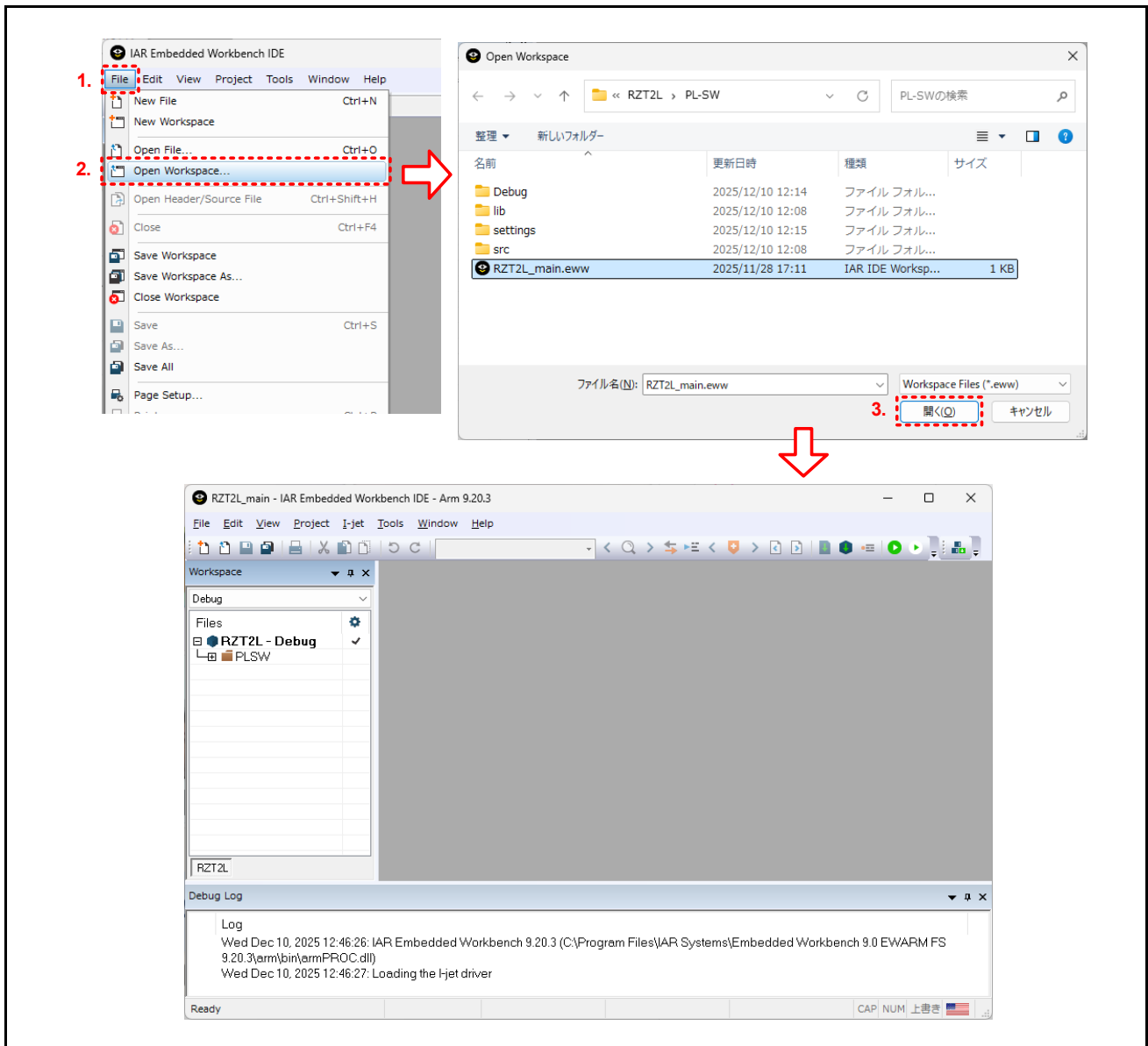


図 5-1 ワークスペース起動

これで、ワークスペース起動操作は終了です。

## 5.2.4 ビルド

以降に、プロジェクトのビルド手順について示します。プロジェクトは、必要なビルド環境設定が実施された状態で提供されます。

- 1) [プロジェクト]→[すべてを再ビルド]を選択し、プログラムのビルドを開始します。（図内、#1, #2）
- 2) ビルドが終了すると、プロジェクトの[ビルド]ウィンドウにエラーとワーニングの合計数が表示されます。

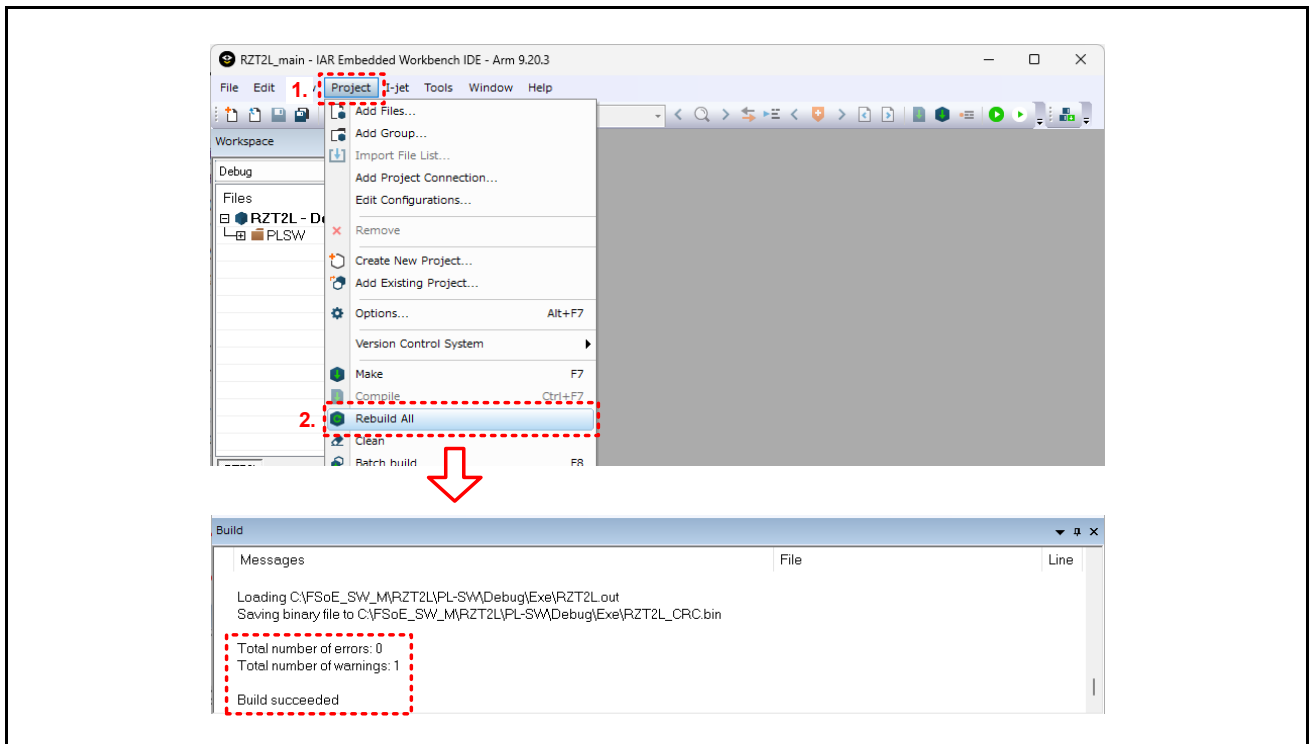


図 5-2 ビルド開始手順

注：ソフトウェアの仕様上、1件（[Pe177]）のワーニングが発生しますが無視してください。

以上で、ビルド操作は終了です。

## 5.2.5 ROMへの書き込み

図5-3に、フラッシュメモリおよびRZ/T2L内蔵RAMへのコード配置アドレスを示します。

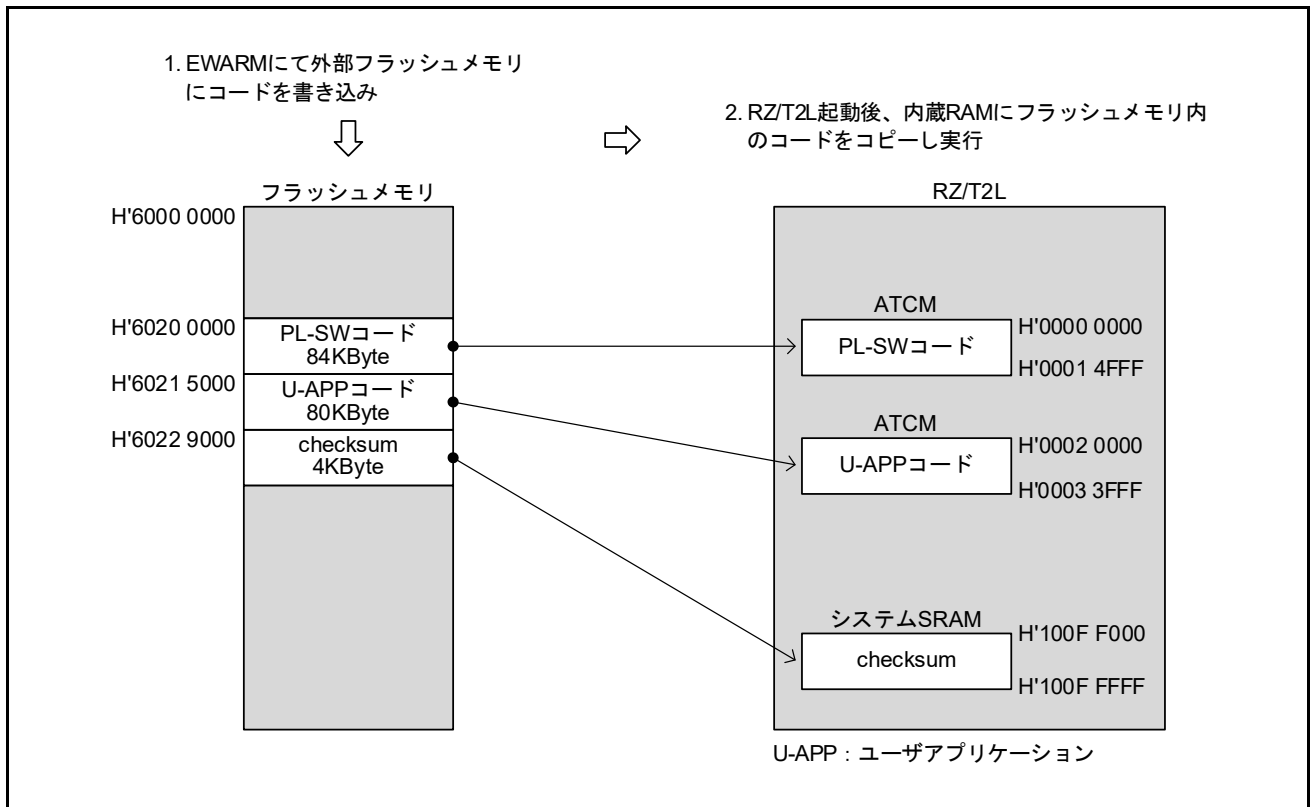


図 5-3 コード配置アドレス

以降にプロジェクトのフラッシュメモリへの書き込み手順について示します。

- 1) flashloaderフォルダ (表7.66, #10) 内のarmフォルダ (表7.66, #11) をEWARMIインストールフォルダ内にある「armフォルダ」に上書きします。

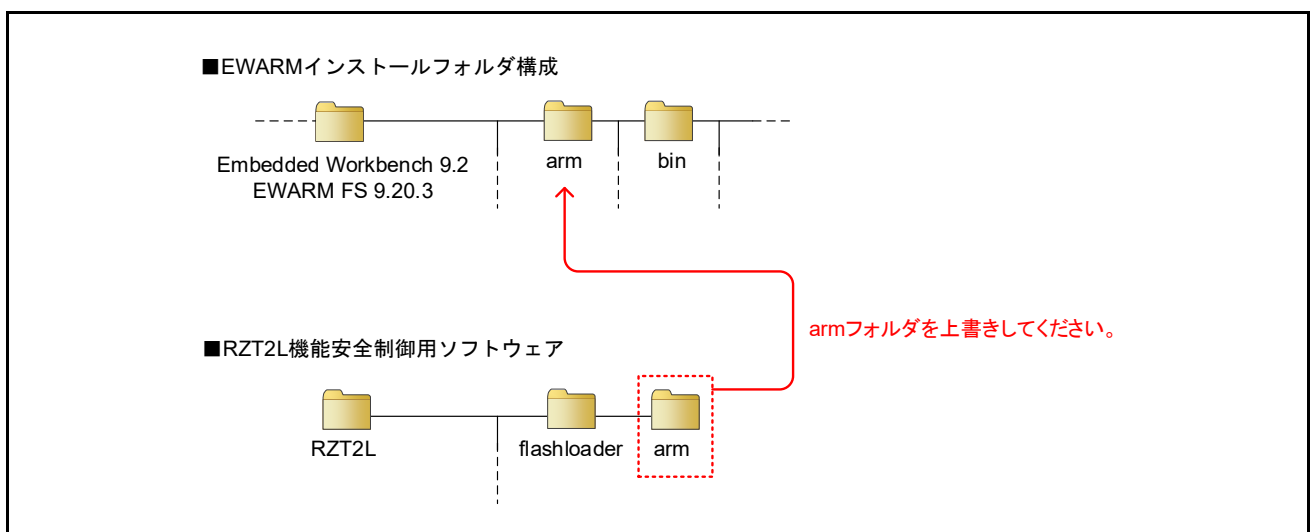


図 5-4 EWARMI へのローダファイルの追加

- 2) 先にRZ/T2L機能安全制御用ソフトウェアのビルドを完了してください。
- 3) RZ/T2L機能安全制御用ソフトウェアのワークスペースを閉じ、ローダプログラム用のワークスペースファイルを開いてください。

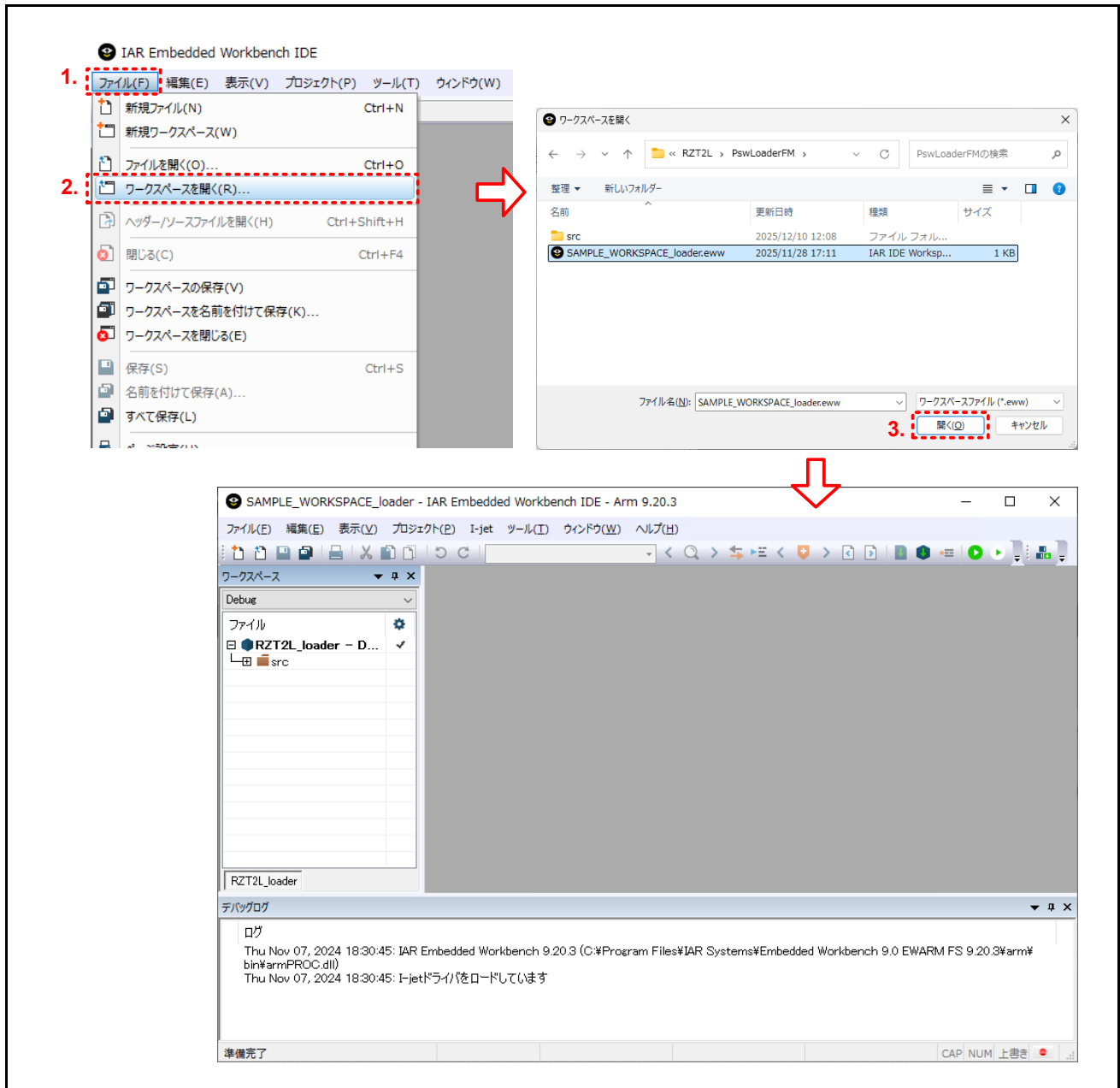


図 5-5 ローダプログラム用ワークスペースの起動

4) ローダプログラムのビルドを実行してください。

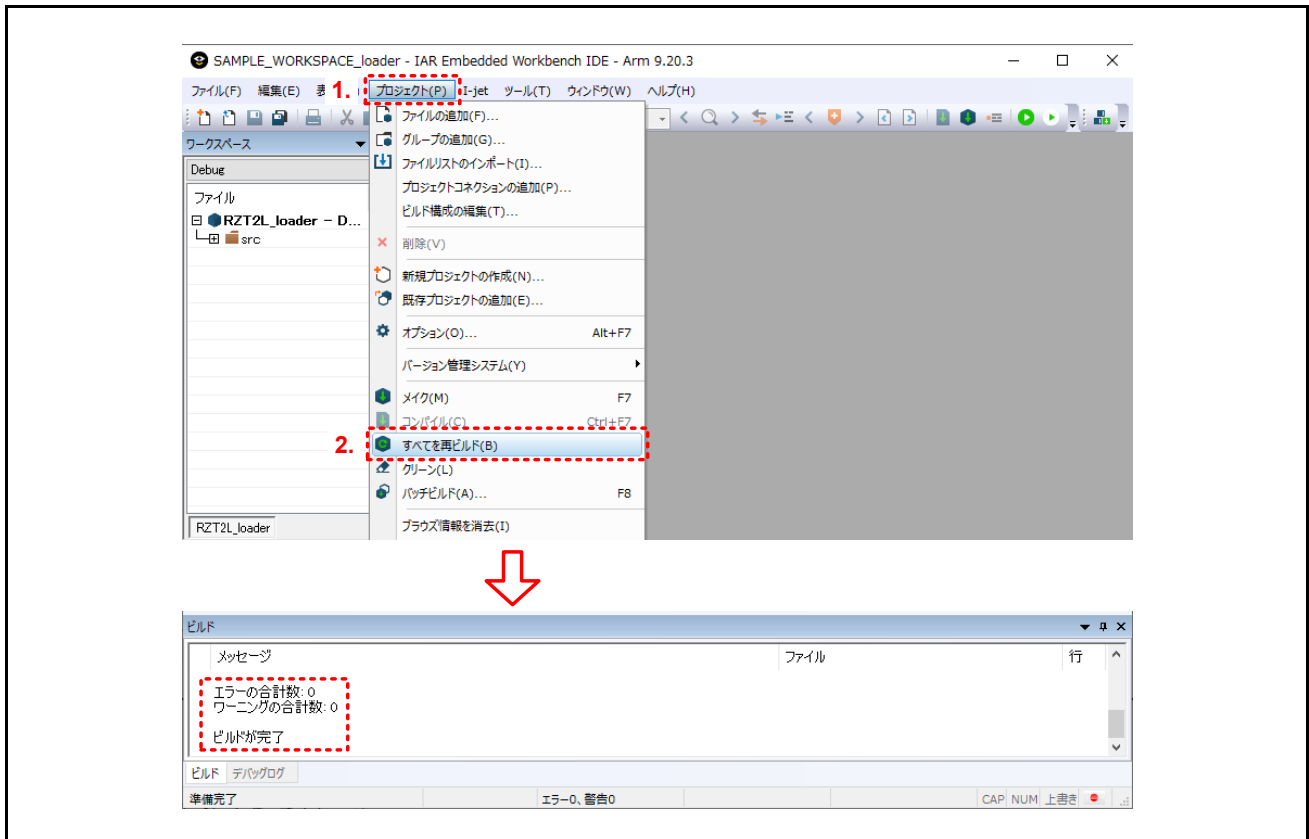


図 5-6 ローダプログラムのビルド

5) フラッシュメモリへのダウンロードを実施してください。

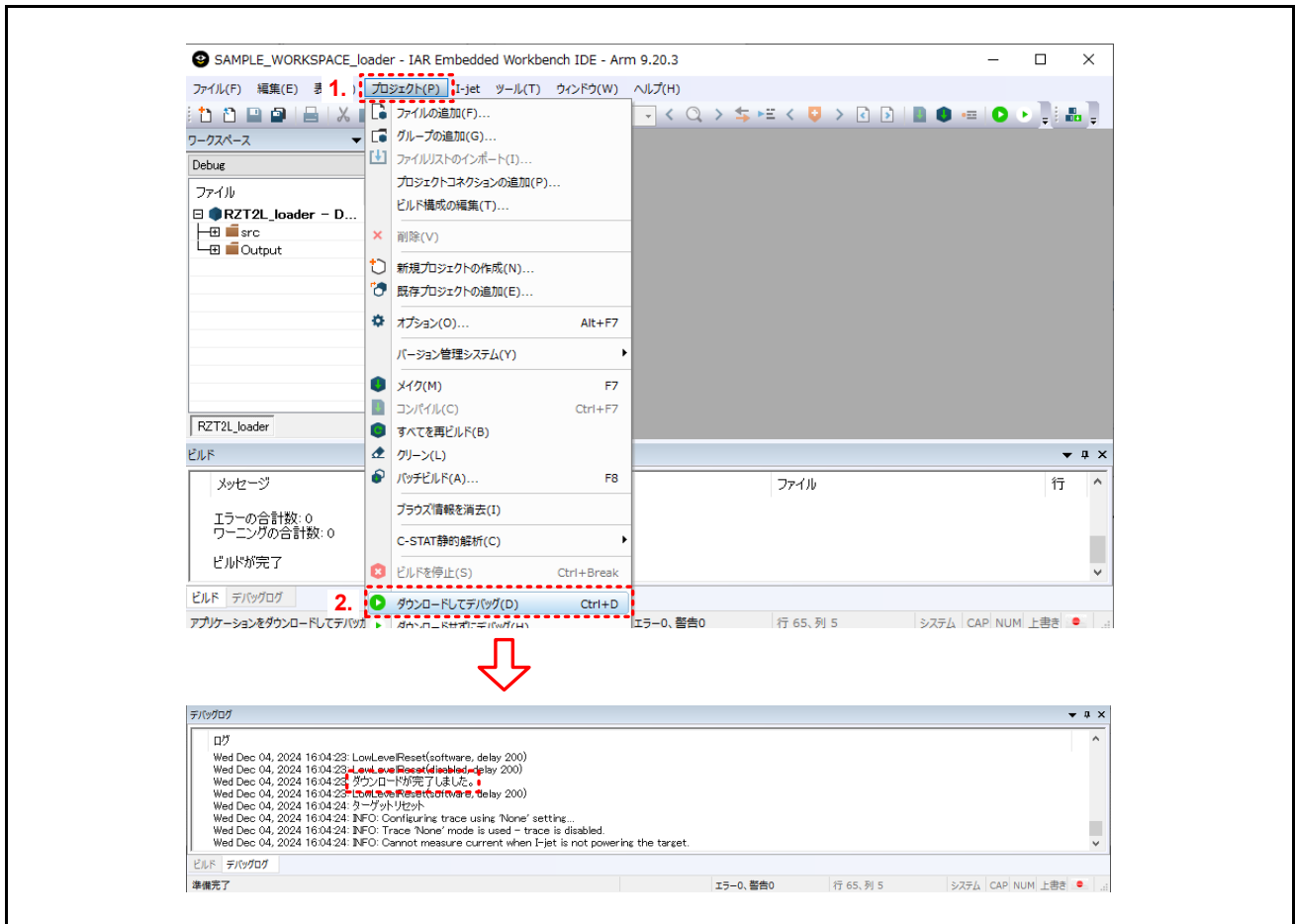


図 5-7 フラッシュメモリへのダウンロード

以上で、ROM書き込みの操作は終了です。

## 6. 実行方法

### 6.1 TwinCAT との接続

#### 6.1.1 ESI ファイルの準備 (初回のみ)

- (1) TwinCAT を起動する前に、本サンプル向け ESI ファイルを特定のフォルダへコピーする必要があります。

コピー元の ESI ファイル(Common¥ESI フォルダに格納)

- “Renesas EtherCAT RZT2 SafetyMotorSolution.xml”

コピー先フォルダ (お使いの環境によっては異なる場合があります)

- C:¥TwinCAT¥3.1¥Config¥lo¥EtherCAT

#### 6.1.2 TwinCAT ドライバのインストール (初回のみ)

- (1) TwinCAT3 を起動します。
- (2) スタートメニューから[TwinCAT3]→[Show Realtime Ethernet Compatible Devices...]を選択します。
- (3) 使用するイーサネットアダプタを選択し、インストールを行います。
- (4) 選択したイーサネットアダプタが[Installed and ready to use devices(realtime capable)]に移動すれば問題ありません。

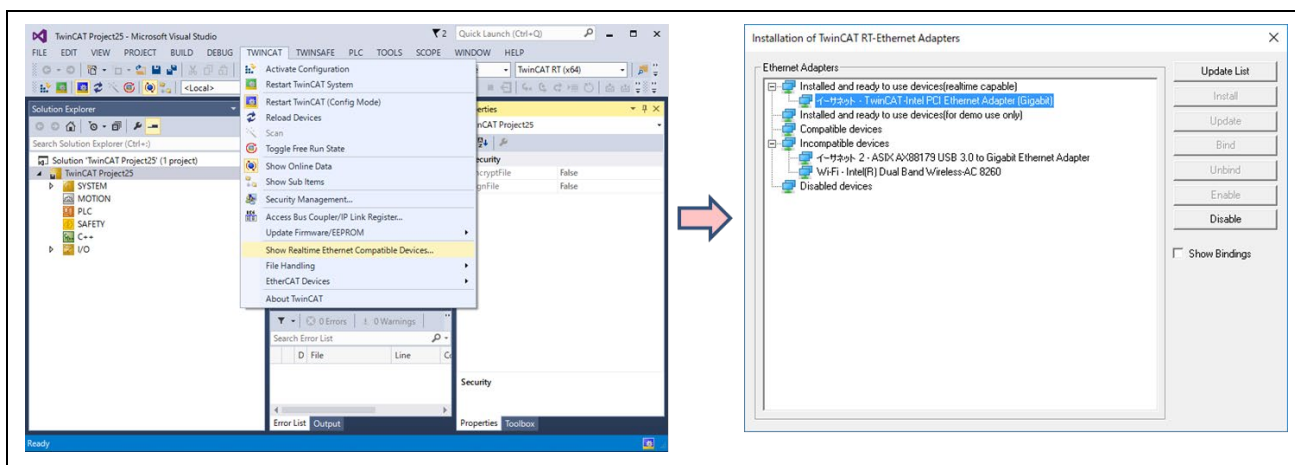


図 6-1. TwinCAT ドライバのインストール

## 6.1.3 デバイスのスキャン・EEPROM への書き込み

- (1) TwinCAT で新規プロジェクトを作成します。
- (2) ソリューションエクスプローラ→I/O →デバイス→「スキャン」を選択します。

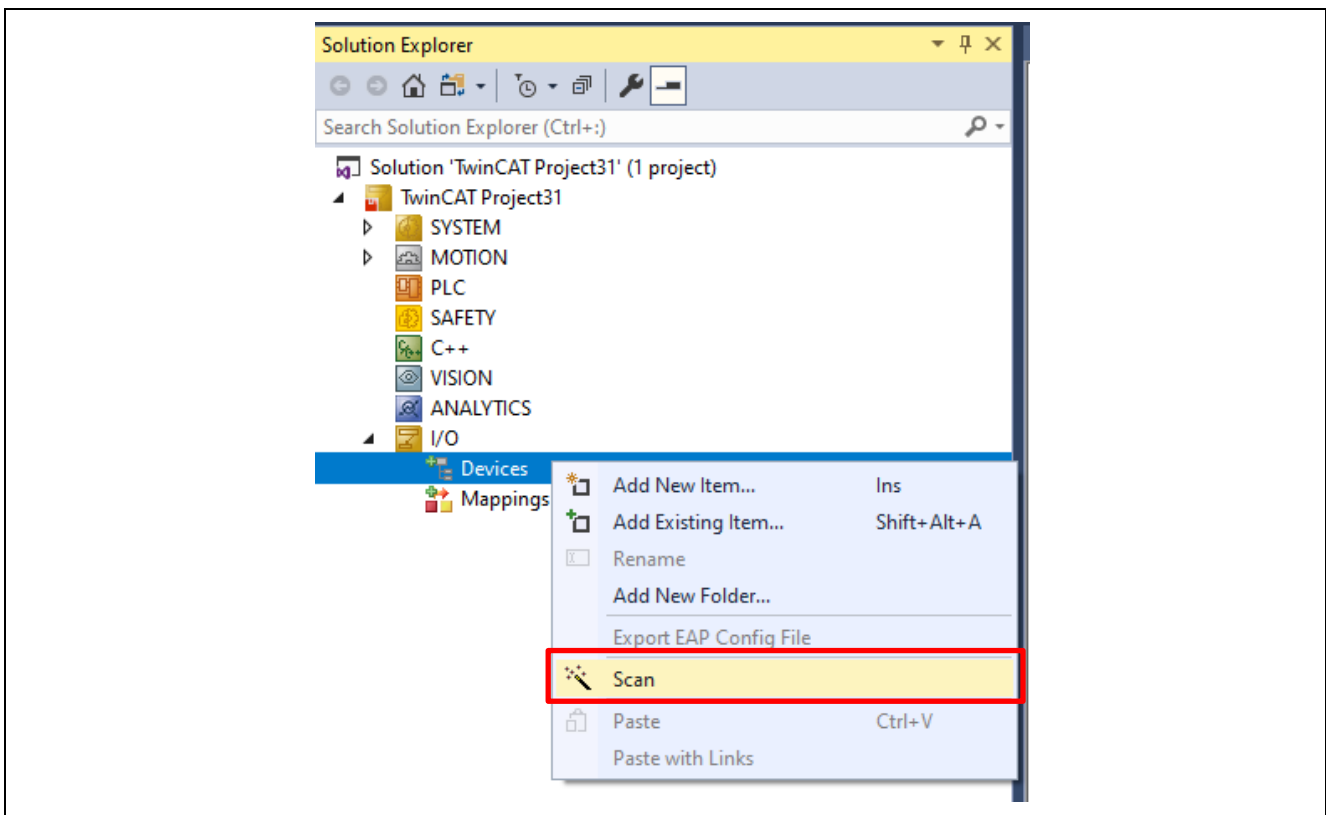


図 6-2. デバイスのスキャン

- (3) RZ/T2M が EtherCAT として認識された場合、下記のように EtherCAT ポートが表示されます。

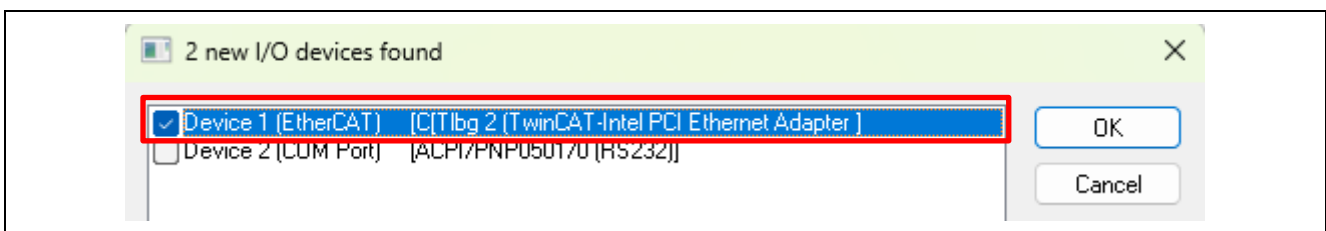


図 6-3. スキャン結果

(4) [Scan for boxes]を実行します。

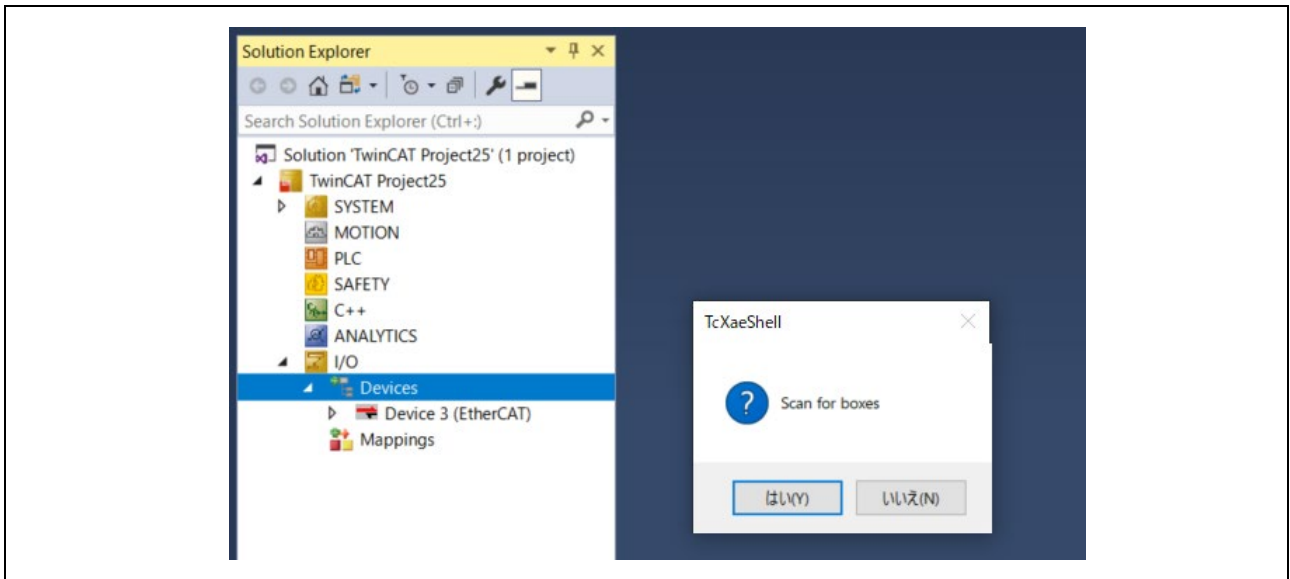


図 6-4. Scan for boxes ウィンドウ

(5) [Activate free run]は OK を押します。

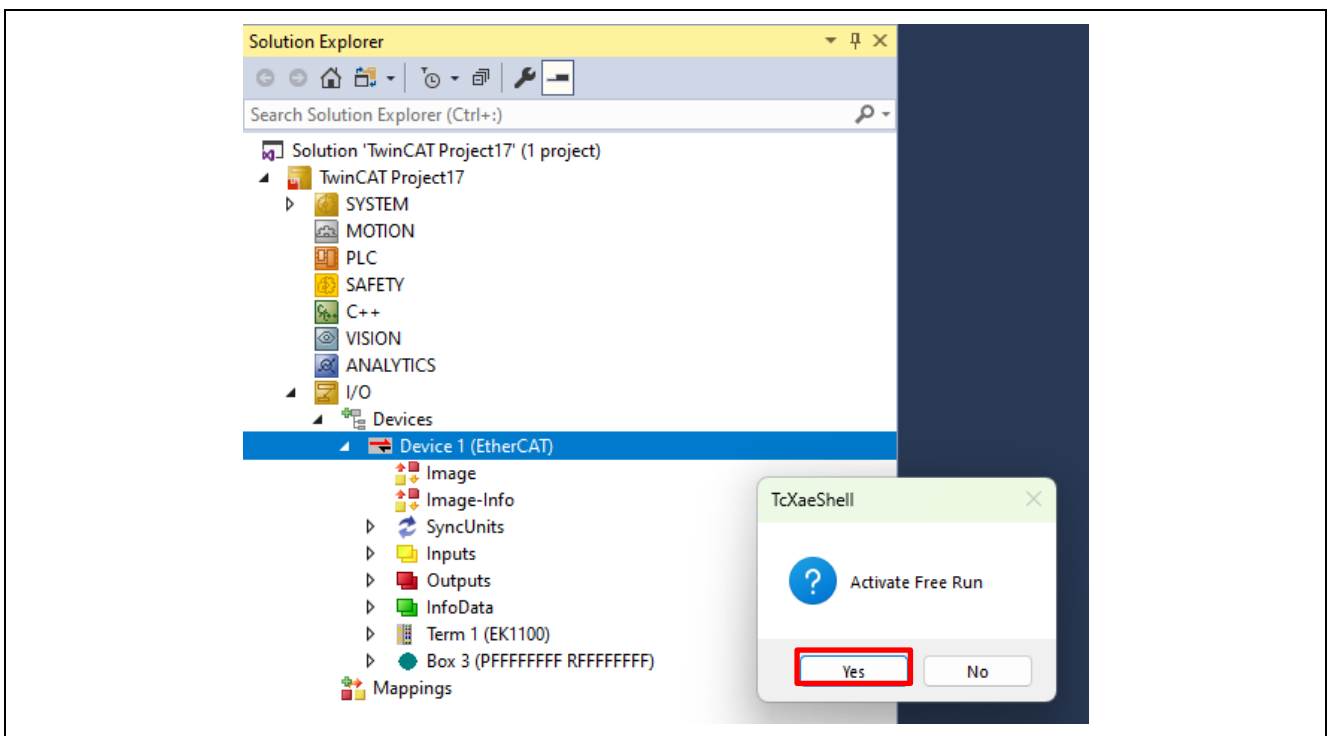


図 6-5. Active Free Run の有効化

(6) Box x (PFFFFFF..)と表示されている場合、セーフティ制御ボードの EEPROM は空の状態です。EEPROM に本サンプル用の EtherCAT SubDevice 情報を書き込む必要があります。

(7) EEPROM に書き込む手順は下記の通りです。

- [Box X] をダブルクリックしてください。設定画面が表示されます。
- [EtherCAT] tab を選択してください。
- [Advanced Setting] ボタンを実行してください。

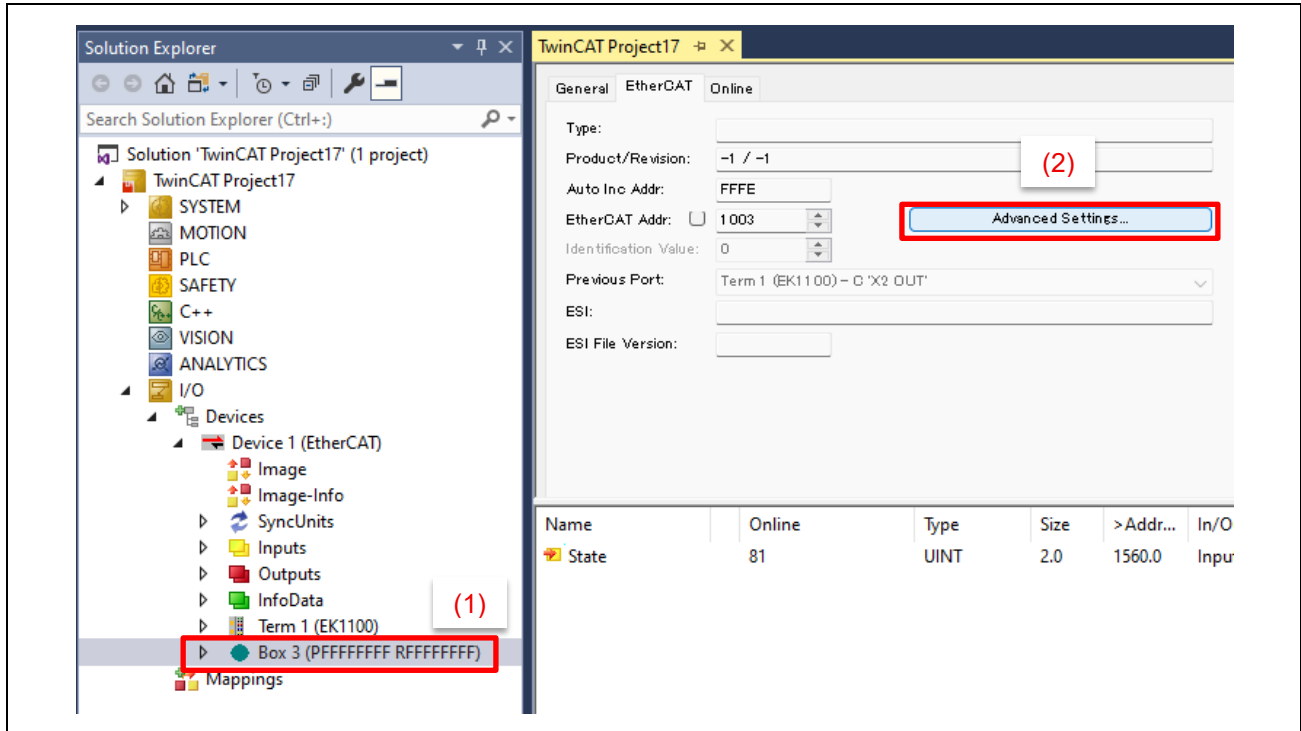


図 6-6 EEPROM 書き込み方法 1

- [ESC Access] → [EEPROM] → [Hex Editor]を選択してください
- [Download from List] を選択してください。

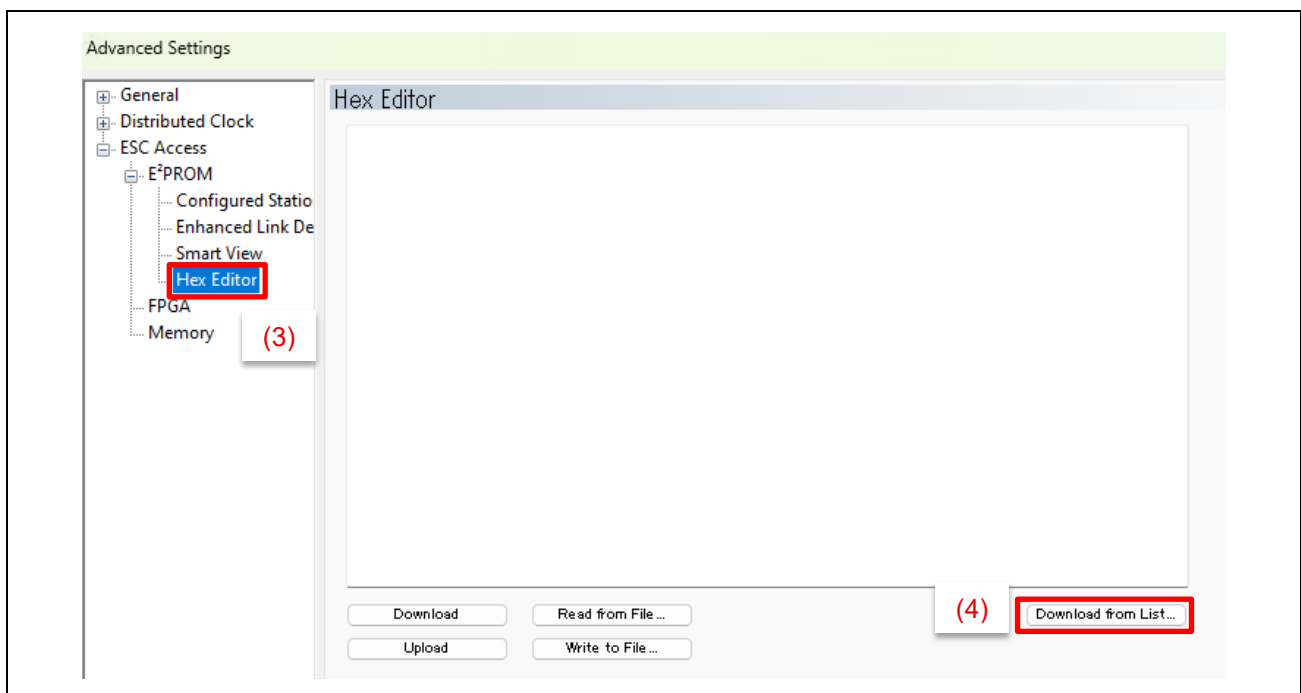


図 6-7. EEPROM 書き込み方法 2

- TwinCAT3 に登録してある ESI ファイルの一覧が現れますので、該当するファイルを選択してください。本サンプルでは、[Renesas EtherCAT RZ/T2 SafetyMotorSolution] を選択します。

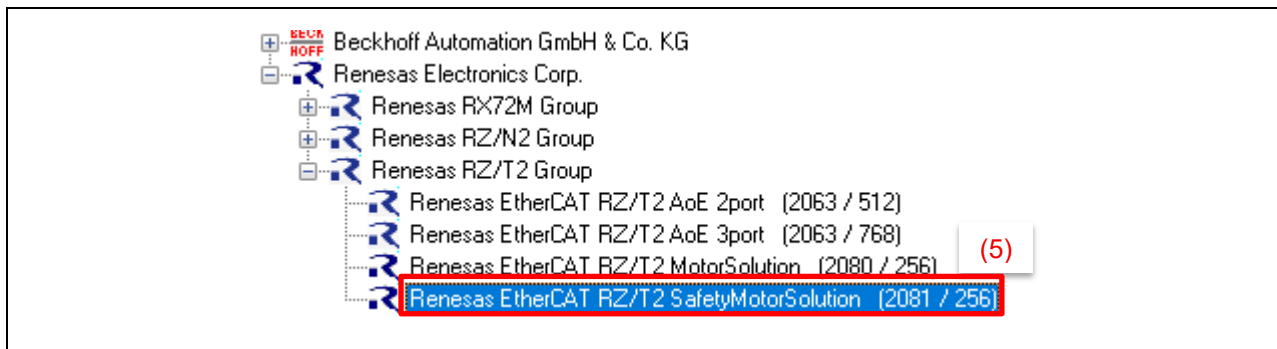


図 6-8. EEPROM 書き込み方法 3

- (8) EEPROM への書き込み完了後、ボードの SW2 を押して RZ/T2M を再起動し、EEPROM を再ロードします。
- (9) TwinCAT 上で[Device x (EtherCAT)]を一旦削除した後、再スキャンを行います。

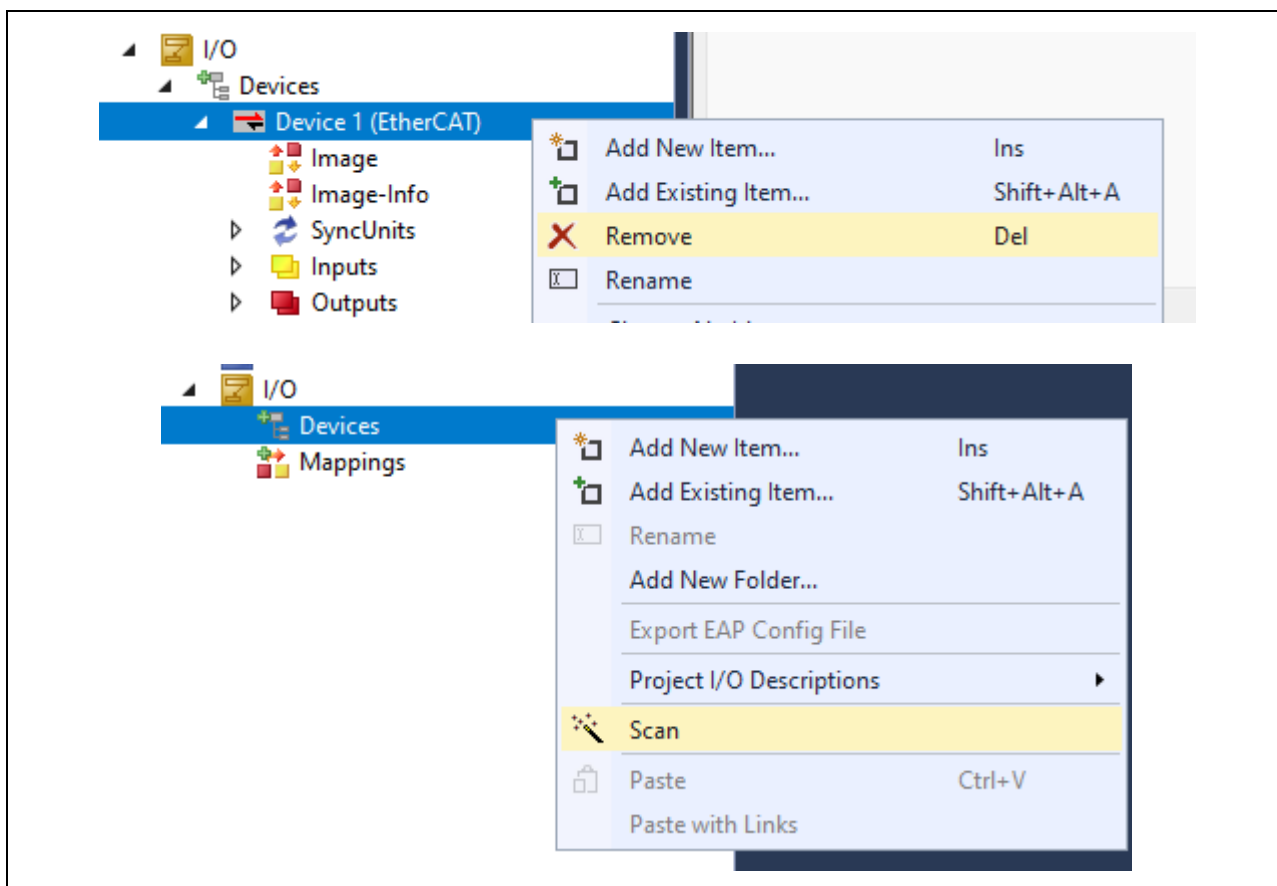


図 6-9. デバイスの再スキャン

(10) 正常にスキャンされた場合、下記の表示が出ます。

EtherCAT drive は動作確認で使用しないため、[Cancel]を選択します。

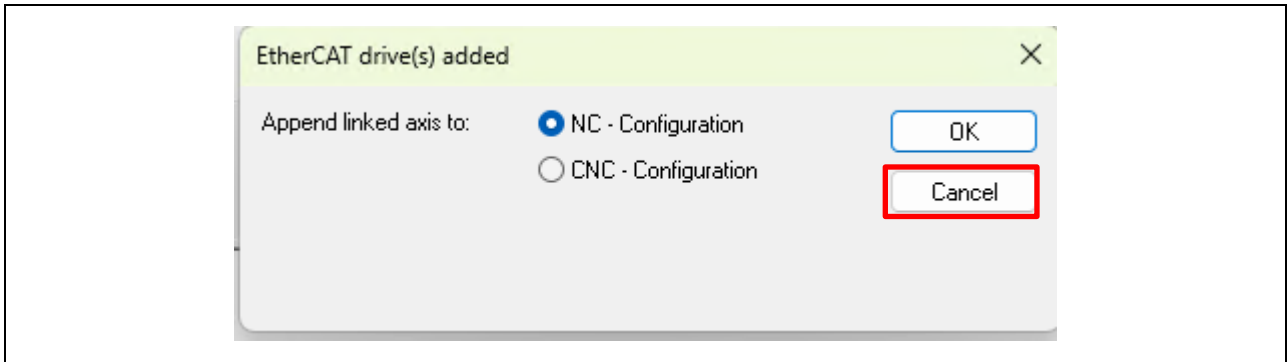


図 6-10. EtherCAT drive の選択

(11) [Reload Devices]を実行し、ネットワークを再起動します。

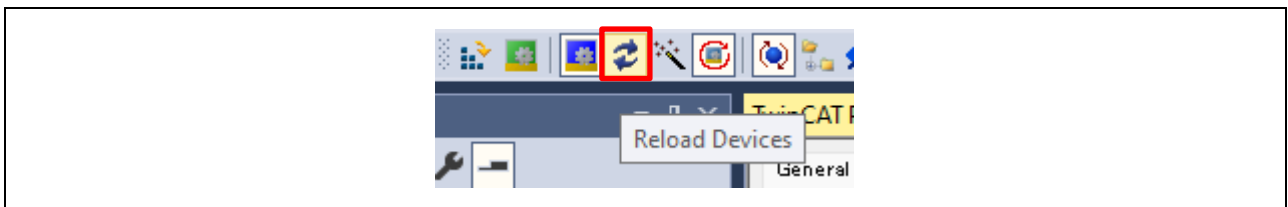


図 6-11. ネットワークの再起動

(12) ツリー上に、下記の画像のようにデバイスが表示されており、[Online]タブで Current State が"OP"になっていることを確認します。

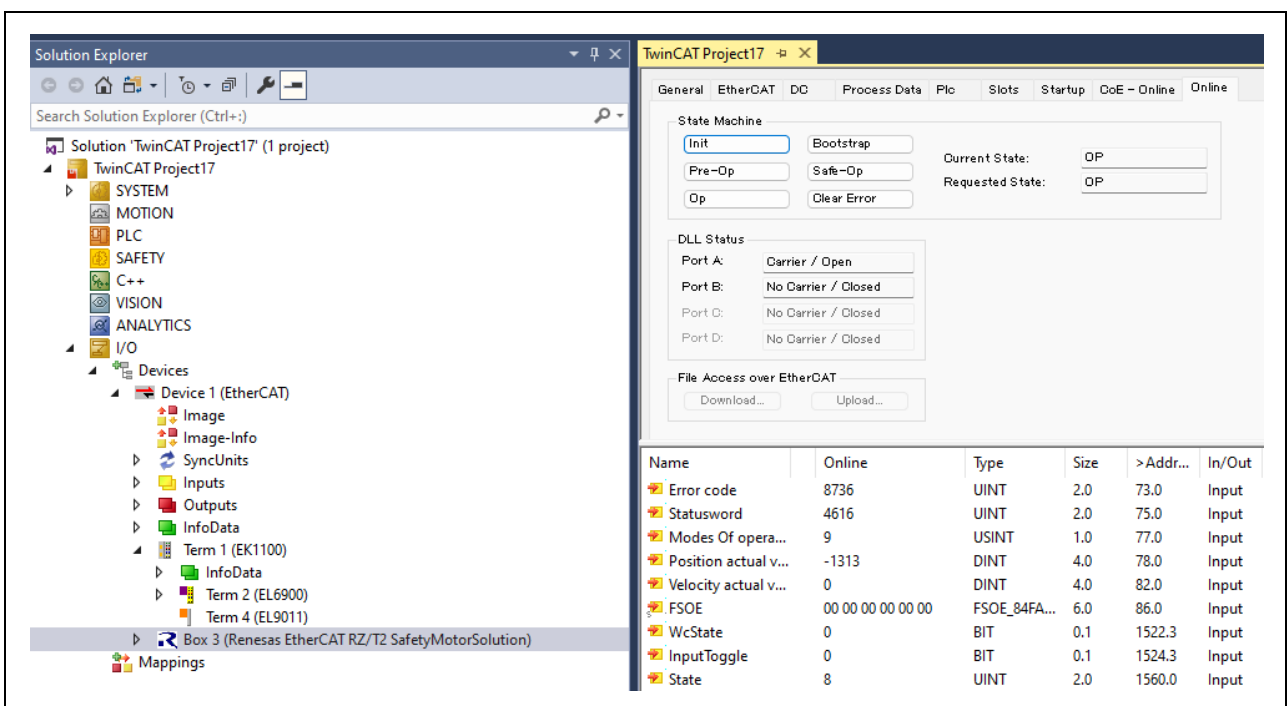


図 6-12. "OP"の確認

## 6.2 FSoE マスタの準備

この章では FSoE マスタプログラムの作成方法を示します。なお、FSoE マスタには指定の機器はありませんが、本資料では Beckhoff Automation 社の下記製品を使用した場合を例として説明を行います。

- ・ EK1100 : EtherCAT Coupler
- ・ EL6900 : EtherCAT Terminal communication interface, TwinSAFE Logic
- ・ EL9011 : Bus end cover for E-bus contacts

なお、この作業は「6.1.3 デバイスのスキャン・EEPROM への書き込み」を行い、システムマネージャツリーに Box が追加されている状態で実施してください。

※6.2 章では TwinCAT4026.19 で作成された Safety プロジェクトおよび PLC プロジェクトを追加します。

お使いの TwinCAT のバージョンが古い場合は正常に追加できない可能性があります。

追加できない場合は付録 : FSoE マスタプログラムの作成手順を参照してください。

### 6.2.1 Safety プロジェクトの追加

作成済みの Safety プロジェクトを追加します。

- 1) TwinCAT3にて[Solution Explorer] → [Safety\_Drive] → [SAFETY] → [Add Existing Item...]を選択してください。
- 2) Safety\_Drive.tfzipファイル（表7.66, #13）を選択してください。
- 3) [開く]をクリックすることで、Safetyプロジェクトが追加されます。

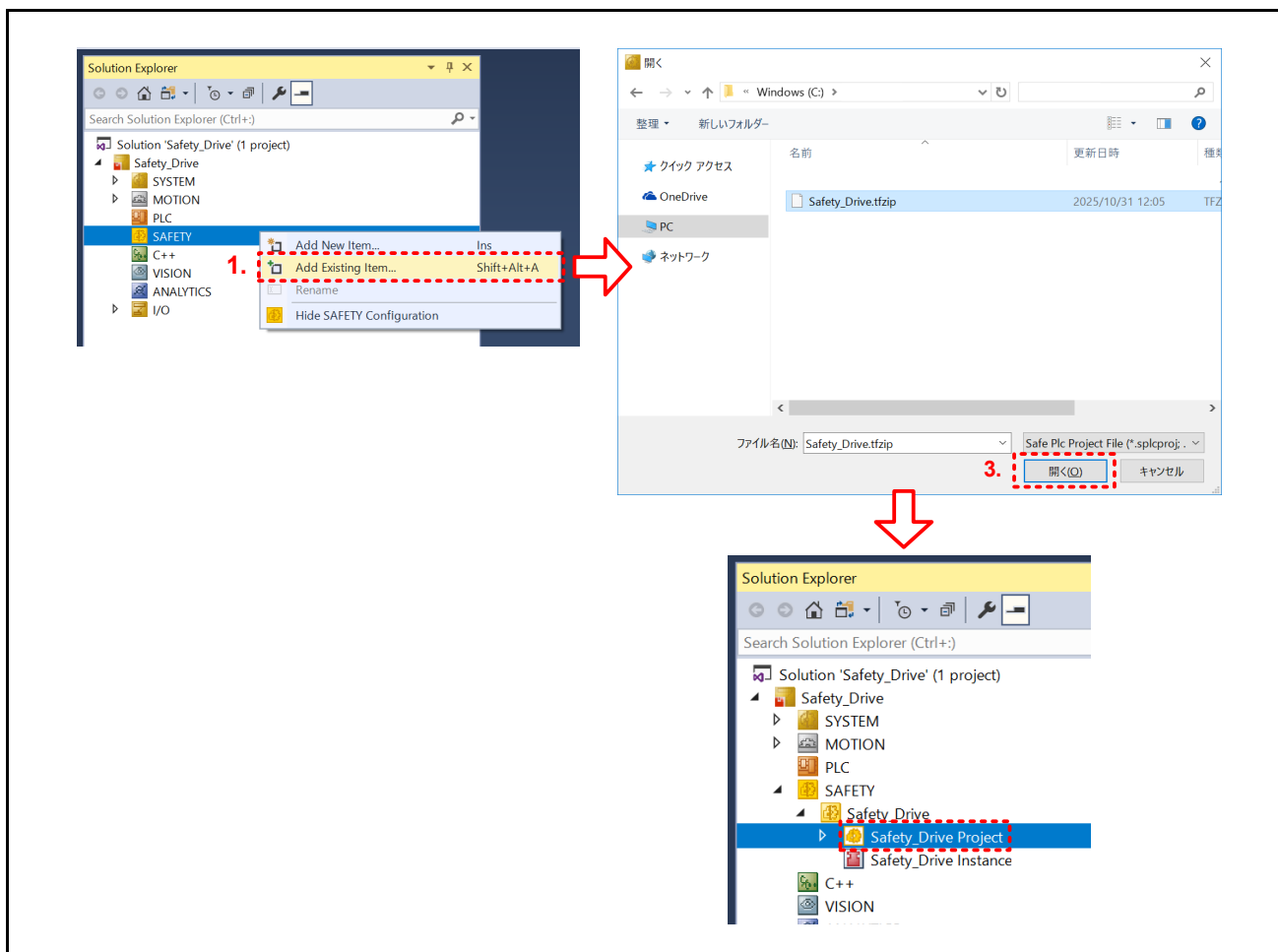


図 6-13. Safety プロジェクト追加手順

## 6.2.2 PLCプロジェクトの追加

以降に、PLCプロジェクトの追加手順を示します。

- 1) TwinCAT3にて[Solution Explorer] → [Safety\_Drive] → [PLC] → [Add Existing Item...]を選択してください。
- 2) Safety\_Drive.tpzipファイル（表7.66, #14）を選択してください。
- 3) [開く] → [OK]をクリックすることで、PLCプロジェクトのロードが行われます。

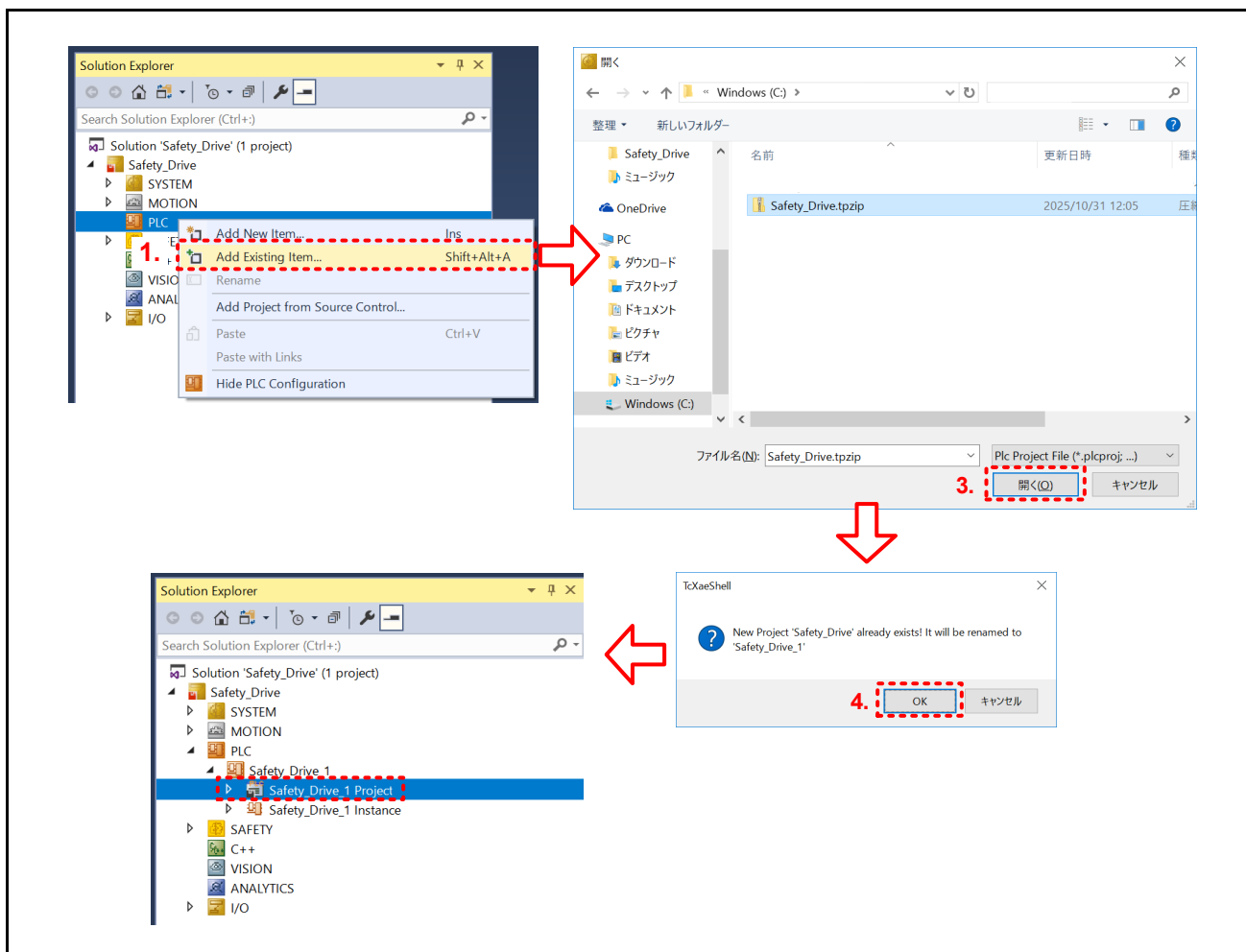


図 6-14. PLC プロジェクトの追加

以上で、PLCプロジェクトの追加作業は完了です。

### 6.2.3 ErrAck 信号の選択

1. TwinCAT3 のシステムマネージャツリーで、[SAFETY]→[safety\_project\_1]→[safety\_project\_1 project]→[TwinsafeGroup1]→[Alias Devices]を順に選択し、その下にある [ErrorAcknowledgement.sds] をダブルクリックします。

1. [Full Name]横のアイコンをクリックします。

2. MAIN.bErrAck を選択して[OK]をクリックします。

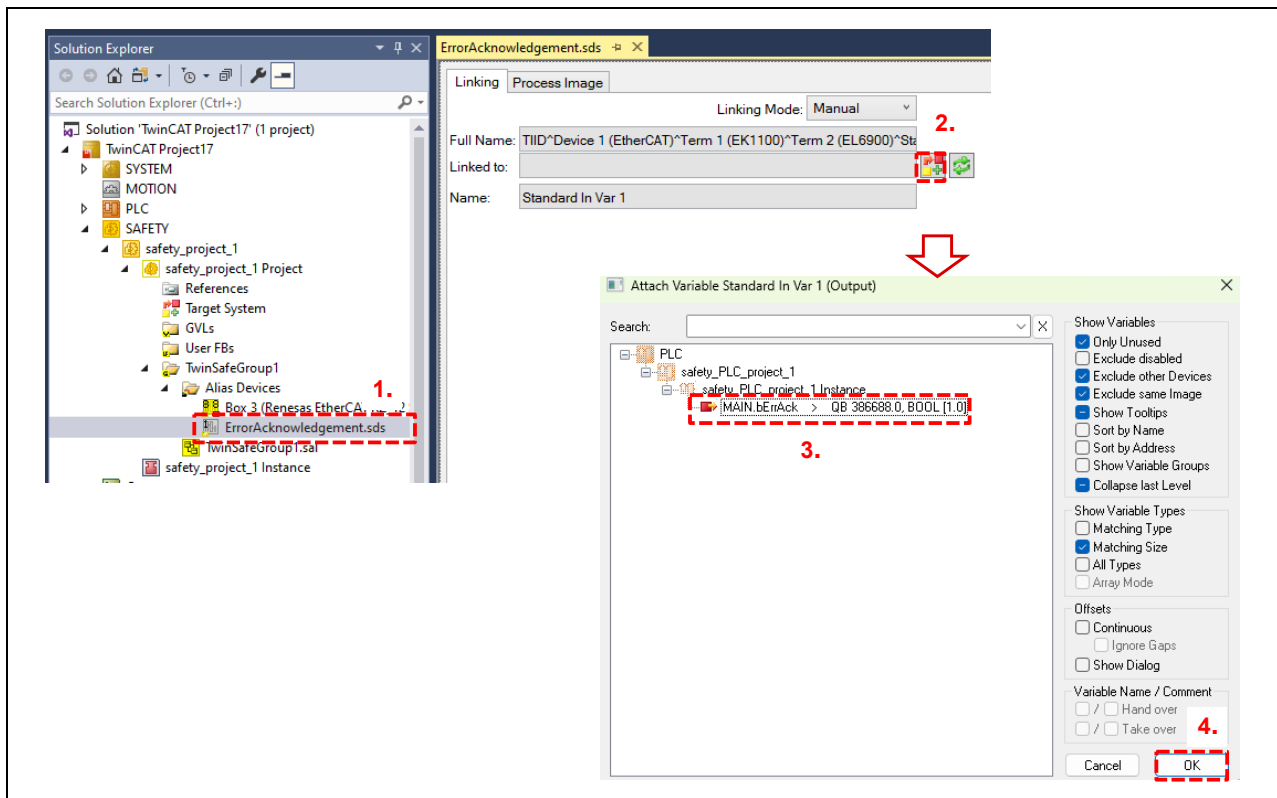


図 6-15. ErrAck 信号の選択

## 6.2.4 FSoE マスタ機器の確認

1. TwinCAT3 のシステムマネージャツリーで、[SAFETY]→[Safety\_Drive]を順に選択し、[Safety\_Drive Project]を右クリックします。
2. [Properties]を選択します。
3. [Target System]タブをクリックし、接続する FSoE マスタ機が正しく選択されていることを確認します。（図では "EL6900" を選択）
4. デバイス情報の「Serial Number」は「6.3」章にて必要となります。
5. [Safe Address]と[Hardware Address]が一致していることを確認します。[Safe Address]と[HardwareAddress]が異なる場合、(a)→(b)の順でアイコンをクリックし、アドレスを一致させてください。

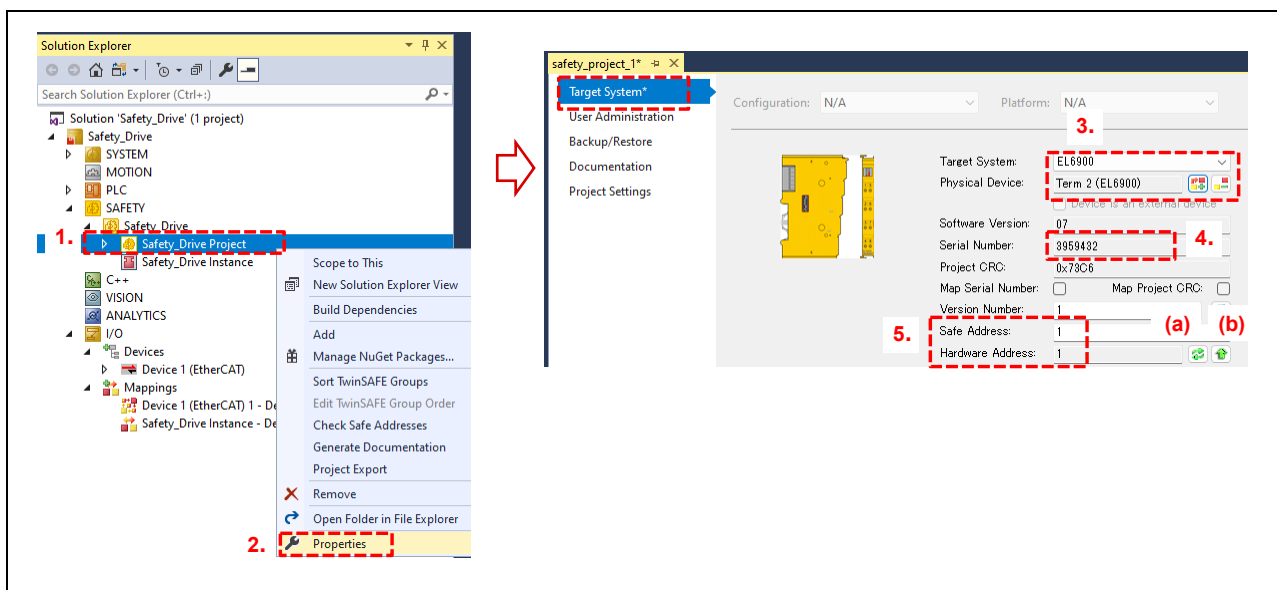


図 6-16. FSoE マスタ機器の確認

### 6.3 FSoE マスタプロジェクトのダウンロード

(1) はじめに、プログラムダウンロードの前に、作成したプログラムに問題がないか検証を行います。

1. [TWINSAFE]タブから[Verify Safety Project]を選択します。
2. 問題がなければ、[Verification Process succeeded]と表示されます。
3. [Verify Complete Safety Project]を選択します。
4. 問題がなければ、[Verification Process succeeded]と表示されます。

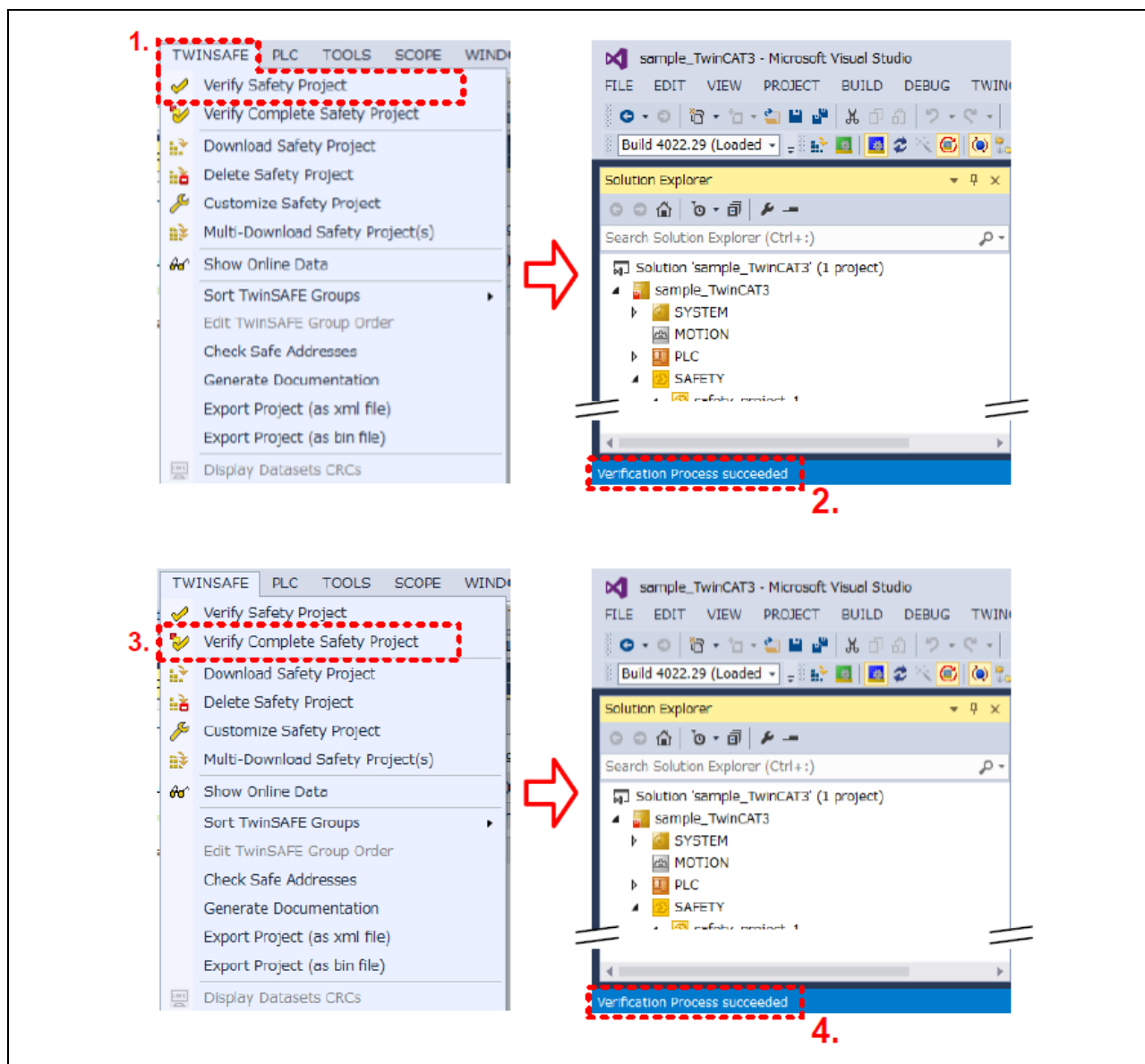


図 6-17. FSoE マスタプログラムの検証

注 : [Verification Process failed]と表示された場合、1つのネットワーク上に同じ FSoE スレーブアドレスが複数存在しないか確認のうえ、6.2章の設定を見直してください。

(2) 作成したプログラムを FSoE マスタ機器にダウンロードします。

1. [TWINSAFE]タブから[Download Safety Project]をクリックします。
2. 確認画面が出たら[はい]をクリックします。これによりダウンロードが行われます。

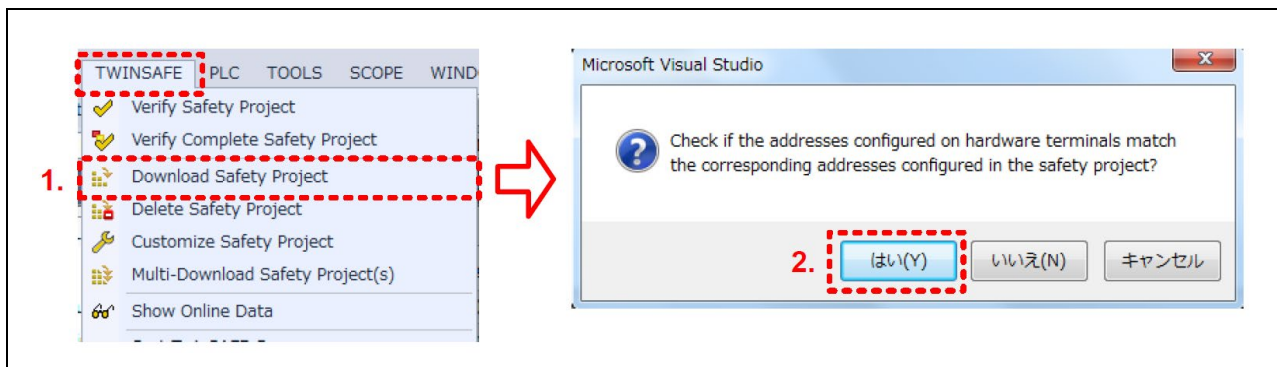


図 6-18. FSoE マスタプログラムのダウンロード開始

(3) ダウンロード結果を確認します。

1. [Login]タブにて下記値を入力し[Next]をクリックします。
  - ・ [Username] : Administrator
  - ・ [Serial Number] : 6.2 章で確認したシリアルナンバー
  - ・ [Password] : TwinSAFE
2. [Download Result]タブにて、全て成功していることを確認して[Next]をクリックします。
3. [Final Verification]タブにて、最終データに問題がないことを確認して、"I have manually ~"の項目にチェックを入れ、[Next]をクリックします。
4. [Activation]タブにて、再度[Password]を入力し、[Finish]をクリックします。

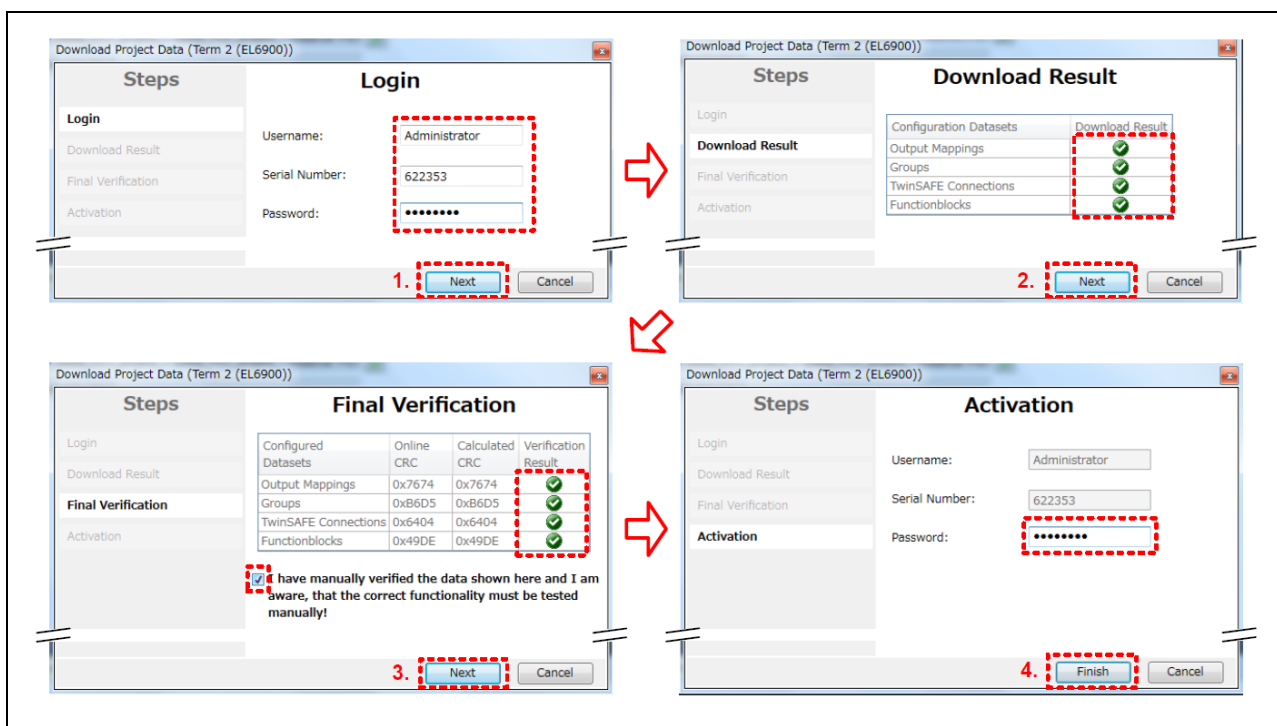


図 6-19. FSoE マスタプログラムのダウンロード

## 6.4 TwinCAT プロジェクトの実行方法

はじめに、Safety Motor Control Reference Kit に電源投入を行い、ボード上のマイコンに書き込んだプログラムを動作させてください。

以降に、TwinCAT プロジェクトを実行するまでの手順を示します。

- (1) はじめに、EtherCAT 通信周期を設定します。本章では、EtherCAT 通信周期を 250us にする場合の設定例を示します。
  1. TwinCAT3 のシステムマネージャツリーで[SYSTEM]→[Real-Time]を順にクリックし、プロパティ画面を開きます。
  2. [Settings]タブをクリックします。
  3. [Base Time]のプルダウンメニューから"250us"を選択します。
  4. TwinCAT3 のシステムマネージャツリーで[SYSTEM]→[Tasks]→[PlcTask]を順にクリックし、プロパティ画面を開きます。
  5. [Task]タブをクリックします。
  6. [Cycle ticks]左側の設定メニューにおいて、スピンドタンで"1"を設定します。
  7. [Cycle ticks]右側の設定メニューの値が、"0.250ms"になっていることを確認します。

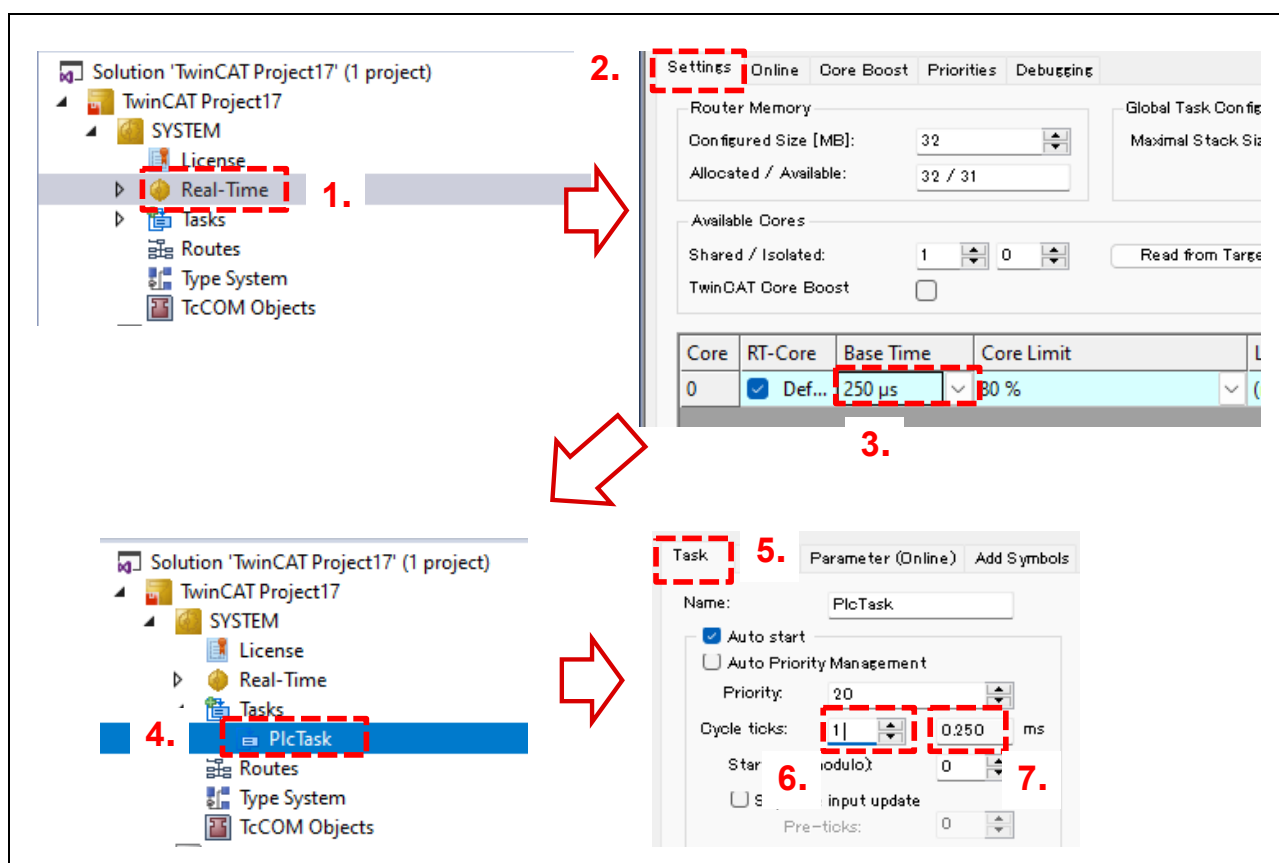


図 6-20. EtherCAT 通信周期の設定

(2) 次に、EtherCAT 同期モードを DC 同期モードに設定する手順を示します。

1. TwinCAT3 のシステムマネージャツリーで[Box X (Renesas EtherCAT RZ/T2 SafetyMotorSolution)] をクリックし、プロパティ画面を開きます。
2. [DC]タブをクリックします。
3. Operation Mode で、[DC-Synchron]を選択します。
4. [Advanced Settings...]をクリックします。
5. [Enable SYNC1]にチェックを入れます。
6. [Shift Time (us):]にて、SYNC1 割り込みの発生タイミングを設定します。
7. [OK]をクリックします。

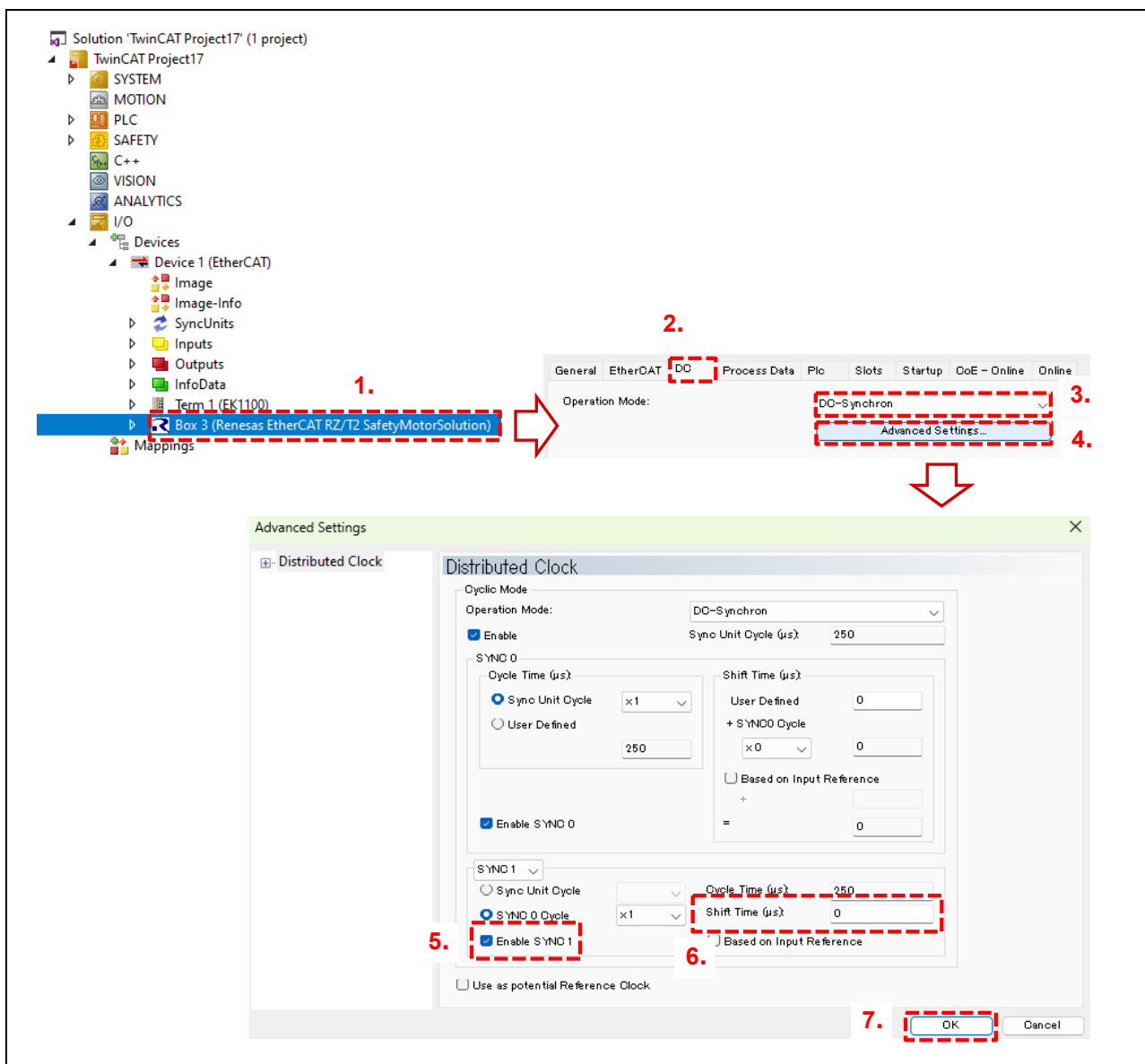


図 6-21. DC 同期モードの設定 1

8. TwinCAT3 のシステムマネージャツリーで[I/O]→[Devices]→[Device X (EtherCAT)]を順にクリックし、プロパティ画面を開きます。
9. [EtherCAT]タブをクリックします。
10. [Advanced Settings...]をクリックし、Advanced Settings ウィンドウを開きます。
11. Advanced Settings ウィンドウにおいて、[Distributed Clocks]をクリックします。
12. Percent of cycle time にて、SYNC0 割り込みの発生タイミングを設定します。
13. [OK]をクリックします。

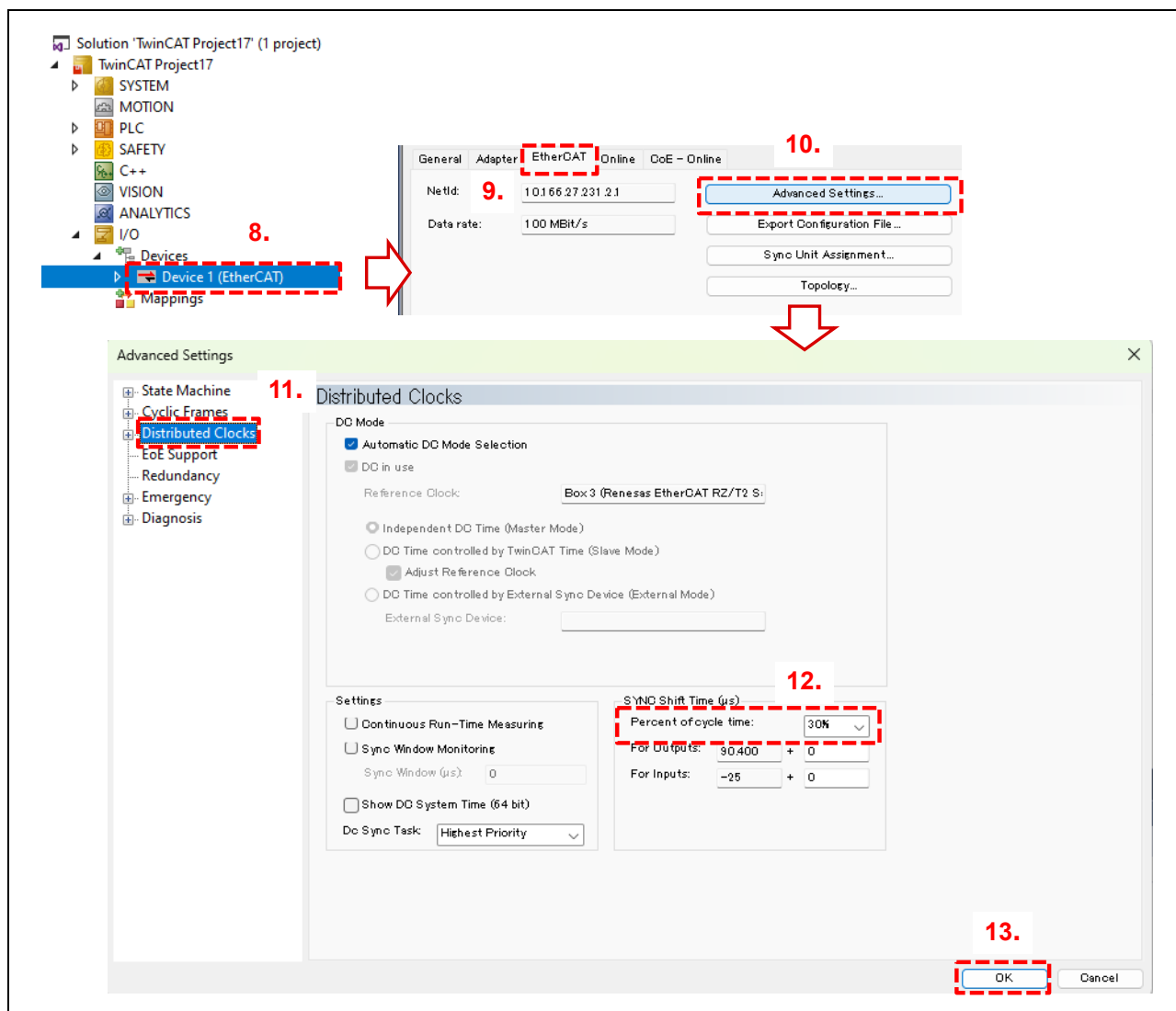


図 6-22. DC 同期モードの設定 2

(3) 次に、EtherCAT ネットワークの設定を有効化します。

1. TwinCAT3 のシステムマネージャツリーで[I/O]→[Devices]→[Device X (EtherCAT)]を順にクリックし、プロパティ画面を開きます。
2. [EtherCAT]タブをクリックします。
3. [Sync Unit Assignment...]をクリックし、Sync Unit Assignment ウィンドウを開きます。
4. Task が<default>に設定されている場合、PlcTask に変更します。
5. [Apply]をクリックした後、[OK]をクリックします。

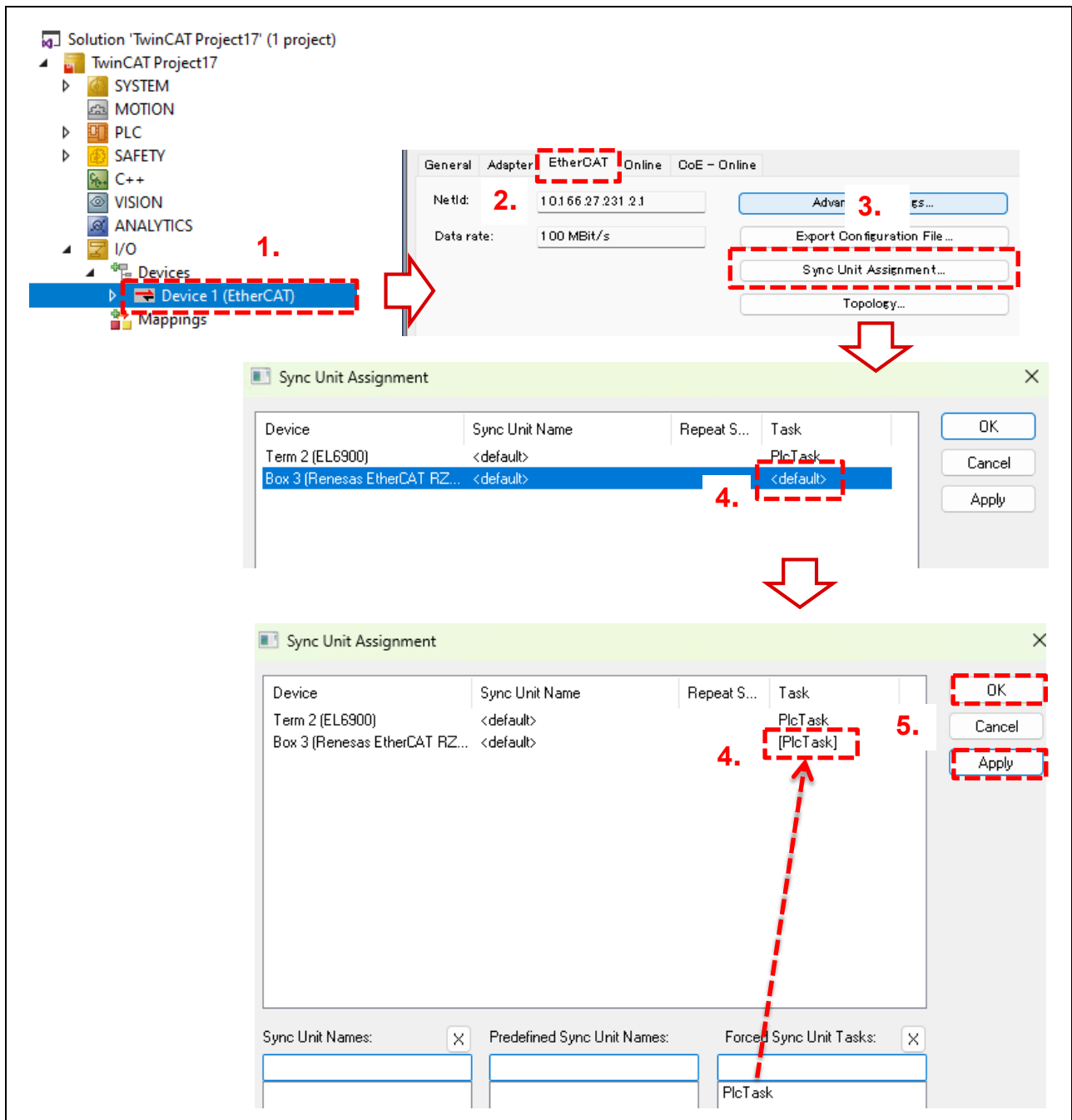


図 6-23. Sync Unit Assignment 設定

6. TwinCAT3 の[Activate Configuration]アイコンをクリックします。
7. "Activate Configuration"ウィンドウが表示されるので、[OK]をクリックします。
8. 続いて、"Restart TwinCAT System in Run Mode"ウィンドウが表示されるので、[OK]をクリックし、TwinCAT3 プロジェクトを Run Mode に切り替えます。

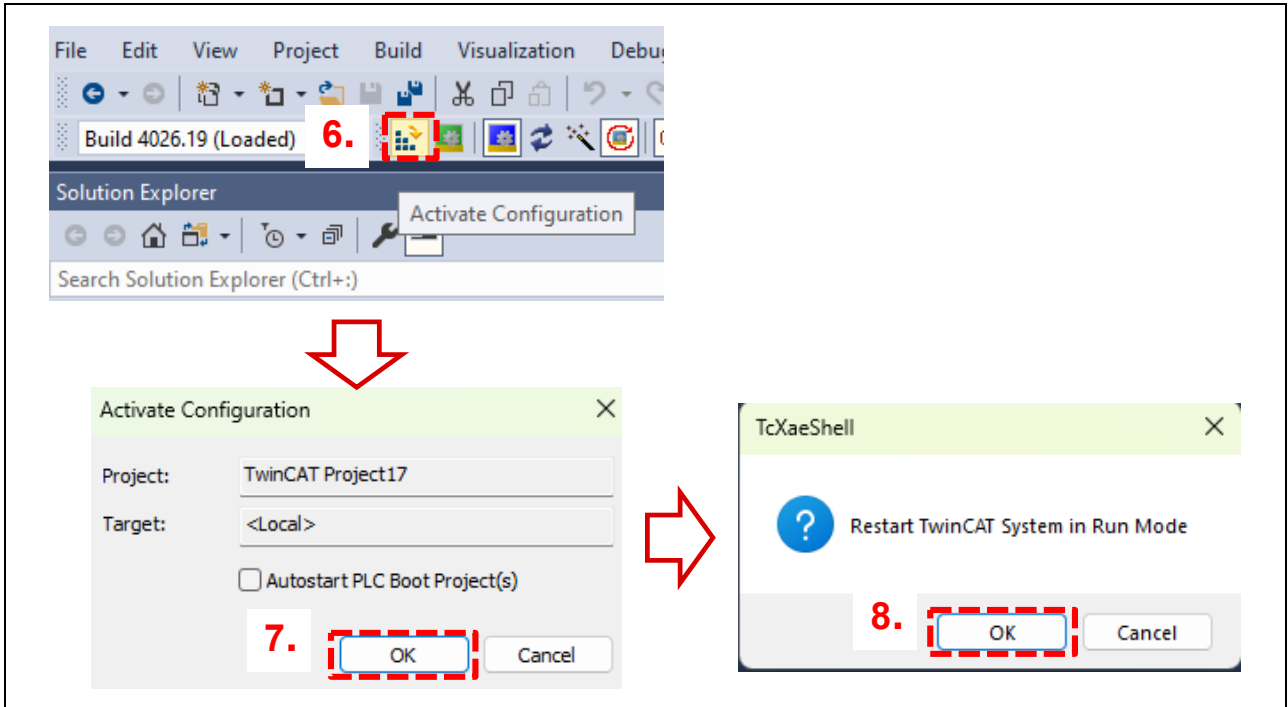


図 6-24. Run mode の開始

(4) 次に、TwinCAT3 と Safety Motor Control Reference Kit の接続状態を確認します。

1. TwinCAT3 のシステムマネージャツリーで、[Box X (Renesa EtherCAT RZ/T2 SafetyMotorSolution)] をダブルクリック、プロパティ画面を開きます。
2. [Online]タブをクリックします。
3. "Current Status"が[OP]になっていれば、TwinCAT3 と Safety Motor Control Reference Kit が接続されている状態です。

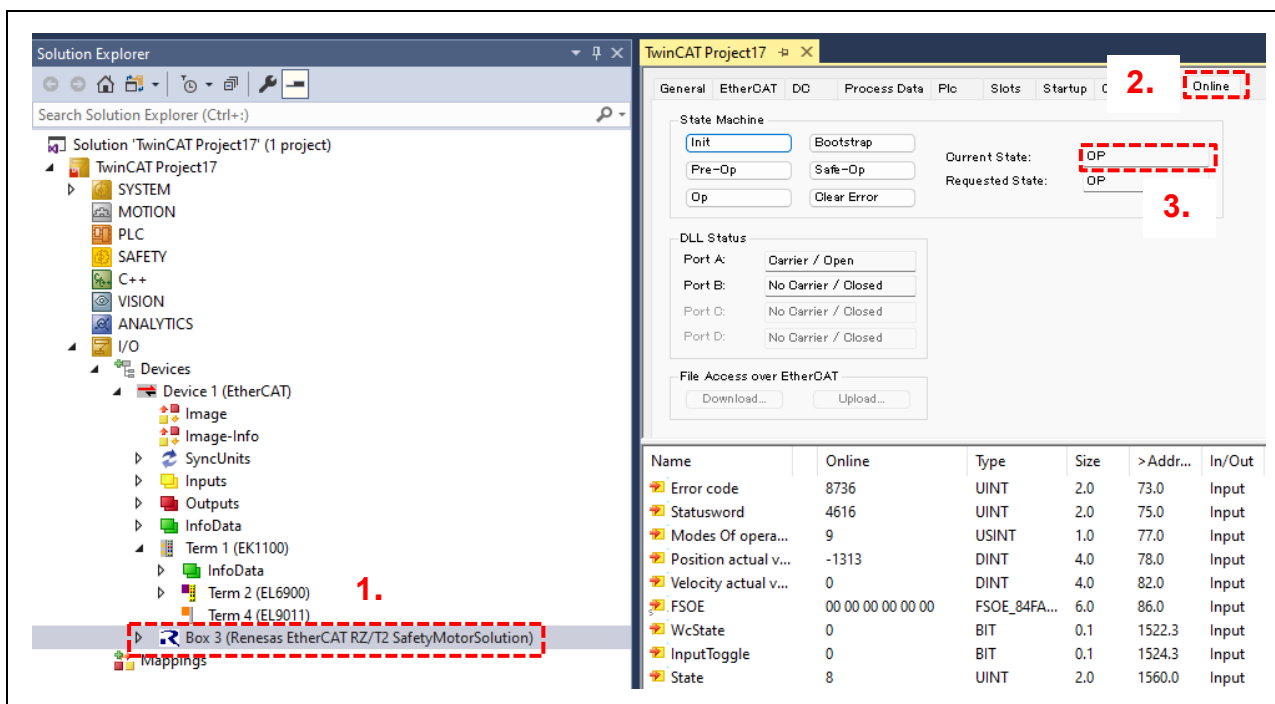


図 6-25. TwinCAT3 と Safety Motor Control Reference Kit の接続状態の確認

(5) 次に、TwinCAT プロジェクトの実行手順を示します。

1. TwinCAT3 において、[Login]ボタンをクリックして PLC にログインします。
2. TwinCAT PLC Control ウィンドウが表示されるので、[Yes]を選択します。
3. [Start]ボタンをクリックし、TwinCAT プロジェクトを実行します。

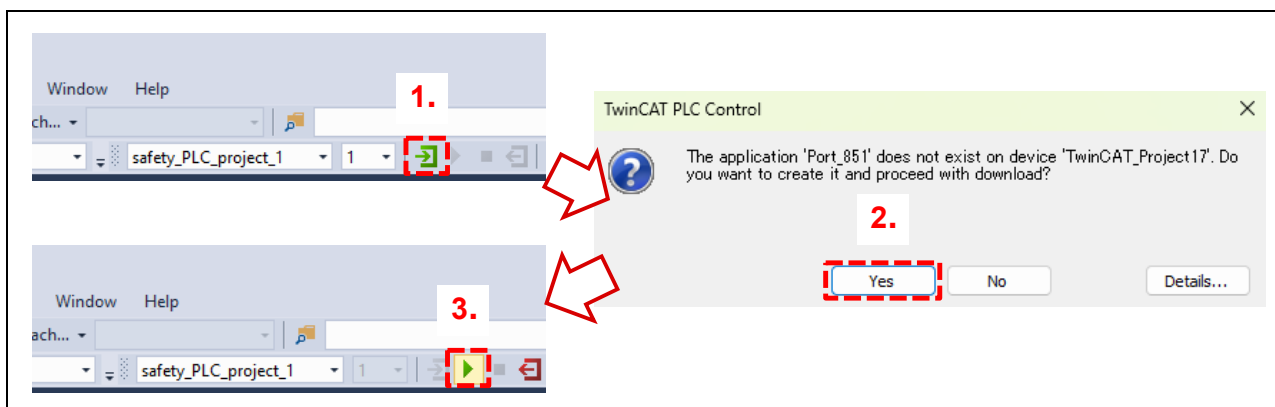


図 6-26. TwinCAT プロジェクトの実行手順

## 6.5 動作確認方法

本章では、Safety Motor Control Reference Kit の動作確認手順を説明します。

### 6.5.1 Safety Drive 機能の選択

#### (1) STO 機能を選択する場合

SW9 と SW11 をすべて OFF の状態で Safety Motor Control Reference Kit を起動してください。

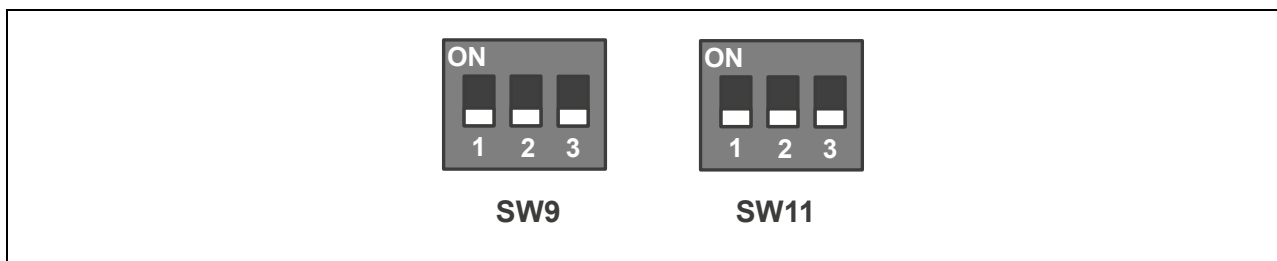


図 6-27. STO 機能の SW 設定

#### (2) SS1-t 機能を選択する場合

SW9-1 と SW11-1 を ON の状態で Safety Motor Control Reference Kit を起動してください。

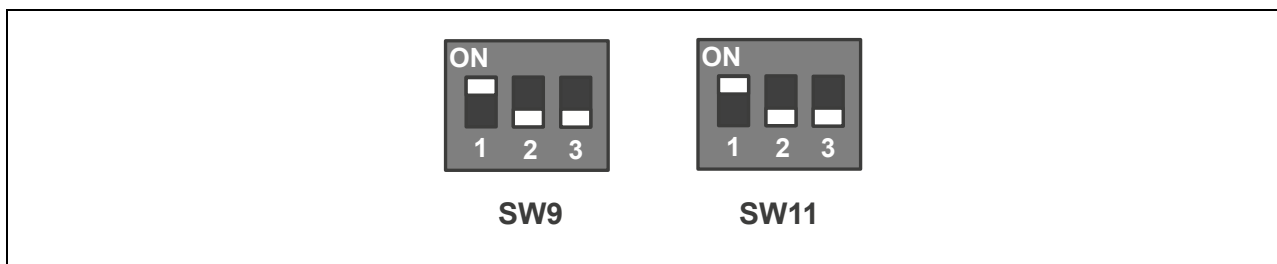


図 6-28. SS1-t 機能の SW 設定

## 6.5.2 CiA402 動作モードの選択

- (1) TwinCAT3 のシステムマネージャツリーで、[Box X (Renesas EtherCAT RZ/T2 SafetyMotorSolution)] をダブルクリック、プロパティ画面を開きます。
- (2) [Slots]タブをクリックします。
- (3) [Slots]タブの左画面の[Motor1]行をクリックします。
- (4) [Slots]タブの右画面から CiA402 動作モードをクリックして選択します。
- (5) [<]ボタンを押して CiA402 動作モードを変更します。
- (6) [Slots]タブの左画面の[Motor1]行を見て、選択した CiA402 動作モードになっていることを確認します。
- (7) TwinCAT3 の[Activate Configuration]アイコンをクリックし、Run mode の再起動を行ってください。

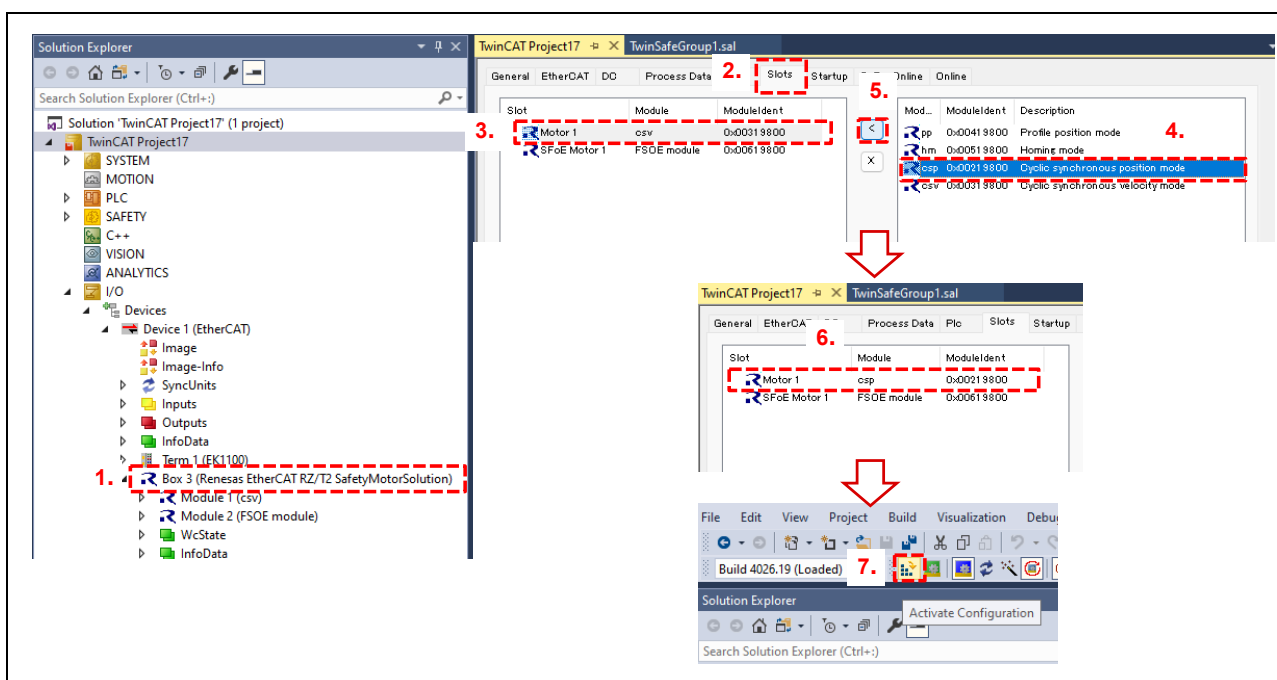


図 6-29. CiA402 動作モードの変更

## 6.5.3 FSoE 通信の確立・モータ停止要求の解除

(1) 6.4 章を実施し、TwinCAT PLC プログラムをスタートします。

(2) 通信状態を確認します。

1. TwinCAT3 のシステムマネージャツリーで、[SAFETY]→[Safety\_Drive]→[Safety\_Drive Project]→[TwinSafeGroup1]を順に選択します。
2. [TwinSafeGroup1.sal]を右クリックします。
3. [Open]を選択すると、TwinSAFE グラフィカルエディタが開きます。
4. TwinCAT3 において、[TWINSAFE]メニューをクリックします。
5. [Show Online Data]をクリックします。
6. TwinSafeGroup が Error であることを確認します。

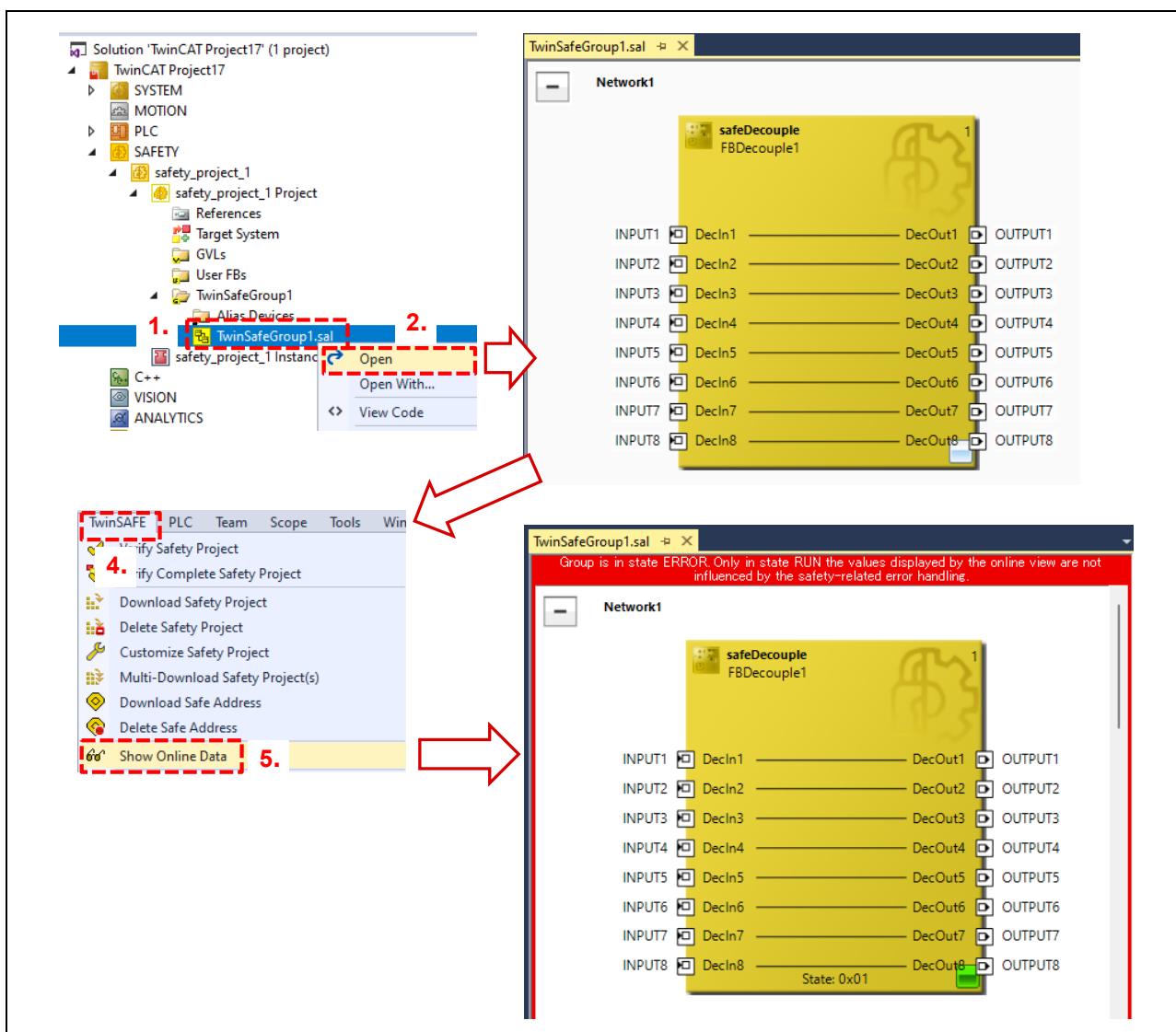


図 6-30. FSoE 通信状態の確認

7. 通信状況はボード上の LED13~LED31 でも確認できます。FSoE の通信状況が Error のときは下記のように動作します。

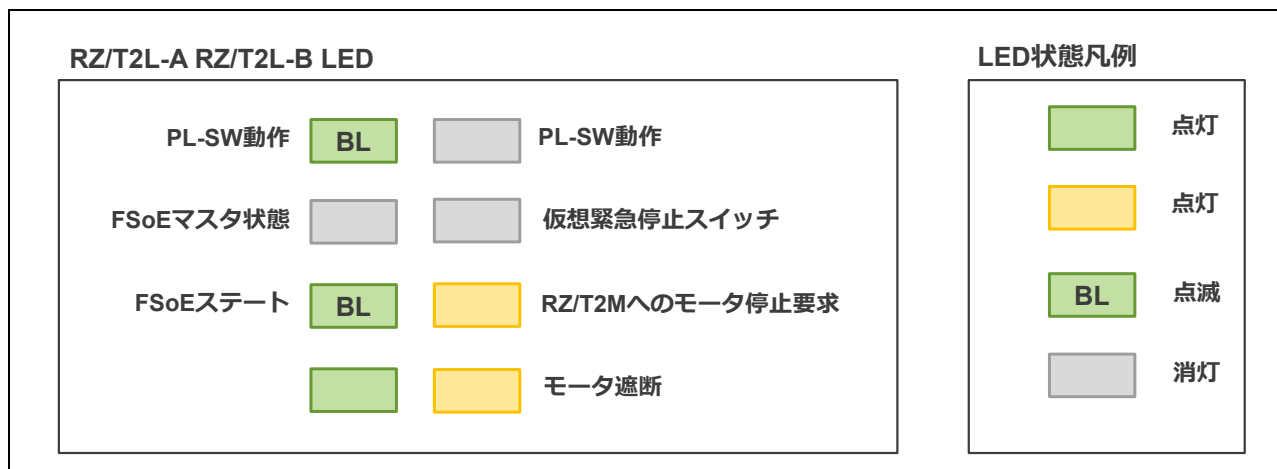


図 6-31. RZ/T2L-A, RZ/T2L-B LED(Error 状態)

(3) 次に、入力用 ErrAckIn 変数に値を書き込みます。

1. TwinCAT3 のシステムマネージャツリーで、[MAIN.bErrAckIn] をクリックし、プロパティ画面を開きます。
2. [Online] タブをクリックします。
3. [Write] をクリックし、"Set Value Dialog" ウィンドウを開きます。
4. [Dec] に "1" を書き込みます。
5. [OK] をクリックします。
6. 再度、"Set Value Dialog" ウィンドウを開き、[Dec] に "0" を書き込み、[OK] をクリックします。

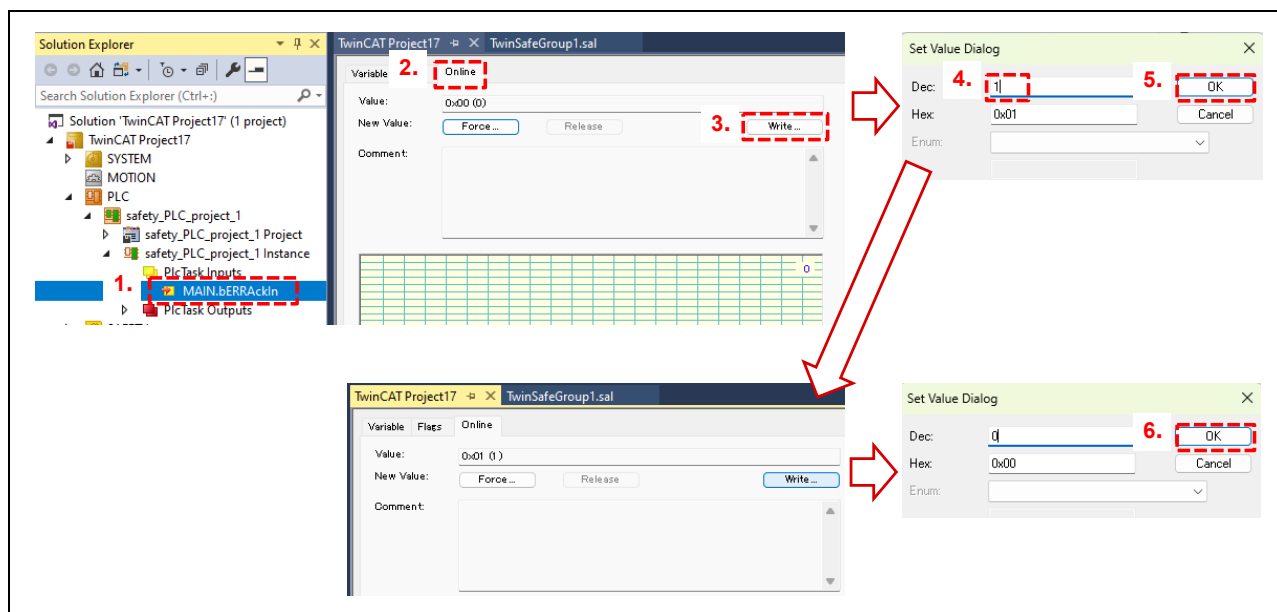


図 6-32. 入力用 ErrAckIn 変数への書き込み

7. TwinCAT 上で通信状態を確認します。問題が無ければ TwinSafeGroup が RUN 状態になります。
8. このとき、ポート上の LED は下記のようにになります。

### TwinSafeGroup状態

#### RZ/T2L-A RZ/T2L-B LED

PL-SW動作	BL	PL-SW動作
FSoEマスタ状態		仮想緊急停止スイッチ
FSoEステート	BL	RZ/T2Mへのモータ停止要求
		モータ遮断

#### LED状態凡例

	点灯
	点灯
BL	点滅
	消灯

図 6-33. TwinSAFEGroup および RZ/T2L-A, RZ/T2L-B LED (RUN 状態)

## (4) 安全信号の入出力を行います。

1. Safety Motor Control Reference Kit 上の SW8 と SW10 を押します。
2. ボードから安全信号が出力された場合、ボード上の LED44 ~ LED51 が全て点灯します。
3. このとき、ファンクションブロックの信号ラインが緑表示になります。
4. また、ポート上の LED は下記のようにになります。RZ/T2M へのモータ停止要求およびモータ遮断 LED が消灯します。

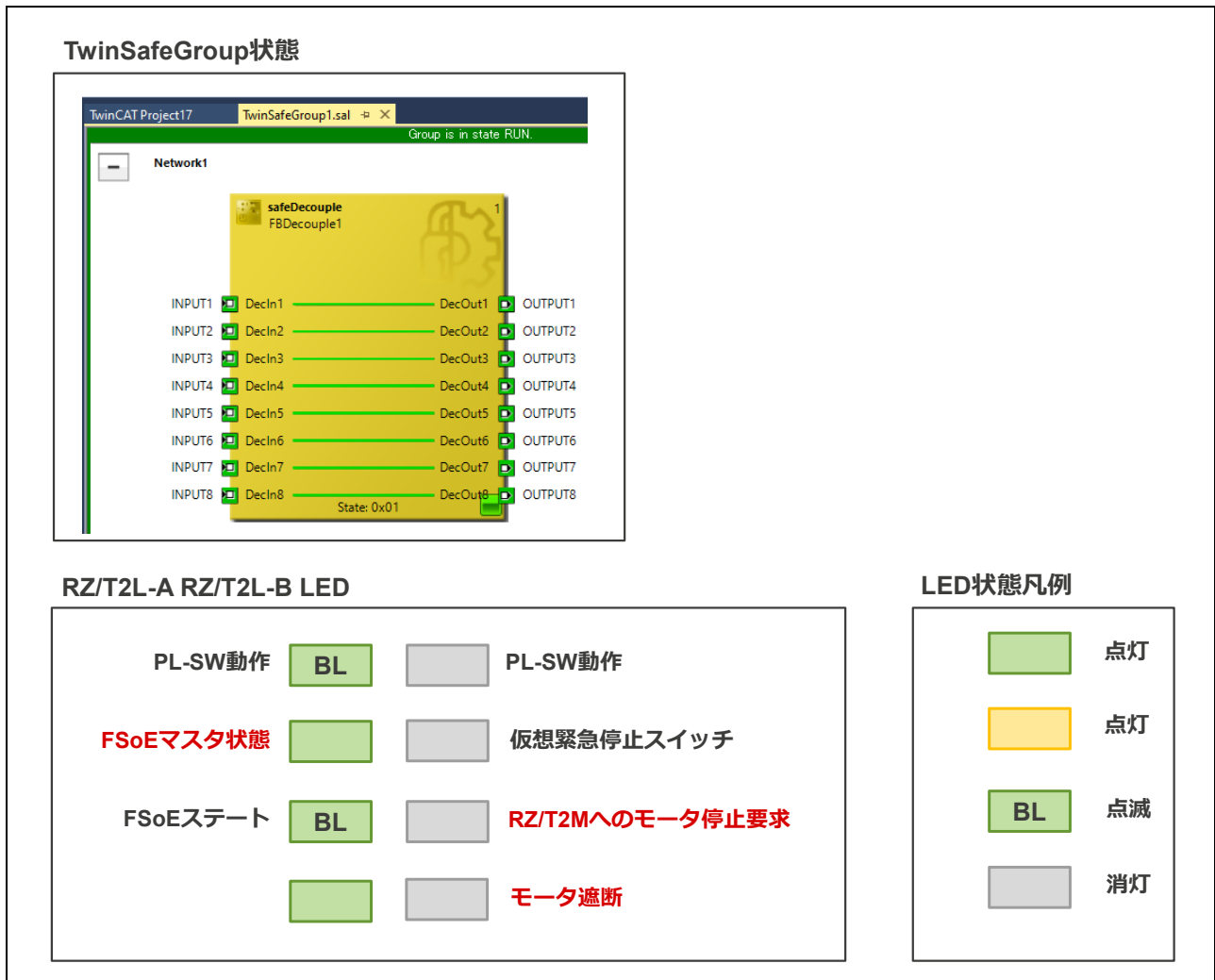


図 6-34. TwinSAFEGroup および RZ/T2L-A, RZ/T2L-B LED (安全信号入出力状態)

## 6.5.4 CiA402 State Machine 制御方法

## 6.5.4.1 Operation enable への遷移方法

- 6.5.3 章の手順を行い、RZ/T2M へのモータ停止要求を解除してください。
- TwinCAT3 のシステムマネージャツリーで、[Box X (Renesa EtherCAT RZ/T2 SafetyMotorSolution)]→[Module 1]→[Inputs]→[Statusword]をダブルクリックします。
- [Flags]タブを開きます。
- [Statusword]の Display mode を Hex に変更します。
- [Online]タブを開きます。
- [Statusword]の 3bit 目が 0 であることを確認します。1 である場合は Fault 状態ですので、Fault から復帰する処理を行う必要があります。6.5.4.2 章の処理を行い、Fault から復帰してください。

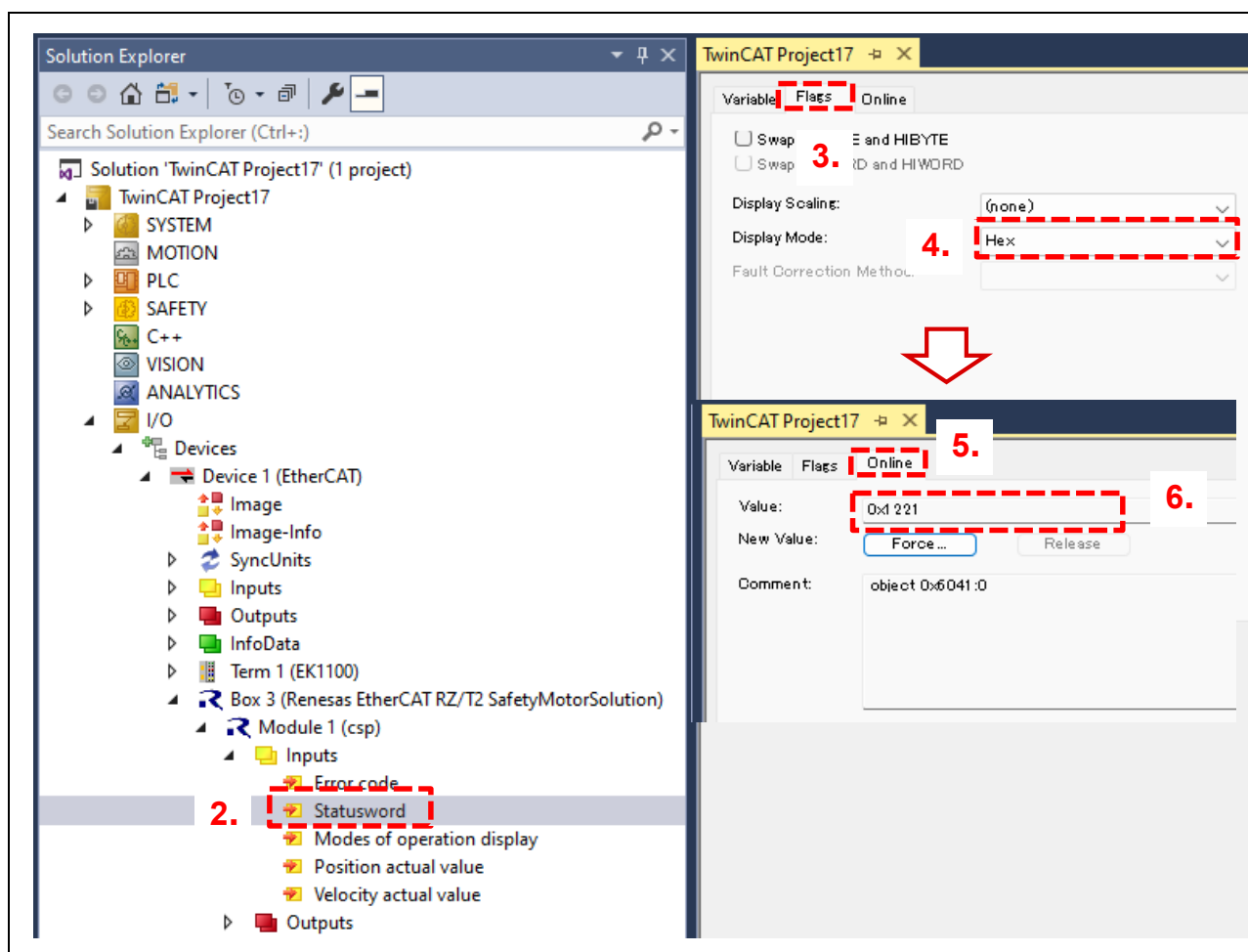


図 6-35. Operation enable への遷移方法 1

7. TwinCAT3 のシステムマネージャツリーで、[Box X (Renesa EtherCAT RZ/T2 SafetyMotorSolution)]→[Module 1]→[Outputs]→[Controlword]をクリックします。
8. [Online]タブを開きます。
9. [Write]ボタンを押します。
10. 15 を書き込みます。

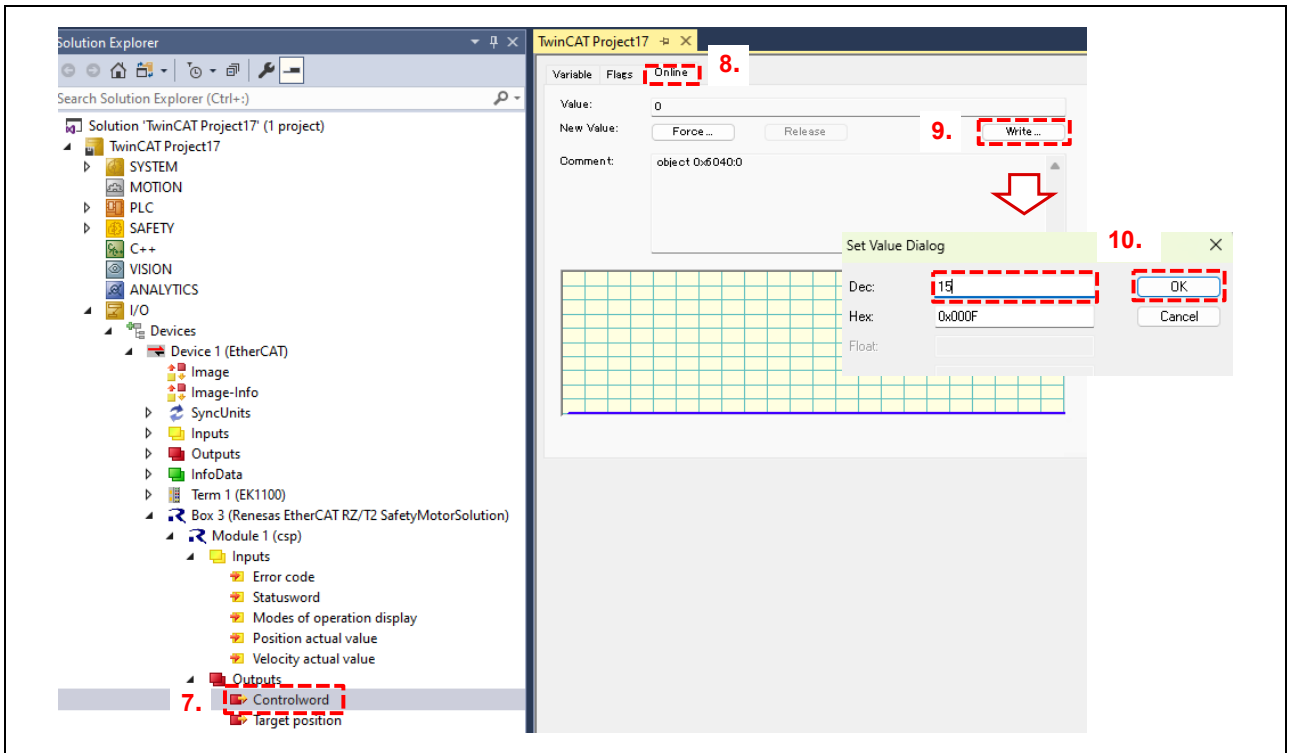


図 6-36. Operation enable への遷移方法 2

11. [Statusword]の値を確認します。2bit目が1になっていれば、Operation enable への遷移が成功しています。
12. ボード上の RZ/T2M 用 LED (LED8 ~ LED15) を見ることも、Operation enable への遷移を確認することができます。下記に確認方法を示します。

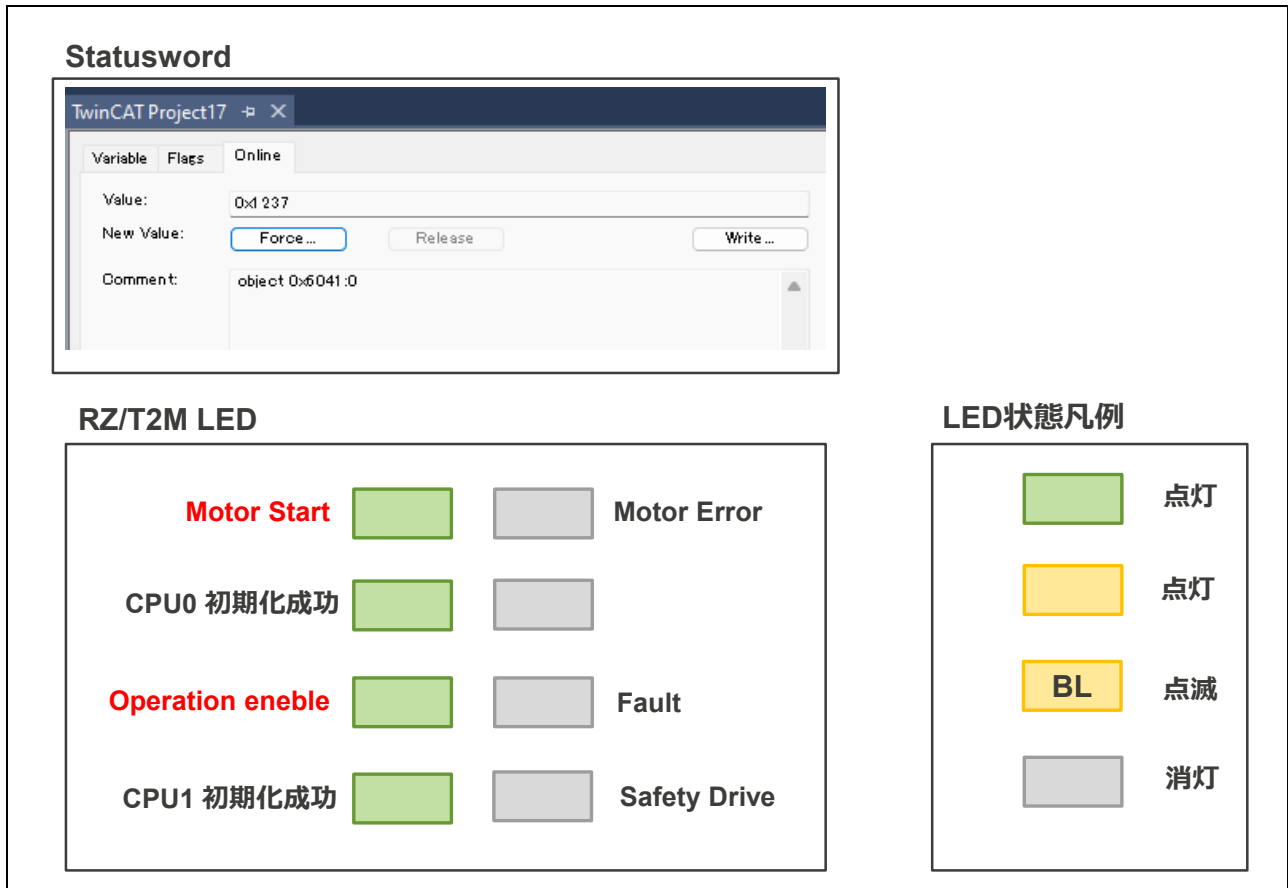


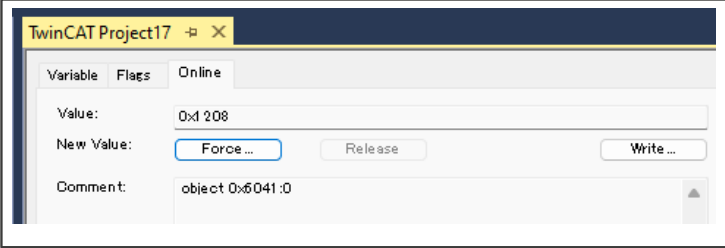
図 6-37. Operation enable 確認方法

注：RZ/T2M へのモータ停止要求を解除されていない場合、Operation enable への遷移は失敗し、Fault へ遷移します。









## 6.5.4.2 Fault からの復帰方法

1. [Statusword]の 3bit 目が 1 である場合は Fault 状態です。また、ボード上の LED14 を見ることで、Fault 状態を判断することができます。





### Statusword



### RZ/T2M LED

Motor Start			Motor Error (※)
CPU0 初期化成功			
Operation enable			Fault
CPU1 初期化成功			Safety Drive

### LED状態凡例

	点灯
	点灯
	点滅
	消灯

(※) Motor Error LED(LED12)は、CPU0でモータ制御システムのエラーが検出された場合に点灯します。

図 6-38. Fault 状態の判別方法

2. TwinCAT3 のシステムマネージャツリーで、[Box X (Renesa EtherCAT RZ/T2 SafetyMotorSolution)]→[Module 1]→[Outputs]→[Controlword]をクリックします。
3. [Online]タブを開きます。
4. [Write...]ボタンを押します。
5. 128 を書き込みます。
6. [Statusword]の 3bit 目が 0 であることを確認します。これで Fault からの復帰は完了です。

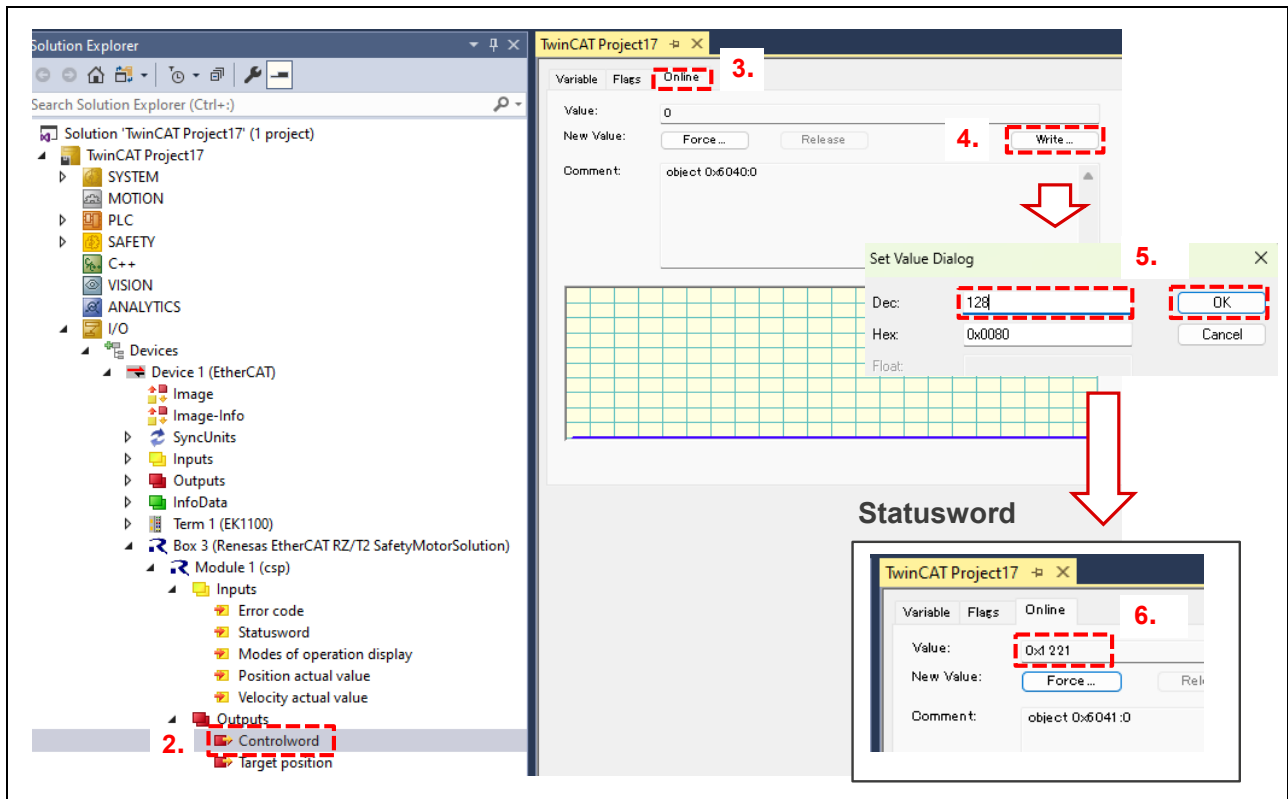


図 6-39. Fault からの復帰方法

## 6.5.5 モータ動作方法

## 6.5.5.1 csp mode

1. 6.5.2 章の手順を行い、モータの CiA402 動作モードを csp mode にします。
2. 6.5.3 章の手順を行い、RZ/T2M へのモータ停止要求を解除します。
3. 6.5.4 章の手順を行い、CiA402 ステートマシンを Operation enable にします。
4. TwinCAT3 のシステムマネージャツリーで、[Target position]をクリックします。
5. [Online]タブを開き、[Write]ボタンを押します。
6. 36000 を書き込み、3600.0 deg に移動するようモータへ位置指令を出します。
7. モータが回転し、3600.0 deg に移動します。
8. TwinCAT3 のシステムマネージャツリーで、[Position actual value]をクリックします。
9. [Online]タブを開き、値が 36000 になっていることを確認します。

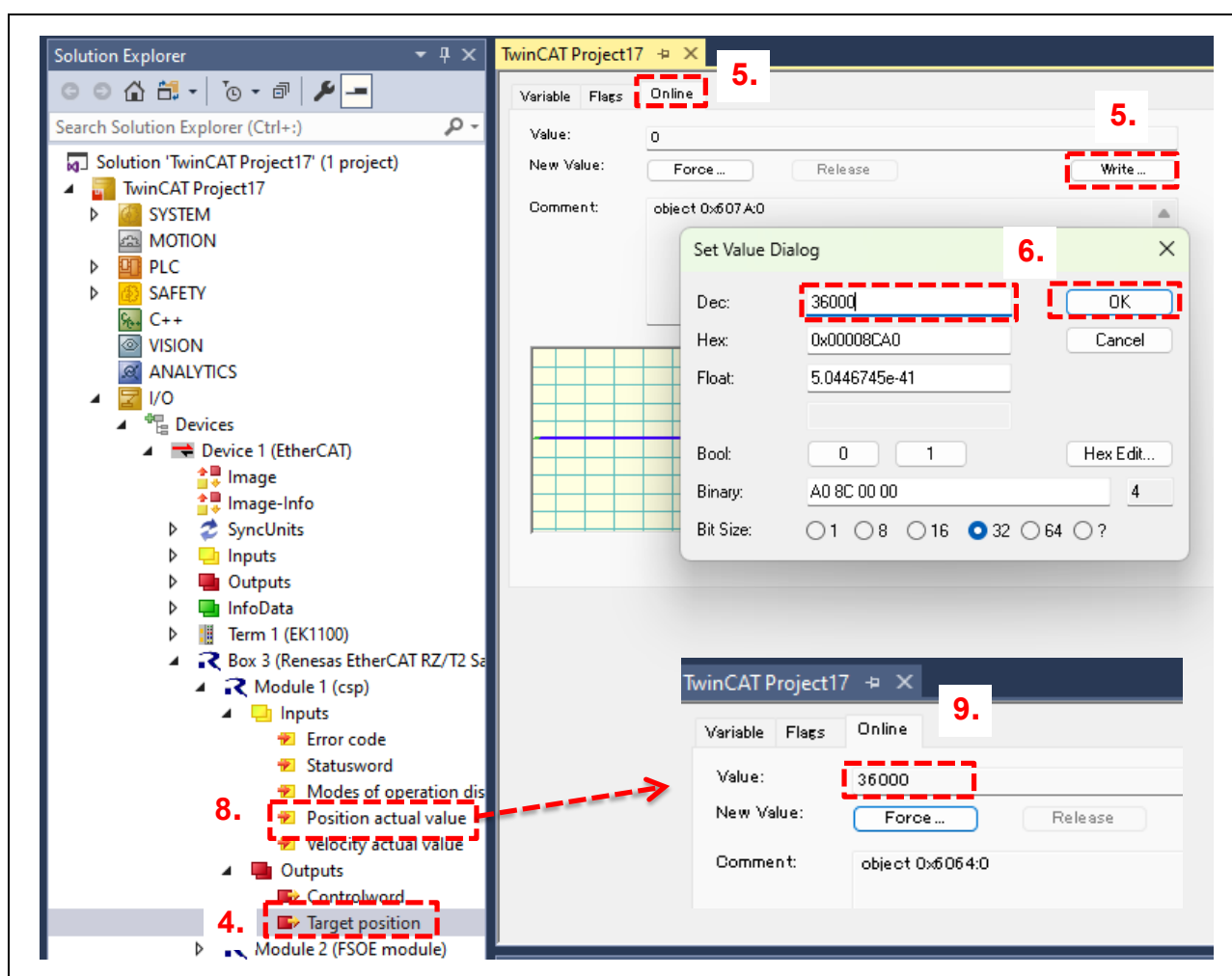


図 6-40. csp モードの動作確認

## 6.5.5.2 pp mode

1. 6.5.2 章の手順を行い、モータの CiA402 動作モードを pp mode にします。
2. 6.5.3 章の手順を行い、RZ/T2M へのモータ停止要求を解除します。
3. 6.5.4 章の手順を行い、CiA402 ステートマシンを Operation enable にします。
4. TwinCAT3 のシステムマネージャツリーで、[Profile velocity]をクリックします。
5. [Online]タブを開き、[Write]ボタンを押します。
6. 30000 を書き込み、速度プロファイルを 3000.0 rpm に設定します。
7. TwinCAT3 のシステムマネージャツリーで、[Profile acceleration]をクリックします。
8. [Online]タブを開き、[Write]ボタンを押します。
9. 150000 を書き込み、加速度プロファイルを 15000.0 rpm/s に設定します。

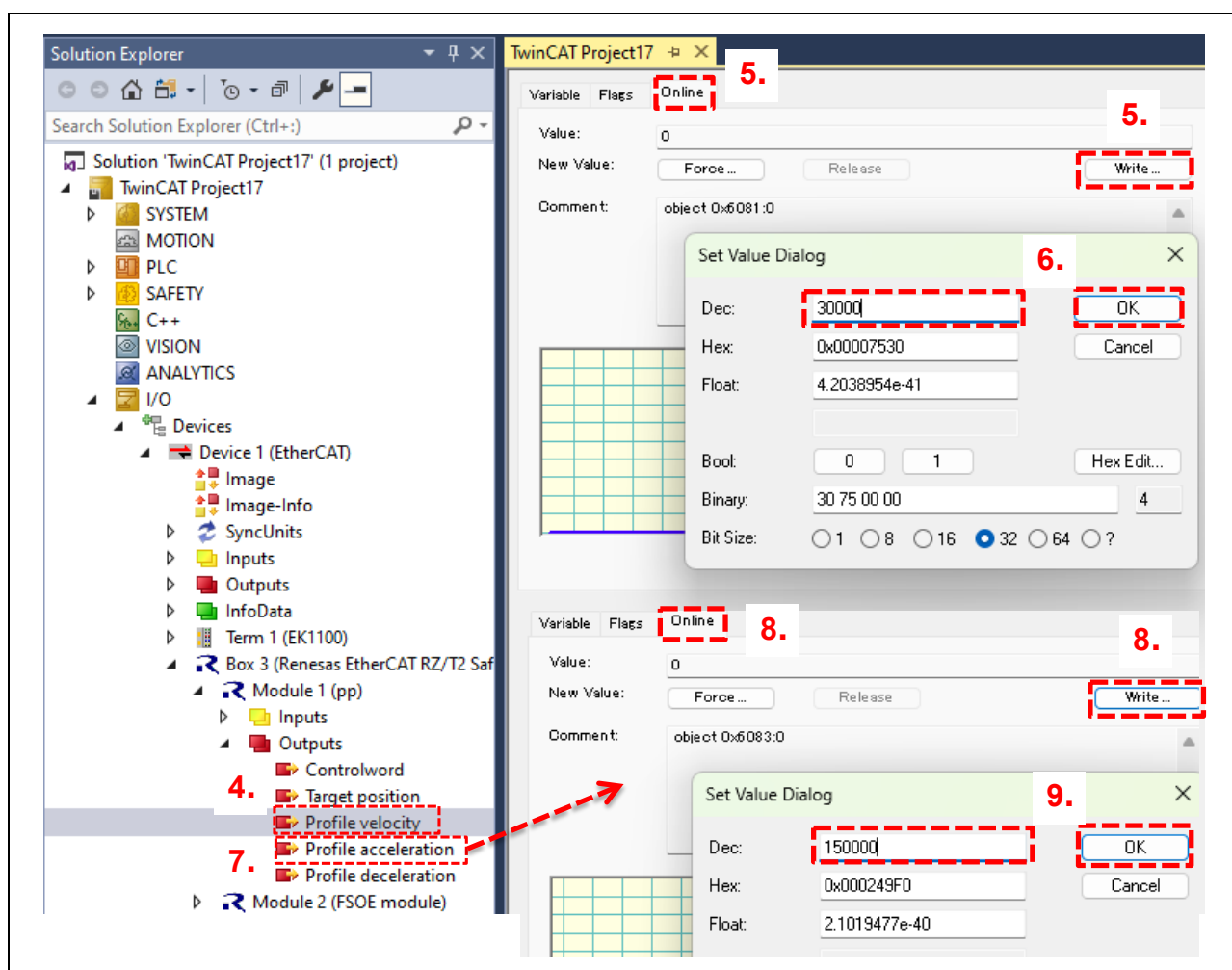


図 6-41. pp モードの動作確認 1

10. TwinCAT3 のシステムマネージャツリーで、[Target position]をクリックします。
11. [Online]タブを開き、[Write]ボタンを押します。
12. 36000 を書き込み、3600.0 deg に移動するようモータへ位置指令を出します。
13. 設定したプロファイルでモータが回転し、3600.0 deg に移動します。
14. TwinCAT3 のシステムマネージャツリーで、[Position actual value]をクリックします。
15. [Online]タブを開き、値が 36000 になっていることを確認します。

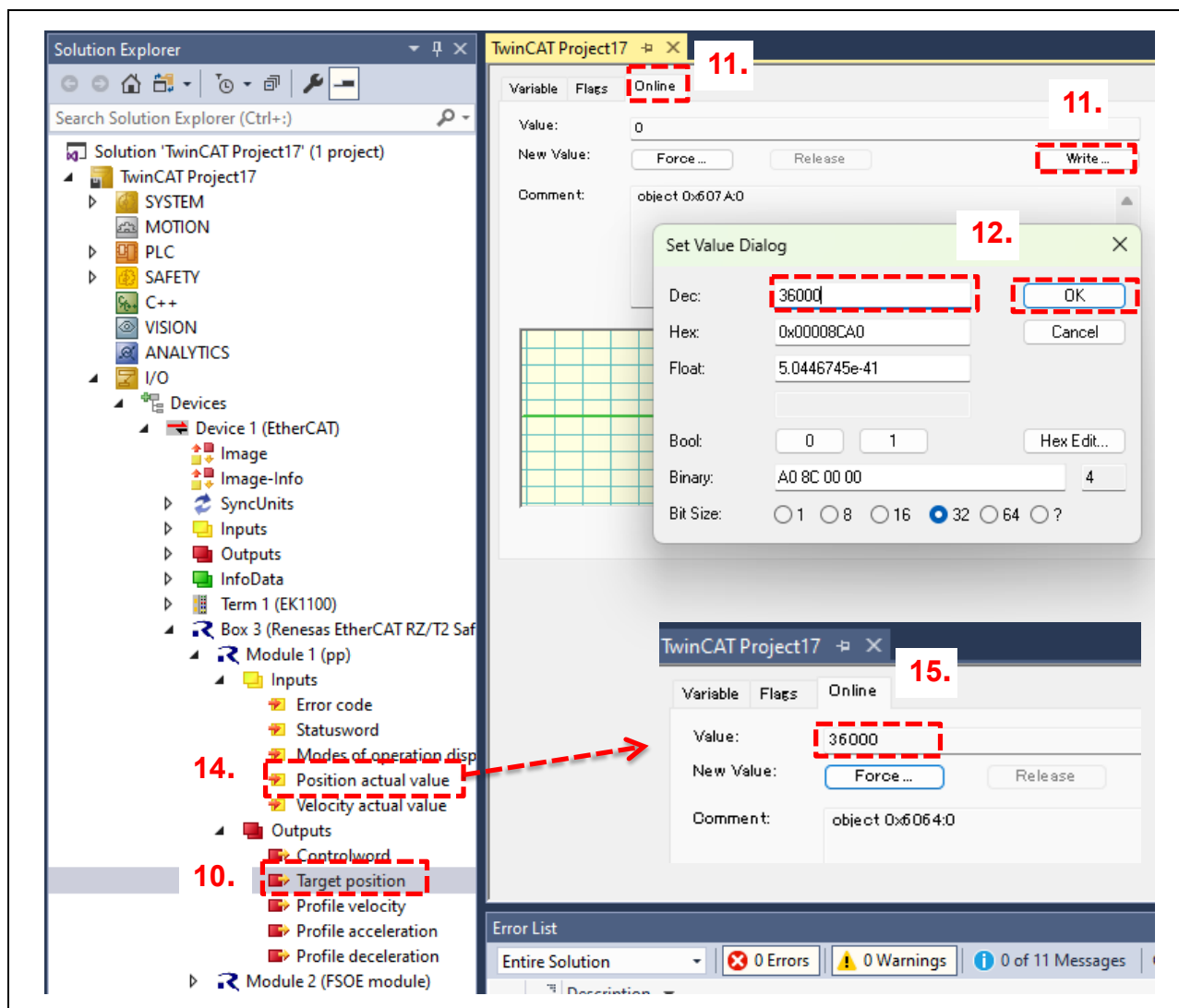


図 6-42. pp モードの動作確認 2

## 6.5.5.3 csv mode

1. 6.5.2 章の手順を行い、モータの CiA402 動作モードを csv mode にします。
2. 6.5.3 章の手順を行い、RZ/T2M へのモータ停止要求を解除します。
3. 6.5.4 章の手順を行い、CiA402 ステータスマシンを Operation enable にします。
4. TwinCAT3 のシステムマネージャツリーで、[Target velocity]をクリックします。
5. [Online]タブを開き、[Write]ボタンを押します。
6. 20000 を書き込み、2000rpm で回転するようモータへ速度指令を出します。
7. 2000rpm でモータが回転します。
8. TwinCAT3 のシステムマネージャツリーで、[Velocity actual value]をクリックします。
9. [Online]タブを開き、値が 20000 付近になっていることを確認します。

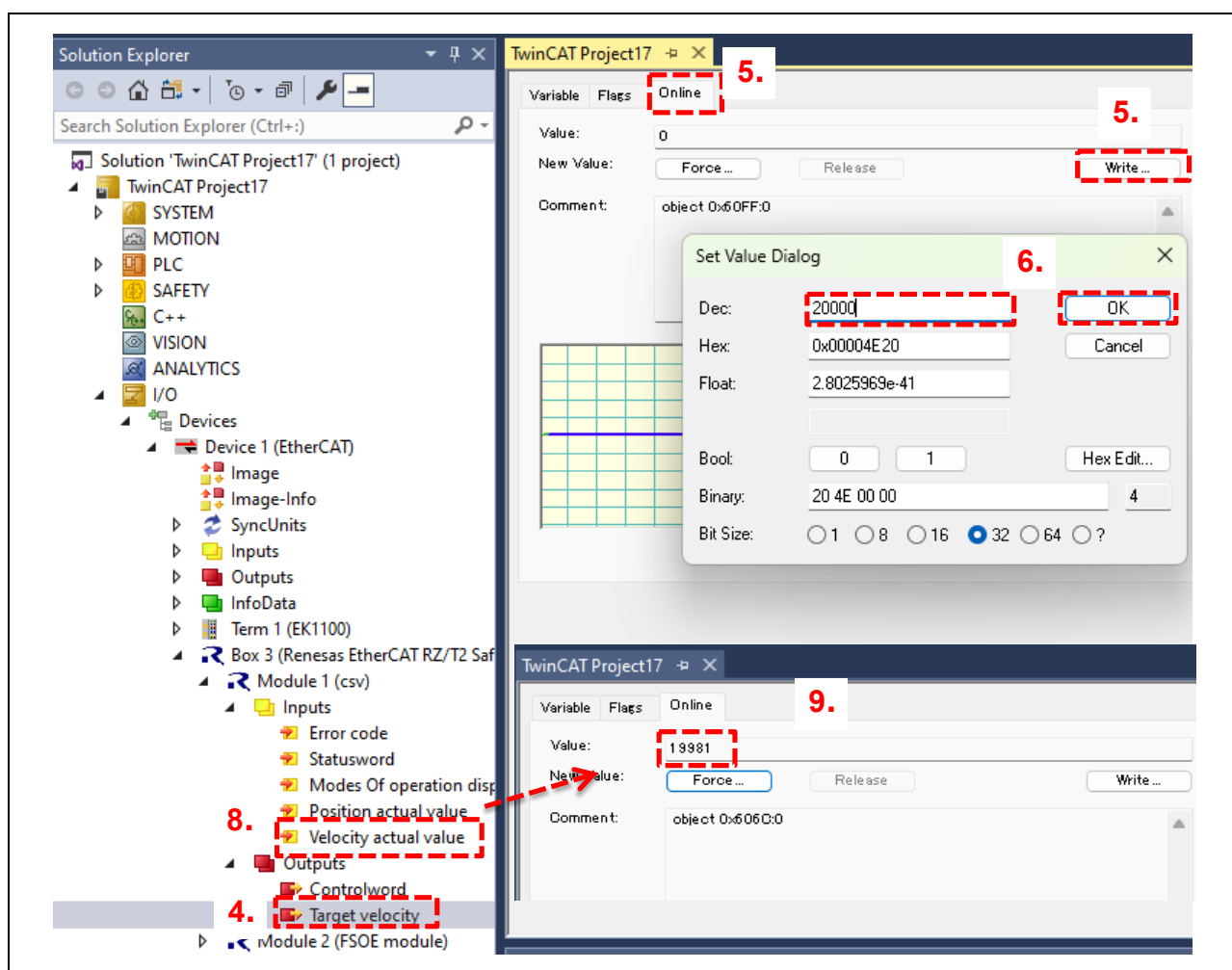


図 6-43. csv モードの動作確認

## 6.5.5.4 hm mode

1. 6.5.2 章の手順を行い、モータの CiA402 動作モードを hm mode にします。
2. 6.5.3 章の手順を行い、RZ/T2M へのモータ停止要求を解除します。
3. 6.5.4 章の手順を行い、CiA402 ステートマシンを Operation enable にします。
4. TwinCAT3 のシステムマネージャツリーで、[Position actual value]をクリックします。
5. [Online]タブを開き、現在の値を確認します。
6. TwinCAT3 のシステムマネージャツリーで、[Home offset]をクリックします。
7. 0 になっていることを確認します。

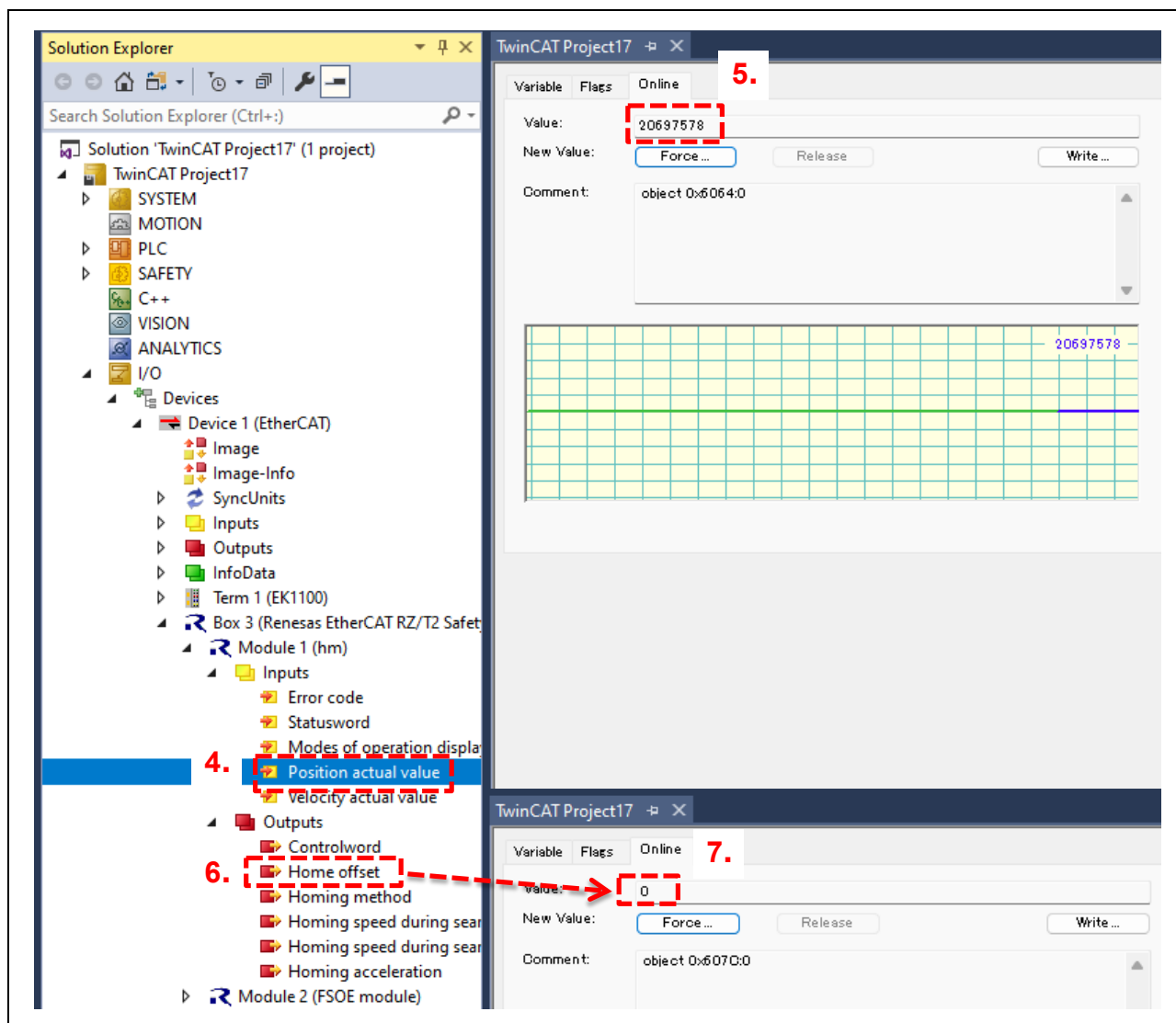


図 6-44. hm モードの動作確認 1

8. TwinCAT3 のシステムマネージャツリーで、[Home method]をクリックします。
9. [Online]タブを開き、[Write]ボタンを押します。
10. 37 を書き込み、Home Method 37 を指定します。

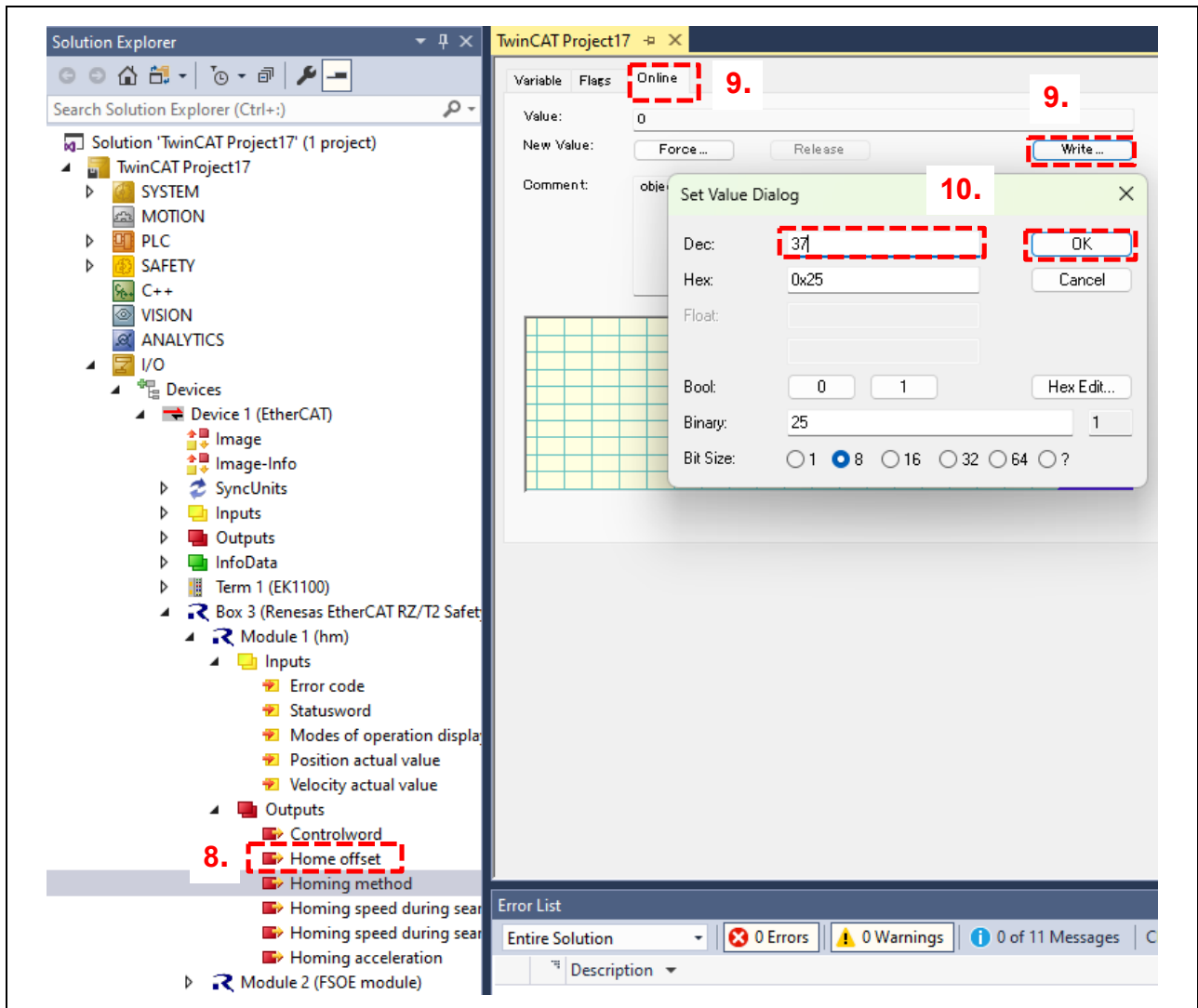


図 6-45. hm モードの動作確認 2

11. TwinCAT3 のシステムマネージャツリーで、[Controlword]をクリックします。
12. [Online]タブを開き、[Write]ボタンを押します。
13. 31 を書き込み、Home Method 37 を実行します。
14. TwinCAT3 のシステムマネージャツリーで、[Position actual value]をクリックします。
15. [Online]タブを開き、値が 0 に変更されていることを確認します。

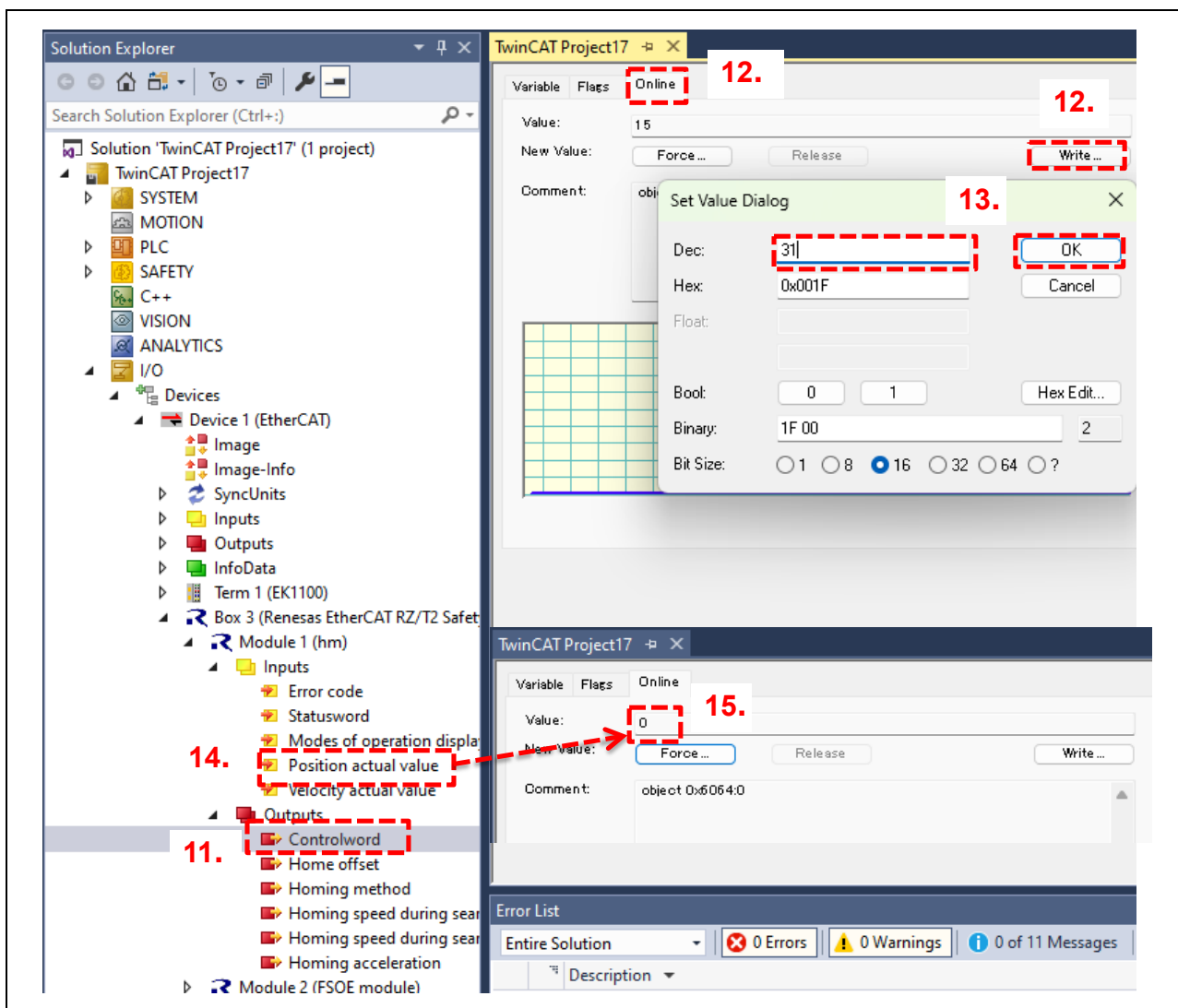


図 6-46. hm モードの動作確認 3

## 6.5.6 Safety Drive 機能の確認

## 6.5.6.1 STO

1. 6.5.1 章に従い、Safety Drive 機能を STO に設定します。
2. 6.5.5.3 章に従い、csv mode でモータを回転させます。
3. ボード上の SW9-2 を ON にして、STO を実行します。
4. STO が実行されると、直ちにモータが止まります。
5. SW9-2 を OFF に戻した後、6.5.3 章と 6.5.4.2 章の手順に従うことで復帰することができます。

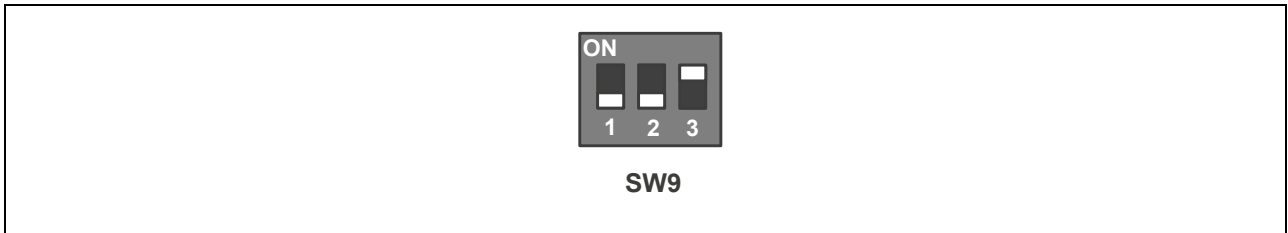


図 6-47. STO 実行時の SW 設定

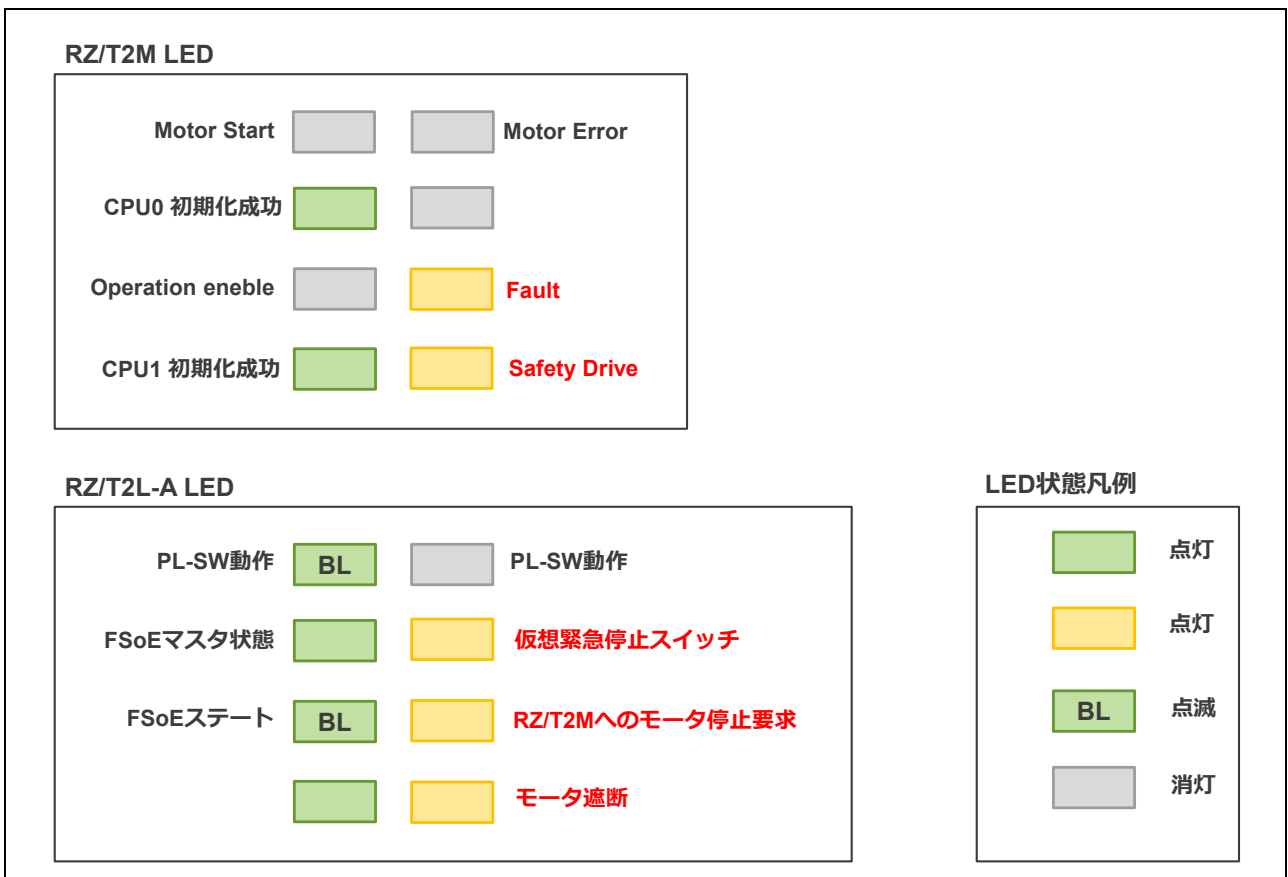


図 6-48. STO 実行後の LED 状態

## 6.5.6.2 SS1-t

1. 6.5.1 章に従い、Safety Drive 機能を SS1-t に設定します。
2. 6.5.5.3 章に従い、csv mode でモータを回転させます。
3. ボード上の SW9-2 を ON にして、SS1-t を実行します。
4. SS1-t が実行されるとモータの回転速度が減速していき、3 秒後に 0 rpm になります。
5. SW9-2 を OFF に戻した後、6.5.3 章と 6.5.4.2 章の手順に従うことで復帰することができます。

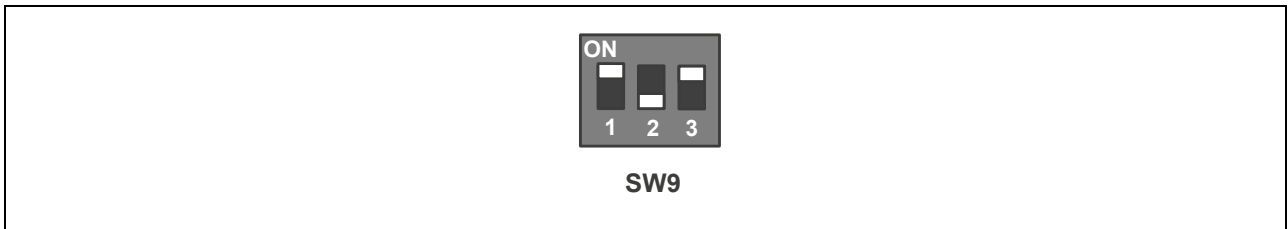


図 6-49. SS1-t 実行時の SW 設定

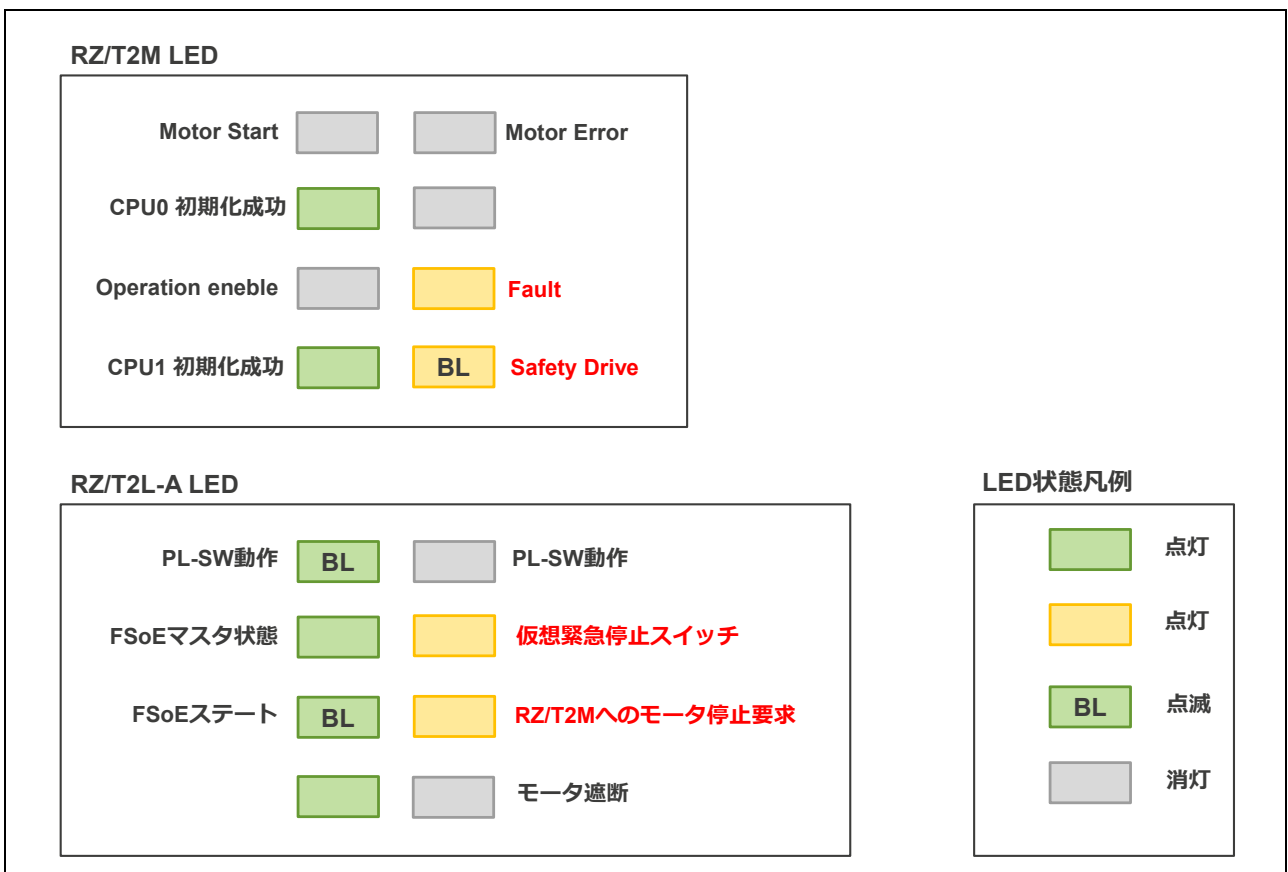


図 6-50. SS1-t 実行中の LED 状態

## 6.5.6.3 RZ/T2L 異常による安全制御

1. 6.5.5.3 章に従い、csv mode でモータを回転させます。
2. ボード上の SW13-16 のいずれかを操作し、RZ/T2L システムに異常を発生させます。
3. モータが直ちに停止されます。
4. ボード上の SW13-16 設定を元に戻し、SW2 を押してシステムを再起動します。その後、6.5.3 章と 6.5.4.2 章の手順に従うことで復帰することができます。



図 6-51. SW13-16

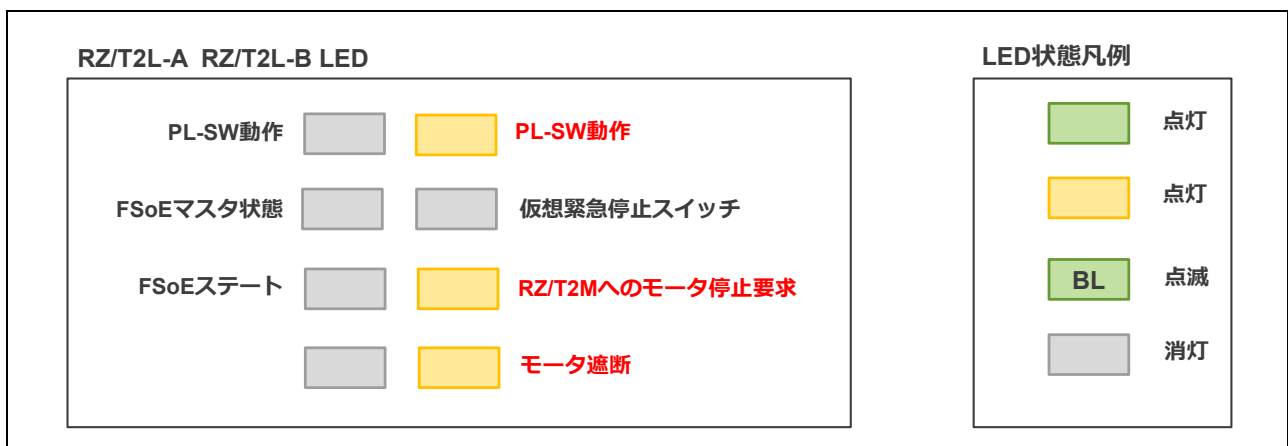


図 6-52. RZ/T2L 異常時の LED 状態

## 6.5.6.4 FSoE マスタ異常による安全制御

1. 6.5.5.3 章に従い、csv mode でモータを回転させます。
2. Safety Motor Control Reference Kit の In Port に接続されている LAN ケーブルを外します。
5. モータが直ちに停止されます。
6. LAN ケーブルを元に戻し、6.5.3 章と 6.5.4.2 章の手順に従うことで復帰することができます。

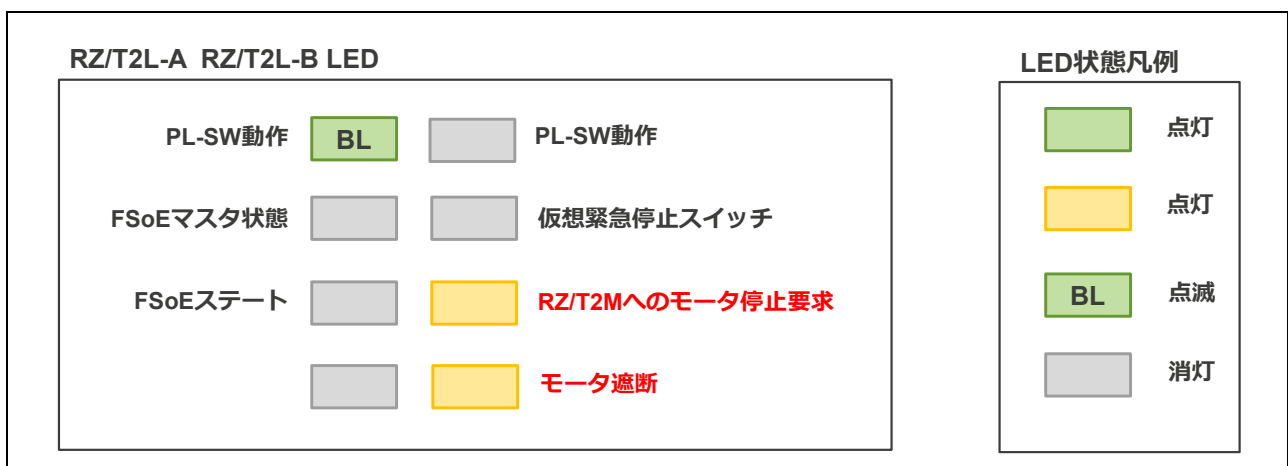


図 6-53. FSoE マスタ異常時の LED 状態

## 7. ソフトウェア仕様

### 7.1 ソフトウェア構成図

図 7-1 にセーフティモータ制御キットのソフトウェア構成図を示します。

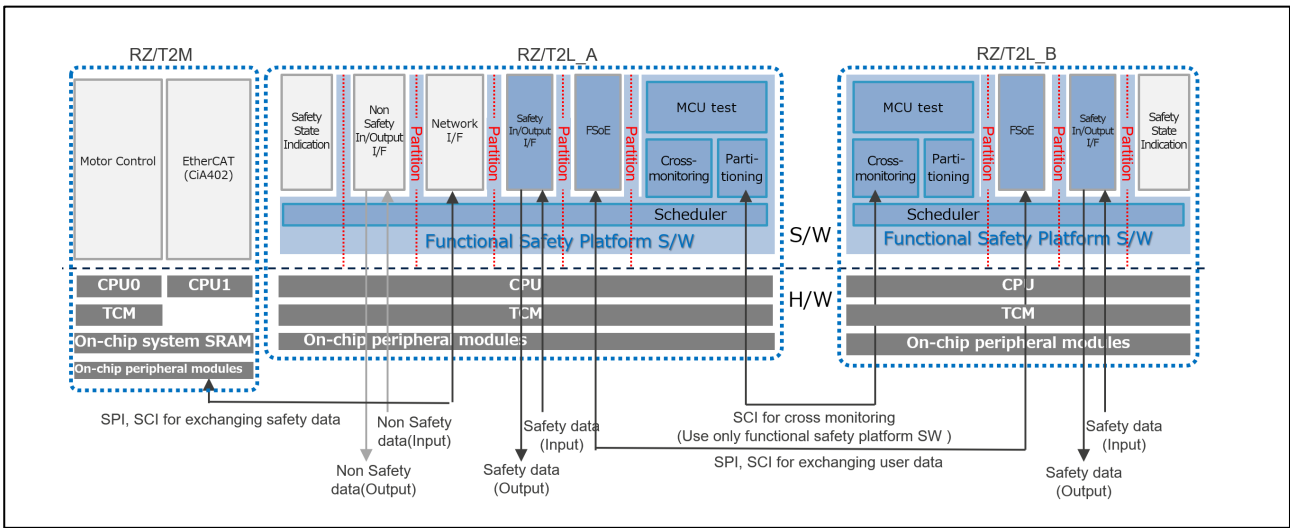


図 7-1. セーフティモータ制御キットのソフトウェア構成図

## 7.2 Motor Control(RZ/T2M CPU0)

### 7.2.1 ソフトウェア全体構成

本サンプルプログラムの RZ/T2M CPU0 ソフトウェアは、『RZ/T2M 永久磁石同期モータのエンコーダベクトル制御 - シリアルエンコーダ(R01AN8003)』をベースに作成しています。

ベースプロジェクトの詳細は R01AN8003 のアプリケーションノートを参照してください。

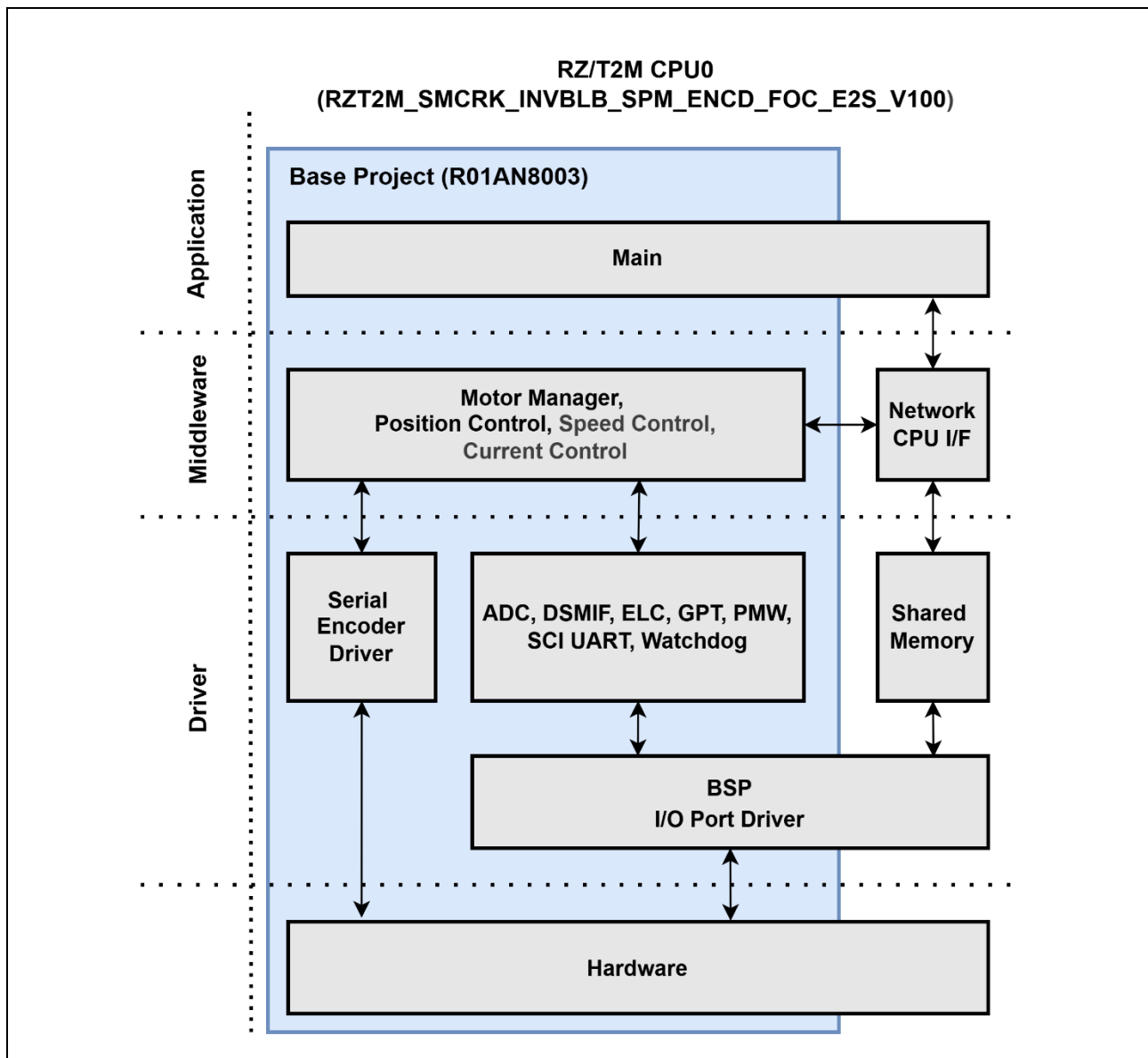


図 7-2. RZ/T2M CPU0 ソフトウェア全体構成

## 7.2.2 ベースプロジェクトからの変更点

ここでは、ベースプロジェクトからの変更点について記述します。

表 7-1. ベースプロジェクトからの変更点

変更のあったフォルダ・ファイル		変更内容	
rzt		Configuration.xml の変更によって、 変更が加えられたファイルや追加 されたファイル・フォルダがある	
rzt_gen			
rzt_cfg			
src	network_cpu_if	r_network_cpu_if.c r_network_cpu_if.h	Network CPU I/F モジュールの追 加 (詳細は 7.2.3 章を参照)
	serial_encoder_driver¥mcu¥rzt2m¥lib¥ecl	r_ecl_rzt2_if.h r_ecl_rzt2_gcc.a	EC lib のバージョンを 2.00 から 3.00 へアップデート
	hal_entry.c		hal_entry 関数内の処理を変更 (詳細は図 7-3 を参照)
configuration.xml		Shared Memory Driver の追加 (設定は表 7-2, 表 7-3, 表 7-4 を 参照) I/O Port P01_6 (LED9) の有効化 CPU1 クロックを 800MHz に設定	
.cproject		“src/network_cpu_if” インクルード パスの追加 RZ/T2M CPU1 オブジェクトファ イルの追加	

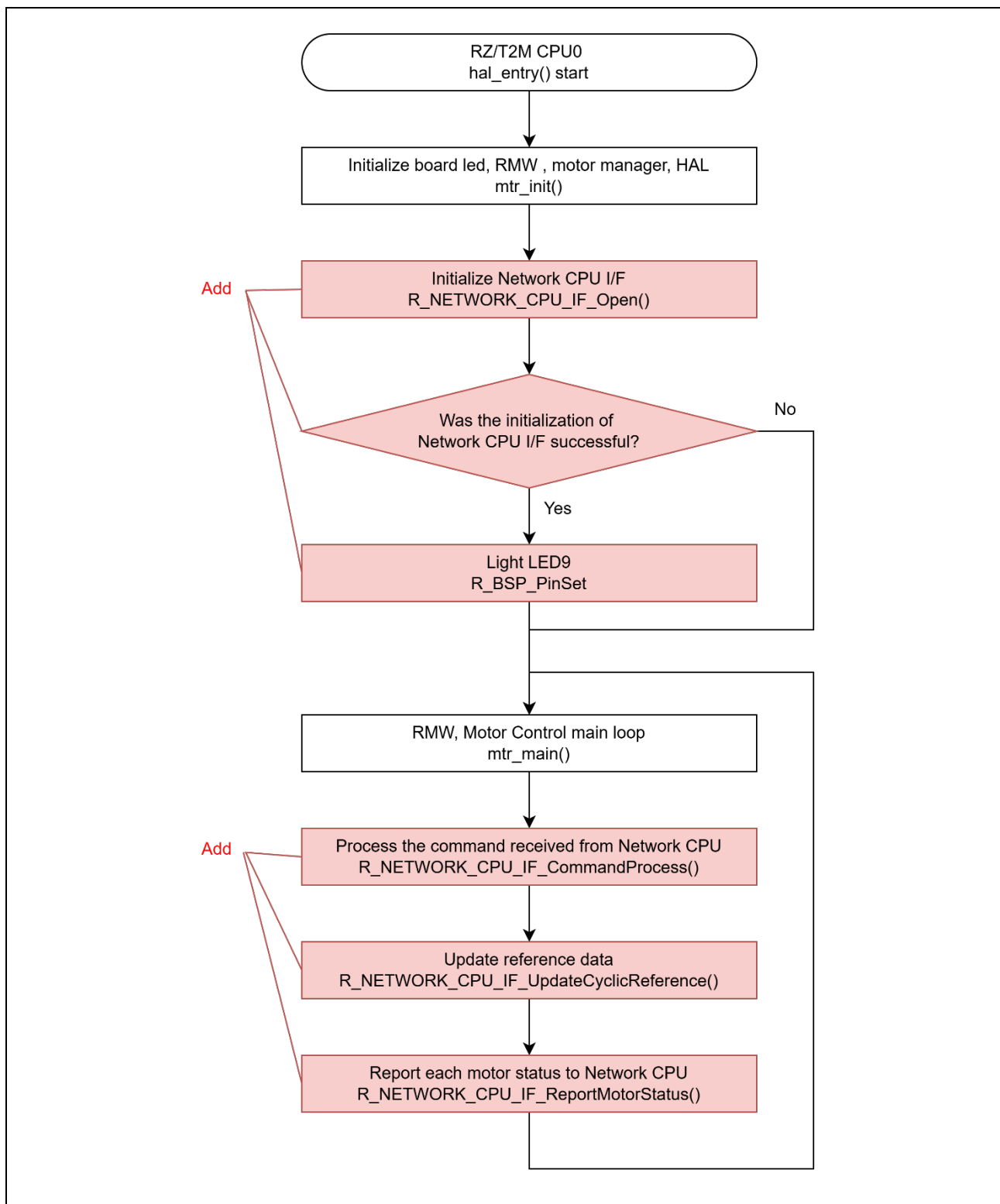


図 7-3. hal\_entry 関数フローチャート

表 7-2. 追加した Shared Memory Driver 設定 (r\_shared\_memory)

プロパティ	設定値
Common - Parameter Checking	Default (BSP)
Module - Name	g_shared_memory0
Module - Shared Memory Address	0x301A0000
Module - Shared Memory Size	1024
Module - Semaphore	0
Module - Callback	shared_memory_callback

表 7-3. 追加した Shared Memory Driver 設定 (r\_icu\_inter\_cpu\_irq, ch0)

プロパティ	設定値
Common - Parameter Checking	Default (BSP)
Common - Multiplex Interrupt	Disabled
Module - Name	g_icu_inter_cpu_irq0
Module - Channel	0
Module - Software Interrupt Priority	Priority6
Module - Callback	r_shared_memory_callback

表 7-4. 追加した Shared Memory Driver 設定 (r\_icu\_inter\_cpu\_irq, ch1)

プロパティ	設定値
Common - Parameter Checking	Default (BSP)
Common - Multiplex Interrupt	Disabled
Module - Name	g_icu_inter_cpu_irq1
Module - Channel	1
Module - Software Interrupt Priority	Disabled
Module - Callback	r_shared_memory_callback

### 7.2.3 Network CPU I/F モジュール

Network CPU I/F モジュールは、Shared Memory Driver を使用して Network CPU (RZ/T2M CPU1)とモータ制御コマンド・指令値・モータステータスの送受信及びモータマネージャ API を呼び出すモジュールです。

#### 7.2.3.1 モジュール構成

モジュール構成図を図 7-4 に示します。

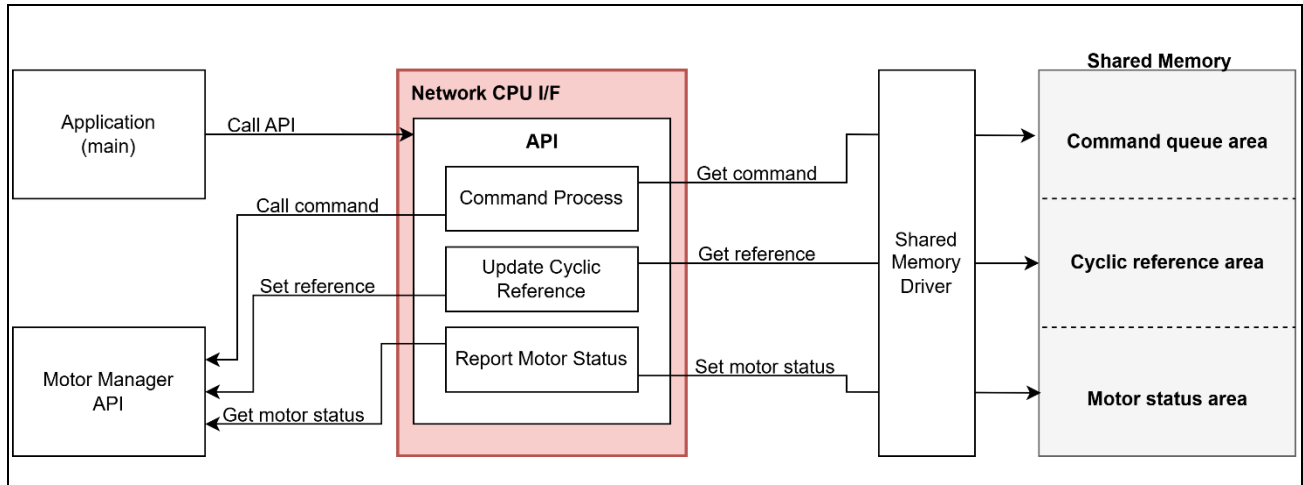


図 7-4. Network CPU I/F モジュール構成図

本モジュールでは主に 3 種類の API を提供します。

#### (1) Command Process

この API は、Shared Memory の Command queue area にモータ制御コマンドが格納されているか確認します。コマンドが存在する場合、該当するモータマネージャ API を呼び出して処理を実行します。

#### (1) Update Cyclic Reference

この API は、Shared Memory の Cyclic reference area に設定されている各種指令値が更新されているかを確認します。指令値が Network CPU によって更新されていた場合、該当するモータマネージャ API を呼び出し、新しい指令値をセットします。

#### (2) Report Motor Status

この API は、モータマネージャ API を呼び出して各種モータステータスを取得し、更新の有無を確認します。ステータスが更新されていた場合、Shared Memory の Motor status area に反映します。

## 7.2.3.2 Shared Memory 割り当て

表 7-5. Shared Memory 割り当て (Command queue area)

スタートアドレス	名称	サイズ [byte]	使用用途
0x301A_0000	Command queue n (n = 0 ... 7)	16 × 8 (=128)	コマンドキュー
0x301A_0080	head	4	キューの先頭
0x301A_0084	tail	4	キューの末尾

表 7-6. Shared Memory 割り当て (Cyclic reference area)

スタートアドレス	名称	サイズ [byte]	使用用途
0x301A_0100	f4_ref_position	4	指令位置
0x301A_0104	f4_ref_speed	4	指令速度
0x301A_0108	f4_ref_torque	4	指令トルク

表 7-7. Shared Memory 割り当て (Motor status area)

スタートアドレス	名称	サイズ [byte]	使用用途
0x301A_0180	f4_pos_deg	4	現在位置
0x301A_0184	f4_speed	4	現在速度
0x301A_0188	u1_status	1	ステータスマシン 0x00 : STOP 0x01 : RUN 0x02 : ERROR
0x301A_0189	reserved1	3	未使用領域
0x301A_018C	u2_error_status	2	エラー状態 0x0000 : エラーなし 0x0001 : HW 過電流エラー 0x0002 : 過電圧エラー 0x0004 : 回転速度エラー 0x0080 : 低電圧エラー 0x0100 : SW の過電流エラー 0x0200 : インバータ過熱エ ラー 0xffff : 不明なエラー
0x301A_018E	reserved2	2	未使用領域
0x301A_0190	u1_ctrl_loop_mode	1	LOOP MODE 0: d 軸電流制御 1: q 軸電流制御 2: 速度制御

			3: 位置制御
0x301A_0191	reserved3	3	未使用領域
0x301A_0194	u1_in_position	1	位置制御完了状態 0: 位置制御未完了状態 1: 位置制御完了状態
0x301A_0195	reserved4	3	未使用領域
0x301A_0198	u1_offset_cal_finished_flag	1	オフセット測定状態 0: 測定中 1: 測定完了
0x301A_0199	reserved5	3	未使用領域

## 7.2.3.3 API

表 7-8. Network CPU I/F API 一覧

API	説明
R_NETWORK_CPU_IF_Open	Network CPU I/F の初期化を行います。 また、Shared memory Driver の初期化を行います。 フローチャート：図 7-5
R_NETWORK_CPU_IF_Close	Network CPU I/F を終了します。 また、Shared memory Driver を終了します。 フローチャート：図 7-6
R_NETWORK_CPU_IF_CommandProcess	Shared Memory の Command queue area にモータ制御コマンドが格納されているか確認します。 コマンドが存在する場合、該当するモータマネージャ API を呼び出して処理を実行します。 フローチャート：図 7-7
R_NETWORK_CPU_IF_UpdateCyclicReference	Shared Memory の Cyclic reference area に設定されている各種指令値が更新されているかを確認します。 指令値が Network CPU によって更新されていた場合、該当するモータマネージャ API を呼び出し、新しい指令値をセットします。 フローチャート：図 7-8
R_NETWORK_CPU_IF_ReportMotorStatus	モータマネージャ API を呼び出して各種モータステータスを取得し、更新の有無を確認します。 ステータスが更新されていた場合、Shared Memory の Motor status area に反映します。 フローチャート：図 7-9

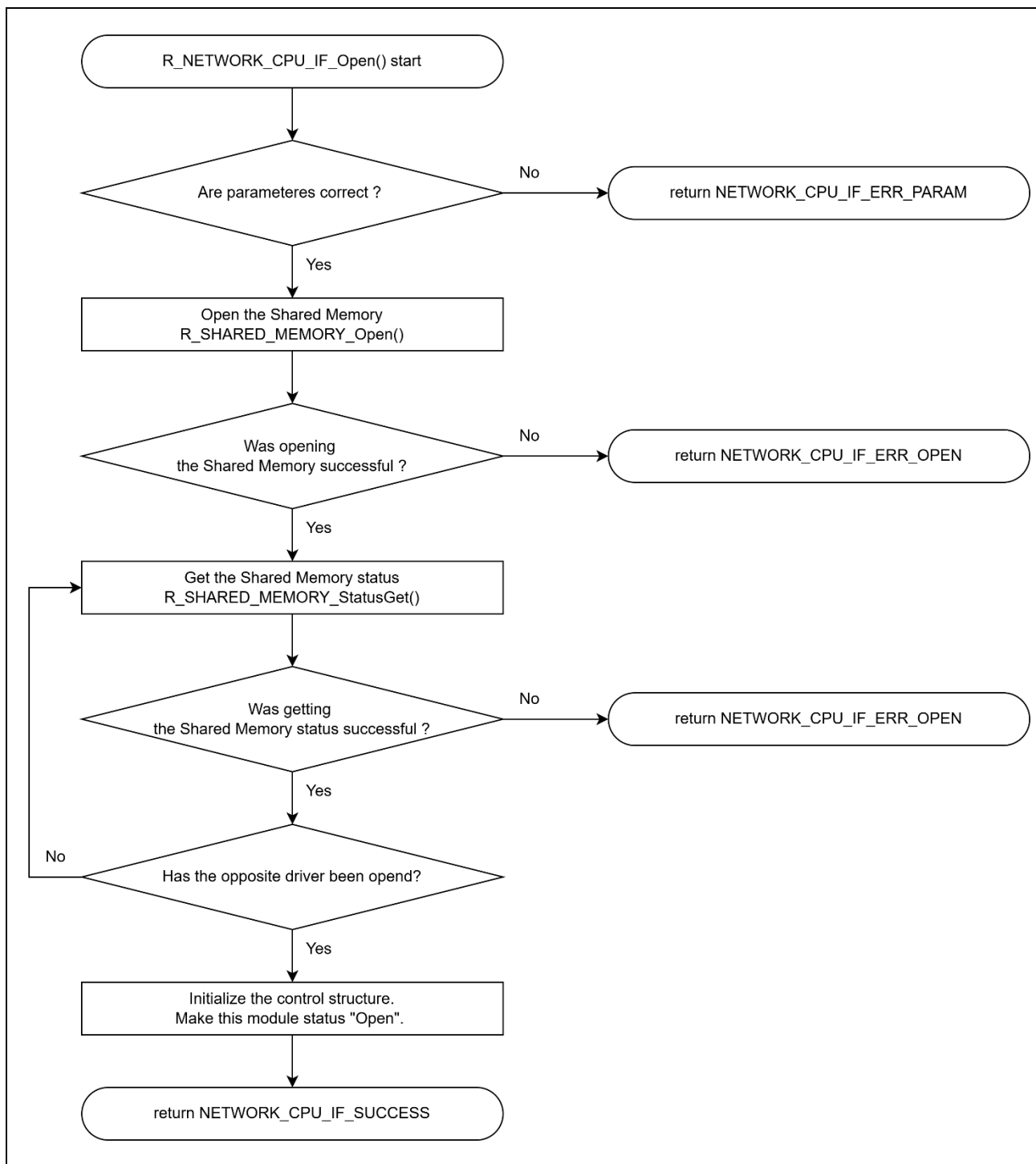


図 7-5. R\_NETWORK\_CPU\_IF\_Open 関数フローチャート

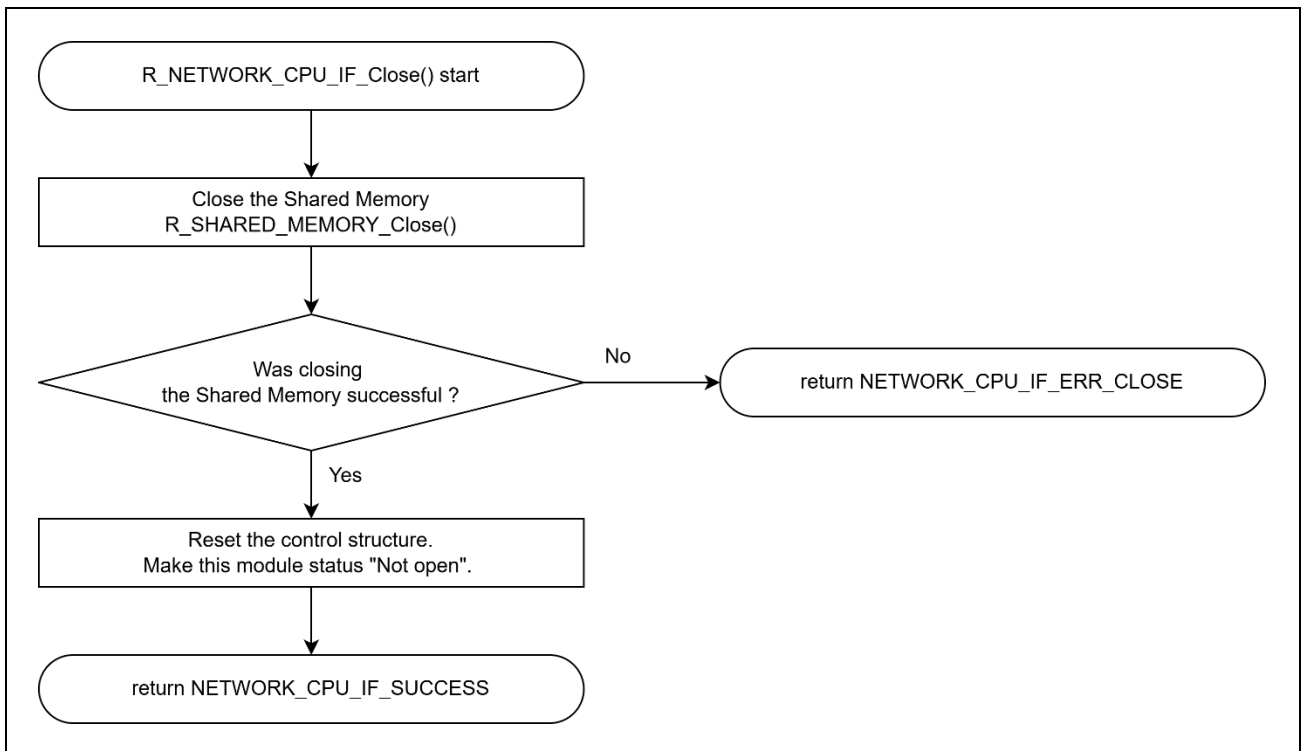


図 7-6. R\_NETWORK\_CPU\_IF\_Close 関数フローチャート

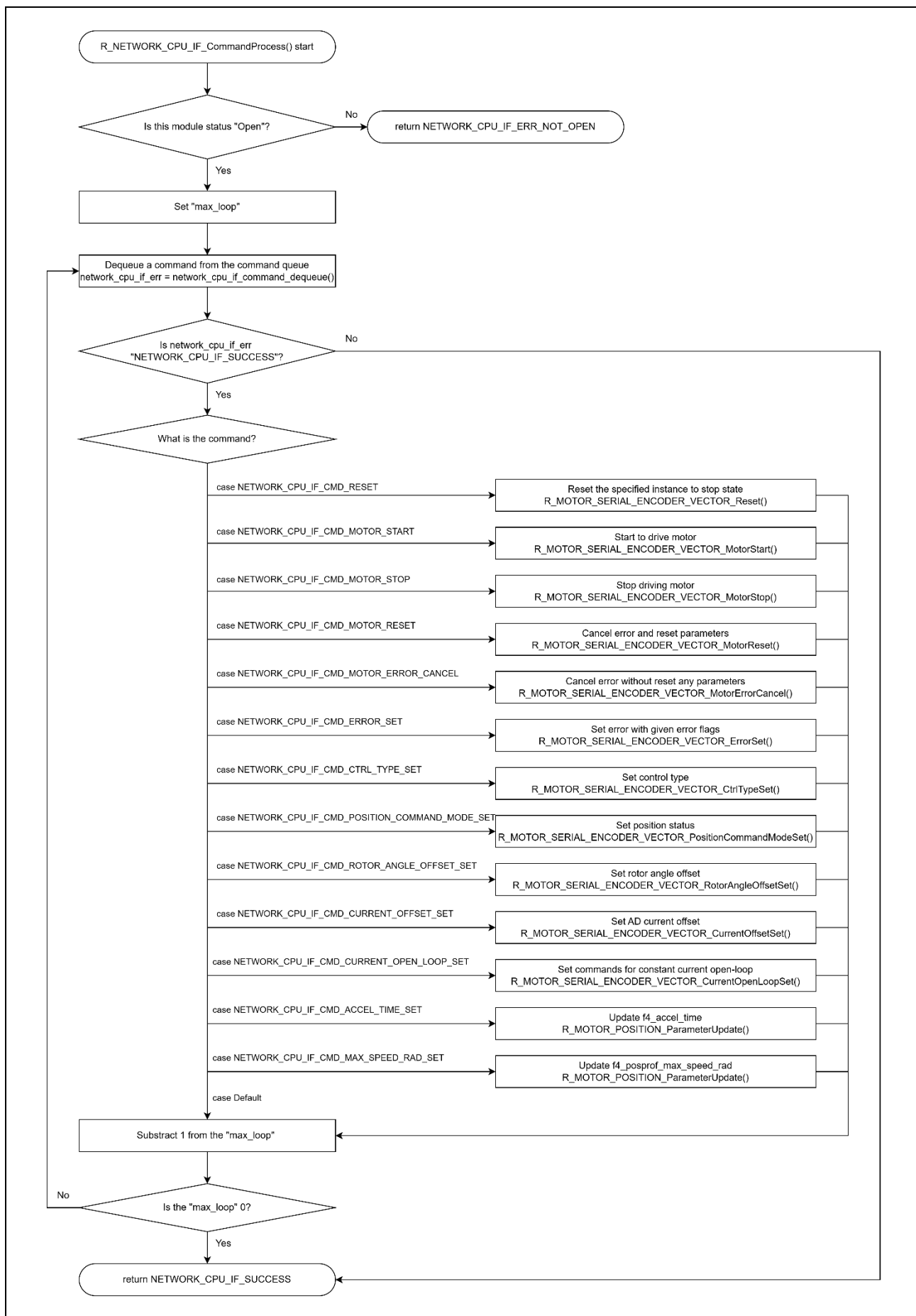


図 7-7. R\_NETWORK\_CPU\_IF\_CommandProcess 関数フローチャート

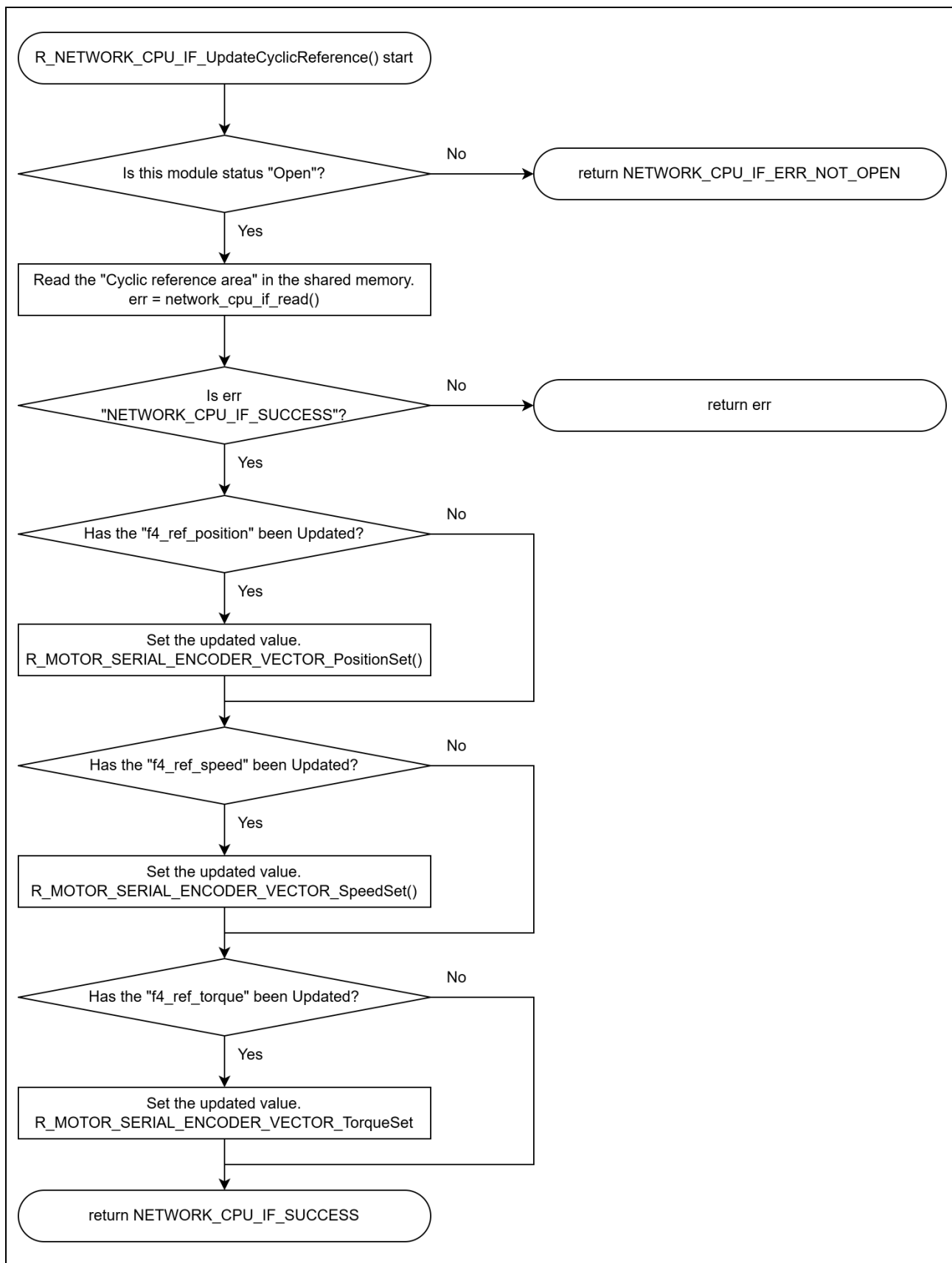


図 7-8. R\_NETWORK\_CPU\_IF\_UpdateCyclicReference 関数フローチャート

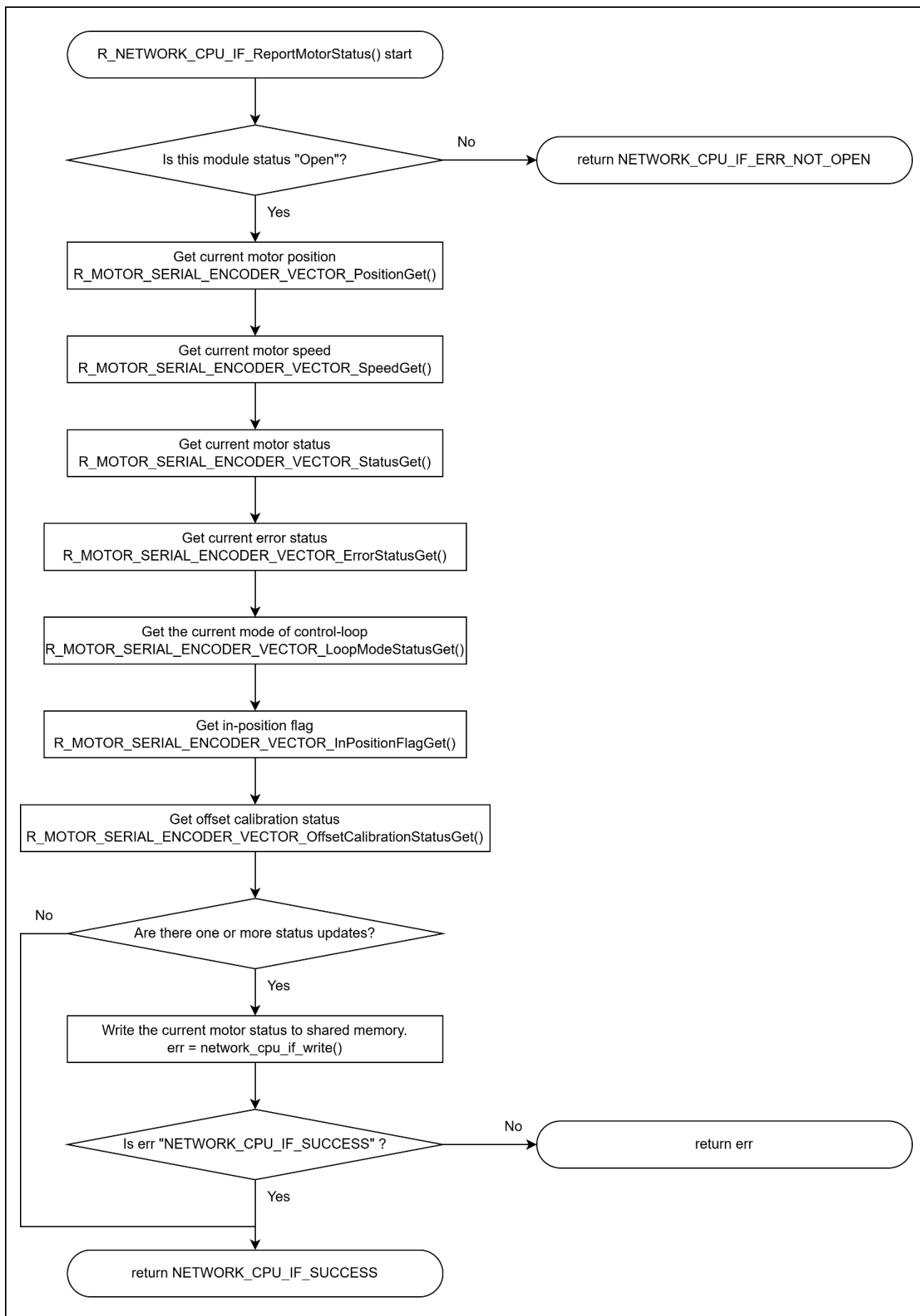


図 7-9. R\_NETWORK\_CPU\_IF\_ReportMotorStatus 関数フローチャート

## 7.2.3.4 プライベート関数

表 7-9. Network CPU I/F プライベート関数 一覧

プライベート関数	説明
network_cpu_if_write	Shared memory にデータを書き込みます。 フローチャート：図 7-10
network_cpu_if_read	Shared memory からデータを読み取ります。 フローチャート：図 7-11
network_cpu_if_command_dequeue	コマンドキューからコマンドを読み取ります。 フローチャート：図 7-12

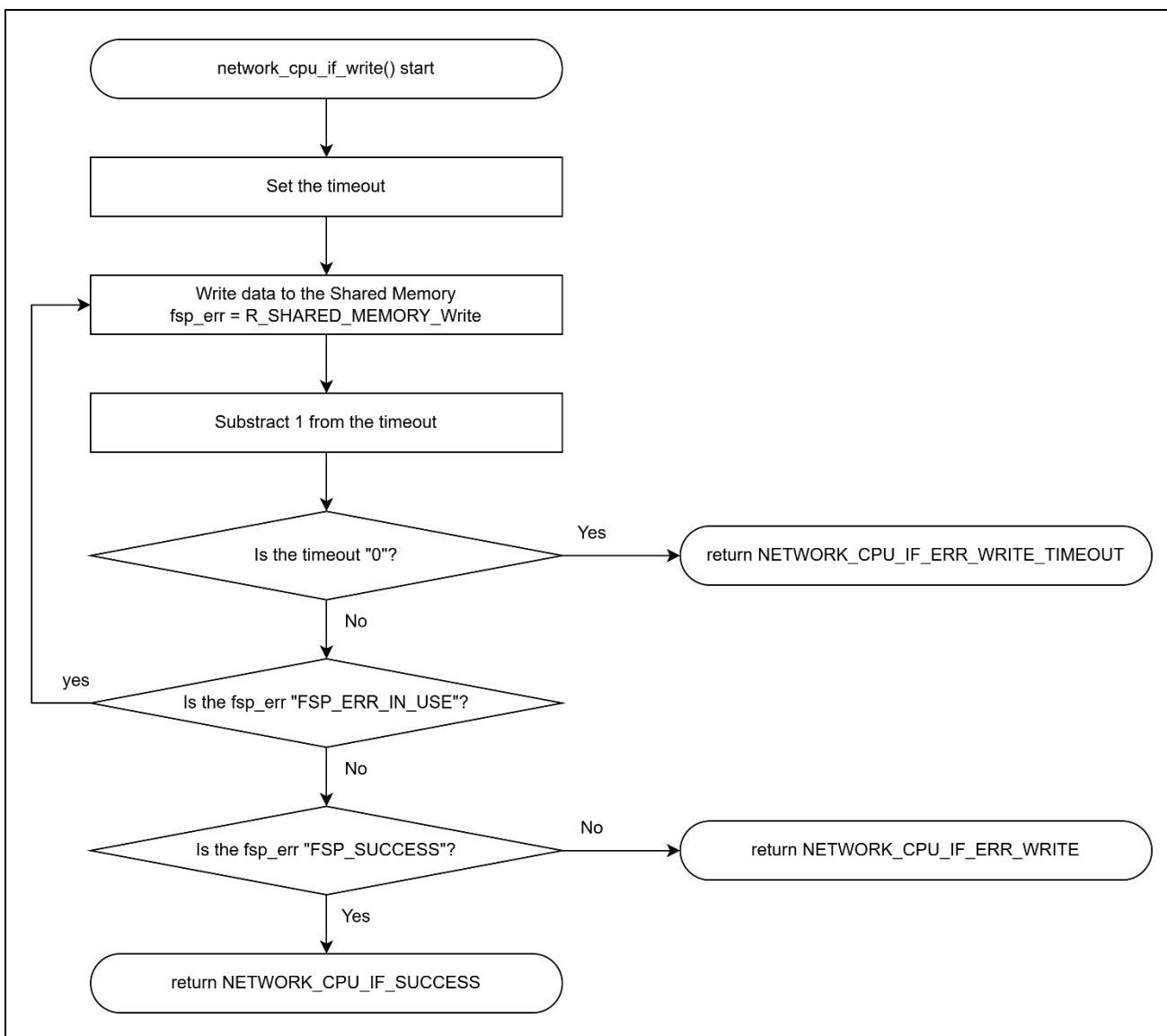


図 7-10. network\_cpu\_if\_write 関数フローチャート

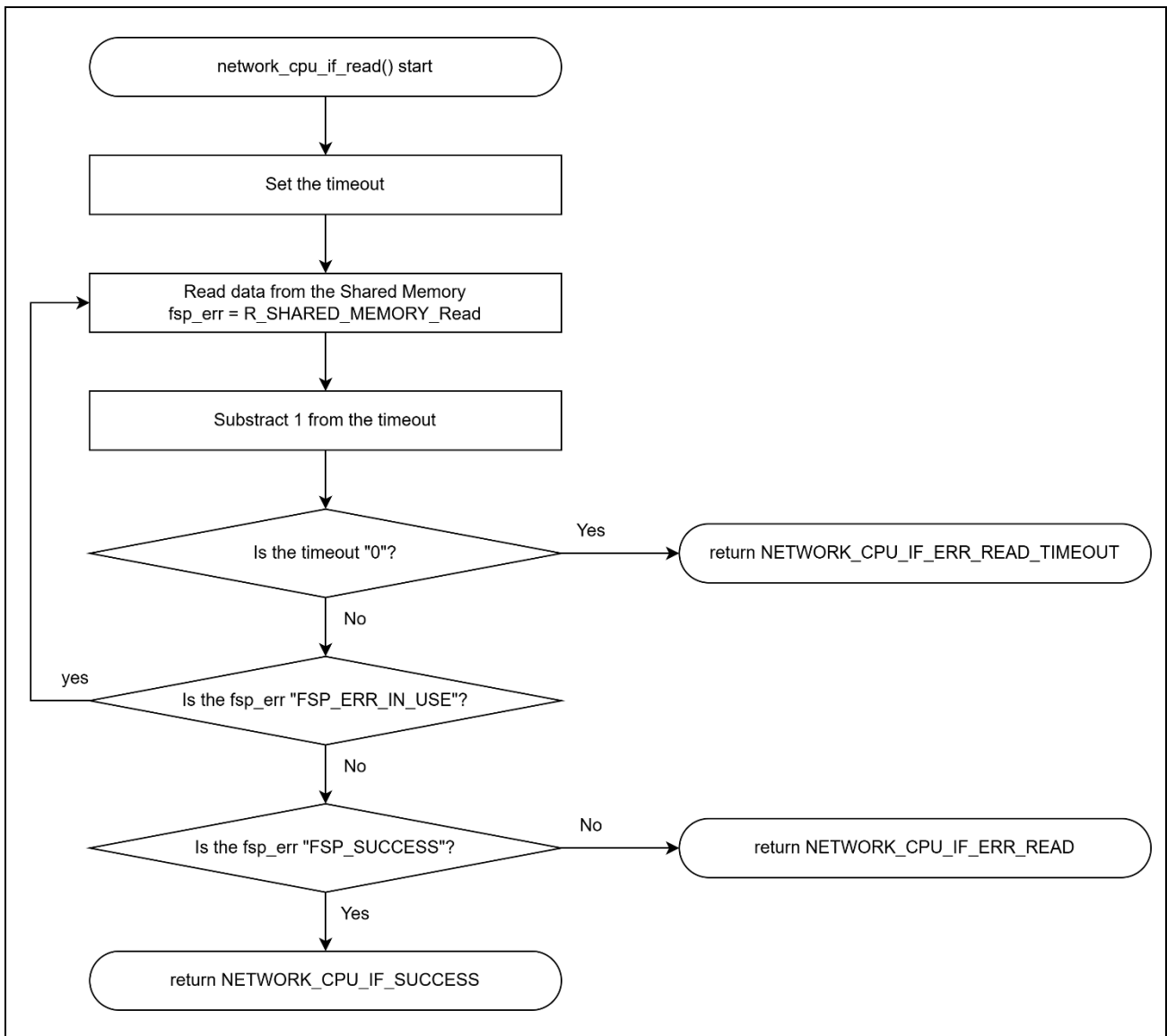


図 7-11. network\_cpu\_if\_read 関数フローチャート

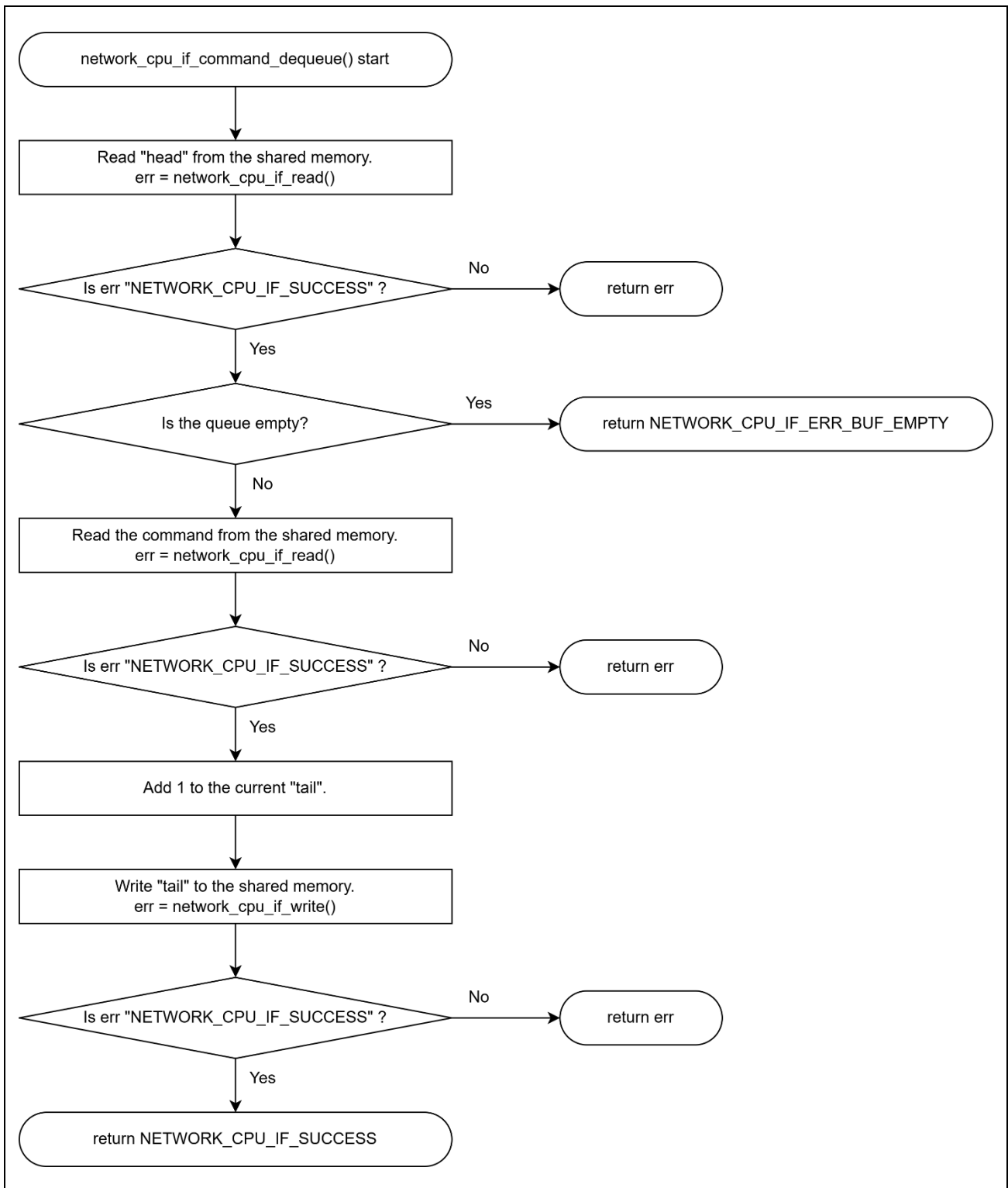


図 7-12. network\_cpu\_if\_command\_dequeue 関数フローチャート

## 7.2.3.5 Callback 関数

表 7-10. Network CPU I/F Callback 関数一覧 (r\_network\_cpu\_if.c)

関数名	説明
shared_memory_callback	<p>これは Shared Memory Driver の Callback 関数です。</p> <p>RZT2M CPU1 側で Shared Memory Driver が Shared Memory へ書き込むと、この割り込みが発生します。</p> <p>Callback 関数内で行われる処理はありません。</p> <p>必要に応じて処理の追加を検討して下さい。</p>

## 7.2.3.6 マクロ定義

表 7-11. Network CPU I/F マクロ定義一覧 (r\_network\_cpu\_if.h)

マクロ名	定義値	備考
CFG_CMD_Q_NUM	8	コマンドキューの数 (2 の累乗にする必要があります)
CFG_CMD_Q_OFFSET	0x0000	Command queue area のオフセットアドレス
CFG_CYCLIC_REF_AREA_OFFSET	0x0100	Cyclic reference area の オフセットアドレス
CFG_MTR_STATUS_AREA_OFFSET	0x0180	Motor status area のオ フセットアドレス
CFG_NETWORK_CPU_IF_TIMEOUT	100	Shared memory 読み書 き時に使用されるタイ ムアウト値
CFG_SHARED_MEMORY_INSTANCE_ADDR	&g_shared_memory0	Shared memory ドライ バのインスタンスへの ポインタ
CFG_MOTOR_CTRL_INSTANCE_ADDR	&g_st_serial_encoder_vector	モータコントロールの インスタンスへのポイ ンタ
NETWORK_CPU_IF_CMD_DATA_MAX_BYTE	12	コマンドキューの最大 データサイズ

## 7.2.3.7 列挙型

表 7-12. Network CPU I/F 列挙型一覧 (r\_network\_cpu\_if.h)

列挙型	定義	備考
network_cpu_if_err_t Network CPU I/F エラーコード	NETWORK_CPU_IF_SUCCESS	エラー無し
	NETWORK_CPU_IF_ERR_OPEN	Open エラー
	NETWORK_CPU_IF_ERR_CLOSE	Close エラー
	NETWORK_CPU_IF_ERR_PARAM	Parameter エラー
	NETWORK_CPU_IF_ERR_NOT_OPEN	Not open エラー
	NETWORK_CPU_IF_ERR_BUF_FULL	コマンドキューが満杯
	NETWORK_CPU_IF_ERR_BUF_EMPTY	コマンドキューが空
	NETWORK_CPU_IF_ERR_WRITE	書き込みエラー
	NETWORK_CPU_IF_ERR_WRITE_TIMEOUT	書き込みタイムアウト
	NETWORK_CPU_IF_ERR_READ	読み取りエラー
	NETWORK_CPU_IF_ERR_READ_TIMEOUT	読み取りタイムアウト
	NETWORK_CPU_IF_ERR_EXCEED_PAYLOAD	データが超過した

列挙型	定義	備考
network_cpu_if_status_t Network CPU I/F ステータス	NETWORK_CPU_IF_STATUS_NOT_OPEN	Not open 状態
	NETWORK_CPU_IF_STATUS_OPEN	Open 状態
network_cpu_if_cmd_t Network CPU I/F コマンド定義	NETWORK_CPU_IF_CMD_NONE	コマンド無し
	NETWORK_CPU_IF_CMD_RESET	Command Process API で 使用されるコマンド (参照 : 図 7-7)
	NETWORK_CPU_IF_CMD_MOTOR_START	
	NETWORK_CPU_IF_CMD_MOTOR_STOP	
	NETWORK_CPU_IF_CMD_MOTOR_RESET	
	NETWORK_CPU_IF_CMD_MOTOR_ERROR_CA NCEL	
	NETWORK_CPU_IF_CMD_ERROR_SET	
	NETWORK_CPU_IF_CMD_CTRL_TYPE_SET	
	NETWORK_CPU_IF_CMD_POSITION_COMMA ND_MODE_SET	
	NETWORK_CPU_IF_CMD_ROTOR_ANGLE_OF FSET_SET	
	NETWORK_CPU_IF_CMD_CURRENT_OFFSET _SET	
	NETWORK_CPU_IF_CMD_CURRENT_OPEN_L OOP_SET	
NETWORK_CPU_IF_CMD_ACCEL_TIME_SET		
NETWORK_CPU_IF_CMD_MAX_SPEED_RAD_ SET		

## 7.2.3.8 構造体・変数情報

表 7-13. Network CPU I/F 構造体定義一覧 (r\_network\_cpu\_if.h)

構造体	変数	備考
st_current_offset_cfg_t Current offset configuration 構造体	f4_iu_offset	U-phase 電流オフセット
	f4_iv_offset	V-phase 電流オフセット
	f4_iw_offset	W-phase 電流オフセット
st_current_open_loop_cfg_t Open loop current configuration 構造体	f4_open_loop_current	Open loop 電流値
	f4_ref_speed_rpm	指令速度 (RPM)
network_cpu_if_cmd_hdr_t コマンドキュー head/tail 構 造体	head	コマンドキューの head
	tail	コマンドキューの tail
network_cpu_if_cmd_q_t コマンドキュー 構造体	cmd	コマンド
	size	データサイズ
	data[NETWORK_CPU_IF_ CMD_DATA_MAX_BYTE]	データ配列

network_cpu_if_cyclic_ref_t Cyclic Reference 構造体	f4_ref_position	指令位置
	f4_ref_speed	指令速度
	f4_ref_torque	指令トルク
network_cpu_if_mtr_status_t Motor Status 構造体	f4_pos_deg	現在位置
	f4_speed	現在速度
	u1_status	ステートマシン
	reserved1	未使用領域
	u2_error_status	エラー状態
	reserved2	未使用領域
	u1_ctrl_loop_mode	LOOP MODE
	reserved3	未使用領域
	u1_in_position	位置制御完了状態
	reserved4	未使用領域
	u1_offset_cal_finished_flag	オフセット測定状態
	reserved5	未使用領域
network_cpu_if_cfg_t Network CPU I/F コンフィグ構造体	p_shared_memory_instance	Shared Memory インスタンスへのポインタ
	timeout	タイムアウト値
	shr_mem_cmd_data_max_size	コマンドデータ最大サイズ
	shr_mem_cmd_q_num	コマンドキューの数
	shr_mem_cmd_q_mask	コマンドキューマスク
	shr_mem_cmd_q_offset	共有メモリのコマンドキュー領域のオフセット
	shr_mem_cmd_q_size	共有メモリのコマンドキュー領域のサイズ
	shr_mem_cmd_head_offset	共有メモリのコマンドキュー head 領域のオフセット
	shr_mem_cmd_head_size	共有メモリのコマンドキュー head 領域のサイズ
	shr_mem_cmd_tail_offset	共有メモリのコマンドキュー tail 領域のオフセット
	shr_mem_cmd_tail_size	共有メモリのコマンドキュー tail 領域のサイズ
	shr_mem_cyclic_ref_area_offset	共有メモリの Cyclic Reference Area のオフセット
	shr_mem_cyclic_ref_area_size	共有メモリの Cyclic Reference Area のサイズ
shr_mem_ref_pos_offset	共有メモリの指令位置領域のオフセット	

	shr_mem_ref_vel_offset	共有メモリの指令速度領域のオフセット
	shr_mem_ref_tor_offset	共有メモリの指令トルク領域のオフセット
	shr_mem_mtr_status_area_offset	共有メモリの Motor Status Area のオフセット
	shr_mem_mtr_status_area_size	共有メモリの Motor Status Area のサイズ
	shr_mem_mtr_pos_offset	共有メモリの現在位置領域のオフセット
	shr_mem_mtr_vel_offset	共有メモリの現在速度領域のオフセット
	shr_mem_mtr_status_offset	共有メモリのステートマシン領域のオフセット
	shr_mem_mtr_err_status_offset	共有メモリのエラー状態領域のオフセット
	shr_mem_mtr_ctrl_loop_mode_offset	共有メモリの LOOP MODE 領域のオフセット
	shr_mem_mtr_in_pos_offset	共有メモリの位置制御完了状態領域のオフセット
	shr_mem_mtr_cal_finished_flag_offset	共有メモリのオフセット測定状態領域のオフセット
	max_loop	CommandProcess API で使用される最大ループ回数
	p_st_serial_encoder_vector_control	Serial encoder vector 制御構造体へのポインタ
network_cpu_if_ctrl_t Network CPU I/F 制御構造体	status	Network CPU I/F のステータス
	cmd_hdr	コマンドキュー head/tail 構造体インスタンス
	timeout_count	タイムアウト発生回数
	position_offset	位置オフセット (未使用)
	cyclic_ref_buf	Cyclic Reference 領域のバッファ
	mtr_status_buf	Motor Status 領域のバッファ
	p_cfg	Network CPU I/F コンフィグ構造体へのポインタ
network_cpu_if_api_t Network CPU I/F API 構造体	Open	オープン関数
	Close	クローズ関数
	CommandProcess	コマンド処理関数
	UpdateCyclicReference	周期的指令値アップデート関数
	ReportMotorStatus	モータステータスレポート関数

## 7.2.3.9 グローバル変数情報

表 7-14. Network CPU I/F グローバル変数一覧 (r\_network\_cpu\_if.c)

型名	インスタンス名	説明
network_cpu_if_ctrl_t	g_network_cpu_if_ctrl	Network CPU I/F 制御構造体インスタンス。 Open 関数で初期化されます。
network_cpu_if_cfg_t	g_network_cpu_if_cfg	Network CPU I/F コンフィグ構造体インスタンス。 インスタンス生成時に各メンバが初期化されます。 初期化以外で各メンバの値を変更することはできません。
network_cpu_if_api_t	g_network_cpu_if_api	Network CPU I/F API 構造体インスタンス。 インスタンス生成時に各メンバが初期化されます。 初期化以外で各メンバの値を変更することはできません。

## 7.3 EtherCAT(CiA402) (RZ/T2M CPU1)

## 7.3.1 基本仕様

表 7-15. RZ/T2M CPU1 基本仕様

機能	項目	説明
EtherCAT SubDevice Stack	PHY とのインターフェース	MII
	MDIO タイプ	GMAC
	通信ポート数	In, Out の 2 ポート
	EtherCAT LED	RUN, ERR, L/A IN, L/A OUT の 4 種類
	EtherCAT Device ID	SW12(8bit)により設定可能
	デバイスプロファイル	CiA402 Drive Profile, FSoE Safety Drive Connection
	メールボックス	CoE をサポート
	同期モード	DC Mode (DC-Synchronous with SM2 event), SM mode (SM-synchronous) の 2 種類をサポート
	DC Mode の Cycle Time	最小 50 $\mu$ s
	割り込み優先度 (PDI, Sync0, Sync1)	Priority 4
	割り込み優先度(Timer)	Priority 6
CiA402 Drive Profile	Mode	Profile position mode, Homing mode, Cyclic synchronous position mode, Cyclic synchronous velocity mode, の 4 種類をサポート
	モータ軸数	1 軸
Fusa CPU Communication	役割	FSoE Master から受信した Safety Master PDU を機能安全 CPU(RZ/T2L-A)へ送信  機能安全 CPU から受信した Safety Slave PDU を FSoE Master へ送信
Safety Drive	停止機能	IEC61800-5-2 : STO, SS1-t に対応(*1)
Motor CPU I/F	役割	Motor 制御 CPU との通信 I/F を提供
	通信手段	Shared Memory
	割り込み優先度 (INTCPU)	Priority 6
Fusa I/F	役割	機能安全 CPU との通信 I/F を提供
	通信手段	調歩同期式(UART)
	割り込み優先度 (ERI, RXI, RXI, TEI)	Priority 2

(\*1) 本ソフトウェアでは、停止機能 SS1-t は Cyclic synchronous velocity mode のみ対応しています。

7.3.2 ソフトウェア全体構成

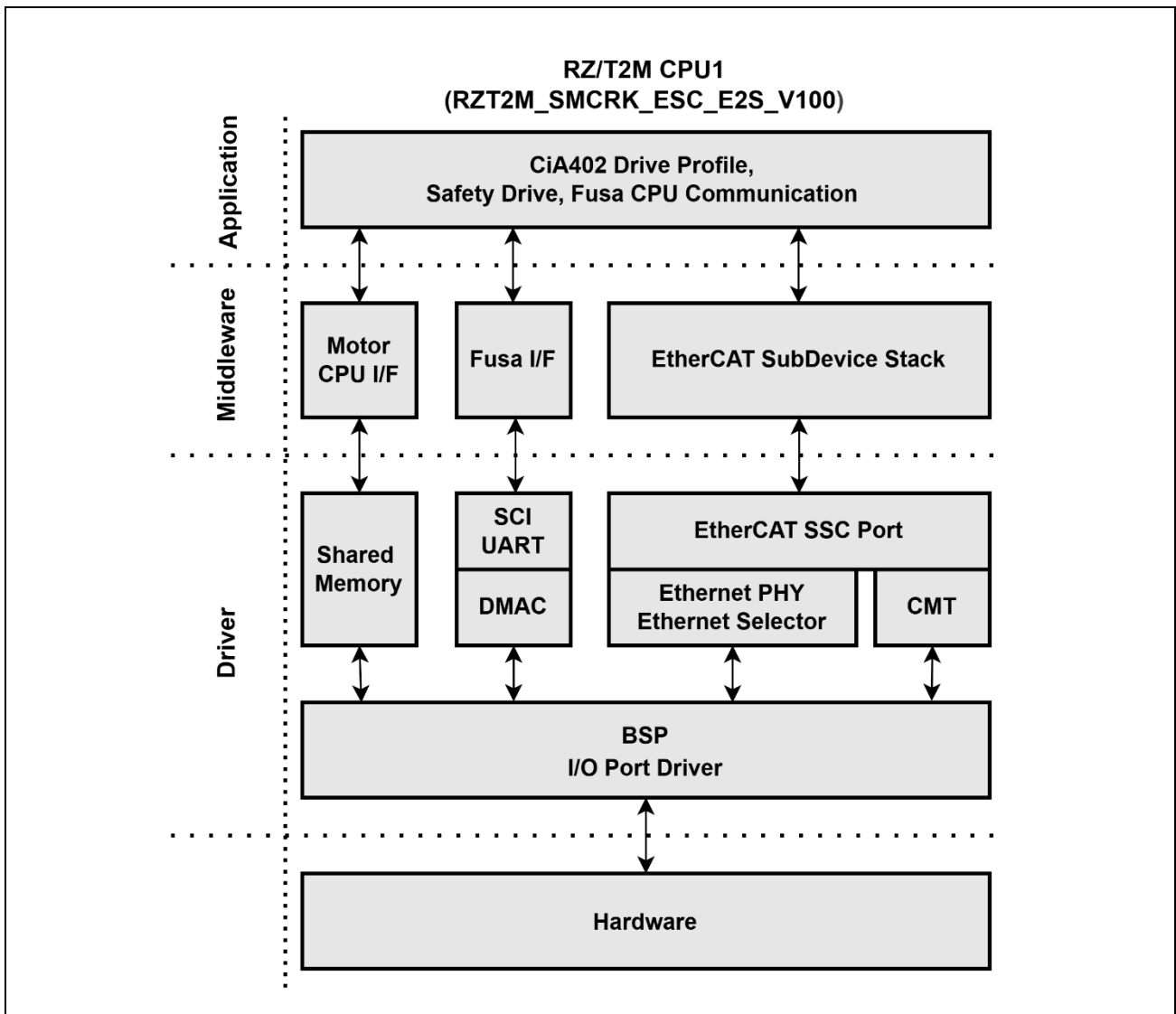


図 7-13. RZ/T2M CPU1 ソフトウェア全体構成

## 7.3.3 src フォルダ構成

表 7-16. RZ/T2M CPU1 src フォルダ構成

フォルダ・ファイル				説明
src	ethercat	Beckhoff/Src	applInterface.h	EtherCAT SubDevice Stack (SSC Tool 生成コード)
			coeappl.c	
			coeappl.h	
			ecatappl.c	
			ecatappl.h	
			ecatcoe.c	
			ecatcoe.h	
			ecatslv.c	
			ecatslv.h	
			ecat_def.h	
			esc.h	
			mailbox.c	
			mailbox.h	
			objdef.c	
			objdef.h	
			sdoserv.c	
			sdoserv.h	
			cia402appl.c	
	cia402appl.h			
			renesas	samplecia402.c
samplecia402.h				
	fusa_if		Fusa I/F モジュール	
	motor_cpu_if		Motor CPU I/F モジュール	
	hal_entry.c		hal_entry 関数	

## 7.3.4 hal\_entry ・メイン処理

hal\_entry 関数のフローチャート図を図 7-14 に示します。

また、hal\_entry 関数で呼び出される関数について表 7-17 に示します。

表 7-17. hal\_entry 関数で呼び出される関数一覧

関数名	説明。
R_MOTOR_CPU_IF_Open	Motor CPU I/F の Open 関数。 詳細は 7.3.8 章を参照してください。
R_FUSA_IF_Open	Fusa I/F の Open 関数。 詳細は 7.3.9 章を参照してください。
RM_ETHERCAT_SSC_PORT_Open	FSP で提供される EtherCAT SSC Port の Open 関数。 詳細は RZT Flexible Software Package Documentation を参照してください。
change_phy_reg	PHY register を変更する関数。 各 PHY(VSC8541)の"GPIO Control 2 register"の値を 0x2000 に変更します。
ecat_ssc_init	EtherCAT SubDevice Stack Code および CiA402 Drive Profile の初期化処理を実行する関数。 図 7-15 にフローチャートを示します。 フローチャート内の各関数・変数の詳細は " Application Note ET9300 (EtherCAT SubDevice Stack Code)" を参照してください。
R_BSP_PinSet	FSP で提供される関数。LED11 を点灯します。 詳細は RZT Flexible Software Package Documentation を参照してください。
MainLoop	EtherCAT SubDevice Stack Code および CiA402 Drive Profile のメイン処理関数。 詳細は " Application Note ET9300 (EtherCAT SubDevice Stack Code)" を参照してください。

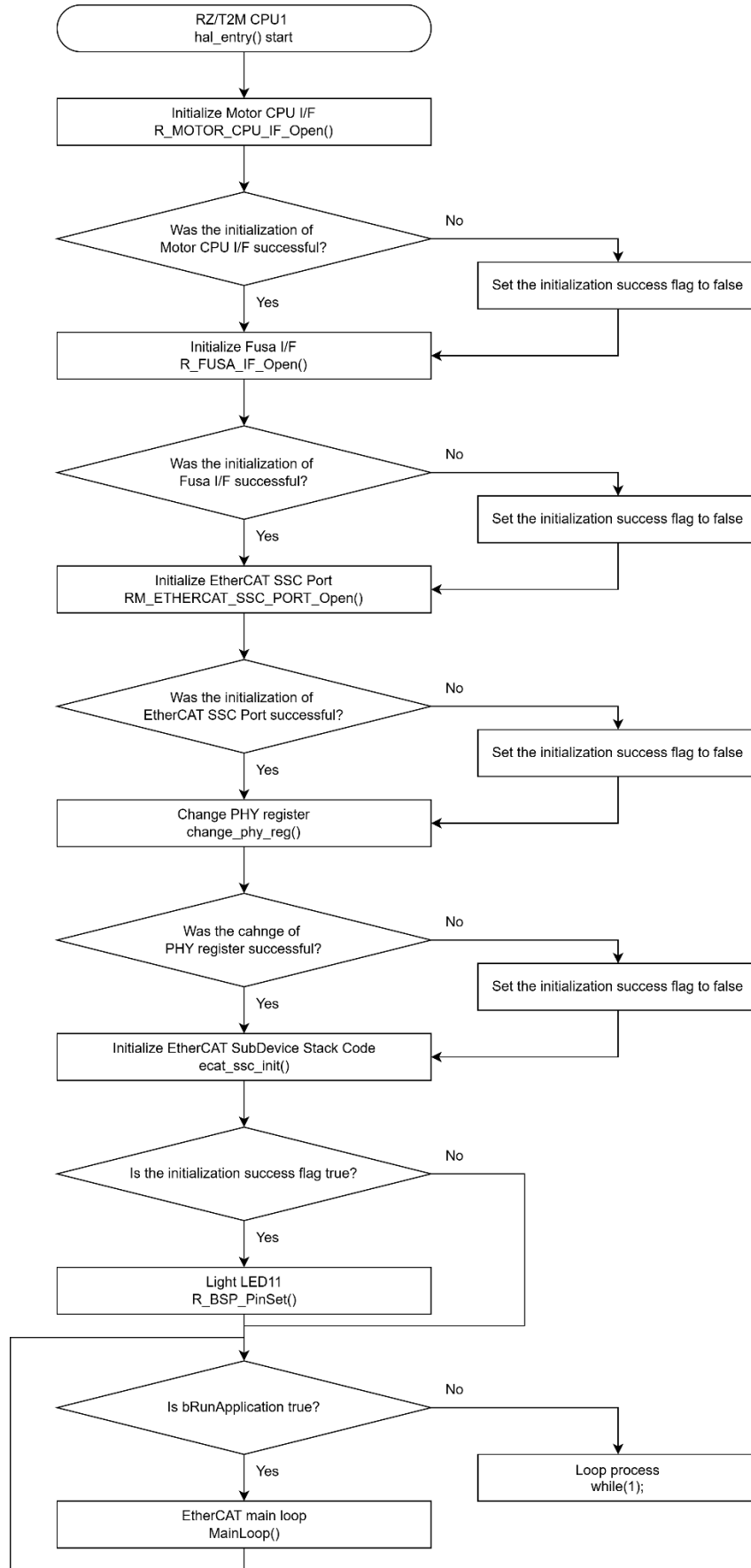


図 7-14. RZ/T2M CPU1 hal\_entry 関数フローチャート

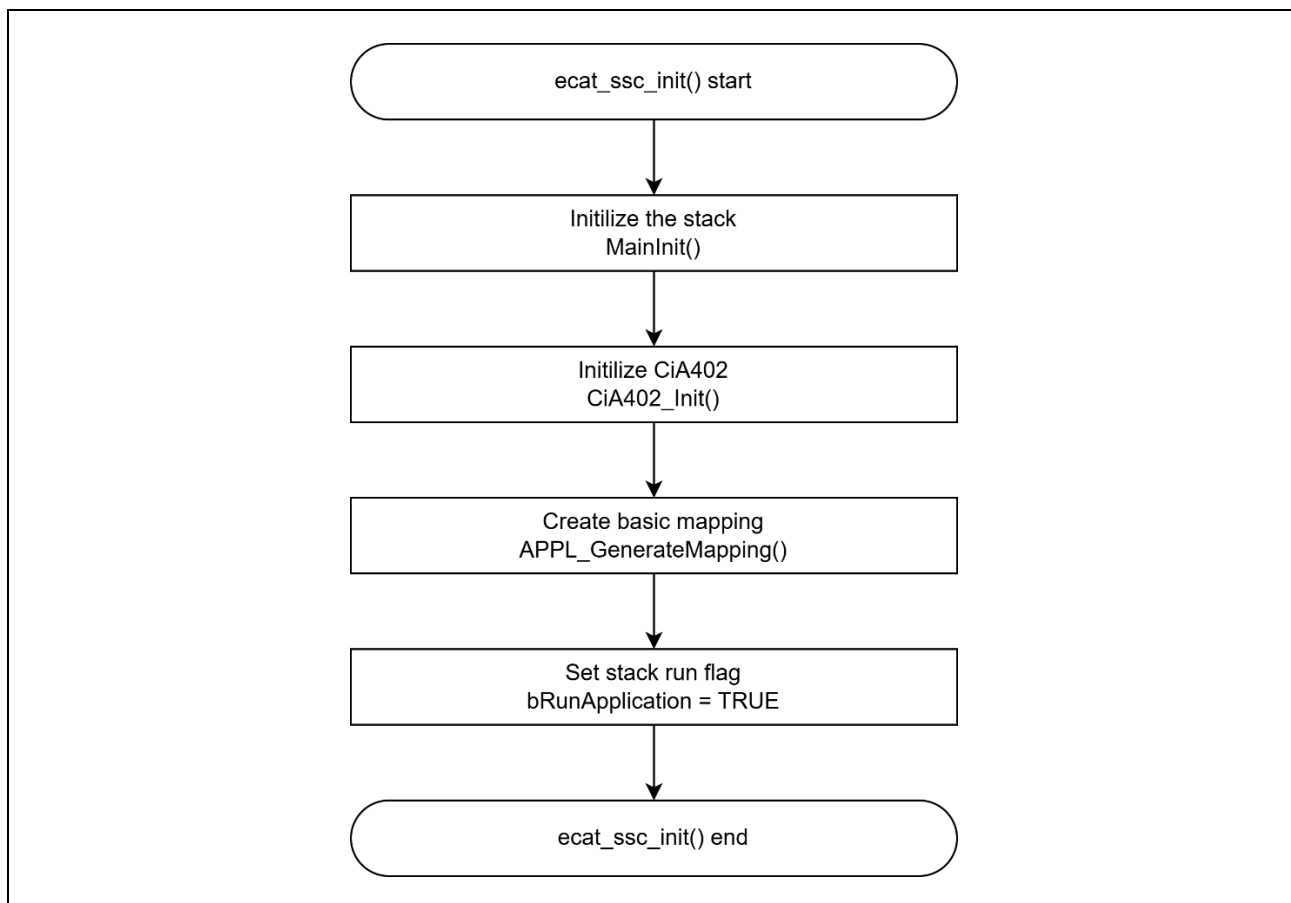


図 7-15. ecat\_ssc\_init 関数フローチャート

### 7.3.5 CiA402 Drive Profile

CiA402 ドライブプロファイルはドライブおよびモーションコントロール用のデバイスプロファイルであり、主にサーボドライブ、正弦波インバータ、およびステッピングモータ用コントローラの機能動作を定義します。

このプロファイルでは、複数の動作モードと対応する設定パラメータがオブジェクトディクショナリとして規定されます。

また、状態ごとの内部および外部動作を規定する有限状態オートマトン (Finite State Automaton: FSA) も含まれます。

状態を変更する場合はコントロールワードオブジェクトを通じて指定することで、現在の状態を示すステータスワードオブジェクトに遷移後の結果が反映されます。

コントロールワードと各種コマンド値（速度など）は RxPDO に割り当てられ、ステータスワードと各種実査値（位置など）は TxPDO に割り当てられます。

詳細については CiA402 規格書の内容を確認してください

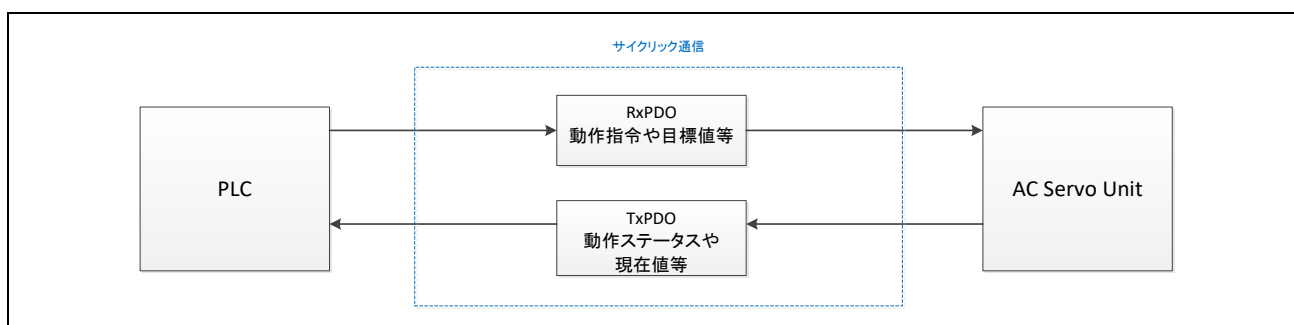


図 7-16. CiA402 Drive Profile 概要

#### 7.3.5.1 CiA402 動作モード

CiA402 として規定されている動作モードのうち、本プログラムでは以下をサポートします。

表 7-18. 動作モード一覧

Operation Mode	Support	Refer for details
Profile position mode	Available	7.3.5.4
Velocity mode (frequency converter)	Not available	-
Profile velocity mode	Not available	-
Profile torque mode	Not available	-
Homing mode	Available	7.3.5.5
Interpolated position mode	Not available	-
Cyclic synchronous position mode	Available	7.3.5.2
Cyclic synchronous velocity mode	Available	7.3.5.3
Cyclic synchronous torque mode	Not available	-
Cyclic synchronous torque mode with commutation angle	Not available	-
Manufacturer specific mode	Not available	-

## 7.3.5.2 Cyclic synchronous position mode

このモードでは、周期的に指令位置を送信し、モータの位置制御を行います。

表 7-19. csp モード RxPDO 一覧

RxPDO	Index	説明
Control Word	0x6040	CiA402 制御用コマンド
Target Position	0x607A	位置指令値 [deg] (*1)

(\*1) モータへの指令値は入力値の 1/10 に変換されます。例) 1234 を入力→モータへの指令値は 123.4。

表 7-20. csp モード TxPDO 一覧

TxPDO	Index	説明
Error Code	0x603F	CiA402 エラーコード
Status Word	0x6041	CiA402 状態
Modes of Operation Display	0x6061	モードの種類 (=8)
Position Actual Value	0x6064	モータ現在位置 [deg] (*2)
Velocity Actual Value	0x606C	モータ現在速度 [rpm] (*2)

(\*2) 実際の値から 10 倍に変換されます。例) 5678 が出力→実際の値は 567.8。

表 7-21. csp モード モータ設定

項目	設定値
LOOP MODE	位置制御
位置制御の位置指令生成のモード	台形駆動動作

## 7.3.5.3 Cyclic synchronous velocity mode

このモードでは、周期的に指令速度を送信し、モータの速度制御を行います。

表 7-22. csv モード RxPDO 一覧

RxPDO	Index	説明
Control Word	0x6040	CiA402 制御用コマンド
Target Velocity	0x60FF	速度指令値 [rpm] (*1)

(\*1) モータへの指令値は入力値の 1/10 に変換されます。例) 1234 を入力→モータへの指令値は 123.4

表 7-23. csv モード TxPDO 一覧

TxPDO	Index	説明
Error Code	0x603F	CiA402 エラーコード
Status Word	0x6041	CiA402 状態
Modes of Operation Display	0x6061	モードの種類 (=9)
Position Actual Value	0x6064	モータ現在位置 [deg] (*2)
Velocity Actual Value	0x606C	モータ現在速度 [rpm] (*2)

(\*2) 実際の値から 10 倍に変換されます。例) 5678 が表示→モータ現在位置は 567.8deg

表 7-24. csv モード モータ設定

項目	設定値
LOOP MODE	速度制御

## 7.3.5.4 Profile position mode

このモードでは、速度・加速度を設定し、モータの位置制御を行います。

表 7-25. pp モード RxPDO 一覧

RxPDO	Index	説明
Control Word	0x6040	CiA402 制御用コマンド
Target Position	0x607A	位置指令値 [deg] (*1)
Profile Velocity	0x6081	プロファイル速度 [rpm] (*1)
Profile Acceleration	0x6083	プロファイル加速度[rpm/s] (*1)
Profile Deceleration	0x6084	プロファイル減速度[rpm/s] (*3)

(\*1) モータへの指令値は入力値の 1/10 に変換されます。例) 1234 を入力→モータへの指令値は 123.4。

(\*3) 本ソフトウェアでは未使用です。

表 7-26. pp モード TxPDO 一覧

TxPDO	Index	説明
Error Code	0x603F	CiA402 エラーコード
Status Word	0x6041	CiA402 状態
Modes of Operation Display	0x6061	モードの種類 (=1)
Position Actual Value	0x6064	モータ現在位置 [deg] (*2)
Velocity Actual Value	0x606C	モータ現在速度 [rpm] (*2)

(\*2) 実際の値から 10 倍に変換されます。例) 5678 が出力→実際の値は 567.8。

表 7-27. pp モード モータ設定

項目	設定値
LOOP MODE	位置制御
位置制御の位置指令生成のモード	台形駆動動作
加減速時間 [s]	(Profile Velocity) / (Profile Acceleration)
台形速度制御の最大速度 [rpm]	(Profile Velocity) / 10

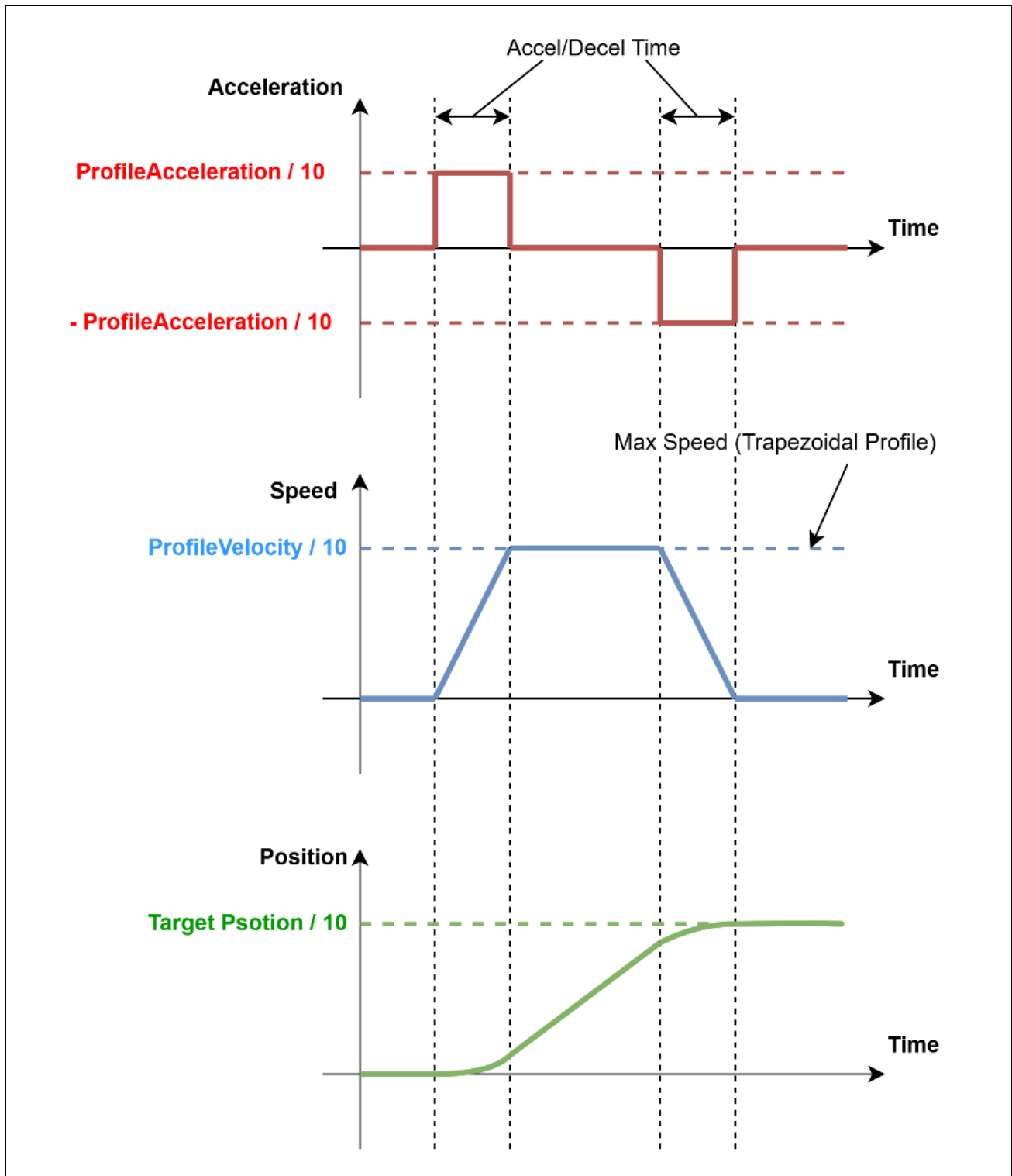


図 7-17. pp mode 台形駆動動作設定

## 7.3.5.5 Homing mode

このモードでは、Homing Method 37 を使用してモータの現在位置を変更します。

表 7-28. Homing モード RxPDO 一覧

RxPDO	Index	説明
Control Word	0x6040	CiA402 制御用コマンド
Home Offset	0x607C	指令現在位置 [deg] (*1)
Home Method	0x6098	Homing 方法
Homing Speed For Switch	0x6099	スイッチ用指令速度 (*3)
Homing Speed For Zero	0x6099	原点用指令速度 (*3)
Homing Acceleration	0x609A	Homing 全体の加減速 (*3)

(\*1) モータへの指令値は入力値の 1/10 に変換されます。例) 1234 を入力→モータへの指令値は 123.4。

(\*3) 本ソフトウェアでは未使用です。

表 7-29. Homing モード TxPDO 一覧

TxPDO	Index	説明
Error Code	0x603F	CiA402 エラーコード
Status Word	0x6041	CiA402 状態
Modes of Operation Display	0x6061	モードの種類 (=6)
Position Actual Value	0x6064	モータ現在位置 [deg] (*2)
Velocity Actual Value	0x606C	モータ現在速度 [rpm] (*2)

(\*2) 実際の値から 10 倍に変換されます。例) 5678 が出力→実際の値は 567.8。

表 7-30. Homing モード モータ設定

項目	設定値
LOOP MODE	位置制御
位置制御の位置指令生成のモード	台形駆動動作

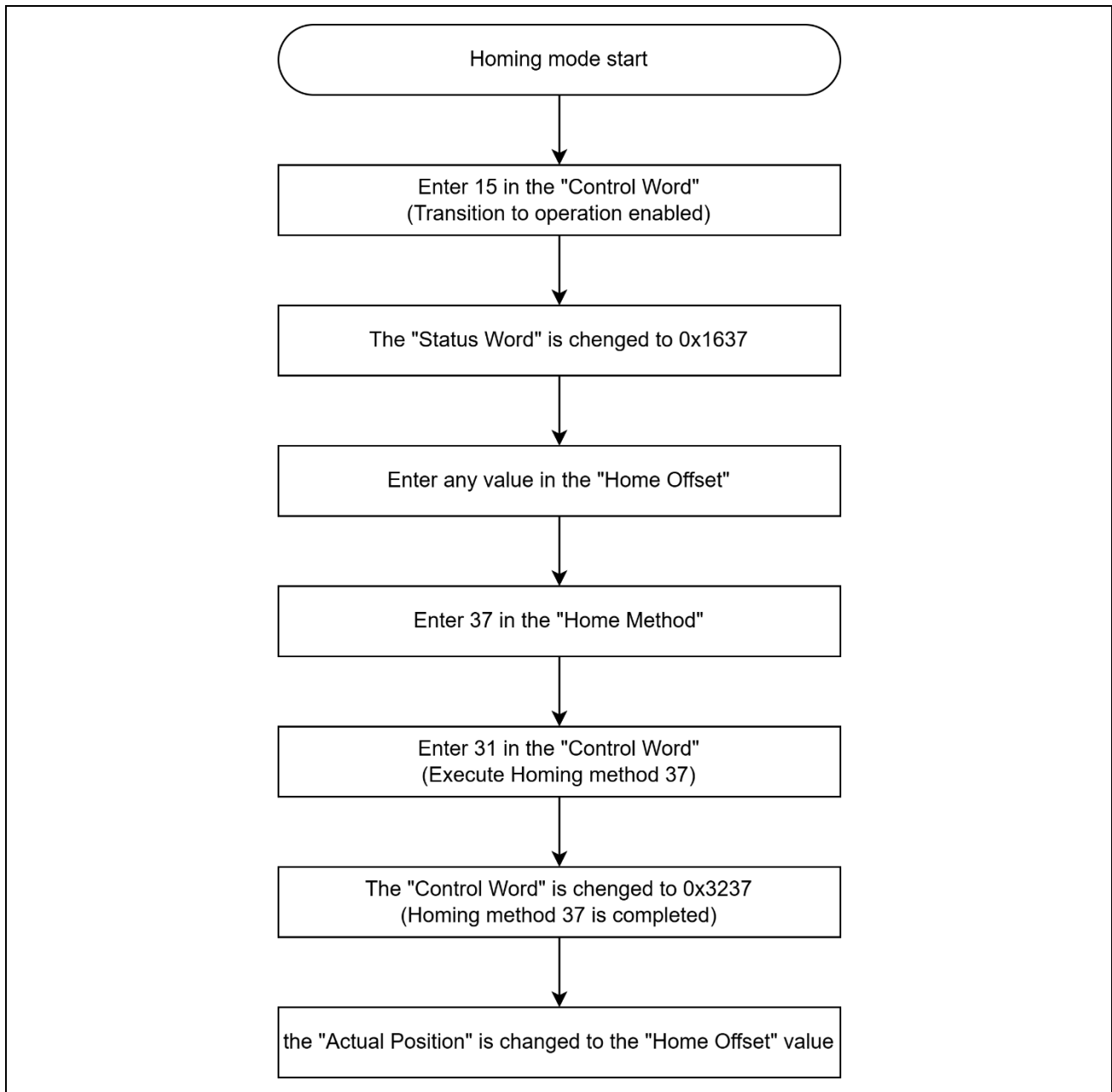


図 7-18. Homing method 37 の実行フローチャート

7.3.5.6 CiA402 状態遷移

CiA402 として規定されている FSA として、サンプルプログラムでは以下をサポートします。

表 7-31 に対応する遷移番号の発生トリガーおよび挙動を示します。

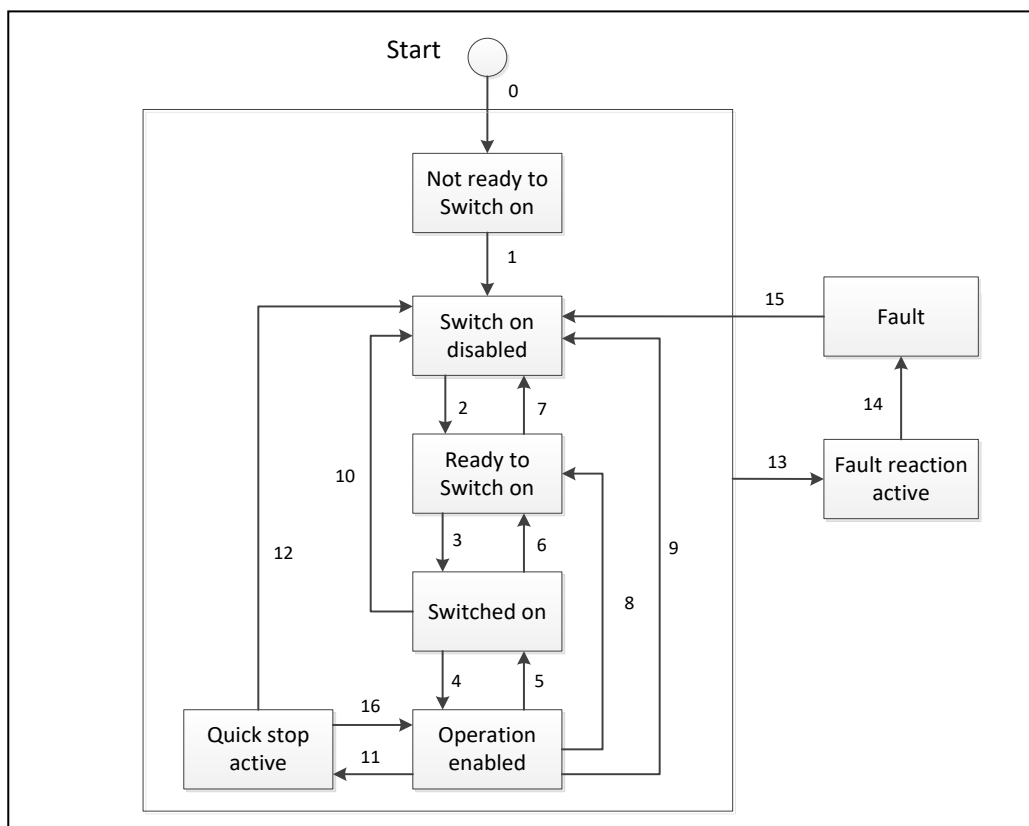


図 7-19. CiA402 状態遷移図

表 7-31. CiA402 各状態遷移

対応する遷移番号	遷移発生トリガー	遷移時の挙動
0, 1	EtherCAT State Machine(ESM)が OP であれば自動でトリガー	何もしない。
2		図 7-20 を参照。
3	Control Word = 7 または Control Word = 15	図 7-21 を参照。
4	Control Word = 15	図 7-22 を参照。
5	Control Word = 7	図 7-23 を参照。
6	Control Word = 6	何もしない。
7	Control Word = 0	何もしない。
8	Control Word = 6	図 7-23 を参照。
9	Control Word = 0	図 7-23 を参照。
10	Control Word = 0	何もしない。
11	Control Word = 2	図 7-23 を参照。

12	Control Word = 0	何もしない。
13	A) 遷移 2 実行時、モータエラーがある場合 B) 遷移 3, 16 実行時、モータ動作が禁止されている場合 C) Operation enabled 時にモータエラーがある場合 D) Operation enabled 時に STO / SS1-t の要求がある場合	図 7-24 を参照。
14	Fault reaction active のとき自動でトリガー。	図 7-25 を参照。
15	Control Word = 128	図 7-26 を参照。
16	Control Word = 15	図 7-27 を参照。

### 7.3.5.7 CiA402 Error Code

Fault reaction active へ遷移する時、発生したエラー内容に応じて Error Code (0x603F)に値が格納されま  
す。表 7-32 に Error Code 一覧を示します。

表 7-32. Error Code(0x603F) 一覧

エラー種類	Error Code	意味
モータエラー	0x2214	HW 過電流エラー(0x0001)が発生した。
	0x3211	過電圧エラー(0x0002)が発生した。
	0x7310	回転速度エラー(0x0004)が発生した。
	0x3221	低電圧エラー(0x0080)が発生した。
	0x2220	SW 過電流エラー(0x0100)が発生した。
	0x4310	温度超過エラー(0x0200)が発生した。
	0x7120	未定義エラー(0xFFFF)が発生した。
モータ停止要求	0xFF00	STO の要求があった。
	0xFF01	SS1-t の要求があった。
モータ動作禁止	0x2250	モータ動作が禁止されている。

## 7.3.5.8 CiA402 Drive Profile 向け関数

ここでは、本サンプルソフトウェアにて実装されている関数について説明します。

SSC Tool が生成する EtherCAT SubDevice Stack Code に実装されている関数については、

” Application Note ET9300 (EtherCAT SubDevice Stack Code)”を参照してください。

表 7-33. CiA402 状態遷移関数一覧 (samplecia402.c)

関数名	説明
CiA402_StateTransition1	本ソフトウェアでは処理はありません。 (*1)
CiA402_StateTransition2	フローチャートを図 7-20 に示します。 (*1)
CiA402_StateTransition3	フローチャートを図 7-21 に示します。 (*1)
CiA402_StateTransition4	フローチャートを図 7-22 に示します。 (*1)
CiA402_StateTransition5	フローチャートを図 7-23 に示します。 (*1)
CiA402_StateTransition6	本ソフトウェアでは処理はありません。 (*1)
CiA402_StateTransition7	本ソフトウェアでは処理はありません。 (*1)
CiA402_StateTransition8	フローチャートを図 7-23 に示します。 (*1)
CiA402_StateTransition9	フローチャートを図 7-23 に示します。 (*1)
CiA402_StateTransition10	本ソフトウェアでは処理はありません。 (*1)
CiA402_StateTransition11	フローチャートを図 7-23 に示します。 (*1)
CiA402_StateTransition12	本ソフトウェアでは処理はありません。 (*1)
CiA402_LocalError	フローチャートを図 7-24 に示します。 (*1)
CiA402_StateTransition14	フローチャートを図 7-25 に示します。 (*1)
CiA402_StateTransition15	フローチャートを図 7-26 に示します。 (*1)
CiA402_StateTransition16	フローチャートを図 7-27 に示します。 (*1)

(\*1) CiA402FSA(図 7-19)にて規定されている状態遷移 1~16 が発生した時に呼び出されます。

表 7-34. CiA402 向け関数一覧 (samplecia402.c)

関数名	説明
APPL_MOTOR_MotionControl_Main	周期的に実行される CiA402_Application 関数から呼ばれるモータ制御関数。 フローチャートを図 7-28 に示します。
check_position_limit	現在位置が Limit を超えてないか確認する関数。 フローチャートを図 7-29 に示します。
int32_to_real32	int32 から real32(float)に変換する関数。
uint32_to_real32	uint32 から real32(float)に変換する関数。
rpm_to_rads	rpm から rad に変換する関数。
update_actual_value	モータの現在位置・速度を取得する関数。 フローチャートを図 7-30 に示します。

process_error_state	モータエラー発生時に呼び出される関数。 フローチャートを図 7-31 に示します。
process_profile_position_mode	pp モード時に呼び出される関数。 フローチャートを図 7-32 に示します。
process_homing_mode	hm モード時に呼び出される関数。 フローチャートを図 7-33 に示します。
process_cyclic_position_mode	csp モード時に呼び出される関数。 フローチャートを図 7-34 に示します。
process_cyclic_velocity_mode	csv モード時に呼び出される関数。 フローチャートを図 7-35 に示します。
APPL_GetDeviceID	EtherCAT ID を取得する関数。

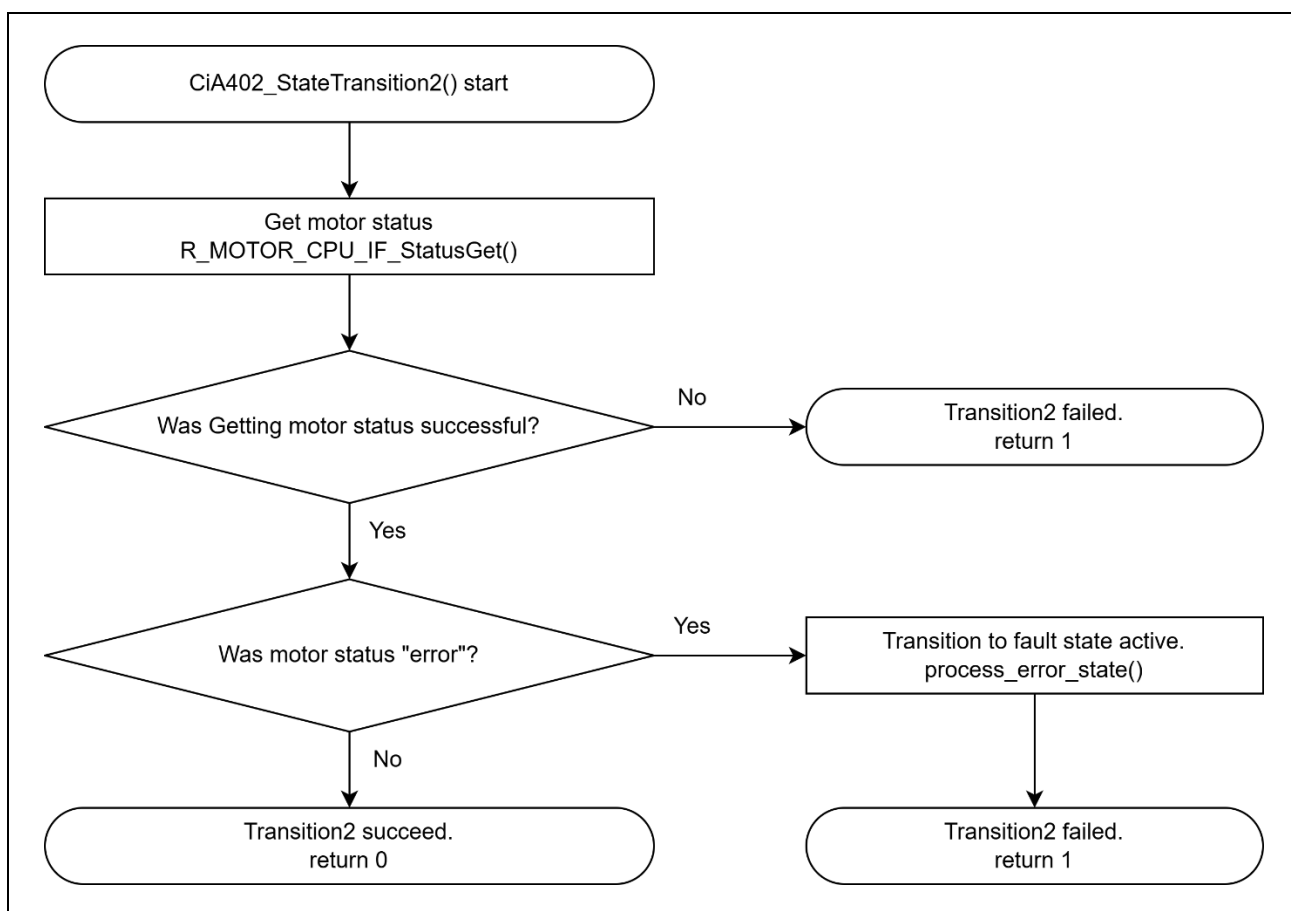


図 7-20. CiA402\_StateTransition2 関数フローチャート

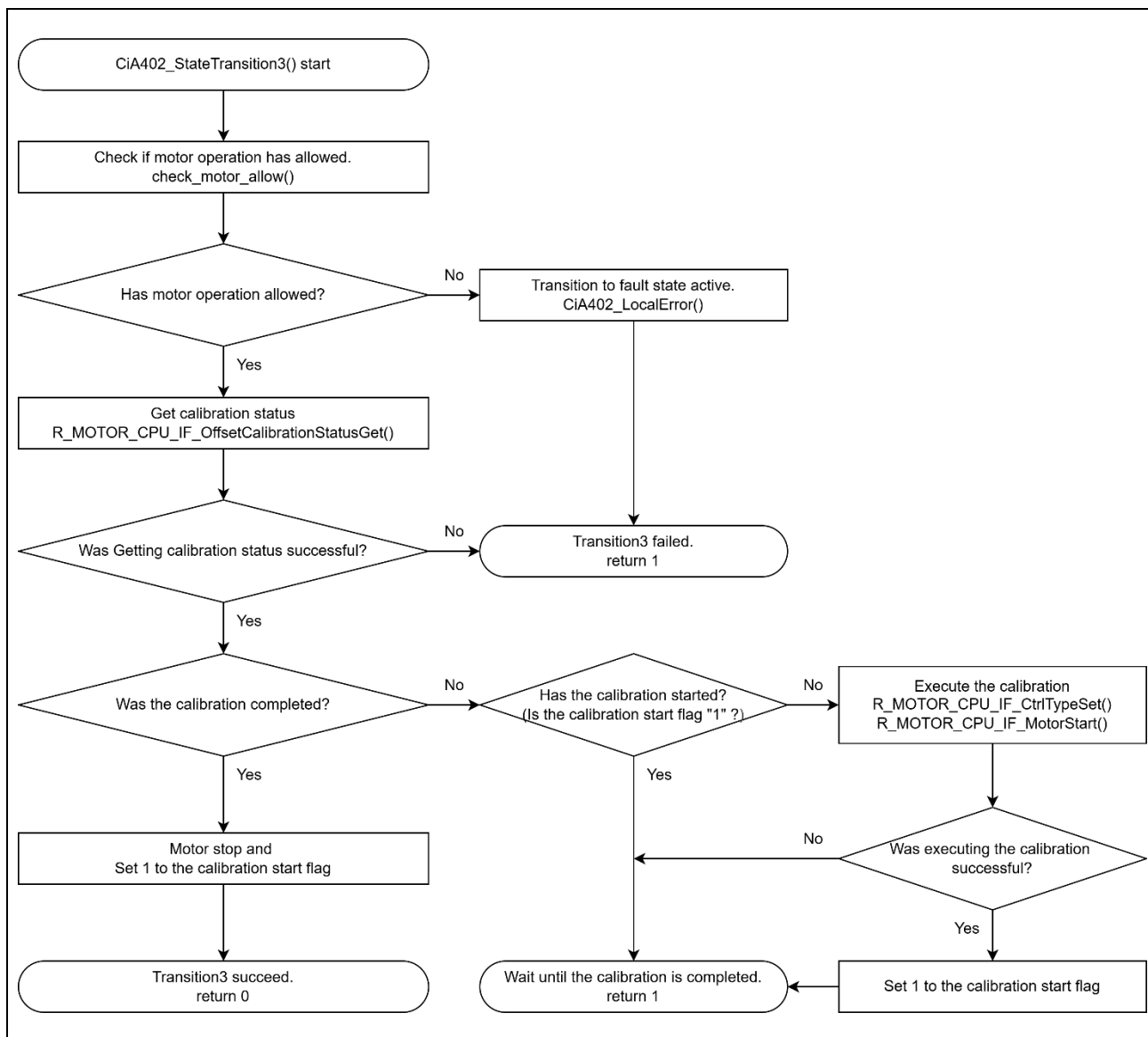


図 7-21. CiA402\_StateTransition3 関数フローチャート

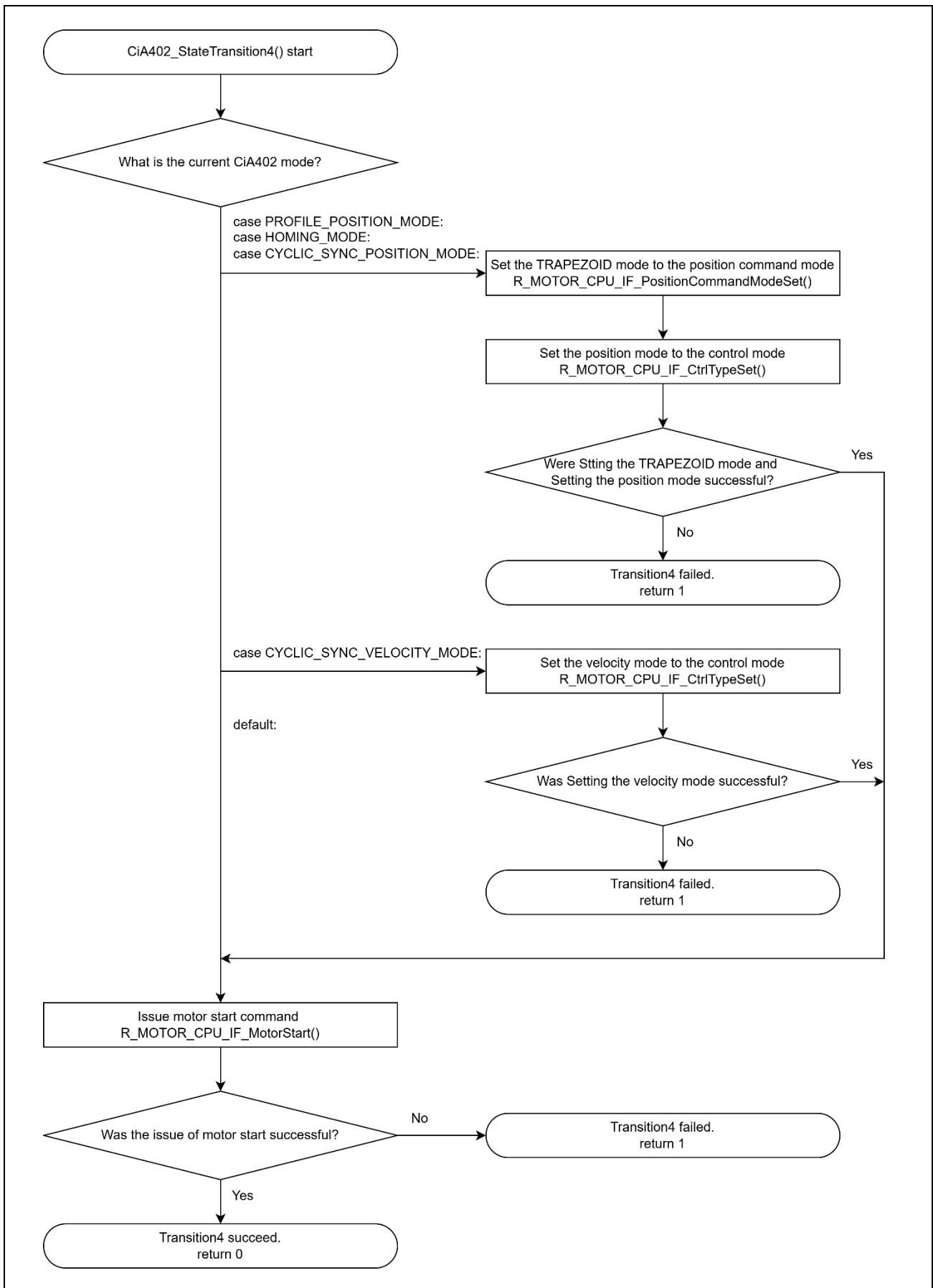


図 7-22. CiA402\_StateTransition4 関数フローチャート

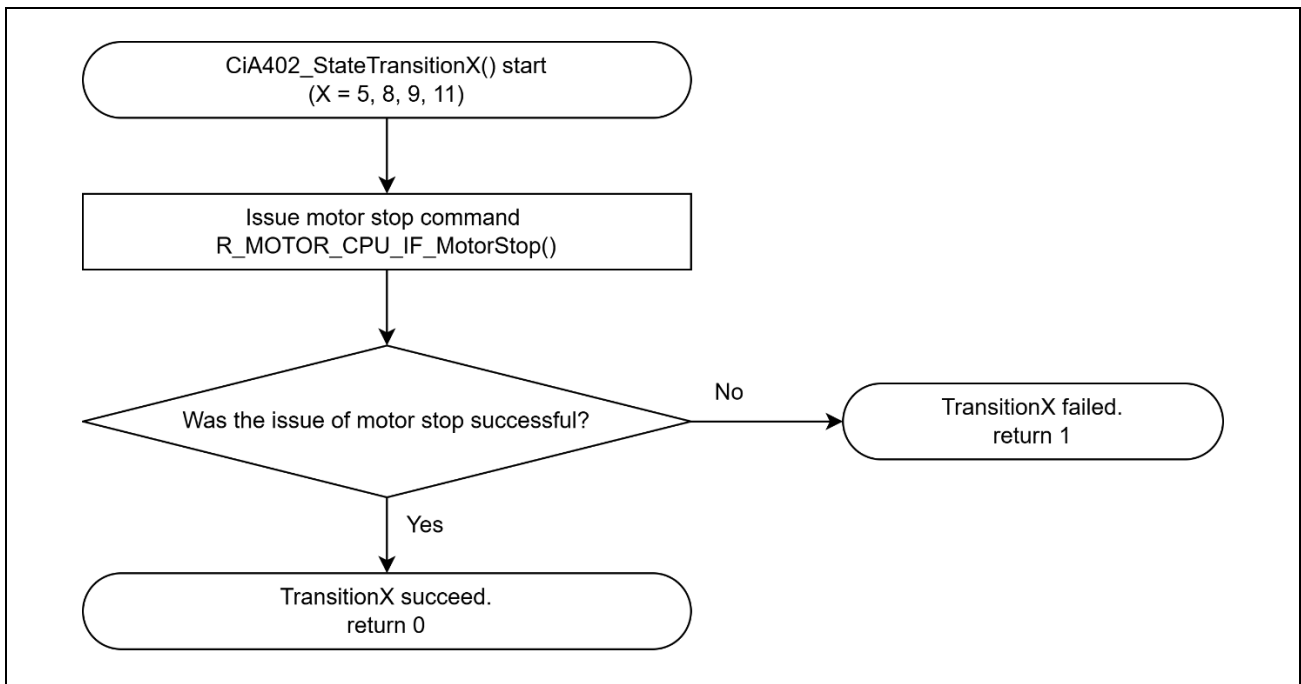


図 7-23. CiA402\_StateTransition5, 8, 9, 11 関数フローチャート

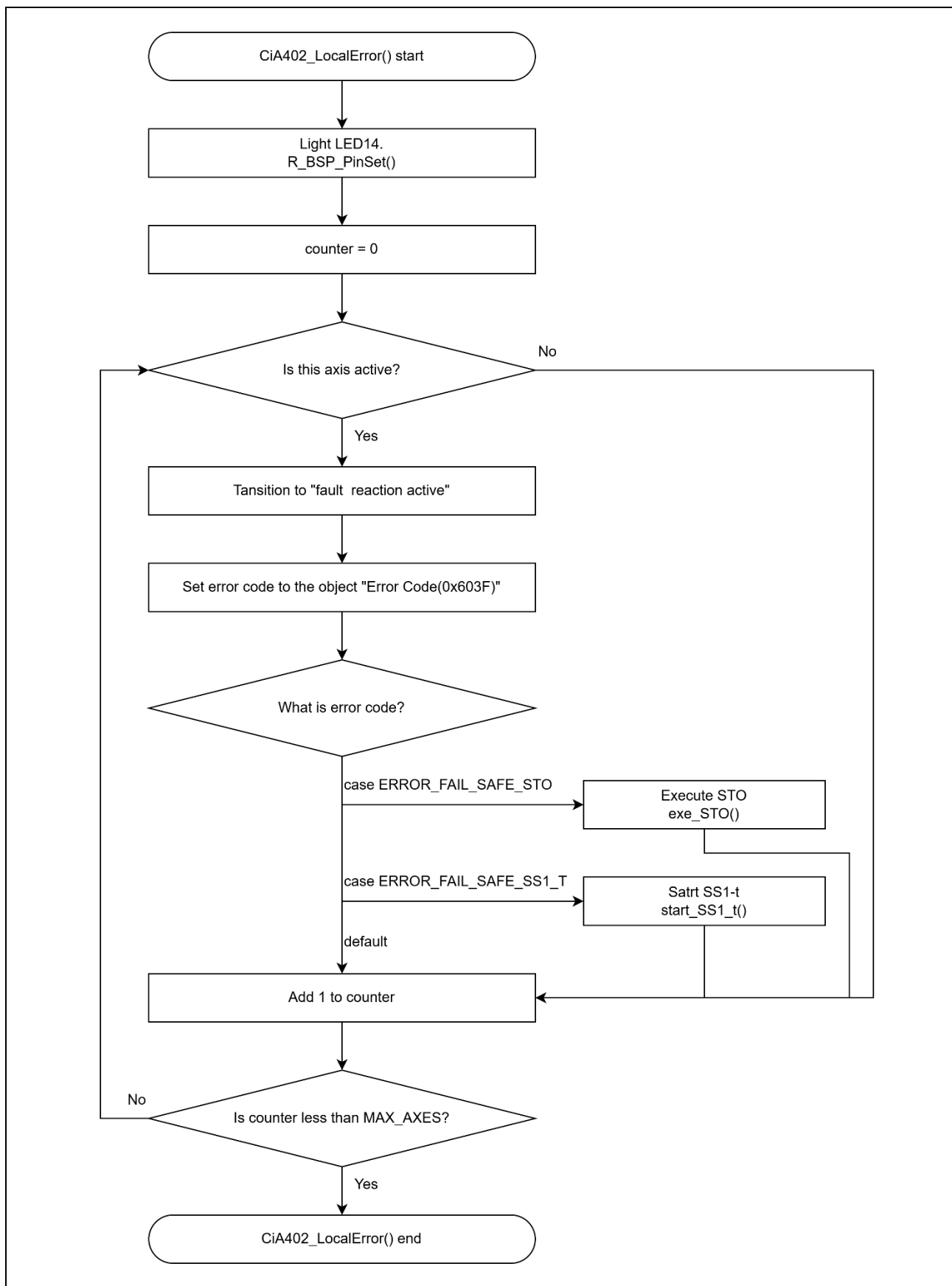


図 7-24. CiA402\_LocalError 関数フローチャート

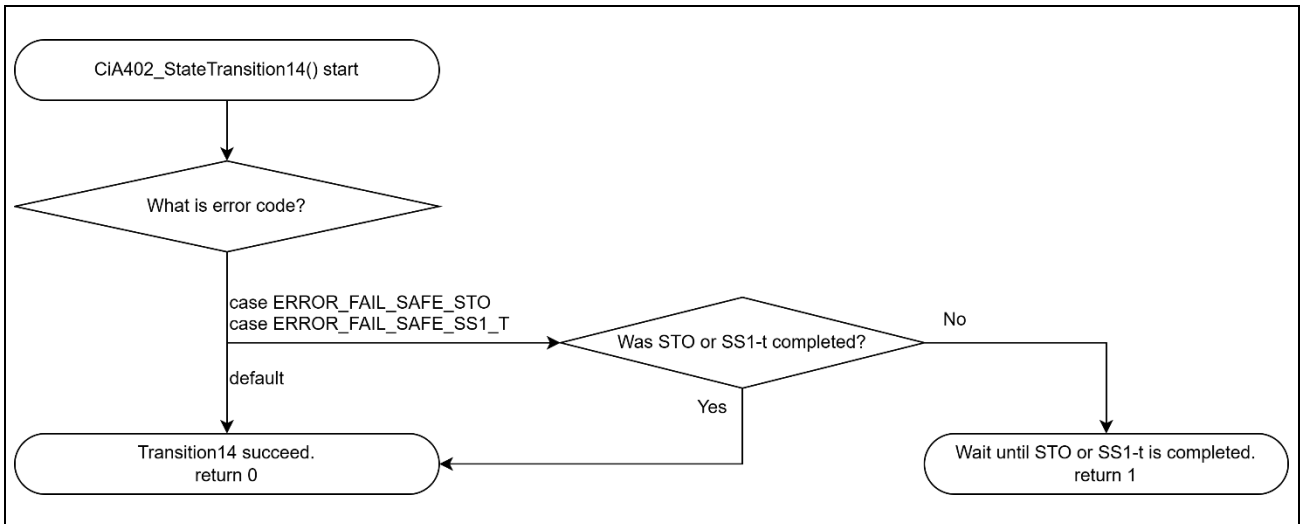


図 7-25. CiA402\_StateTransition14 関数フローチャート

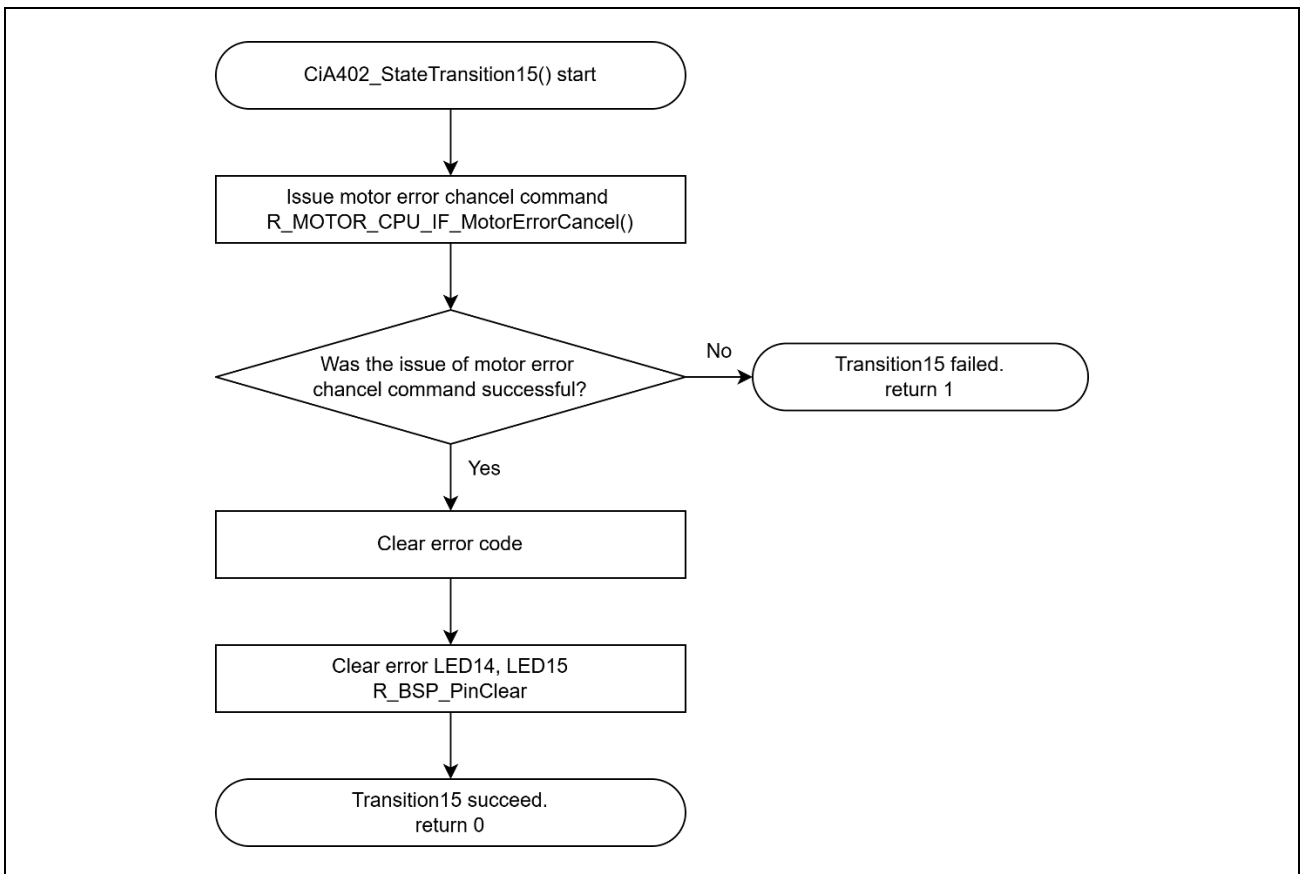


図 7-26. CiA402\_StateTransition15 関数フローチャート

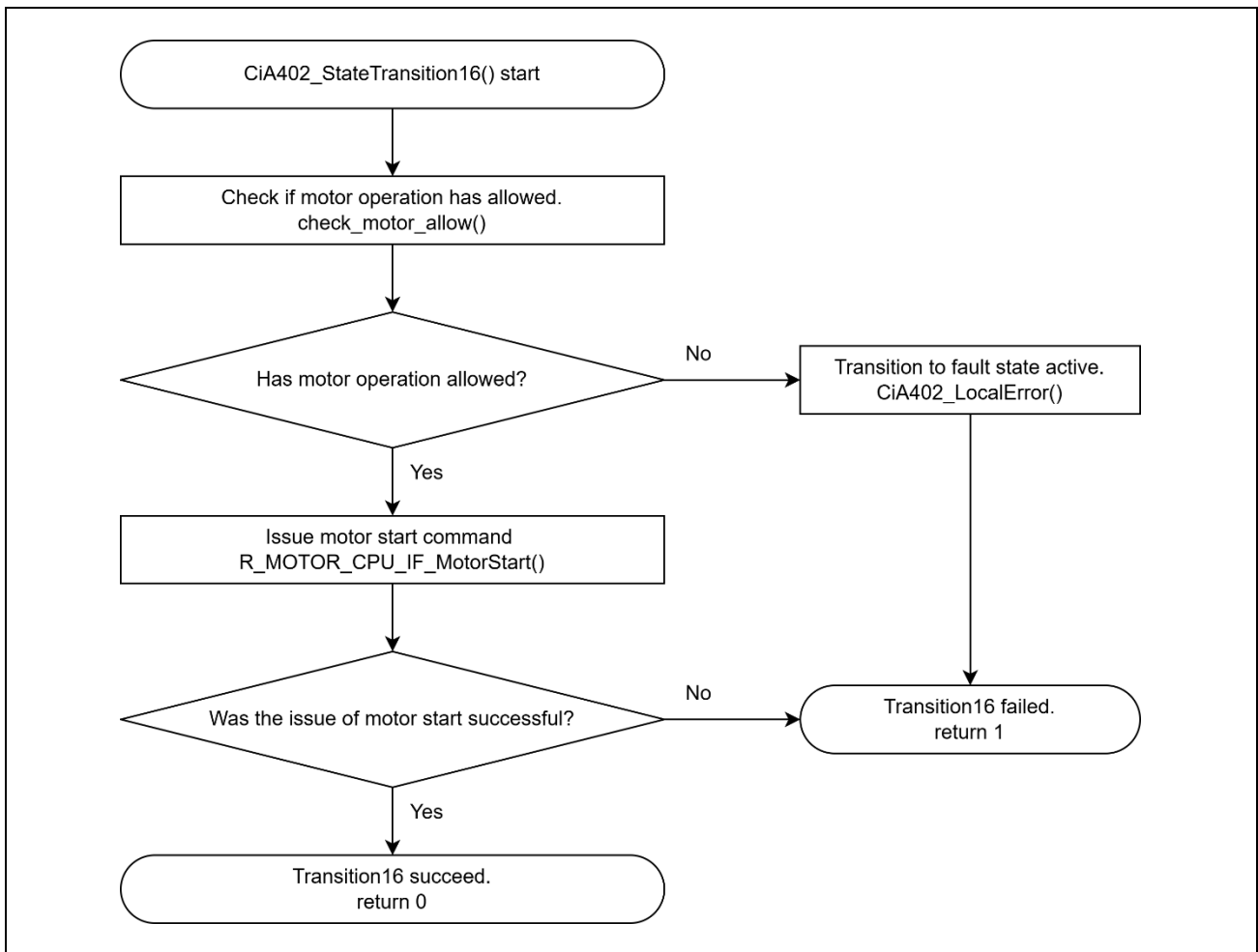


図 7-27. CiA402\_StateTransition16 関数フローチャート

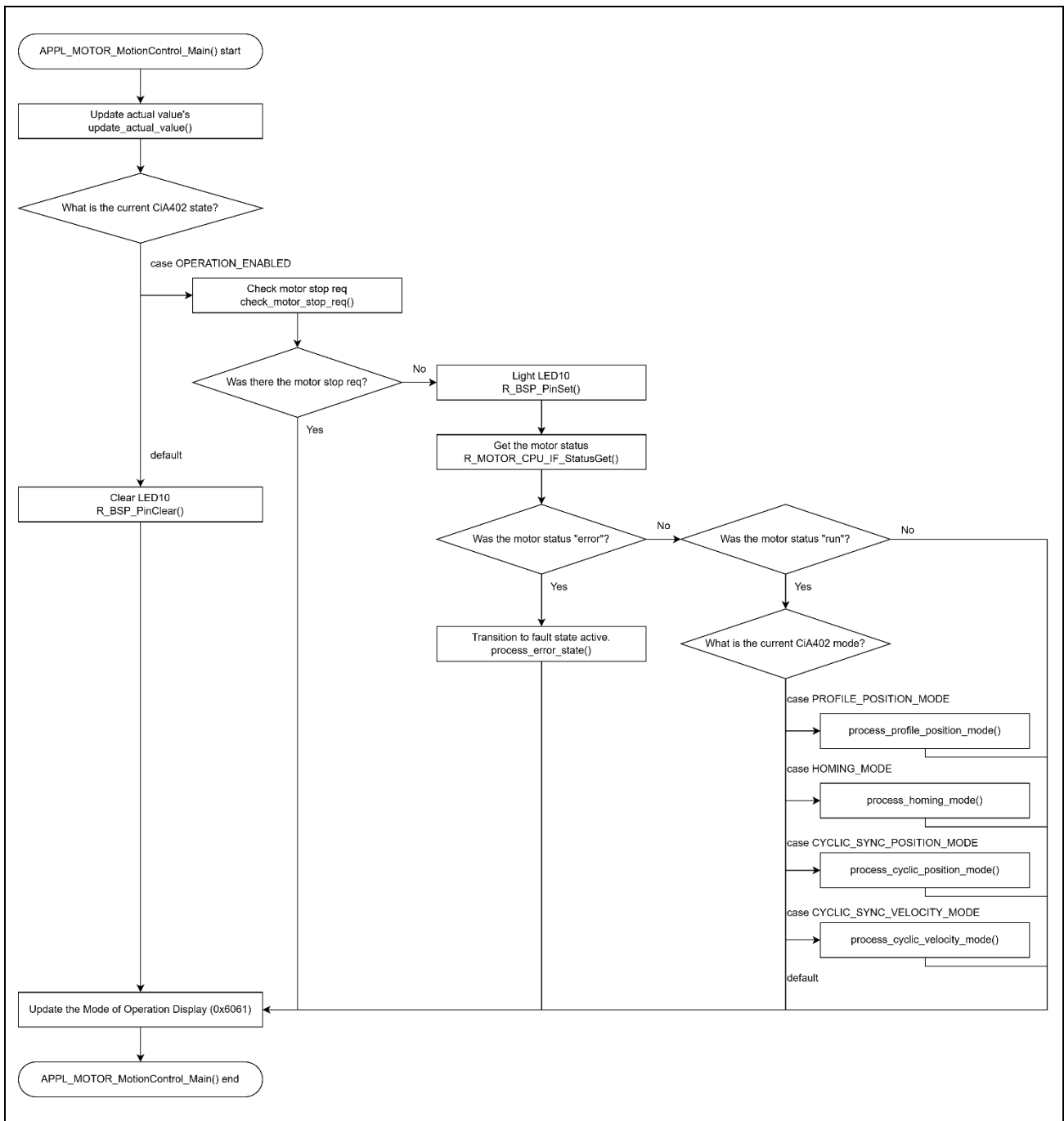


図 7-28. APPL\_MOTOR\_MotionControl\_Main 関数フローチャート

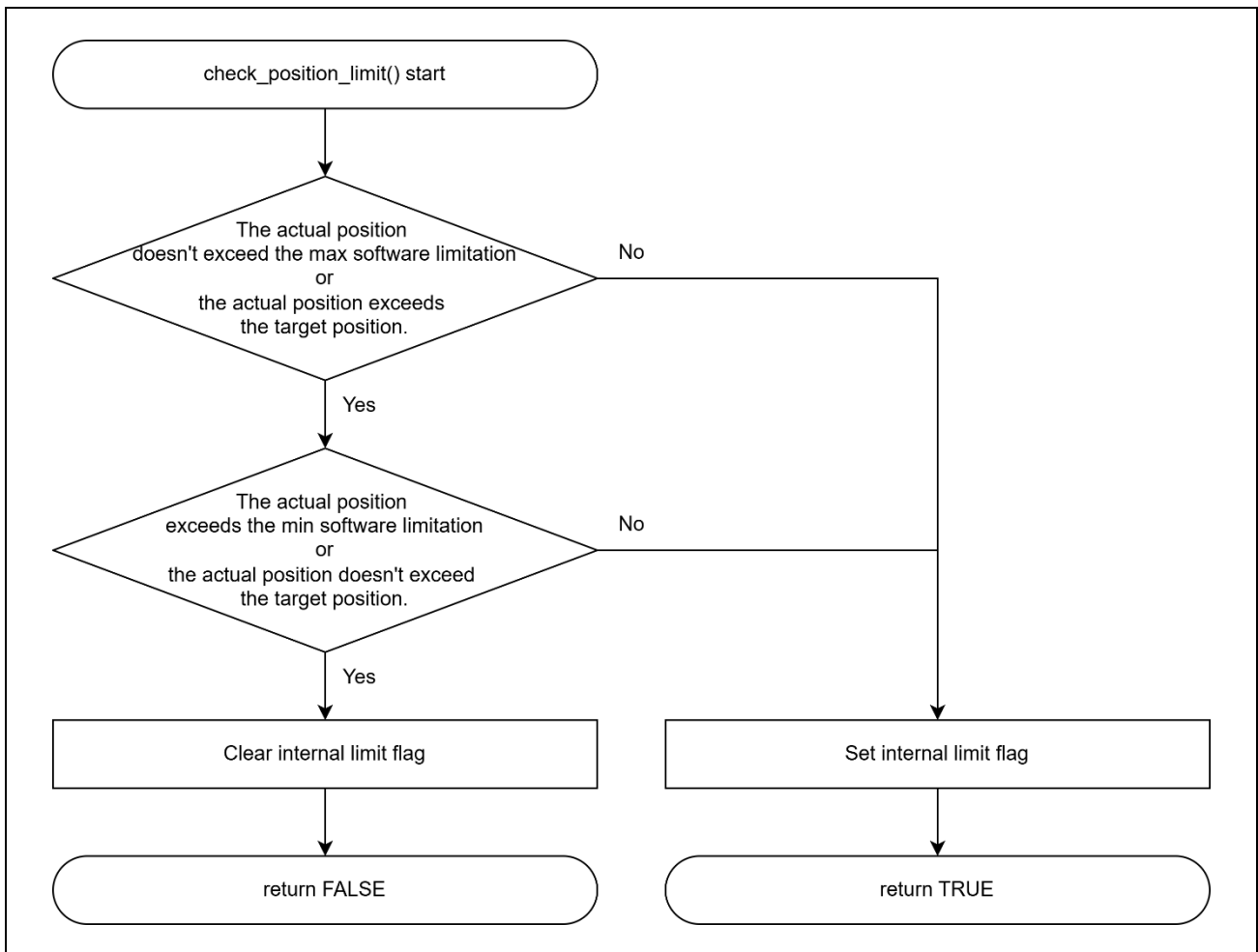


図 7-29. check\_position\_limit 関数フローチャート

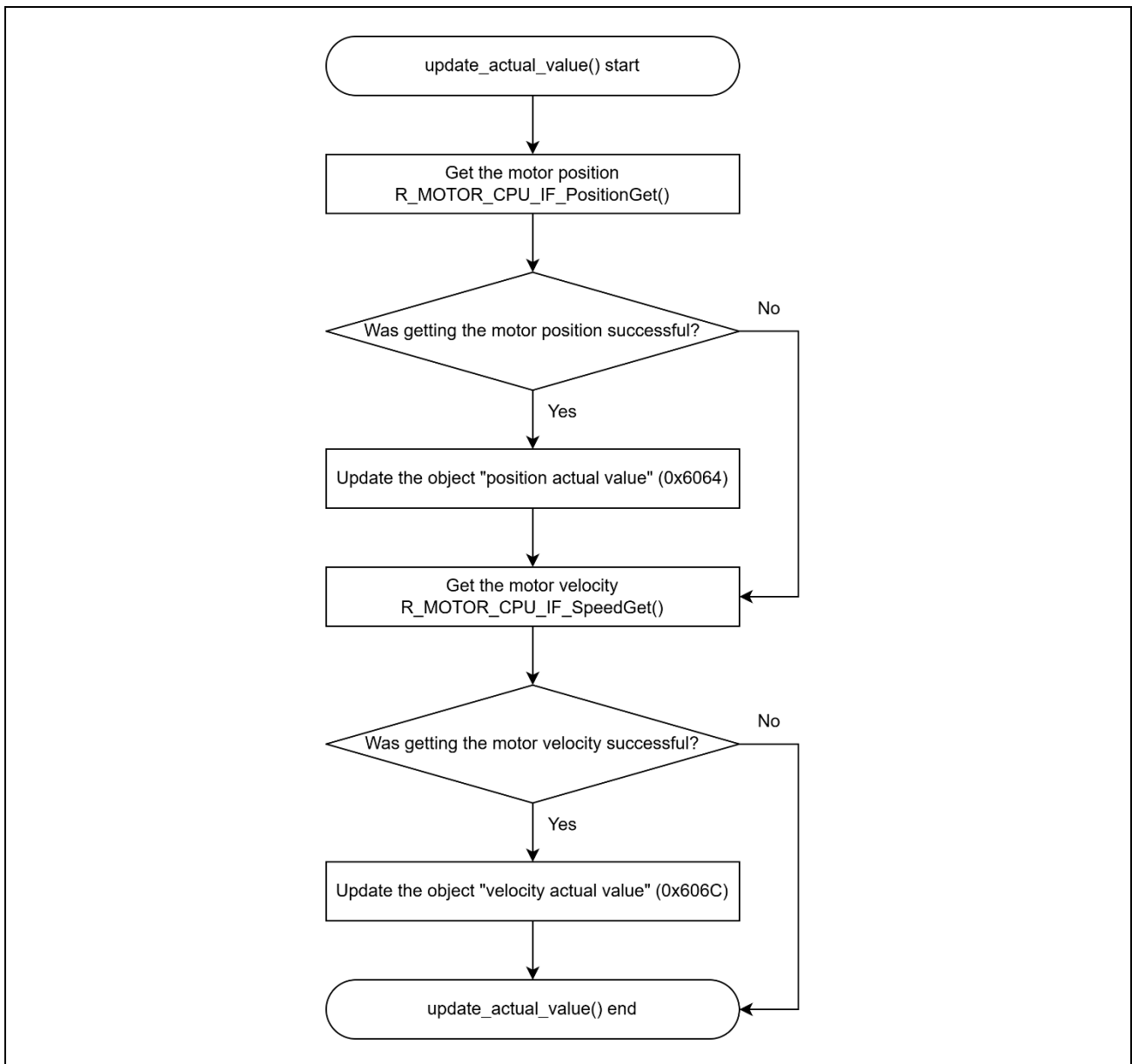


図 7-30. update\_actual\_value 関数フローチャート

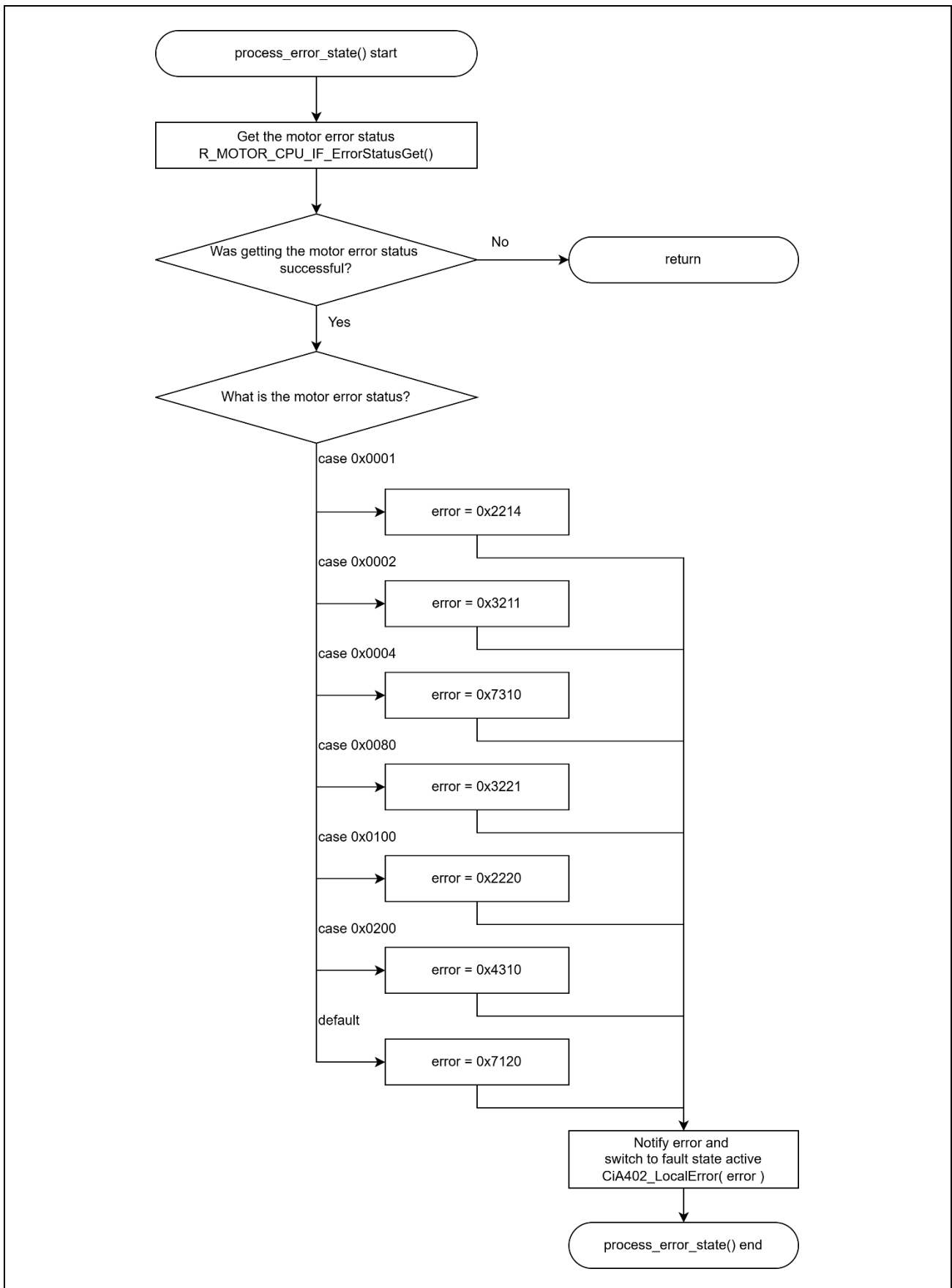


図 7-31. process\_error\_state 関数フローチャート

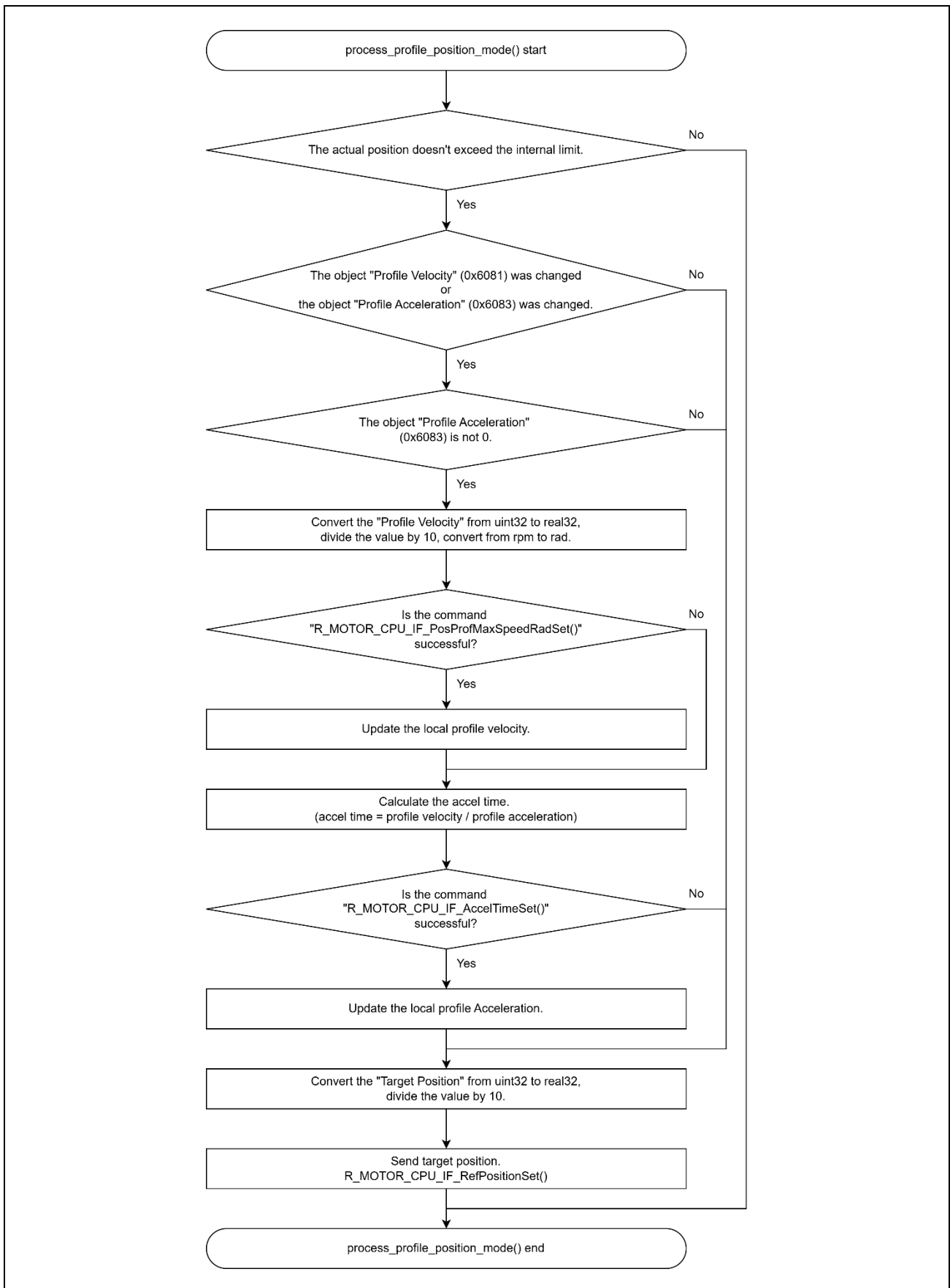


図 7-32. process\_profile\_position\_mode 関数フローチャート

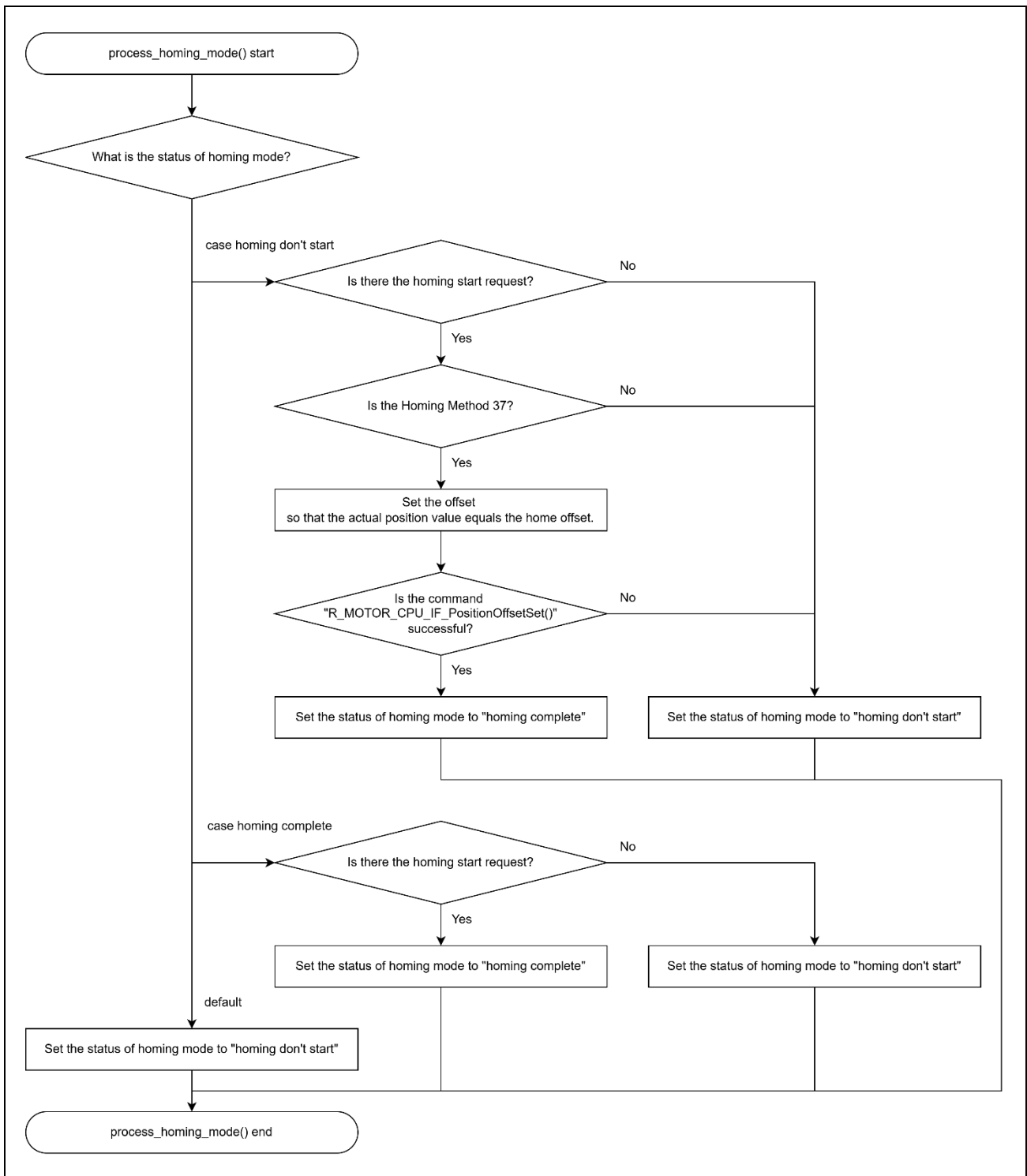


図 7-33. process\_homing\_mode 関数フローチャート

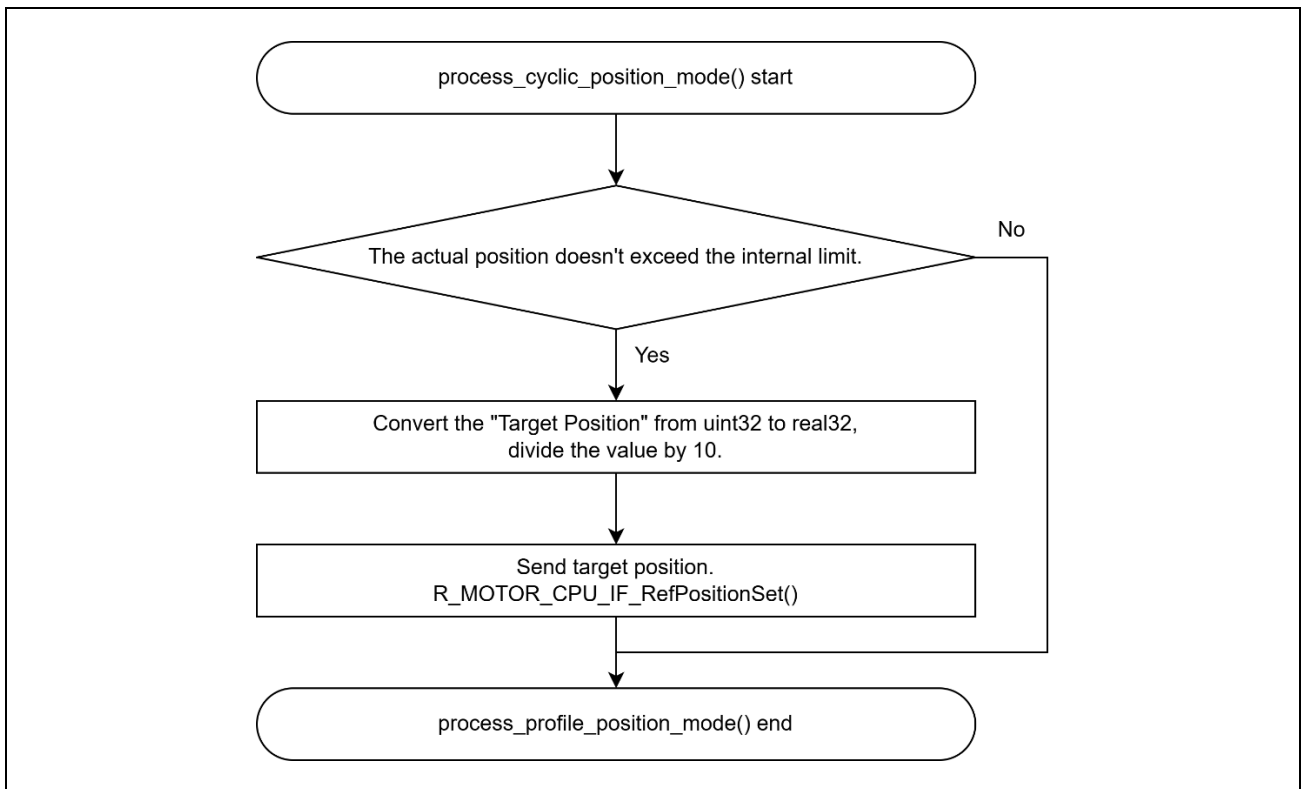


図 7-34. process\_cyclic\_position\_mode 関数フローチャート

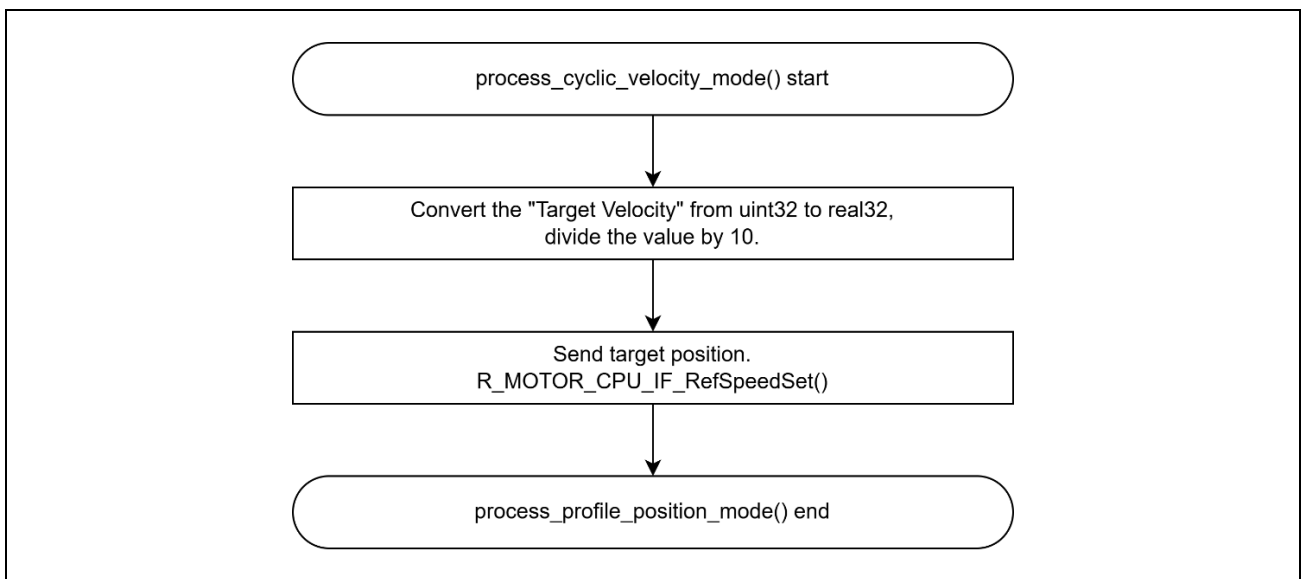


図 7-35. process\_cyclic\_velocity\_mode 関数フローチャート

## 7.3.5.9 マクロ定義

ここでは、本サンプルソフトウェアにて実装されているマクロ定義について説明します。SSC Tool が生成する EtherCAT SubDevice Stack Code に実装されているマクロ定義については、

” Application Note ET9300 (EtherCAT SubDevice Stack Code)”を参照してください。

表 7-35. CiA402 Drive Profile 向けマクロ定義一覧 (samplecia402.c)

マクロ名	定義値	備考
CFG_POS_MODE	MOTOR_CPU_IF_POS_MODE_TRAPEZOID	モータ位置制御モードの設定

## 7.3.5.10 SSC tool 生成後ファイルからの変更点

ここでは、SSC Tool が生成する EtherCAT SubDevice Stack Code から変更した内容について述べます。変更前のソースコードの詳細は、”Application Note ET9300 (EtherCAT SubDevice Stack Code)”を参照してください。

表 7-36. SSC tool 生成後ファイルからの変更点一覧

ファイル名	変更点	詳細
cia402appl.c	インクルードパスを追加	hal_data.h、samplecia402.h、r_fusa_if.h を追加
	CiA402_Init 関数の処理を変更	構造体 TCiA402Axis に追加したメンバの初期化処理を追加 追加したオブジェクトの初期化処理を追加
	CiA402_StateMachine 関数の処理を変更	CiA402 状態遷移時に表 7-33 で示される各 CiA402 遷移関数が呼ばれるように変更
	CiA402_LocalError 関数および CiA402_DummyMotionControl 関数を削除	CiA402_LocalError 関数は samplecia402.c で定義する
	CiA402_Application 関数の処理を変更	CiA402 状態遷移時に表 7-33 で示される各 CiA402 遷移関数が呼ばれるように変更 DC モードだけでなく SM モードでもモータ動作が可能ないように変更 CiA402_DummyMotionControl 関数ではなく APPL_MOTOR_MotionControl_Main 関数を呼び出すように変更
	Write0xF030 関数の処理を変更	PDO Assign および PDO mapping の変更に合わせて処理を変更
	APPL_GenerateMapping 関数の処理を変更	プロセスデータサイズを適切に計算できるように処理を変更
	APPL_InputMapping 関数の処理を変更	PDO Assign および PDO mapping の変更に合わせて処理を変更
	APPL_OutputMapping 関数の処理を変更	
APPL_Application 関数の処理を変更	FSoE_Application 関数呼び出し処理を追加	

	APPL_GetDeviceID 関数を削除	APPL_Application 関数は samplecia402.c で定義する
cia402appl.h	マクロ定義を追加	表 7-38 に追加したマクロ定義を示します。
	マクロ定義の設定値を変更	マクロ定義 MAX_AXES の設定値を 1 に変更
	オブジェクトディクショナリの定義を追加・変更	追加・変更後のオブジェクトディクショナリ一覧を表 7-39 に示します。
	構造体 TCiA402Axis にメンバを追加	追加したメンバを表 7-37 に示します。
ecatappl.c	インクルードパスを追加	samplecia402.h を追加
	extern 宣言を追加	LocalAxes 構造体の宣言を追加
	MainLoop 関数の処理を変更	1ms 毎に exe_SS1_t 関数を呼び出す処理を追加
ecatcoe.h	構造体のアラインメントを 1 に固定	-
ecatslv.c	StartInputHandler 関数の処理を変更	AL イベントマスクレジスタの 2 ビット目と 3 ビット目を 1 にする処理を削除
mailbox.h	構造体のアラインメントを 1 に固定	-
	マクロ定義の設定値を変更	マクロ定義 MBX_HEADER_SIZE の設定値を"sizeof(TMBXHEADER)"に変更
sdoserv.h	構造体のアラインメントを 1 に固定	-

表 7-37. 構造体 TCiA402Axis に追加したメンバー一覧 (cia402appl.h)

メンバ名	役割
bFSoEIsActive	FSoE モジュールアクティブフラグ
bCalibration	キャリブレーション実行フラグ
bSTOCompleted	STO 完了フラグ
bSS1tStarted	SS1-t 完了フラグ
u32LocalProfVel	以前の profile velocity 値を格納するメンバ
u32LocalProfAcc	以前の profile acceleration 値を格納するメンバ
i32SS1tStartVel	SS1-t 実行開始時のモータ速度の値を格納するメンバ
i32SS1tTime	SS1-t 実行開始時から終了までの時間を格納するメンバ
i32SS1tcnt	SS1-t 処理で使用するカウント用メンバ

表 7-38. 追加したマクロ定義一覧 (cia402appl.h)

マクロ名	定義値	備考
STATUSWORD_HOMING_MASK	0x2400	ステータスワードのホーミング用ビットマスク
STATUSWORD_HOMING_NOT_STARTED	0x0400	ホーミング処理がまだスタートしていない
STATUSWORD_HOMING_COMPLETED	0x2000	ホーミング処理が完了した
CONTROLWORD_HOMING_START	0x0010	ホーミング処理スタートビット
HOMING_METHOD_37	0x0025	Homing Method 37
ERROR_FAIL_SAFE_STO	0xFF00	STO 用エラーコード
ERROR_FAIL_SAFE_SS1_T	0xFF01	SS1-t 用エラーコード
MAX_FSOE_MODULES	MAX_AXES	FSoE モジュールの数
MAX_MODULE	MAX_AXES + MAX_FSOE_MODULES	モジュールの合計
PP_MODULE_ID	0x00419800	pp モードを示す ID
HM_MODULE_ID	0x00519800	hm モードを示す ID
FSOE_MODULE_ID	0x00619800	FSoE モジュールを示す ID
ENTRIES_1600	5	PDO 0x1600 のマッピング数
ENTRIES_1601	6	PDO 0x1601 のマッピング数
ENTRIES_1602	2	PDO 0x1602 のマッピング数
ENTRIES_1603	2	PDO 0x1603 のマッピング数
ENTRIES_1604	11	PDO 0x1604 のマッピング数
ENTRIES_1A00	5	PDO 0x1A00 のマッピング数
ENTRIES_1A01	5	PDO 0x1A01 のマッピング数
ENTRIES_1A02	5	PDO 0x1A02 のマッピング数
ENTRIES_1A03	5	PDO 0x1A03 のマッピング数
ENTRIES_1A04	11	PDO 0x1A04 のマッピング数

## 7.3.5.11 Object Dictionary

表 7-39. オブジェクトディクショナリー一覧

Index	ObjectCode	SI	DataType	Access	Object Name
<b>0x1nnn Communication Area</b>					
0x1000	VAR	-	UDINT	RO	Device type
0x1001	VAR	-	USINT	RO	Error Register
0x1008	VAR	-	STRING(43)	RO	Device Name
0x1009	VAR	-	STRING(4)	RO	Manufacturer Hardware Version
0x100A	VAR	-	STRING(4)	RO	Manufacturer Software Version
0x1018	RECORD	0	USINT	RO	Identity Object
		1	UDINT	RO	Vendor ID
		2	UDINT	RO	Product Code
		3	UDINT	RO	Revision Number
		4	UDINT	RO	Serial Number
0x10F1	RECORD	0	USINT	RO	Error Settings
		1	UDINT	RO	Local Error Reaction
		2	UINT	RW	Sync Error Counter Limit
0x10F8	VAR	-	ULINT	RO	Timestamp Object
0x1600	RECORD	0	USINT	RO	pp RxPDO
		1-5	UDINT	RO	SubIndex 00x
0x1601	RECORD	0	USINT	RO	hm RxPDO
		1-6	UDINT	RO	SubIndex 00x
0x1602	RECORD	0	USINT	RO	csp RxPDO
		1-2	UDINT	RO	SubIndex 00x
0x1603	RECORD	0	USINT	RO	csv RxPDO
		1-2	UDINT	RO	SubIndex 00x
0x1604	RECORD	0	USINT	RO	FSOE RxPDO
		1-11	UDINT	RO	SubIndex 0xx
0x1A00	RECORD	0	USINT	RO	pp TxPDO
		1-5	UDINT	RO	SubIndex 00x
0x1A01	RECORD	0	USINT	RO	hm TxPDO
		1-5	UDINT	RO	SubIndex 00x
0x1A02	RECORD	0	USINT	RO	csp TxPDO
		1-5	UDINT	RO	SubIndex 00x
0x1A03	RECORD	0	USINT	RO	csv TxPDO
		1-5	UDINT	RO	SubIndex 00x
0x1A04	RECORD	0	USINT	RO	FSOE TxPDO
		1-11	UDINT	RO	SubIndex 0xx

0x1C00	ARRAY	0	USINT	RO	Sync Manager Type
		1-4	USINT	RO	SubIndex 00x
0x1C12	ARRAY	0	USINT	RO *	RxPDO Assign
		1-2	UINT	RO *	SubIndex 00x
0x1C13	ARRAY	0	USINT	RO *	TxPDO Assign
		1-2	UINT	RO *	SubIndex 00x
0x1C32	RECORD	0	USINT	RO	SM output parameter
		1	UINT	RO *	Synchronization Type
		2	UDINT	RO	Cycle Time
		4	UINT	RO	Synchronization Types supported
		5	UDINT	RO	Minimum Cycle Time
		6	UDINT	RO	Calc and Copy Time
		8	UINT	RW	Get Cycle Time
		9	UDINT	RO	Delay Time
		10	UDINT	RW	Sync0 Cycle Time
		11	UINT	RO	SM-Event Missed
		12	UINT	RO	Cycle Time Too Small
		13	UINT	RO	Shift Time Too Short Counter
		32	BOOL	RO	Sync Error
		0x1C33	RECORD	0	USINT
1	UINT			RO *	Synchronization Type
2	UDINT			RO	Cycle Time
4	UINT			RO	Synchronization Types supported
5	UDINT			RO	Minimum Cycle Time
6	UDINT			RO	Calc and Copy Time
8	UINT			RW	Get Cycle Time
9	UDINT			RO	Delay Time
10	UDINT			RW	Sync0 Cycle Time
11	UINT			RO	SM-Event Missed
12	UINT			RO	Cycle Time Too Small
13	UINT			RO	Shift Time Too Short Counter
32	BOOL			RO	Sync Error
<b>0x6nnn CiA402 Drive Profile Area (ETG.6010)</b>					
0x603F	VAR	-	UINT	RO	Error Code
0x6040	VAR	-	UINT	RW	Control Word
0x6041	VAR	-	UINT	RO	Status Word
0x605A	VAR	-	INT	RW	Quick stop option code
0x605B	VAR	-	INT	RW	Shutdown option code
0x605C	VAR	-	INT	RW	Disable operation option code

0x605D	VAR	-	INT	RW	Halt option code
0x605E	VAR	-	INT	RW	Fault reaction option code
0x6060	VAR	-	SINT	RW	Modes of Operation
0x6061	VAR	-	SINT	RO	Modes of operation display
0x6062	VAR	-	DINT	RO	Position demand value
0x6063	VAR	-	DINT	RO	Position actual internal value
0x6064	VAR	-	DINT	RO	Position actual value
0x6065	VAR	-	UDINT	RW	Following error window
0x6066	VAR	-	UINT	RW	Following error time out
0x6067	VAR	-	UDINT	RW	Position window
0x606C	VAR	-	DINT	RO	Velocity actual value
0x6072	VAR	-	UINT	RW	Max Torque
0x6077	VAR	-	INT	RO	Torque actual value
0x607A	VAR	-	DINT	RW	Target position
0x607B	RECORD	0	USINT	RO	Position range limit
		1	DINT	RW	Min position range limit
		2	DINT	RW	Max position range limit
0x607C	VAR	-	DINT	RW	Home Offset
0x607D	RECORD	0	USINT	RO	Software position limit
		1	DINT	RW	Min position limit
		2	DINT	RW	Max position limit
0x607F	VAR	-	UDINT	RW	Max profile velocity
0x6080	VAR	-	UDINT	RW	Max motor speed
0x6081	VAR	-	UDINT	RW	Profile velocity
0x6083	VAR	-	UDINT	RW	Profile acceleration
0x6084	VAR	-	UDINT	RW	Profile deceleration
0x6085	VAR	-	UDINT	RW	Quick stop deceleration
0x6091	RECORD	0	UDINT	RW	Gear ratio
		1	UDINT	RW	Motor revolutions
		2	UDINT	RW	Shaft revolutions
0x6099	RECORD	0	USINT	RO	Homing speeds
		1	UDINT	RW	Speed during search for switch
		2	UDINT	RW	Speed during search for zero
0x609A	VAR	-	UDINT	RW	Homing acceleration
0x60B0	VAR	-	DINT	RW	Position offset
0x60B1	VAR	-	DINT	RW	Velocity offset
0x60B2	VAR	-	INT	RW	Torque offset
0x60B8	VAR	-	UINT	RW	Touch probe function
0x60B9	VAR	-	UINT	RO	Touch probe status

0x60BA	VAR	-	DINT	RO	Touch probe position 1 positive value
0x60BB	VAR	-	DINT	RO	Touch probe position 1 negative value
0x60C2	RECORD	0	USINT	RO	Interpolation time period
		1	USINT	RW	Interpolation time period value
		2	SINT	RW	Interpolation time index
0x60D0	RECORD	0	USINT	RO	Touch probe source
		1	INT	RW	Touch probe 1 source
0x60E0	VAR	-	UINT	RW	Positive torque limit value
0x60E1	VAR	-	UINT	RW	Negative torque limit value
0x60F4	VAR	-	DINT	RO	Following error actual value
0x60FD	VAR	-	UDINT	RO	Digital inputs
0x60FE	RECORD	0	USINT	RO	Digital outputs
		1	UDINT	RW	Physical outputs
		2	UDINT	RW	Bit mask
0x60FF	VAR	-	DINT	RW	Target velocity
0x6402	VAR	-	UINT	RW	Motor Type
0x6502	VAR	-	UDINT	RO	Supported drive modes
<b>0x67nn FSoE Safety Drive Connection (ETG.6100)</b>					
0x6760	RECOED	0	USINT	RO	FSOE Slave Frame Elements
		1	USINT	RO	Command
		2	GAP	None	SubIndex 002
		3	UINT	RO	Connection ID
		4	UINT	RO	CRC0
0x6761	RECOED	0	USINT	RO	FSOE Inputs
		1-8	BOOL	RO	SubIndex 00x
0x6770	RECOED	0	USINT	RO	FSOE Main Frame Elements
		1	USINT	RO	Command
		2	GAP	None	SubIndex 002
		3	UINT	RO	Connection ID
		4	UINT	RO	CRC0
0x6771	RECOED	0	USINT	RO	FSOE Outputs
		1-8	BOOL	RW	SubIndex 00x
0x6780	RECOED	0	USINT	RO	Safety Parameter
		1	USINT	RW	Sensor Test Channel 1 active
		2	USINT	RW	Sensor Test Channel 2 active
0x6790	RECOED	0	USINT	RO	FSOE Info Data
		1	UINT	RO	FSOE Connection ID
		2	UINT	RO	Device Type
		3	UDINT	RO	Vendor ID

		4	UDINT	RO	Product Code
		5	UDINT	RO	Revision Number
		6	UDINT	RO	Serial Number
		7	UINT	RO	Network Flags
		8	UINT	RO	Network Port
		9	UINT	RO	Network Segment Address
0x6791	RECOED	0	USINT	RO	FSoE Communication Parameter
		1	STRING(2)	RO	Version
		2	UINT	RO	Safe Address
		3	UINT	RO	Connection
		4	UINT	RO	Watchdog Time
		5	ARRAY [0..5] OF BYTE	RO	Unique ID
		6	UINT	RO	Connection Type
		7	UINT	RO	Communication Parameter Length
		8	UINT	RO	Application Parameter Length
0x67A0	RECOED	0	USINT	RO	FSoE Connection Diagnosis
		1	UINT	RO	Connection State
		2	UINT	RO	Connection Diagnosis
0xF000	RECOED	0	USINT	RO	Modular Device Profile
		1	UINT	RO	Index distance
		2	UINT	RO	Maximum number of modules
0xF010	ARRAY	0	USINT	RO	Module List
		1-2	UDINT	RO	SubIndex 00x
0xF030	ARRAY	0	USINT	RO	Configured Module Ident List
		1-2	UDINT	RO	SubIndex 00x
0xF050	ARRAY	0	USINT	RO	Detected modules
		1-2	UDINT	RO	SubIndex 00x
0xF980	RECOED	0	USINT	RO	Safe Address
		1	UINT	RO	FSoE Address

\* This is RW only if its status is PreOP.

7.3.6 Safety Drive

本ソフトウェアでは、システムの異常検出により RZ/T2M が制御するモータを停止する機能を実装しています。モータ停止機能は IEC61800-5-2 の STO, SS1-t に対応しています。

この機能で使用する I/O ポートについて表 7-40 に示します。

表 7-40. Safety Drive 機能用 I/O Port 一覧

I/O Port	説明
P14_2	このポートは RZ/T2L-A によって制御されます。 “L”の場合、モータ動作禁止です。“H”の場合、モータ動作可能です。 “H”から“L”に遷移した場合、モータ停止要求があったと見なし、P14_3 のレベルに応じて STO もしくは SS1-t を実行します。
P14_3	モータ停止機能を選択する I/O ポートです。 このポートは RZ/T2L-A によって制御されます。 “L”の場合、モータ停止機能は STO です。 “H”の場合、モータ停止機能は SS1-t です。

モータ動作開始から STO / SS1-t 発生、復帰までの流れを図 7-36、図 7-37 に示します。

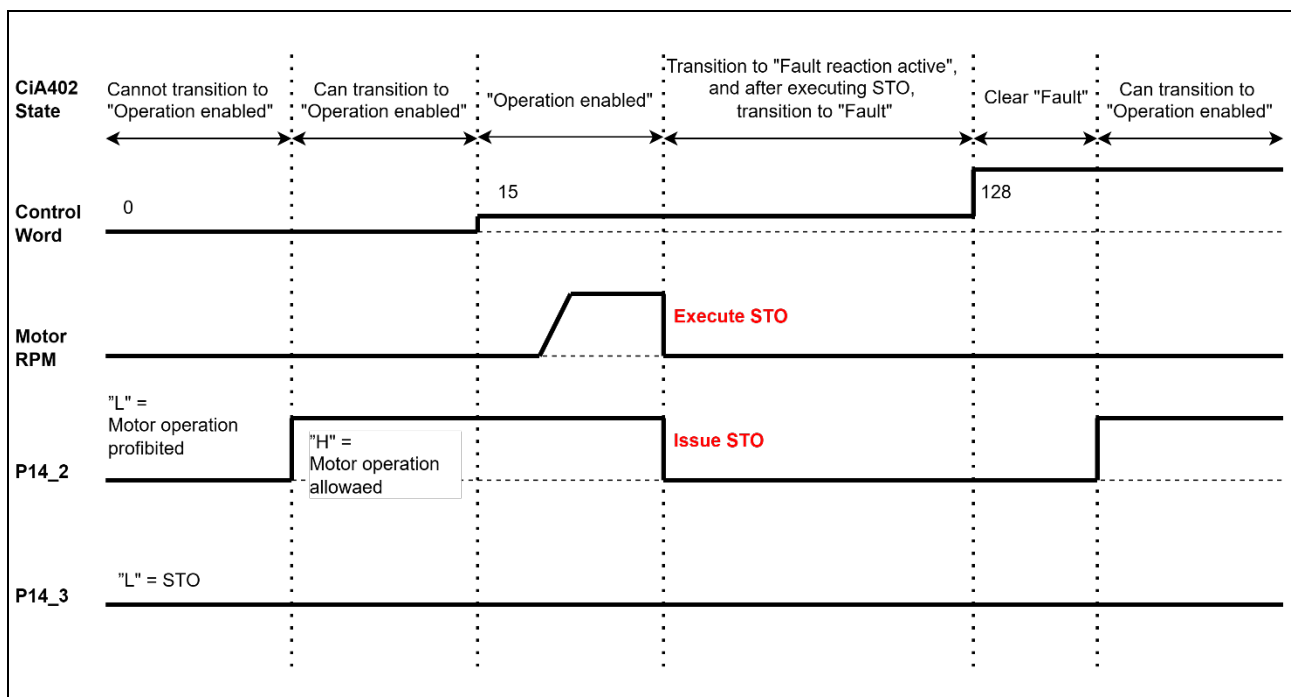


図 7-36. モータ動作・STO・復帰の流れ

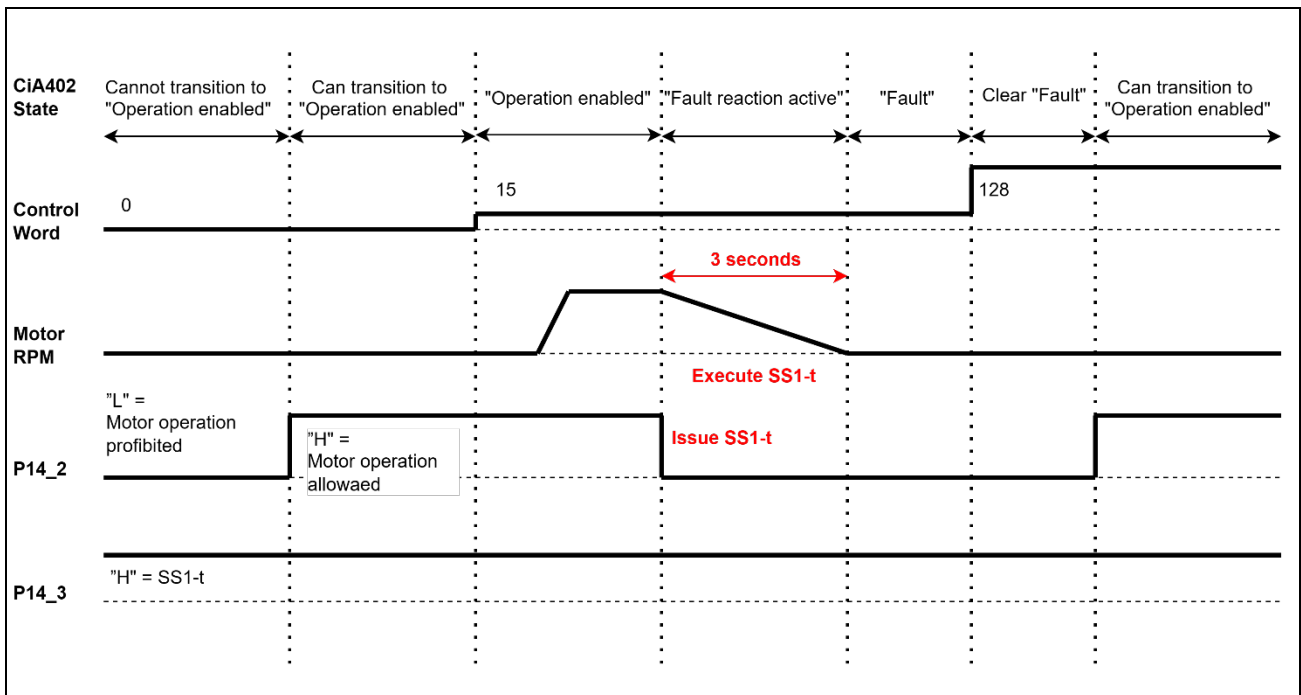


図 7-37. モータ動作・SS1-t・復帰の流れ

## 7.3.6.1 Safety Drive 向け関数

表 7-41. Safety Drive 向け関数一覧 (samplecia402.c)

関数名	説明
check_motor_allow	P14_2 のレベルを読み取り、モータ動作可能か確認する関数。
check_motor_stop_req	モータ動作中にモータ停止要求があるかを確認する関数。 モータ停止要求があった場合、CiA402_LocalError 関数を呼び出して"Fault reaction active"に遷移し、STO もしくは SS1-t 処理を開始します。
exe_STO	STO を実行する関数。 モータ制御 CPU にモータストップコマンドを送信し、モータを停止します。 また、STO 実行完了を示すために LED15 を点灯させ、構造体 TCiA402Axis の STO 完了フラグに TRUE をセットします。
start_SS1_t	SS1-t 処理を開始する関数。 構造体 TCiA402Axis の各 SS1-t 用メンバの設定を行います。 また、SS1-t 開始フラグに TRUE をセットします。
exe_SS1_t	SS1-t 処理を行う関数です。 構造体 TCiA402Axis の SS1-t 開始フラグに TRUE がセットされていると MainLoop 関数から 1ms 毎に呼ばれます。 この関数では、SS1-t が実行中であることを示すために 100ms 毎に LED15 を点滅させます。 また、1ms 毎に速度を計算してモータ制御 CPU へ送信する速度指定を更新します。 SS1-t 処理が終了した場合、exe_STO 関数を呼び出します。

## 7.3.6.2 マクロ定義

表 7-42. Safety Drive 向けマクロ定義一覧 (samplecia402.c)

マクロ名	定義値	備考
CFG_SS1T_TIME	3000	SS1-t 実行開始から終了までの時間[ms]を設定します。

### 7.3.7 Fusa CPU Communication

本モジュールでは、FSoE Master から受信した Safety Master PDU を機能安全 CPU(RZ/T2L-A)へ送信します。また、機能安全 CPU から受信した Safety Slave PDU を FSoE Master へ送信します。

機能安全 CPU との通信には、Fusa I/F モジュールを使用します。Fusa I/F モジュールの詳細については、7.3.9 Fusa I/F を参照して下さい。

機能安全 CPU の仕様および通信の仕様については『RZ ファミリ FSoE アプリケーションソフトウェア ソフトウェア開発ハンドブック (R30UZ0176JJ0110)』を参照してください。

#### 7.3.7.1 FSoE PDO 設定

表 7-43. FSoE RxPDO 一覧

RxPDO	Index	Size[bit]
FSoE output Command	0x6770:1	8
FSoE output data 1	0x6771:1	1
FSoE output data 2	0x6771:2	1
FSoE output data 3	0x6771:3	1
FSoE output data 4	0x6771:4	1
FSoE output data 5	0x6771:5	1
FSoE output data 6	0x6771:6	1
FSoE output data 7	0x6771:7	1
FSoE output data 8	0x6771:8	1
FSoE output CRC0	0x6770:4	16
FSoE output ConnectionID	0x6770:3	16

表 7-44. FSoE TxPDO 一覧

TxPDO	Index	Size[bit]
FSoE input Command	0x6760:1	8
FSoE input data 1	0x6761:1	1
FSoE input data 2	0x6761:2	1
FSoE input data 3	0x6761:3	1
FSoE input data 4	0x6761:4	1
FSoE input data 5	0x6761:5	1
FSoE input data 6	0x6761:6	1
FSoE input data 7	0x6761:7	1
FSoE input data 8	0x6761:8	1
FSoE input CRC0	0x6760:4	16
FSoE input ConnectionID	0x6760:3	16

## 7.3.7.2 機能安全 CPU との送受信に使用されるオブジェクト

表 7-45. 機能安全 CPU への送信に使用されるオブジェクト一覧

パケット名	オブジェクト	Index	Size[byte]
Safety Master PDU	FSoE output Command	0x6770:1	1
	FSoE output data 1 ~ 8	0x6771:1 ~ 8	1
	FSoE output CRC0	0x6770:4	2
	FSoE output ConnectionID	0x6770:3	2

表 7-46. 機能安全 CPU からの受信に使用されるオブジェクト一覧

パケット名	オブジェクト	Index	Size[byte]
Safety Slave PDU	FSoE input Command	0x6760:1	1
	FSoE input data 1 ~ 8	0x6761:1 ~ 8	1
	FSoE input CRC0	0x6760:4	2
	FSoE input ConnectionID	0x6760:3	2
Communication parameter	Connection ID	0x6791:3	2
	Connection State	0x67A0:1	2
	Connection Diagnosis	0x67A0:2	2
	Watchdog Time	0x6791:4	2
	Communication Parameter Length	0x6791:7	2
	Application Parameter Length	0x6791:8	2

7.3.7.3 関数

表 7-47. 関数一覧 (samplecia402.c)

関数名	説明
FSoE_Application	周期的に実行される APPL_Application 関数から呼ばれます。 フローチャートを図 7-38 に示します。

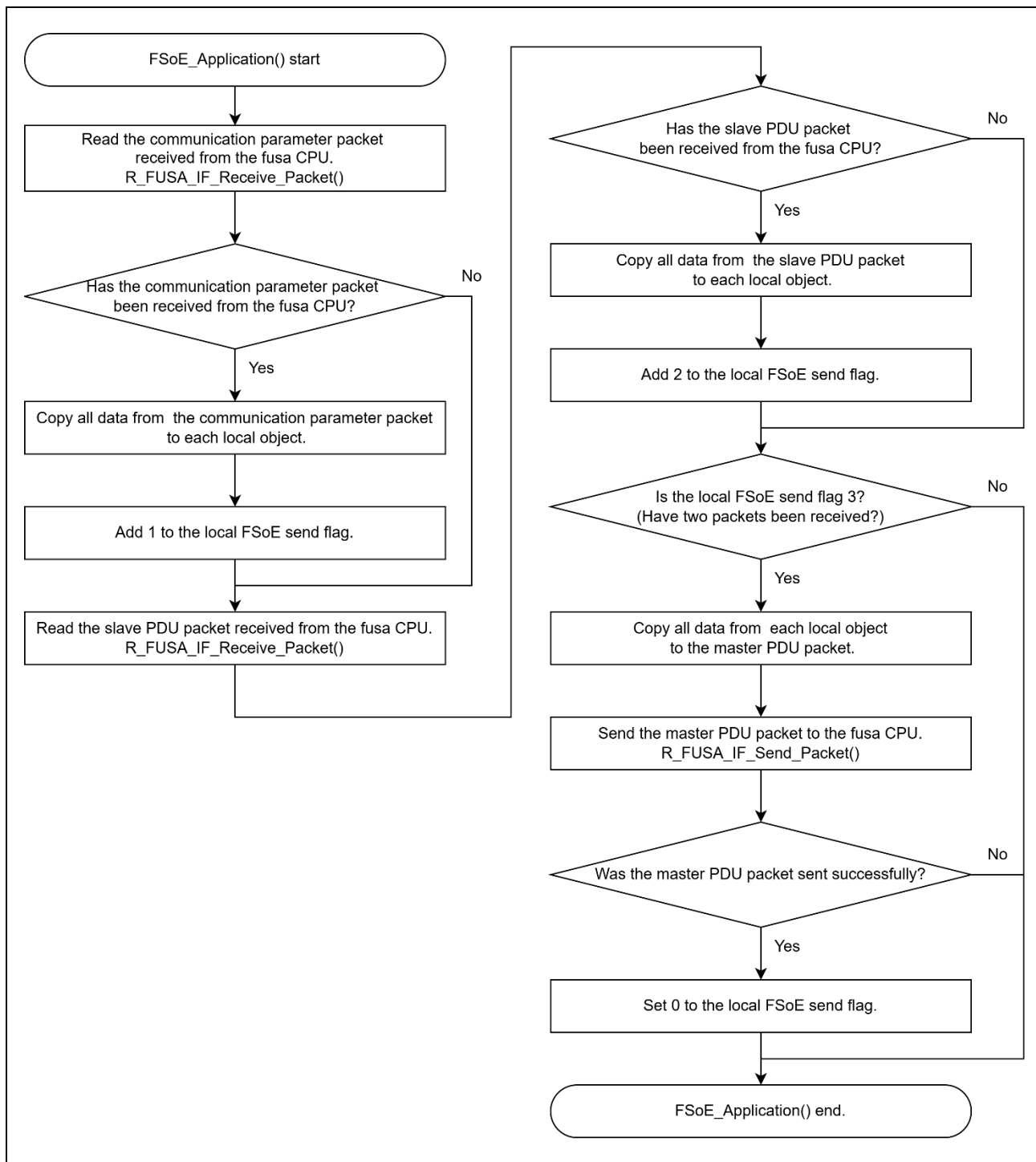


図 7-38. FSoE\_Application 関数フローチャート

## 7.3.7.4 構造体・グローバル変数情報

表 7-48. 構造体一覧 (r\_samplecia402.h)

構造体	変数	
fsoe_params_t Communication parameter packet 構造体	connection_id	FSoE_Appl
	connection_state	
	connection_diagnosis	
	watchdog_time	
	com_parameter_length	
	appl_parameter_length	

表 7-49. グローバル変数一覧 (r\_samplecia402.c)

型名	インスタンス名	説明
UINT8	u8FSOEsendFlag	FSoE 送信フラグ。 FSoE_Application 関数内で使用されます。 初期値は 3 です。

### 7.3.8 Motor CPU I/F

Motor CPU I/F モジュールは、Shared Memory Driver を使用して Motor CPU (RZ/T2M CPU0)とモータ制御コマンドの送信・指令値の更新・モータステータスの取得を行うモジュールです。

#### 7.3.8.1 モジュール構成

モジュール構成図を図 7-39 に示します。

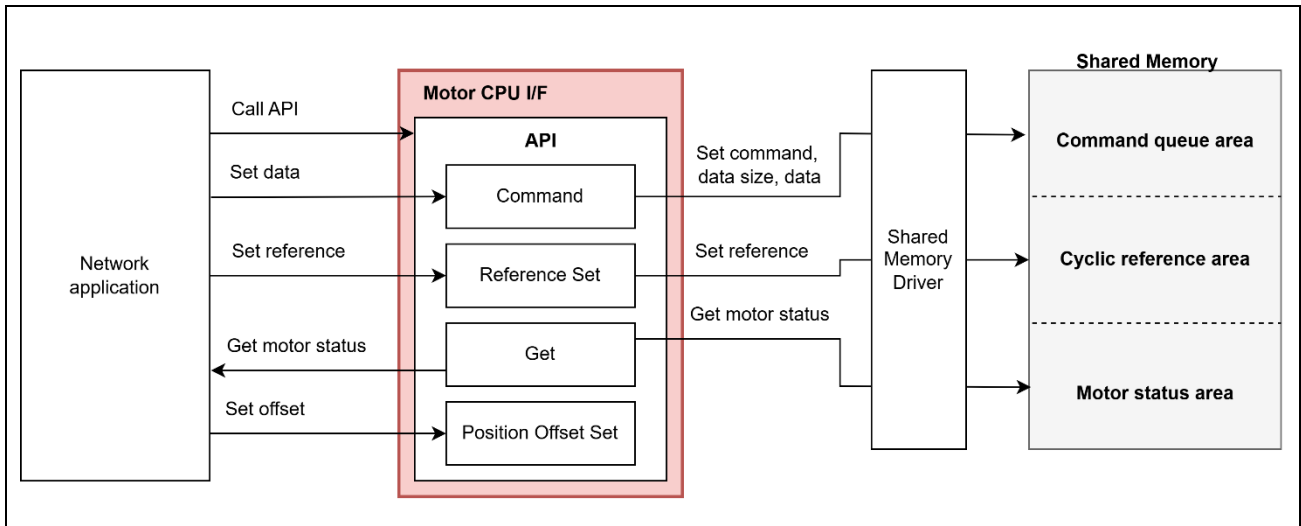


図 7-39. Motor CPU I/F モジュール構成図

本モジュールでは主に 4 種類の API を提供します。

#### (1) Command Set

この API は、Shared Memory の Command queue area にモータ制御コマンド・データサイズ・データを格納します。

#### (2) Reference Set

この API は、Shared Memory の Cyclic reference area に設定されている各種指令値を更新します。

#### (3) Get Motor Status

この API は、Shared Memory の Motor status area に格納されている各種モータステータスを取得します。

#### (4) Position Offset Set

この API は、Home 位置の設定を目的として用意されている API です。位置情報にオフセットを設定します。

#### 7.3.8.2 Shared Memory 割り当て

7.2.3.2 章に記載されている表と同様です。

## 7.3.8.3 API

表 7-50. Motor CPU I/F API 一覧

API	説明
R_MOTOR_CPU_IF_Open	Motor CPU I/F の初期化を行います。 また、Shared Memory Driver の初期化を行います。 フローチャート：図 7-40
R_MOTOR_CPU_IF_Close	Motor CPU I/F を終了します。 また、Shared Memory Driver を終了します。 フローチャート：図 7-41
R_MOTOR_CPU_IF_Reset R_MOTOR_CPU_IF_MotorStart R_MOTOR_CPU_IF_MotorStop R_MOTOR_CPU_IF_MotorReset R_MOTOR_CPU_IF_MotorErrorCancel R_MOTOR_CPU_IF_ErrorSet R_MOTOR_CPU_IF_CtrlTypeSet R_MOTOR_CPU_IF_PositionCommandModeSet R_MOTOR_CPU_IF_RotorAngleOffsetSet R_MOTOR_CPU_IF_CurrentOffsetSet R_MOTOR_CPU_IF_CurrentOpenLoopSet R_MOTOR_CPU_IF_AccelTimeSet R_MOTOR_CPU_IF_PosProfMaxSpeedRadSet	Command Set API です。 Shared Memory の Command queue area にモータ制御コマンド・データサイズ・データを格納します。 フローチャート：図 7-42
R_MOTOR_CPU_IF_RefPositionSet R_MOTOR_CPU_IF_RefSpeedSet R_MOTOR_CPU_IF_RefTorqueSet	Reference Set API です。 Shared Memory の Cyclic reference area に設定されている各種指令値を更新します。 フローチャート：図 7-43
R_MOTOR_CPU_IF_PositionGet R_MOTOR_CPU_IF_SpeedGet R_MOTOR_CPU_IF_StatusGet R_MOTOR_CPU_IF_ErrorStatusGet R_MOTOR_CPU_IF_LoopModeStatusGet R_MOTOR_CPU_IF_InPositionFlagGet R_MOTOR_CPU_IF_OffsetCalibrationStatusGet	Get Motor Status API です。 Shared Memory の Motor status area に格納されている各種モータステータスを取得します。 フローチャート：図 7-44
R_MOTOR_CPU_IF_PositionOffsetSet	Position Offset Set API です。 位置情報にオフセットを設定します。 フローチャート：図 7-45

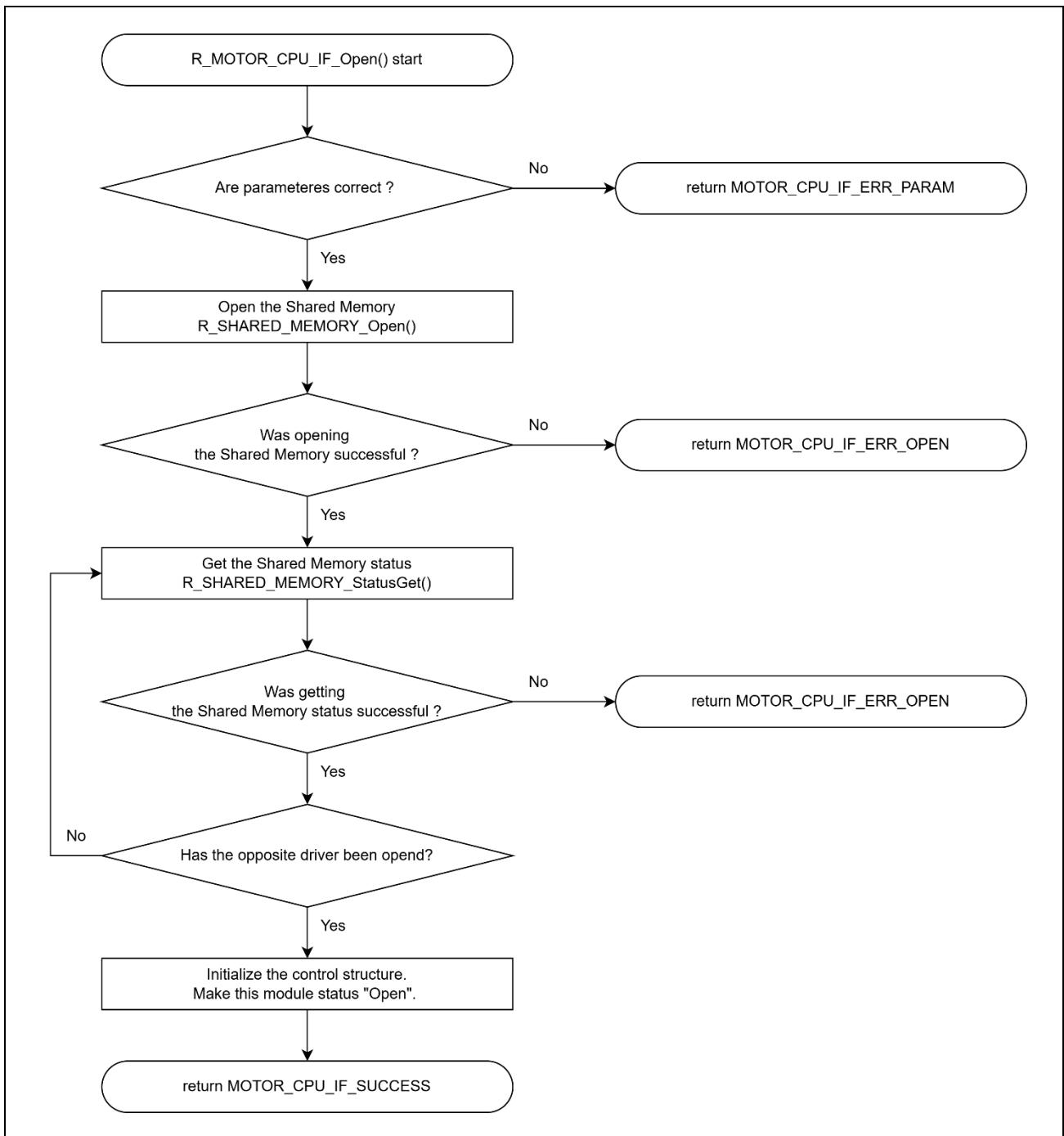


図 7-40. R\_MOTOR\_CPU\_IF\_Open 関数フローチャート

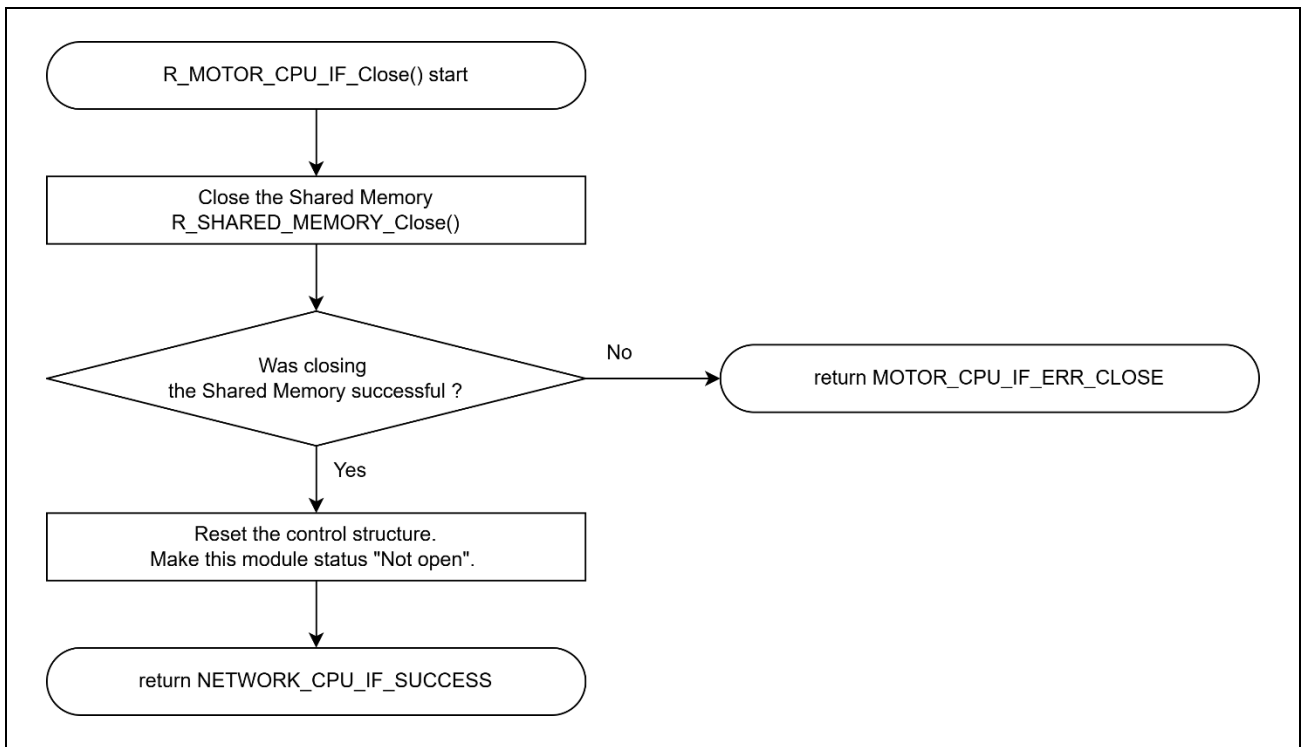


図 7-41. R\_MOTOR\_CPU\_IF\_Close 関数フローチャート

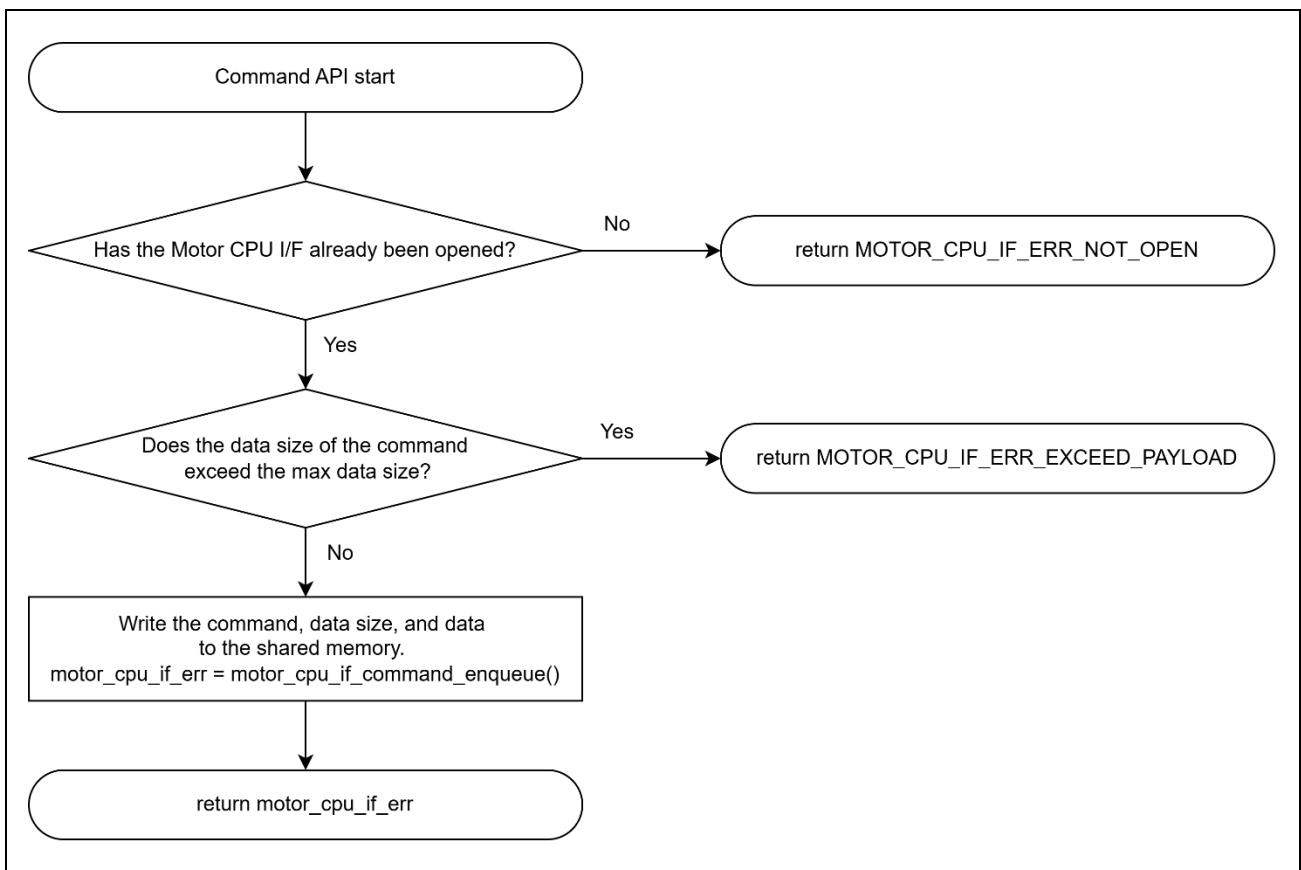


図 7-42. R\_MOTOR\_CPU\_IF Command Set API フローチャート

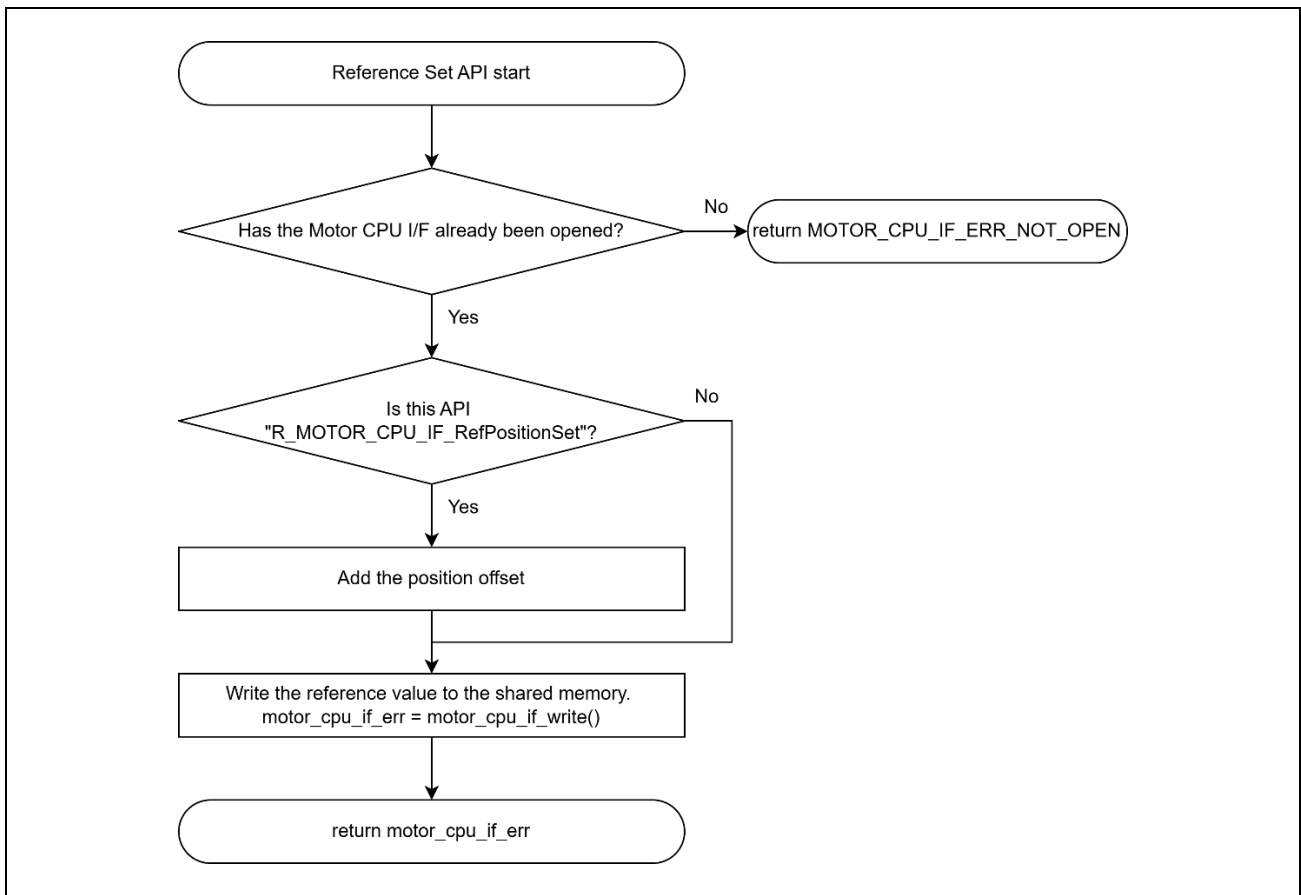


図 7-43. R\_NETWORK\_CPU\_IF Reference Set API フローチャート

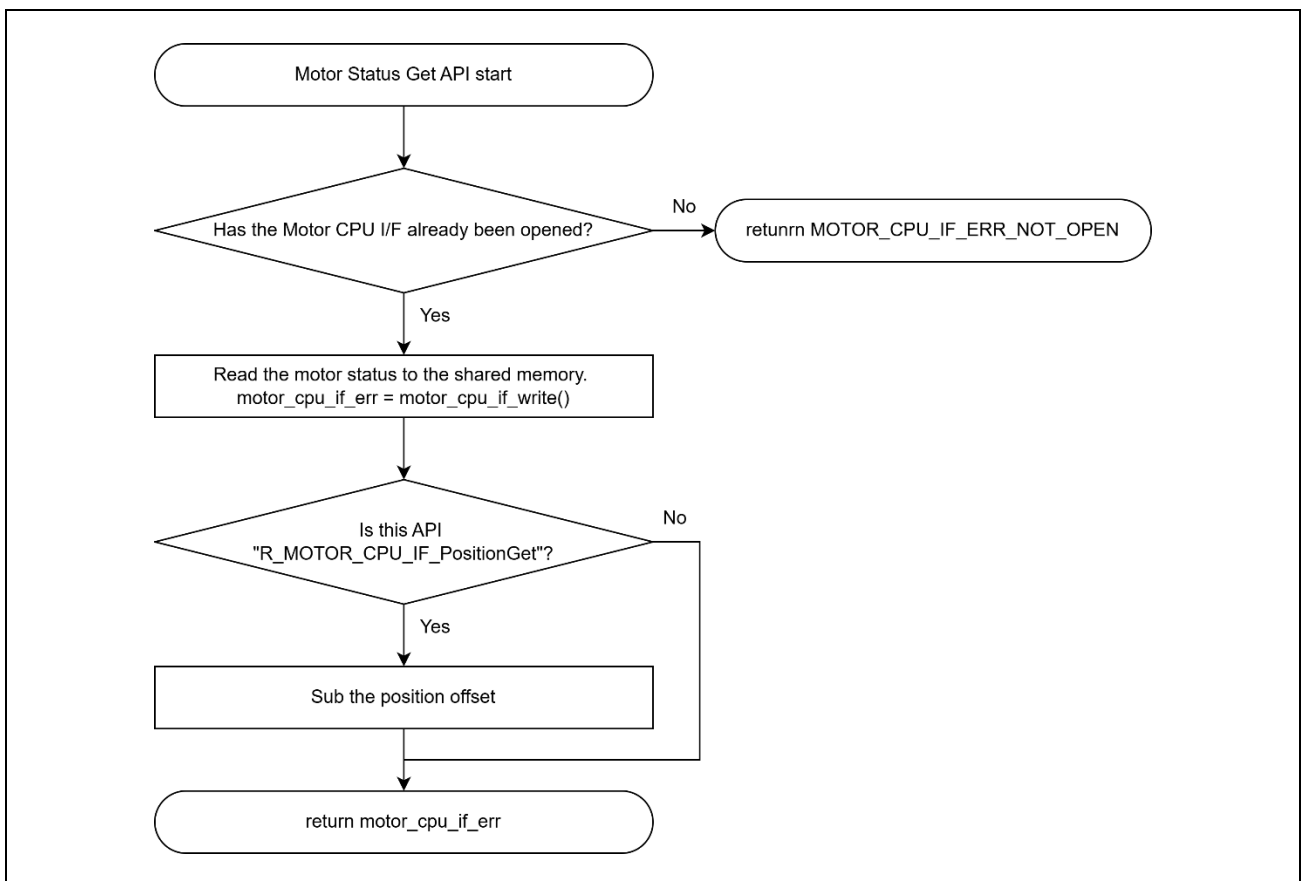


図 7-44. R\_NETWORK\_CPU\_IF Motor Status Get API フローチャート

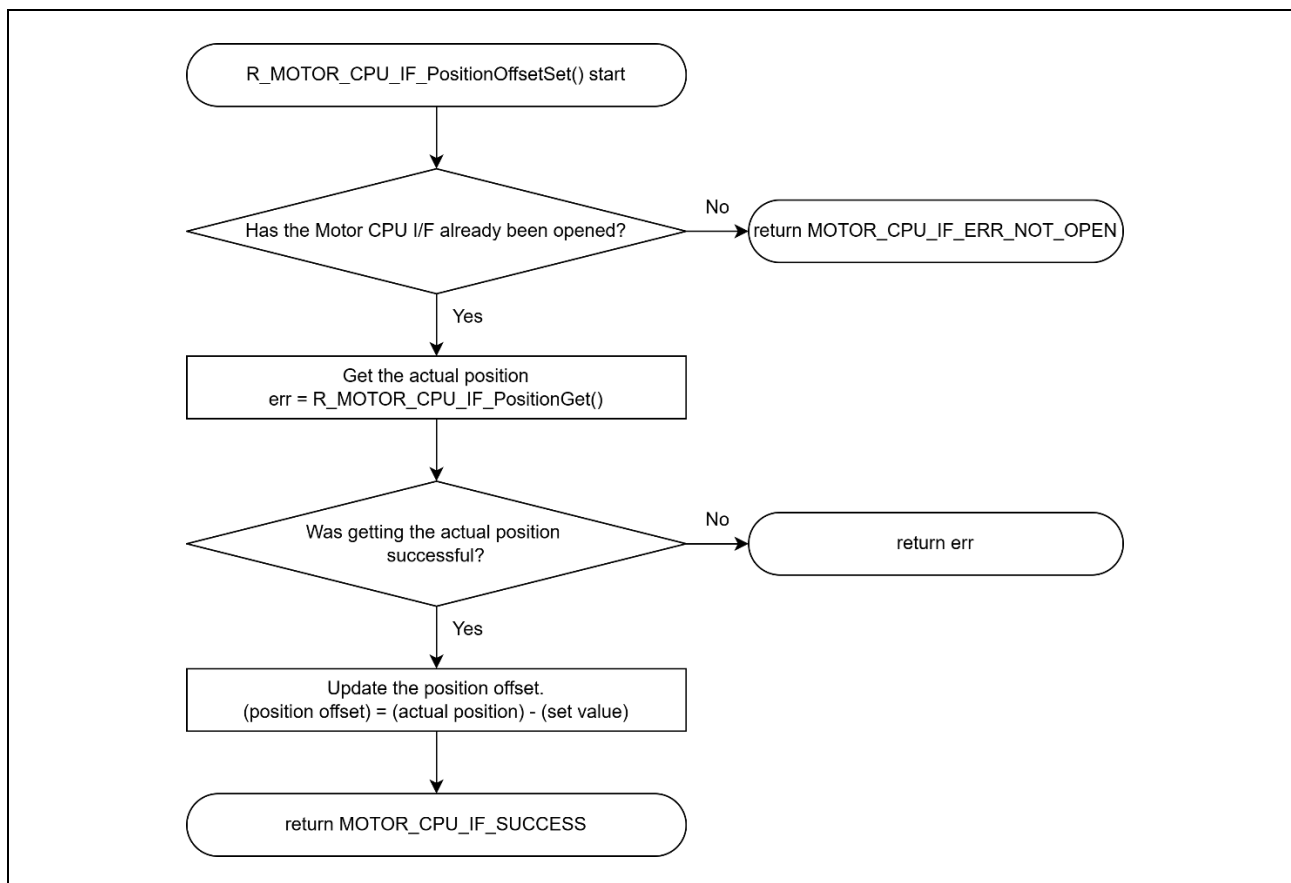


図 7-45. R\_MOTOR\_CPU\_IF\_PositionOffsetSet 関数フローチャート

## 7.3.8.4 プライベート関数

表 7-51. Motor CPU I/F プライベート関数 一覧

プライベート関数	説明
motor_cpu_if_write	Shared memory にデータを書き込みます。 フローチャート：図 7-46
motor_cpu_if_read	Shared memory からデータを読み取ります。 フローチャート：図 7-47
motor_cpu_if_command_enqueue	コマンドキューからコマンドを書き込みます。 フローチャート：図 7-48

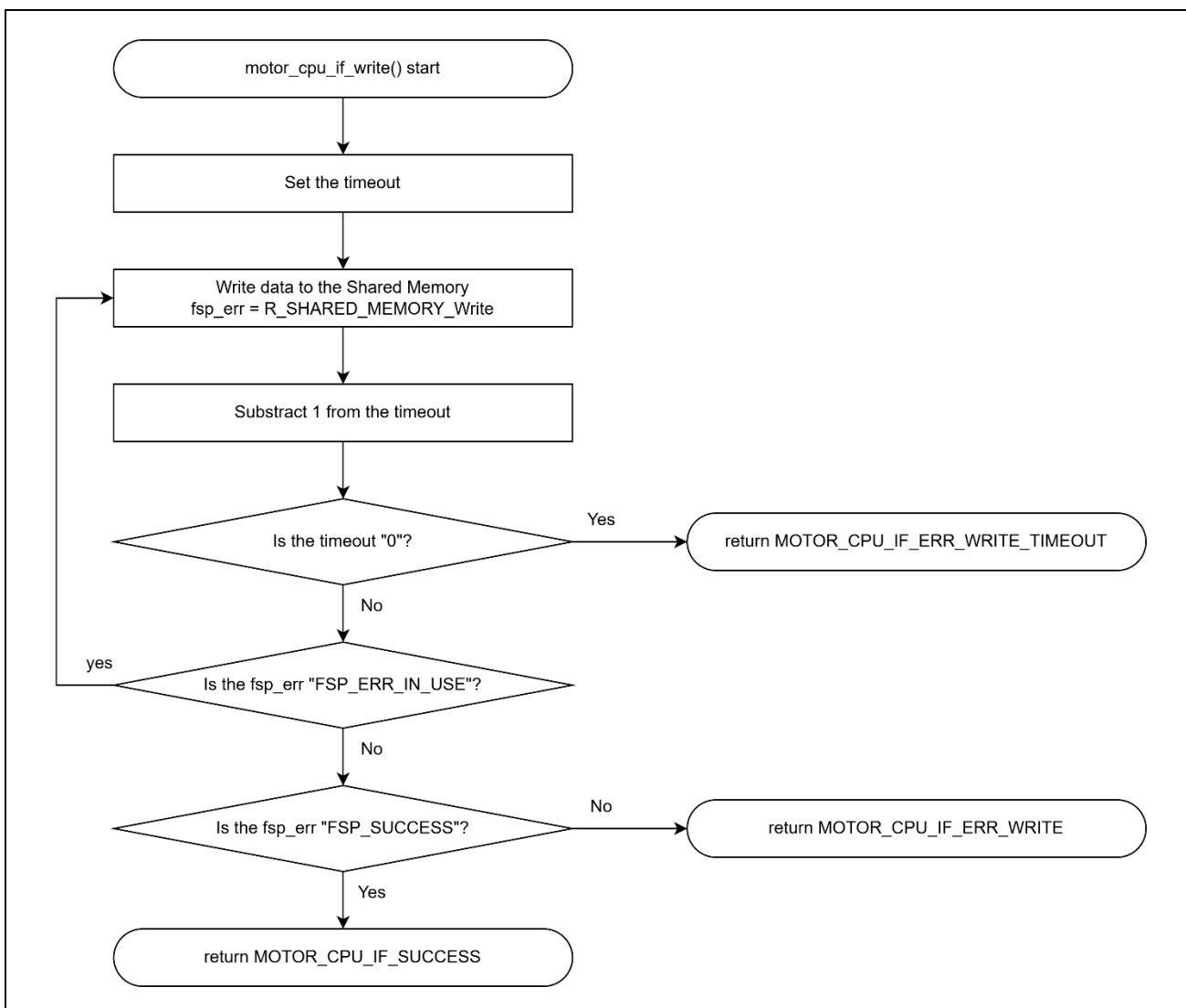


図 7-46. motor\_cpu\_if\_write 関数フローチャート

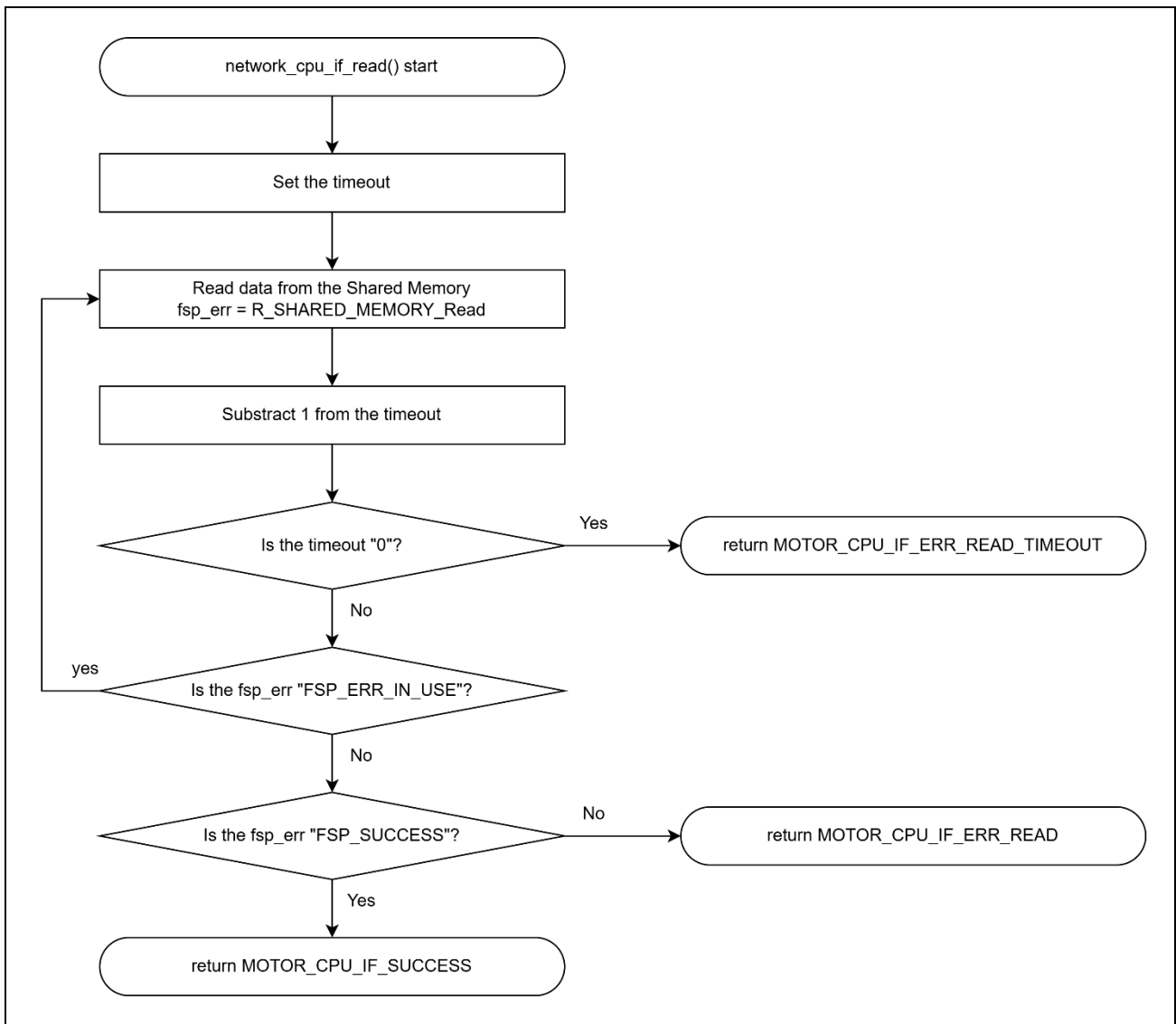


図 7-47. motor\_cpu\_if\_read 関数フローチャート

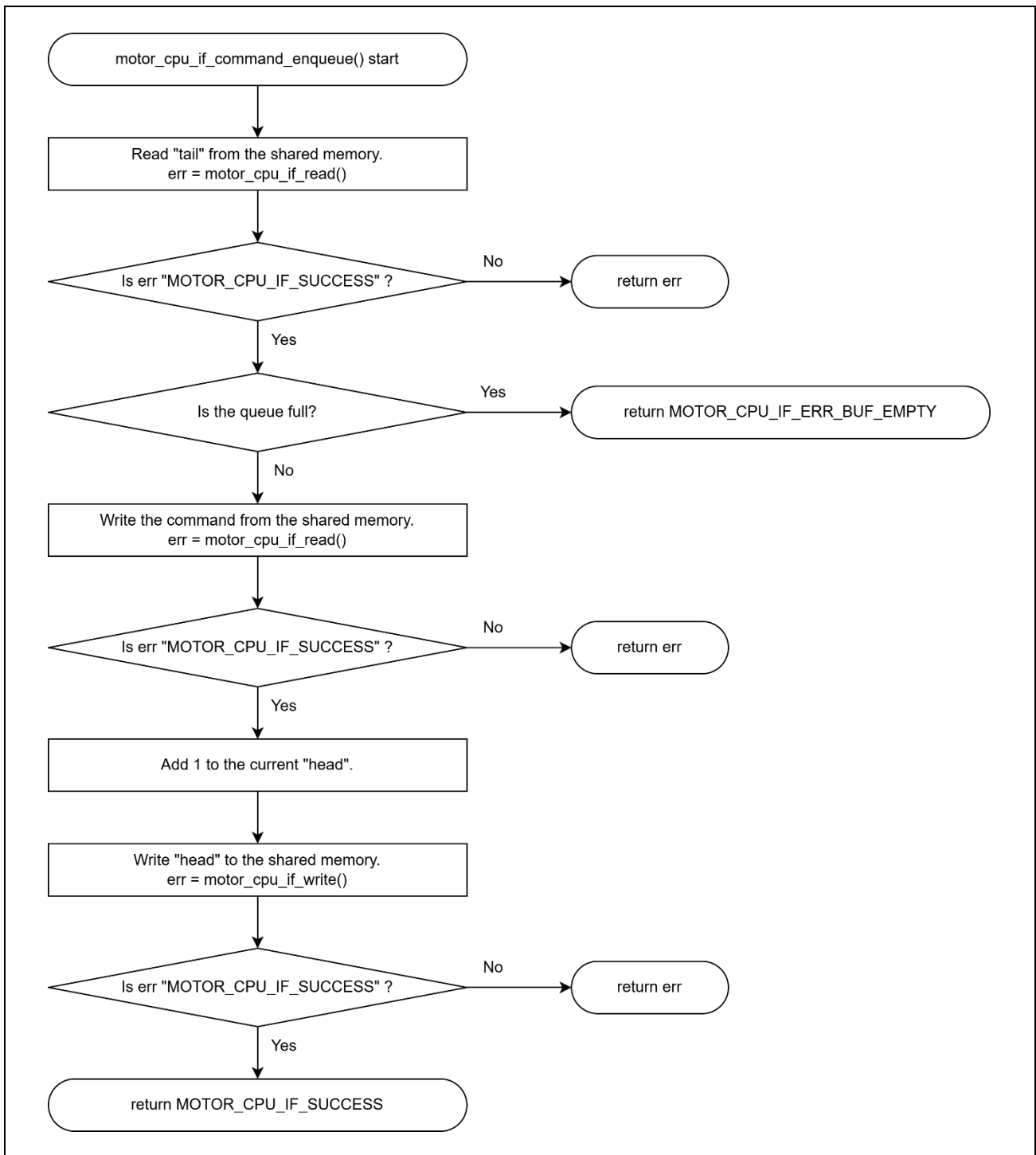


図 7-48. motor\_cpu\_if\_command\_enqueue 関数フローチャート

## 7.3.8.5 Callback 関数

表 7-52. Motor CPU I/F Callback 関数一覧 (r\_motor\_cpu\_if.c)

関数名	説明
shared_memory_callback	<p>これは Shared Memory Driver の Callback 関数です。</p> <p>RZT2M CPU0 側で Shared Memory Driver が Shared Memory へ書き込むと、この割り込みが発生します。</p> <p>Callback 関数内で行われる処理はありません。</p> <p>必要に応じて処理の追加を検討して下さい。</p>

## 7.3.8.6 マクロ定義

表 7-53. Motor CPU I/F マクロ定義一覧 (r\_motor\_cpu\_if.h)

マクロ名	定義値	備考
CFG_CMD_Q_NUM	8	コマンドキューの数 (2 の累乗にする必要があります)
CFG_CMD_Q_OFFSET	0x0000	Command queue area の オフセットアドレス
CFG_CYCLIC_REF_AREA_OFFSET	0x0100	Cyclic reference area の オフセットアドレス
CFG_MTR_STATUS_AREA_OFFSET	0x0180	Motor status area のオフ セットアドレス
CFG_MOTOR_CPU_IF_TIMEOUT	100	Shared memory 読み書 き時に使用されるタイム アウト値
CFG_SHARED_MEMORY_INSTANCE_ADDR	&g_shared_memory0	Shared memory ドライ バのインスタンスへのポ インタ
MOTOR_CPU_IF_CMD_DATA_MAX_BYTE	12	コマンドキューの最大 データサイズ

## 7.3.8.7 列挙型

表 7-54. Motor CPU I/F 列挙型一覧 (r\_motor\_cpu\_if.h)

列挙型	定義	備考
motor_cpu_if_err_t Motor CPU I/F エラーコード	MOTOR_CPU_IF_SUCCESS	エラー無し
	MOTOR_CPU_IF_ERR_OPEN	Open エラー
	MOTOR_CPU_IF_ERR_CLOSE	Close エラー
	MOTOR_CPU_IF_ERR_PARAM	Parameter エラー
	MOTOR_CPU_IF_ERR_NOT_OPEN	Not open エラー
	MOTOR_CPU_IF_ERR_BUF_FULL	コマンドキューが満杯
	MOTOR_CPU_IF_ERR_BUF_EMPTY	コマンドキューが空
	MOTOR_CPU_IF_ERR_WRITE	書き込みエラー
	MOTOR_CPU_IF_ERR_WRITE_TIMEOUT	書き込みタイムアウト
	MOTOR_CPU_IF_ERR_READ	読み取りエラー
	MOTOR_CPU_IF_ERR_READ_TIMEOUT	読み取りタイムアウト
	MOTOR_CPU_IF_ERR_EXCEED_PAYLOAD	データが超過した

列挙型	定義	備考
motor_cpu_if_status_t Motor CPU I/F ステータス	MOTOR_CPU_IF_STATUS_NOT_OPEN	Not open 状態
	MOTOR_CPU_IF_STATUS_OPEN	Open 状態
motor_cpu_if_cmd_t Motor CPU I/F コマンド定義	MOTOR_CPU_IF_CMD_NONE	コマンド無し
	MOTOR_CPU_IF_CMD_RESET	Command Set API で使用され るコマンド
	MOTOR_CPU_IF_CMD_MOTOR_START	
	MOTOR_CPU_IF_CMD_MOTOR_STOP	
	MOTOR_CPU_IF_CMD_MOTOR_RESET	
	MOTOR_CPU_IF_CMD_MOTOR_ERROR_CANCEL	
	MOTOR_CPU_IF_CMD_ERROR_SET	
	MOTOR_CPU_IF_CMD_CTRL_TYPE_SET	
	MOTOR_CPU_IF_CMD_POSITION_COMMAND_MODE_SET	
	MOTOR_CPU_IF_CMD_ROTOR_ANGLE_OFFSET_SET	
	MOTOR_CPU_IF_CMD_CURRENT_OFFSET_SET	
	MOTOR_CPU_IF_CMD_CURRENT_OPEN_LOOP_SET	
	MOTOR_CPU_IF_CMD_ACCEL_TIME_SET	
MOTOR_CPU_IF_CMD_MAX_SPEED_RAD_SET		
motor_cpu_if_pos_mode_t Motor CPU I/F 位置制御 モード	MOTOR_CPU_IF_POS_MODE_STEP	ステップモード
	MOTOR_CPU_IF_POS_MODE_TRAPEZOID	台形モード
motor_cpu_if_ctrl_type_t Motor CPU I/F モータ制御 タイプ	MOTOR_CPU_IF_CTRL_TYPE_POS	位置制御
	MOTOR_CPU_IF_CTRL_TYPE_SPEED	速度制御
	MOTOR_CPU_IF_CTRL_TYPE_TORQUE	トルク制御
	MOTOR_CPU_IF_CTRL_TYPE_VOLTAGE	オープンループ 制御
	MOTOR_CPU_IF_CTRL_TYPE_CURRENT	
	MOTOR_CPU_IF_CTRL_TYPE_ADJ_CRNT_OFFSET	電流オフセット
MOTOR_CPU_IF_CTRL_TYPE_ADJ_POS_OFFSET	位置オフセット	
motor_cpu_if_motor_err_code_t Motor CPU I/F モータエ ラーコード	MOTOR_CPU_IF_MOTOR_ERROR_NONE	エラー無し
	MOTOR_CPU_IF_MOTOR_ERROR_OVER_CURRENT_HW	HW 過電流エ ラー
	MOTOR_CPU_IF_MOTOR_ERROR_OVER_VOLTAGE	過電圧エラー
	MOTOR_CPU_IF_MOTOR_ERROR_OVER_SPEED	回転速度エラー
	MOTOR_CPU_IF_MOTOR_ERROR_UNDER_VOLTAGE	定電圧エラー
	MOTOR_CPU_IF_MOTOR_ERROR_OVER_CURRENT_SW	SW の過電流エ ラー

	MOTOR_CPU_IF_MOTOR_ERROR_OVER_TEMPERATURE	インバータ加熱エラー
	MOTOR_CPU_IF_MOTOR_ERROR_UNKNOWN	不明なエラー
motor_cpu_if_motor_state_t	MOTOR_CPU_IF_STATEMACHINE_STATE_STOP	STOP モード
Motor CPU I/F モータステータス	MOTOR_CPU_IF_STATEMACHINE_STATE_RUN	RUN モード
	MOTOR_CPU_IF_STATEMACHINE_STATE_ERROR	ERROR モード

## 7.3.8.8 構造体・変数情報

表 7-55. Motor CPU I/F 構造体定義一覧 (r\_motor\_cpu\_if.h)

構造体	変数	備考
st_current_offset_cfg_t Current offset configuration 構造体	f4_iu_offset	U-phase 電流オフセット
	f4_iv_offset	V-phase 電流オフセット
	f4_iw_offset	W-phase 電流オフセット
st_current_open_loop_cfg_t Open loop current configuration 構造体	f4_open_loop_current	Open loop 電流値
	f4_ref_speed_rpm	指令速度 (RPM)
motor_cpu_if_cmd_hdr_t コマンドキュー head/tail 構造体	head	コマンドキューの head
	tail	コマンドキューの tail
motor_cpu_if_cmd_q_t コマンドキュー 構造体	cmd	コマンド
	size	データサイズ
	data[MOTOR_CPU_IF_CMD_DATA_MAX_BYTE]	データ配列
motor_cpu_if_cyclic_ref_t Cyclic Reference 構造体	f4_ref_position	指令位置
	f4_ref_speed	指令速度
	f4_ref_torque	指令トルク
motor_cpu_if_mtr_status_t Motor Status 構造体	f4_pos_deg	現在位置
	f4_speed	現在速度
	u1_status	ステートマシン
	reserved1	未使用領域
	u2_error_status	エラー状態
	reserved2	未使用領域
	u1_ctrl_loop_mode	LOOP MODE
	reserved3	未使用領域
	u1_in_position	位置制御完了状態
	reserved4	未使用領域
	u1_offset_cal_finished_flag	オフセット測定状態

	reserved5	未使用領域
motor_cpu_if_cfg_t Motor CPU I/F コンフィグ構造体	p_shared_memory_instance	Shared Memory インスタンスへのポインタ
	timeout	タイムアウト値
	shr_mem_cmd_data_max_size	コマンドデータ最大サイズ
	shr_mem_cmd_q_num	コマンドキューの数
	shr_mem_cmd_q_mask	コマンドキューマスク
	shr_mem_cmd_q_offset	共有メモリのコマンドキュー領域のオフセット
	shr_mem_cmd_q_size	共有メモリのコマンドキュー領域のサイズ
	shr_mem_cmd_head_offset	共有メモリのコマンドキューhead領域のオフセット
	shr_mem_cmd_head_size	共有メモリのコマンドキューhead領域のサイズ
	shr_mem_cmd_tail_offset	共有メモリのコマンドキューtail領域のオフセット
	shr_mem_cmd_tail_size	共有メモリのコマンドキューtail領域のサイズ
	shr_mem_cyclic_ref_area_offset	共有メモリのCyclic Reference Areaのオフセット
	shr_mem_cyclic_ref_area_size	共有メモリのCyclic Reference Areaのサイズ
	shr_mem_ref_pos_offset	共有メモリの指令位置領域のオフセット
	shr_mem_ref_vel_offset	共有メモリの指令速度領域のオフセット
	shr_mem_ref_tor_offset	共有メモリの指令トルク領域のオフセット
	shr_mem_mtr_status_area_offset	共有メモリのMotor Status Areaのオフセット
	shr_mem_mtr_status_area_size	共有メモリのMotor Status Areaのサイズ
shr_mem_mtr_pos_offset	共有メモリの現在位置領域のオフセット	
shr_mem_mtr_vel_offset	共有メモリの現在速度領域のオフセット	
shr_mem_mtr_status_offset	共有メモリのステートマシン領域のオフセット	
shr_mem_mtr_err_status_offset	共有メモリのエラー状態領域のオフセット	

	shr_mem_mtr_ctrl_loop_mode_offset	共有メモリの LOOP MODE 領域のオフセット	
	shr_mem_mtr_in_pos_offset	共有メモリの位置制御完了状態領域のオフセット	
	shr_mem_mtr_cal_finished_flag_offset	共有メモリのオフセット測定状態領域のオフセット	
motor_cpu_if_ctrl_t Motor CPU I/F 制御構造体	status	Motor CPU I/F のステータス	
	cmd_hdr	コマンドキュー head/tail 構造体インスタンス	
	timeout_count	タイムアウト発生回数	
	position_offset	位置オフセット	
	p_cfg	Motor CPU I/F コンフィグ構造体へのポインタ	
motor_cpu_if_api_t Motor CPU I/F API 構造体	Open	オープン関数	
	Close	クローズ関数	
	Reset	コマンド送信関数	
	MotorStart		
	MotorStop		
	MotorReset		
	MotorErrorCancel		
	ErrorSet		
	CtrlTypeSet		
	PositionCommandModeSet		
	RotorAngleOffsetSet		
	CurrentOffsetSet		
	CurrentOpenLoopSet		
	AccelTimeSet		
	PosProfMaxSpeedRadSet		
	PositionOffsetSet		位置オフセット設定関数
	RefPositionSet		参照値セット関数
	RefSpeedSet		
	RefTorqueSet		
	PositionGet	モータステータス取得関数	
	SpeedGet		
	StatusGet		
	ErrorStatusGet		
LoopModeStatusGet			
InPositionFlagGet			
OffsetCalibrationStatusGet			

## 7.3.8.9 グローバル変数情報

表 7-56. Motor CPU I/F グローバル変数一覧 (r\_motor\_cpu\_if.c)

型名	インスタンス名	説明
motor_cpu_if_ctrl_t	g_motor_cpu_if_ctrl	Motor CPU I/F 制御構造体インスタンス。 Oepn 関数で初期化されます。
motor_cpu_if_cfg_t	g_motor_cpu_if_cfg	Motor CPU I/F コンフィグ構造体インスタンス。 インスタンス生成時に各メンバが初期化されます。 初期化以外で各メンバの値を変更することはできません。
motor_cpu_if_api_t	g_motor_cpu_if_api	Motor CPU I/F API 構造体インスタンス。 インスタンス生成時に各メンバが初期化されます。 初期化以外で各メンバの値を変更することはできません。

### 7.3.9 Fusa I/F

Fusa I/F モジュールは、機能安全 CPU (RZ/T2L-A)と UART 通信を行うモジュールです。

機能安全 CPU の仕様および通信の仕様については『RZ ファミリ FSoE アプリケーションソフトウェア ソフトウェア開発ハンドブック (R30UZ0176JJ0110)』を参照してください。

#### 7.3.9.1 モジュール構成

モジュール構成図を図 7-49 に示します。

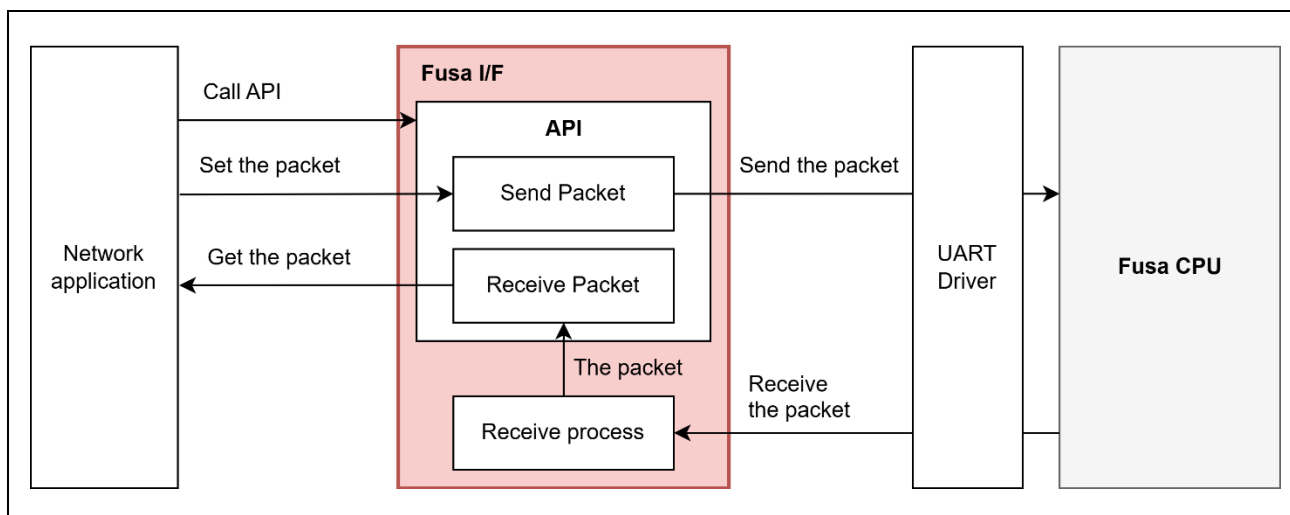


図 7-49. Fusa I/F モジュール構成図

#### 7.3.9.2 送受信パケット

Fusa I/F モジュールでは、送受信されるパケットのコマンドコードとサイズ情報を Fusa I/F コンフィグ 構造体インスタンス (g\_fusa\_if\_cfg) に登録して使用します。下記に登録されている送受信パケットについて述べます。

表 7-57. 登録されている受信パケット一覧

#	パケット名称	コマンドコード	サイズ情報	受信するデータ
1	起動要求	0x0101	0x0000	無し
2	Safety Slave PDU	0x0202	0x0006	表 7-46 の Safety Slave PDU 行に記載されているオブジェクト
3	Communication parameter	0x0203	0x000C	表 7-46 の Communication parameter 行に記載されているオブジェクト

表 7-58. 登録されている送信パケット一覧

#	パケット名称	コマンドコード	サイズ情報	送信するデータ
1	起動要求受付確認	0x0102	0x0000	無し
2	Safety Master PDU	0x0201	0x0006	表 7-45 の Safety Master PDU 行に記載されているオブジェクト

## 7.3.9.3 API

表 7-59. Fusa I/F API 一覧

API	説明
R_FUSA_IF_Open	UART Driver の初期化を行います。 また、起動要求パケットを受信されるのを待ち、その受信が完了した場合は起動要求受付確認パケットを送信します。 フローチャート： 図 7-50
R_FUSA_IF_Close	Fusa I/F を終了します。 また、UART Driver を終了します。 フローチャート： 図 7-51
R_FUSA_IF_Send_Packet	Fusa CPU へパケットを送信する関数。 送信したいパケットのコマンドコードを予めセットした構造体ポインタを引数にとります。 そのコマンドコードが本モジュールに登録されていない場合、パケットを送信しません。 また、ひとつ前のパケットの送信が完了していない場合、パケットを送信しません。 フローチャート： 図 7-52
R_FUSA_IF_Receive_Packet	Fusa CPU から受け取ったパケットを取得する関数。 取得したいパケットのコマンドコードを予めセットした構造体ポインタを引数にとります。 そのコマンドコードが本モジュールに登録されていない場合、パケットを取得しません。 また、そのパケットの受信が完了していない場合は取得できません。 フローチャート： 図 7-53

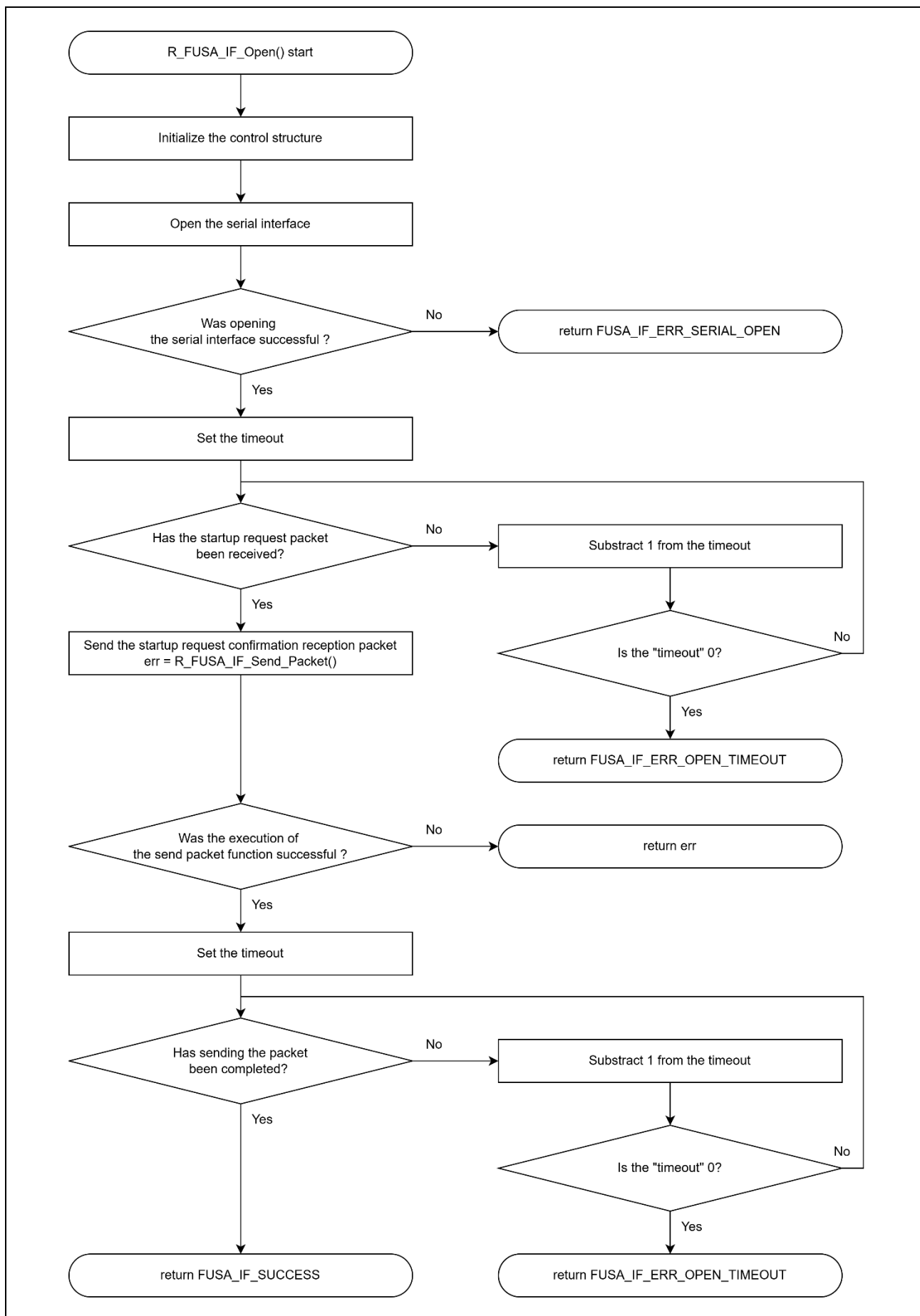


図 7-50. R\_FUSA\_IF\_Open 関数フローチャート

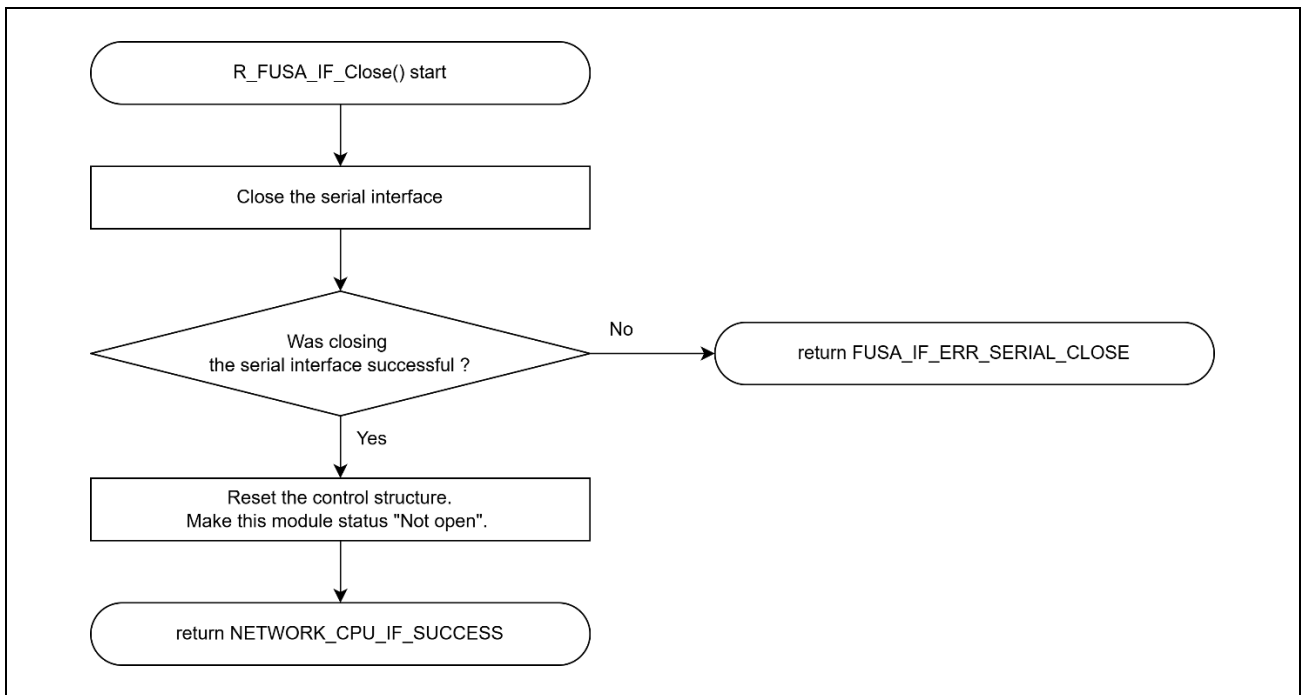


図 7-51. R\_FUSA\_IF\_Close 関数フローチャート

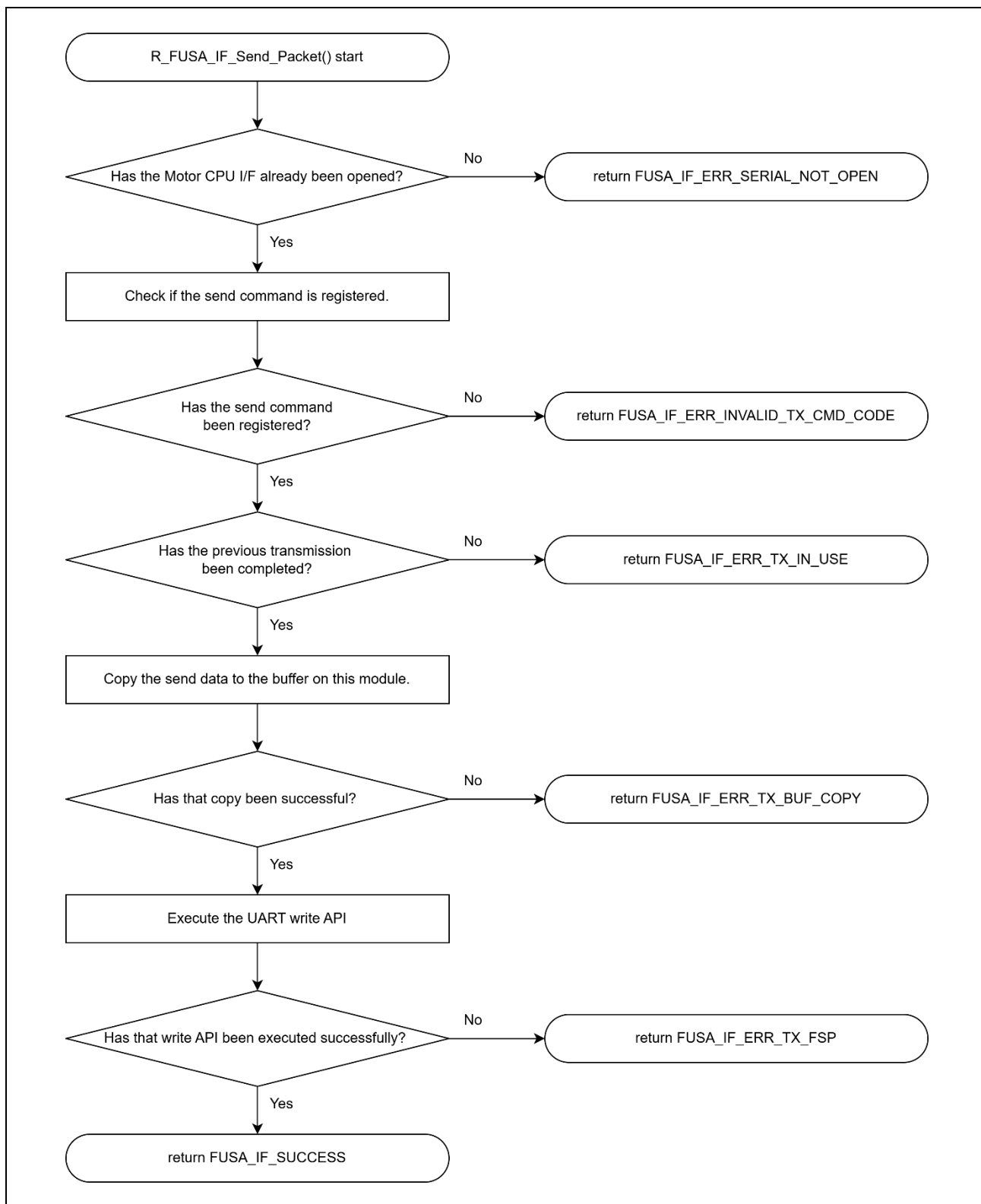


図 7-52. R\_FUSA\_IF\_Send\_Packet 関数フローチャート

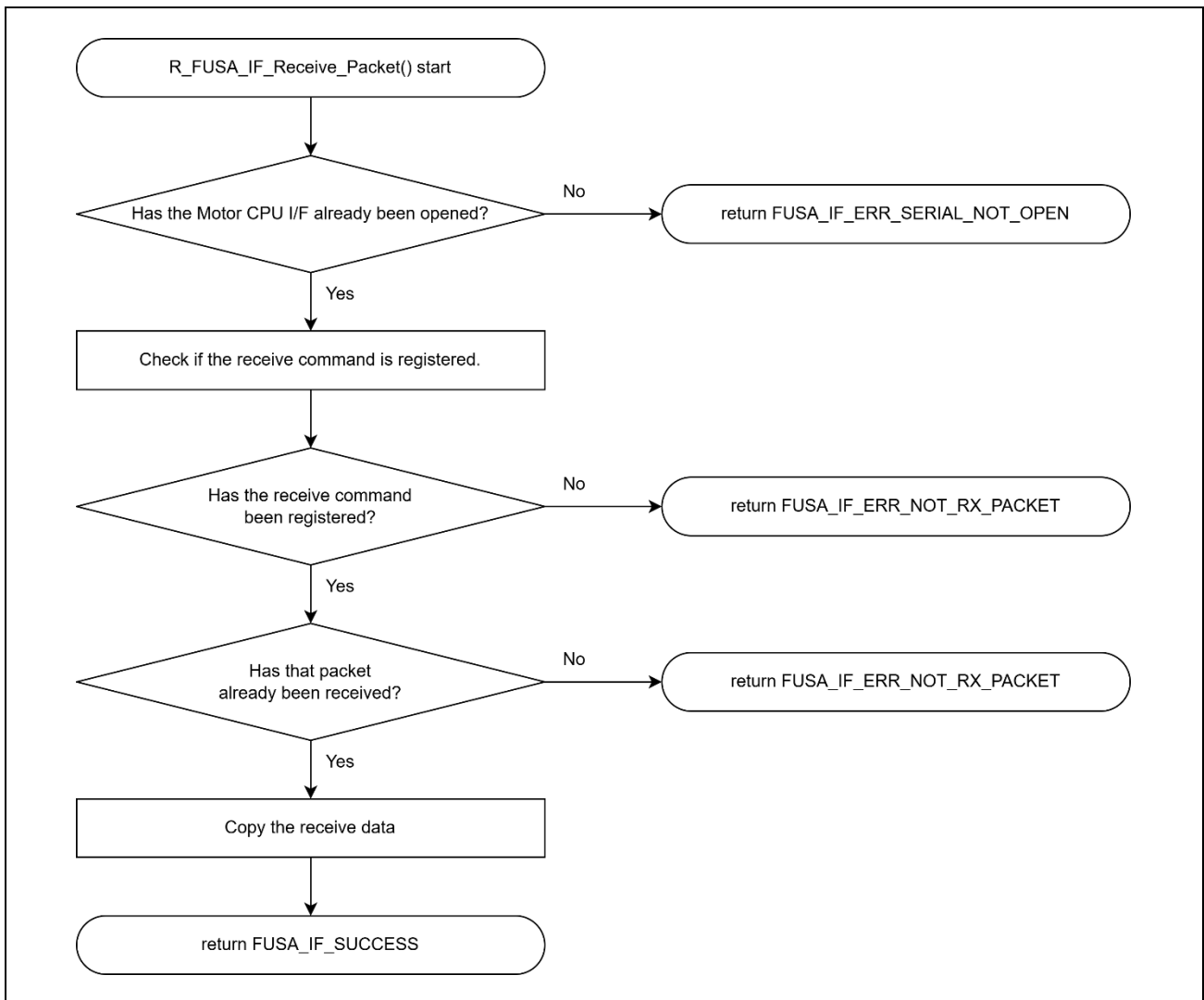


図 7-53. R\_FUSA\_IF\_Receive\_Packet 関数フローチャート

## 7.3.9.4 プライベート関数

表 7-60. Fusa I/F プライベート関数 一覧

プライベート関数	説明
fusa_if_receive_process	Fusa CPU から受け取ったデータを解析し、問題が無ければパケットととして保存する関数です。

## 7.3.9.5 Callback 関数

表 7-61. Fusa I/F Callback 関数一覧 (r\_network\_cpu\_if.c)

関数名	説明
fusa_if_callback	<p>これは UART Driver の Callback 関数です。</p> <p>Callback 関数内では下記の処理が行われます。</p> <p>(1) 1byte データを受信したとき</p> <ul style="list-style-type: none"><li>● fusa_if_receive_process 関数を呼び出します。</li></ul> <p>(2) 送信処理が完了したとき</p> <ul style="list-style-type: none"><li>● 送信処理が完了したことを Fusa I/F に知らせます。</li></ul> <p>必要に応じて処理の追加を検討して下さい。</p>

## 7.3.9.6 マクロ定義

表 7-62. Fusa I/F マクロ定義一覧 (r\_fusa\_if.h)

マクロ名	定義値	備考
CFG_FUSA_IF_UART_INSTANCE_ADD R	&g_uart0	UART ドライバのインスタ ンスへのポインタ
CFG_FUSA_IF_CMD_RX_NUM	3	登録する受信パケットの数
CFG_FUSA_IF_CMD_TX_NUM	2	登録する送信パケットの数
CFG_FUSA_IF_CMDCODE_RX_START _REQ	0x0101	起動要求のコマンドコード
CFG_FUSA_IF_CMDCODE_RX_PDU	0x0202	Safety Slave PDU のコマ ンドコード
CFG_FUSA_IF_CMDCODE_RX_PARAM	0x0102	通信パラメータのコマ ンドコード
CFG_FUSA_IF_CMDCODE_TX_START _REQ	0x0102	起動要求受付確認のコマ ンドコード
CFG_FUSA_IF_CMDCODE_TX_PDU	0x0201	Safety Master PDU のコマ ンドコード
CFG_FUSA_IF_DATASIZE_RX_START _REQ	0	起動要求のデータサイズ
CFG_FUSA_IF_DATASIZE_RX_PDU	6	Safety Slave PDU のデー タサイズ
CFG_FUSA_IF_DATASIZE_RX_PARAM	12	通信パラメータのデー タサイズ
CFG_FUSA_IF_DATASIZE_TX_START _REQ	0	起動要求受付確認のデー タサイズ
CFG_FUSA_IF_DATASIZE_TX_PDU	6	Safety Master PDU のデー タサイズ
CFG_FUSA_IF_DATASIZE_MAX	CFG_FUSA_IF_DATASIZE_R X_PARAM	最大のデータサイズ
CFG_FUSA_IF_RXCMD0_CMDCODE	CFG_FUSA_IF_CMDCODE_R X_START_REQ	受信パケット0のコマ ンドコード
CFG_FUSA_IF_RXCMD0_DATASIZE	CFG_FUSA_IF_DATASIZE_R X_START_REQ	受信パケット0のデー タサイズ
CFG_FUSA_IF_RXCMD1_CMDCODE	CFG_FUSA_IF_CMDCODE_R X_PDU	受信パケット1のコマ ンドコード
CFG_FUSA_IF_RXCMD1_DATASIZE	CFG_FUSA_IF_DATASIZE_R X_PDU	受信パケット1のデー タサイズ
CFG_FUSA_IF_RXCMD2_CMDCODE	CFG_FUSA_IF_CMDCODE_R X_PARAM	受信パケット2のコマ ンドコード
CFG_FUSA_IF_RXCMD2_DATASIZE	CFG_FUSA_IF_DATASIZE_R X_PARAM	受信パケット2のデー タサイズ
CFG_FUSA_IF_TXCMD0_CMDCODE	CFG_FUSA_IF_CMDCODE_T X_START_REQ	送信パケット0のコマ ンドコード

CFG_FUSA_IF_TXCMD0_DATASIZE	CFG_FUSA_IF_DATASIZE_TX_START_REQ	送信パケット0のデータサイズ
CFG_FUSA_IF_TXCMD1_CMDCODE	CFG_FUSA_IF_CMDCODE_TX_PDU	送信パケット1のコマンドコード
CFG_FUSA_IF_TXCMD1_DATASIZE	CFG_FUSA_IF_DATASIZE_TX_PDU	送信パケット1のデータサイズ
CFG_OPEN_TIMEOUT	0x0FFFFFFF	タイムアウト値

## 7.3.9.7 列挙型

表 7-63. Fusa I/F 列挙型一覧 (r\_fusa\_if.h)

列挙型	定義	備考
fusa_if_err_t Fusa I/F エラーコード	FUSA_IF_SUCCESS	エラー無し
	FUSA_IF_ERR_INVALID_PARAMETER	Parameter エラー
	FUSA_IF_ERR_SERIAL_OPEN	Open エラー
	FUSA_IF_ERR_SERIAL_CLOSE	Close エラー
	FUSA_IF_ERR_SERIAL_NOT_OPEN	Not open エラー
	FUSA_IF_ERR_INVALID_RX_CMD_CODE	無効な受信コマンドコード
	FUSA_IF_ERR_NOT_RX_PACKET	パケットを受信していない
	FUSA_IF_ERR_INVALID_TX_CMD_CODE	無効な送信コマンドコード
	FUSA_IF_ERR_TX_IN_USE	送信中
	FUSA_IF_ERR_TX_BUF_COPY	送信用バッファへのコピー失敗
	FUSA_IF_ERR_TX_FSP	FSP 関数でエラーが発生
	FUSA_IF_ERR_OPEN_TIMEOUT	Open タイムアウト
fusa_if_status_t Fusa I/F ステータス	FUSA_IF_STATUS_NOT_OPEN	Not open 状態
	FUSA_IF_STATUS_OPEN	Open 状態
fusa_if_rx_status_t Fusa I/F 受信ステータス	FUSA_IF_RX_NO_DATA	データ未受信
	FUSA_IF_RX_RECEIVED_CMD1	コマンドコードの1byte目を受信済み
	FUSA_IF_RX_RECEIVED_CMD2	コマンドコード受信済み
	FUSA_IF_RX_RECEIVED_SIZE1	データサイズの1byte目を受信済み
	FUSA_IF_RX_RECEIVED_SIZE2	データサイズ受信済み

## 7.3.9.8 構造体・変数情報

表 7-64. Fusa I/F 構造体一覧 (r\_fusa\_if.h)

構造体	変数	備考
fusa_if_cmd_list_t Command list 構造体	command_code	コマンドコード情報
	data_size	データサイズ情報
fusa_if_cfg_t Fusa I/F コンフィグ構造体	p_uart_instance	UART インスタンスへのポインタ
	open_timeout	タイムアウト値
	rx_cmd_num	受信パケットの数
	tx_cmd_num	送信パケットの数
	rx_cmd_list[CFG_FUSA_IF_CMD_RX_NUM]	受信パケットリスト
	tx_cmd_list[CFG_FUSA_IF_CMD_TX_NUM]	送信パケットリスト
fusa_if_packet_t Fusa I/F パケット構造体	command_code	コマンドコード
	data_size	データサイズ
	data[CFG_FUSA_IF_DATASIZE_MAX]	データ
fusa_if_ctrl_t Fusa I/F 制御構造体	fusa_if_status	Fusa I/F のステータス
	transfer_complete	送信完了フラグ
	tx_tmp	送信用バッファ
	rx_status	受信ステータス
	rx_data_pos	受信データ位置
	rx_tmp	受信用バッファ
	rx_buf[CFG_FUSA_IF_CMD_RX_NUM]	各受信パケットの保存バッファ
	rx_complete_flag[CFG_FUSA_IF_CMD_RX_NUM]	受信パケットの受信フラグ
fusa_if_api_t Fusa I/F API 構造体	Open	オープン関数
	Close	クローズ関数
	send_packet	パケットの送信
	receive_packet	パケットの受信

## 7.3.9.9 グローバル変数情報

表 7-65. Fusa I/F グローバル変数一覧 (r\_fusa\_if.c)

型名	インスタンス名	説明
fusa_if_ctrl_t	g_fusa_if_ctrl	Fusa I/F 制御構造体インスタンス。 Open 関数で初期化されます。
fusa_if_cfg_t	g_fusa_if_cfg	Fusa I/F コンフィグ構造体インスタンス。 インスタンス生成時に各メンバが初期化されます。 初期化以外で各メンバの値を変更することはできません。
fusa_if_api_t	g_fusa_if_api	Fusa I/F API 構造体インスタンス。 インスタンス生成時に各メンバが初期化されます。 初期化以外で各メンバの値を変更することはできません。

## 7.4 機能安全制御 (RZ/T2L-A, RZ/T2L-B)

### 7.4.1 概要

RZ/T2L機能安全制御用ソフトウェアの機能概要を以下に示します。

- ・ RZ/T2LによるHFT = 1構成の機能安全システムを実現
- ・ RTK0EF0190D01001BJのRZ/T2Mが制御するモータに対する安全制御 (IEC61800-5-2 : STO, SS1-t) を実施
- ・ ETG.5100ES (R) V1.2.0 準拠のFSoEプロトコル処理を実施

RZ/T2Lで安全部を構築するために、本ソフトウェアには下記機能安全ソリューション製品が使用されています。

1. RZ(CR52 core)ファミリ セルフテストソフトウェアキット : RTK0EF0102F01001SJ Ver.1.30  
同梱 自己診断ソフトウェア : Ver.1.01
2. RZ/T2Lグループ SIL3システムソフトウェアキット : RTK0EF0154F01001SJ Ver.1.03  
同梱 機能安全プラットフォームソフトウェア : Ver.1.00
3. RZファミリ FSoEアプリケーションソフトウェアキット : RTK0EF0129F01001SJ Ver.1.12  
同梱 FSoEアプリケーションソフトウェア : Ver.1.10

これらソフトウェアの詳細については、各製品に同梱されているドキュメントを参照ください。

表7.66に、RZ/T2L機能安全制御用ソフトウェアのフォルダ構成を示します。

表 7.66 RZ/T2L 機能安全制御用ソフトウェアのフォルダ構成

#	フォルダ / ファイル名	説明
1	¥RZT2L	RZ/T2L用プロジェクトフォルダ
2	¥FSoE_SW_M	RZ/T2L-A (メイン) 用プロジェクトフォルダ
3	¥RZT2L	
4	¥PL-SW	
5	¥PswLoaderFM	
6	¥FSoE_SW_S	
7	¥RZT2L	
8	¥PL-SW	
9	¥PswLoaderFS	
10	¥flashloader	EWARM用RTK0EF0190D01001BJ搭載ROM書き込み ファイルフォルダ
11	¥arm	
12	¥TwinCAT3	FSoEマスタ用プロジェクト (注)
13	Safety_Drive.tfzip	
14	Safety_Drive.tpzip	
15	¥rewrite_folder	セーフティモータ制御ボード用のパッチプログラム RZ/T2L_A用のパッチプログラムフォルダ
16	¥FSoE_SW_M	
17	¥FSoE_SW_S	

注：本プロジェクトはEtherCATマスタにTwinCAT3 (Build 4024.6)、FSoEマスタにBeckhoff Automation社の下記機器を使用した場合に使用可能です。それ以外の環境を使用する場合には、ユーザにて7.4.7章に記載されている機能をFSoEマスタに対してプログラムしてください。

- ・ EK1100 : EtherCAT Coupler
- ・ EL6900 : EtherCAT Terminal communication interface, TwinSAFE Logic
- ・ EL9011 : Bus end cover for E-bus contacts

## 7.4.2 機能概要

表7.67に、RZ/T2L機能安全制御用ソフトウェアの概要を示します。

表 7.67 RZ/T2L 機能安全制御用ソフトウェア 概要

項目	仕様	参照
モータ 安全制御部	モータ停止モード選択スイッチ (SW9_1 および SW11_1) による停止モード (STO/SS1-t) の状態確認とモータ制御 MPU (RZ/T2M) への通知	7.4.4
	モータ安全状態解除と RZ/T2M への通知 [モータ安全状態解除の条件] 下記 2 つの条件が満たされた状態で、2 つのシステム起動スイッチ (SW8 および SW10) を押下することでモータの安全状態は解除されます (スイッチ押下順は任意)。 - RZ/T2L による FSoE 通信機能が正常に動作し、FailSafeData ステートになっている場合 - FSoE マスタから受信した SafeData により、RTK0EF0190D01001BJ の 2 つの疑似緊急停止スイッチ (SW9_3, SW11_3) の OFF が確認された場合	7.4.5
	モータ安全制御実施と RZ/T2M への通知 [モータ安全制御実施の条件] - FSoE マスタから受信した SafeData にて、疑似緊急停止スイッチ (SW9_3, SW11_3) の ON が確認された場合 - RZ/T2L の FSoE が ProcessData ステート以外に遷移した場合 - PL-SW による診断の結果、安全制御用 RZ/T2L に異常が検出された場合	7.4.6
FSoE 処理部	ETG.5100ES (R) V1.2.0 準拠の protocols 処理	-
	FSoE 応答性能 : 339 us	図 7-55
	FSoE スレーブアドレス : 0x0010 固定	-
	SafeData サイズ : 1 バイト (PDU サイズ : 6 バイト)	7.4.7
	[Safety Slave PDU 側 : SafeData 仕様] 安全入力コネクタ (CON17, CON18) と疑似緊急スイッチ (SW9_3, SW11_3) から SafeData を生成  SafeData[7] = ! CON17_5 SafeData[6] = ! CON17_4 SafeData[5] = ! CON17_3 SafeData[4] = ! CON17_2 & SW9_3 SafeData[3] = ! CON18_5 SafeData[2] = ! CON18_4 SafeData[1] = ! CON18_3 SafeData[0] = ! CON18_2 & SW11_3	[Safety Master PDU 側 : SafeData 仕様] 受信した SafeData をそのまま安全出力コネクタ (CON15, CON16) に出力  SafeData[7] = CON15_5 SafeData[6] = CON15_4 SafeData[5] = CON15_3 SafeData[4] = CON15_2 SafeData[3] = CON16_5 SafeData[2] = CON16_4 SafeData[1] = CON16_3 SafeData[0] = CON16_2
その他	LED にて安全部の動作状態を表示	7.4.8

7.4.3 ソフトウェア構成

図7-54に、RZ/T2L機能安全制御用ソフトウェアの構成を示します。以降、機能安全プラットフォームソフトウェアは "PL-SW"、FSoEアプリケーションソフトウェアは "FSoEアプリケーション" と表記します。

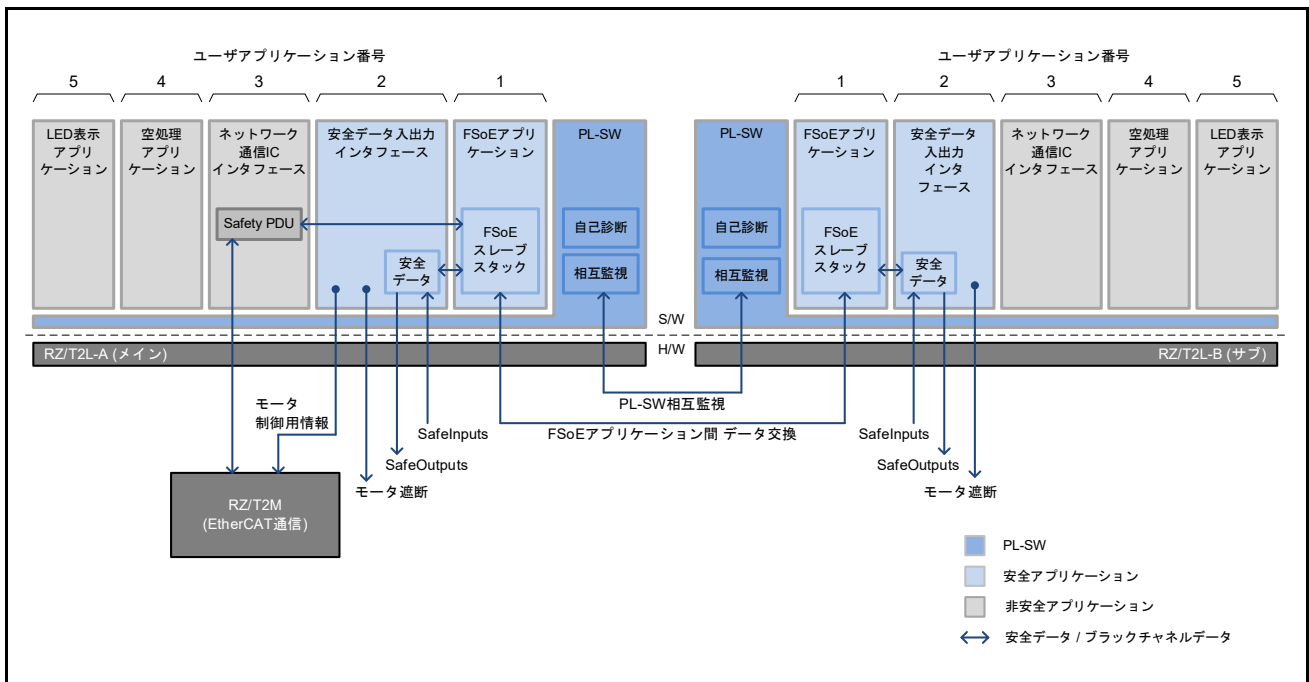


図 7-54 RZ/T2L 機能安全制御用ソフトウェア構成

表7.68に、各ユーザアプリケーションの概要を示します。

表 7.68 各ユーザアプリケーションの概要 (1/2)

App No	ユーザアプリケーション名	概要
1	FSoE アプリケーション	FSoE スレーブ用プロトコルスタック処理を行います。対となる RZ/T2L 内の FSoE アプリケーションとシリアル通信によりデータ交換を行います。
2	安全データ入出力インタフェース	安全入力ポート、安全出力ポートを制御するソフトウェアです。FSoE アプリケーションと安全データの受け渡しを行います。RZ/T2M に対して、モータ制御用の情報を通知します。
3	ネットワーク通信 IC インタフェース	[RZ/T2L-A (メイン)] EtherCAT 通信を行う RZ/T2M と Safety PDU の送受信を行います。また、FSoE アプリケーションと Safety PDU および通信パラメータの受け渡しを行います。 [RZ/T2L-B (サブ)] FSoE アプリケーションと通信パラメータの受け渡しを行います。
4	空処理アプリケーション	ユーザ処理追加用の空タスクです。
5	LED 表示アプリケーション	本ソフトウェアの動作状態を LED で表示する評価用アプリケーションです。

図7-55に、RZ/T2L機能安全制御用ソフトウェアの動作シーケンスを示します。各ユーザアプリケーションの動作シーケンスはPL-SWにより制御されています。この動作シーケンスにおいて、PL-SW1周期で1通信分のFSoEデータの処理を行います。

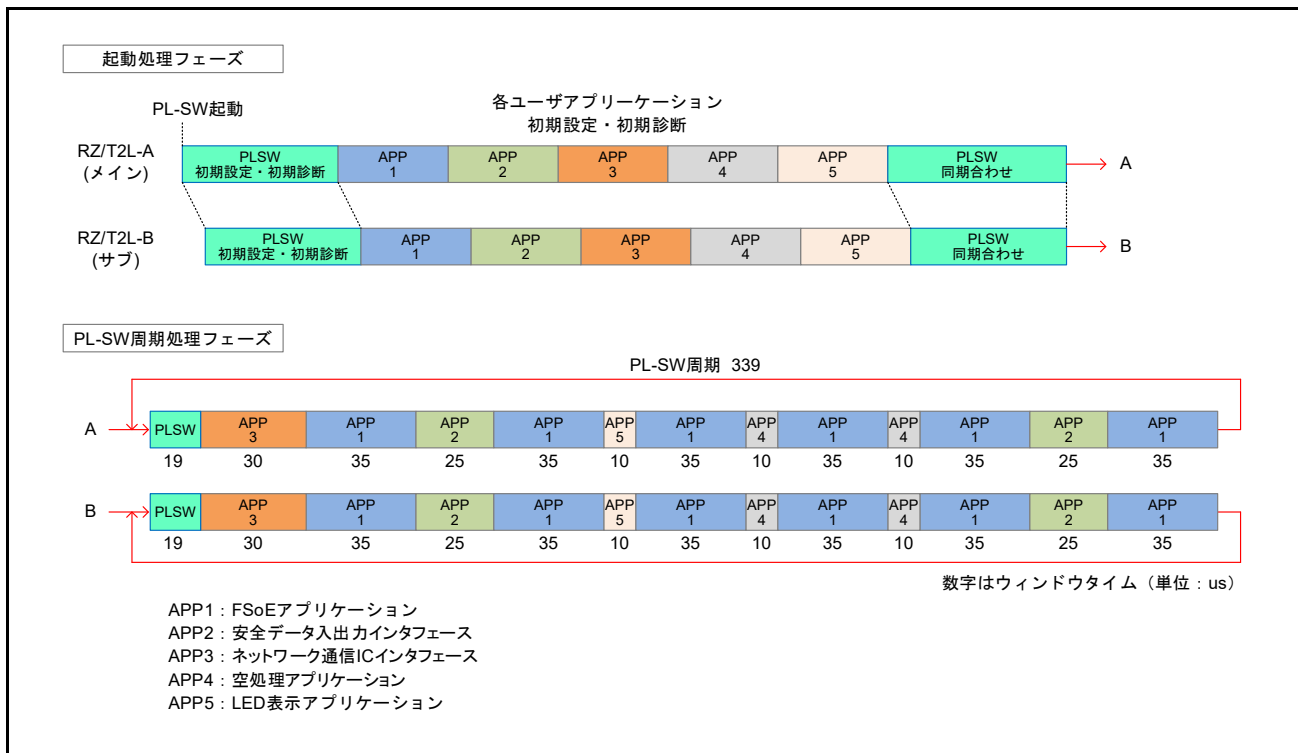


図 7-55 RZ/T2L 機能安全制御用ソフトウェア 動作シーケンス

表7.69に、PL-SWおよび各ユーザアプリケーションの配置を示します。

表 7.69 PL-SW および各ユーザアプリケーションの配置

メモリ	アドレス	配置ソフトウェア	永久故障診断対象	ROM 書き込み対象
ATCM	H'0000 0000 - H'0001 4FFF	PL-SW コード	✓	✓
	H'0001 5000 - H'0001 AFFF	PL-SW データ	✓	
	:	:		
	H'0002 0000 - H'0002 4FFF	APP 1 コード	✓	✓
	H'0002 5000 - H'0002 5FFF	APP 2 コード	✓	✓
	H'0002 6000 - H'0002 6FFF	APP 3 コード		✓
	H'0002 7000 - H'0002 7FFF	APP 4 コード		✓
	H'0002 8000 - H'0002 8FFF	APP 5 コード		✓
	H'0002 9000 - H'0003 3FFF	APP 6 ~ 16 コード (未使用)		✓
H'0003 4000 - H'0007 FFFF	未使用			
システム SRAM	H'1000 0000 - H'1000 0FFF	APP 1 データ	✓	
	H'1000 1000 - H'1000 1FFF	APP 2 データ	✓	
	H'1000 2000 - H'1000 2FFF	APP 3 データ		
	H'1000 3000 - H'1000 3FFF	APP 4 データ		
	H'1000 4000 - H'1000 4FFF	APP 5 データ		
	H'1000 5000 - H'1000 FFFF	APP 6 ~ 16 データ (未使用)		
	:	:		
	H'100F F000 - H'100F FFFF	未使用		
		PL-SW checksum		✓

表7.70に、RZ/T2L機能安全制御用ソフトウェアの使用資源を示します。リファレンスボード "RTK0EF0190D01001BJ" のコネクタ (CONxx)、スイッチ (SWxx) の情報については、ボードのユーザーズマニュアルを参照ください。

表7.70 RZ/T2Lセーフティドライブソフトウェア使用資源

項目	使用モジュール	使用端子	用途	
PL-SW	SCI1	TXD1 : P02_2	相互監視 (RZ/T2L-A ⇔ RZ/T2L-B)	
		RXD1 : P02_0		
	GPT7, 9, 10	GTIOC10A : P13_2 GTETRGB : P24_0		
	CRC1	なし	コードデータ診断	
FSoE アプリケーション	SCI0	TXD0 : P16_0	FSoE 関連データ交換 (RZ/T2L-A ⇔ RZ/T2L-B)	
		RXD0 : P16_1		
	DMAC0 ch0-1	なし		
	CRC0	なし		
データ入出力 インタフェース	PORT	P01_3~0	SafeInputs	RZ/T2L-A : CON17_5~2 ピン RZ/T2L-B : CON18_5~2 ピン
		P13_7~4	SafeOutputs	RZ/T2L-A : CON15_5~2 ピン RZ/T2L-B : CON16_5~2 ピン
		P10_4	モータ停止モード (STO / SS1-t)	RZ/T2L-A : SW9_1 (注1) RZ/T2L-B : SW11_1 (注1)
		P10_3	未使用	RZ/T2L-A : SW9_2 (注2) RZ/T2L-B : SW11_2 (注2)
		P10_2	疑似緊急停止 スイッチ	RZ/T2L-A : SW9_3 (注3) RZ/T2L-B : SW11_3 (注3)
		P10_0	システム起動 スイッチ	RZ/T2L-A : SW8 RZ/T2L-B : SW10
		P14_3	RZ/T2M へのモータ停止モード (STO / SS1-t) 通知	
		P14_2	RZ/T2M へのモータ動作許可通知	
		P07_2-1	モータ遮断	
ネットワーク通信 IC インタフェース	SCI4	TXD4 : P18_4	Safety PDU 交換 (RZ/T2M ⇔ RZ/T2L-A)	
		RXD4 : P18_5		
	DMAC0 ch2-3	なし		
LED 表示 アプリケーション	PORT	P17_7~5, P17_3 P21_7~6, P21_2~1	ステータス LED 制御	

注1：動作モードはRZ/T2L起動時に参照されます。動作中に動作モードの変更を行っても無視されます。

注2：SW9\_2およびSW11\_2は、常に "OFF" のまま使用してください。

注3：RZ/T2Lリセット解除時、疑似緊急停止スイッチ (SW9\_3, SW11\_3) は必ずOFFにしておいてください。

## 7.4.4 モータ停止モード選択

リファレンスボード "RTK0EF0190D01001BJ" のスイッチにてモータ停止モードの選択が可能です。PL-SWの起動処理フェーズでモード選択スイッチの状態が確認され、その情報がRZ/T2Mに通知されます。その後スイッチの設定を変更してもモータ停止モードは変化しません。

図7-56に、モータ停止モード関連の信号制御について示します。

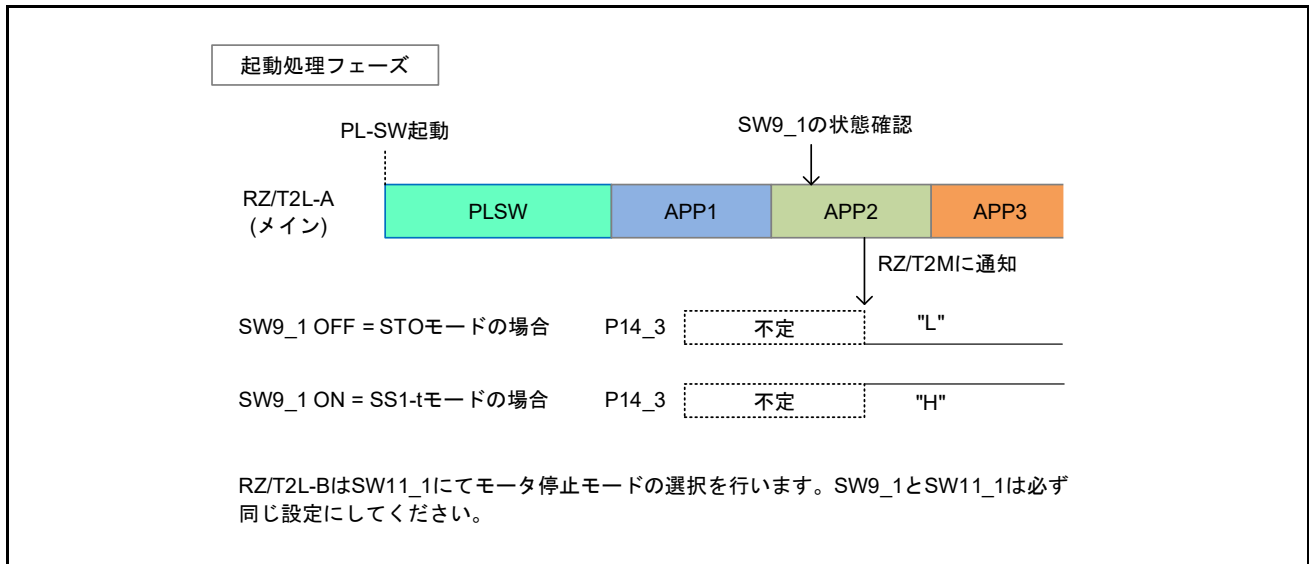


図 7-56 モータ停止モード関連の信号制御

## 7.4.5 安全状態解除

以下に、モータの安全状態（RZ/T2Lによるモータ停止）の解除条件を示します。

[モータ安全状態解除条件]

1. RZ/T2L による FSoE 通信機能が正常に動作し、FailSafeData ステートになっている場合
2. FSoE マスタから受信した SafeData により、RTK0EF0190D01001BJ の 2 つの疑似緊急停止スイッチ（SW9\_3, SW11\_3）の OFF（SafeData[4] = SafeData[0] = "1"）が確認された場合
3. 上記 2 つの条件が満たされた状態で、2 つのシステム起動スイッチ（SW8, SW10）が押下された場合（スイッチ押下順は不問）

システム起動スイッチは、上記 "1" および "2" の条件が満たされている間のみサンプリングされます。上記 "1" および "2" の条件達成の状態はステータスLEDにて確認可能です。ステータスLEDの仕様については「7.4.8. ステータスLED仕様」を参照ください。

図7-57に、モータ安全状態解除フローを示します。

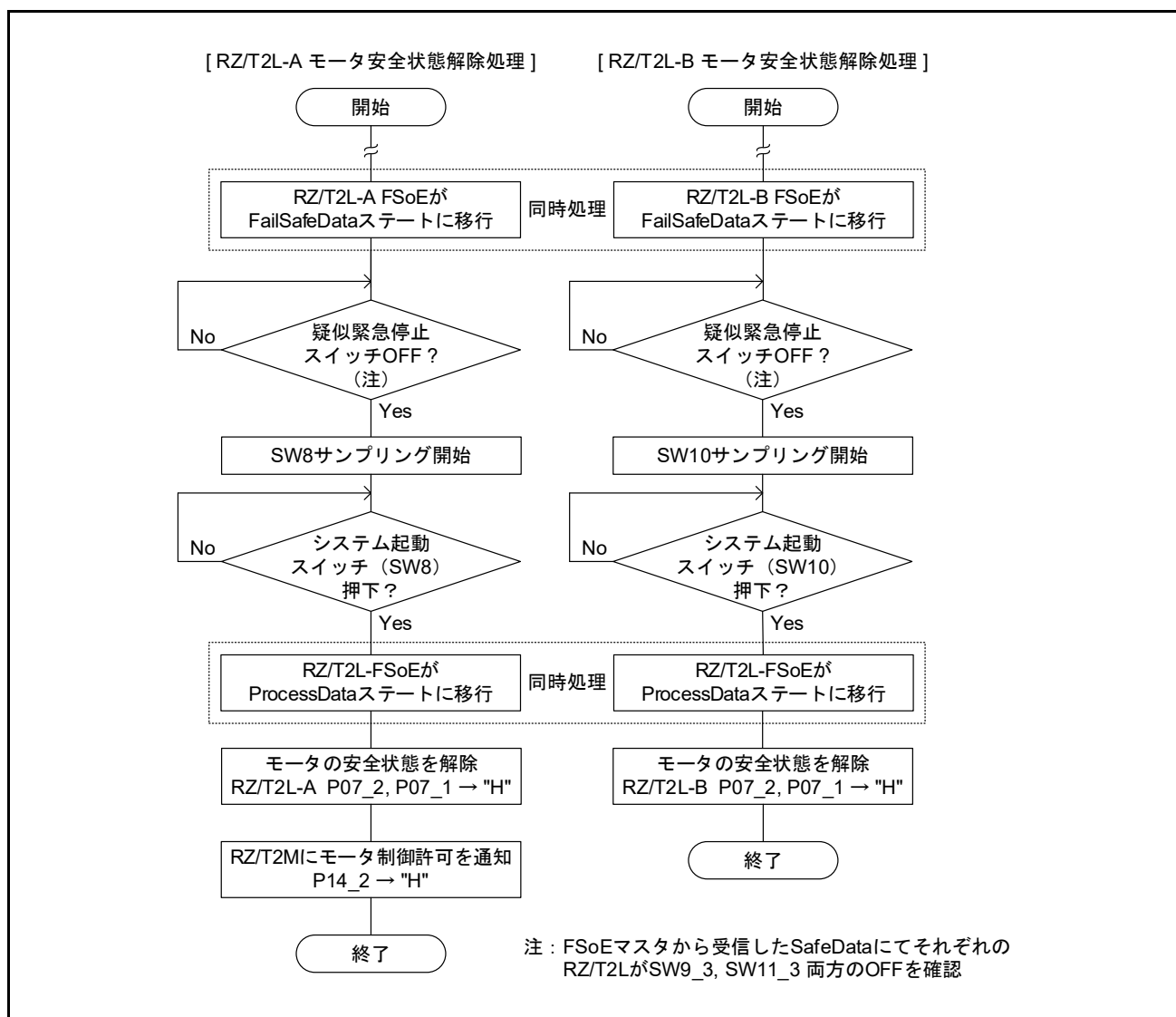


図 7-57 モータ安全状態解除フロー

## 7.4.6 安全制御の実行

以下に、モータの安全制御（RZ/T2Lによるモータ停止）の条件を示します。

## [モータ安全制御条件]

1. FSoE マスタから受信した SafeData により、疑似緊急停止スイッチの ON（SafeData[4] もしくは SafeData[0] が "0" であること）が確認された場合
2. FSoE 通信異常により RZ/T2L の FSoE ステートが ProcessData ステート以外に遷移した場合
3. PL-SW による診断の結果、安全制御用 RZ/T2L に異常が検出された場合

図7-58に、モータ安全制御シーケンスを示します。

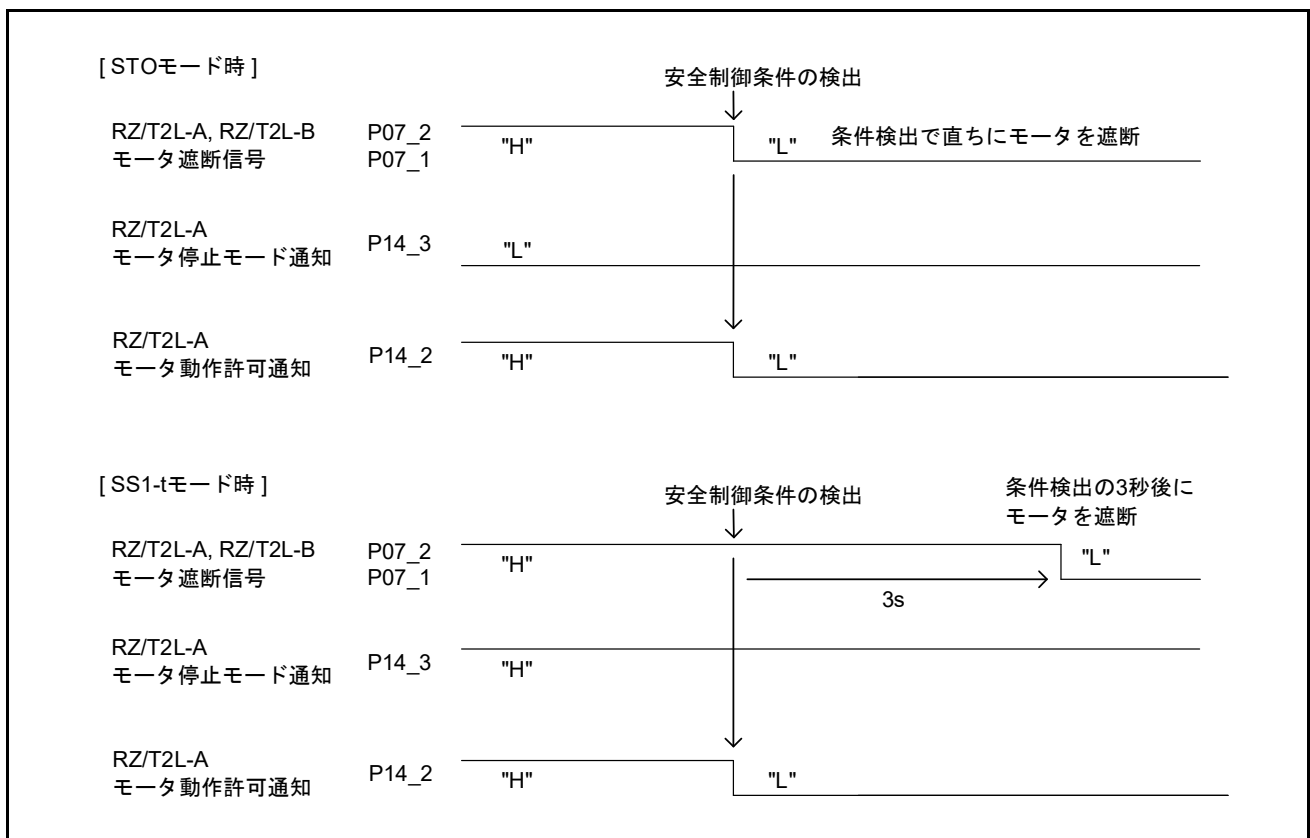


図 7-58 モータ安全制御シーケンス

## 7.4.7 SafeData仕様

RZ/T2L機能安全制御用ソフトウェアは、リファレンスボード "RTK0EF0190D01001BJ" の安全入力コネクタ (CON17, CON18) への入力値と、疑似緊急停止スイッチ (SW9\_3, SW11\_3) の値からSafety Slave PDUのSafeDataを生成します。また、Safety Master PDUから生成したSafeDataの値を "RTK0EF0190D01001BJ" の安全出力コネクタ (CON15, CON16) に出力します。なお、本ソフトウェアは、FSoEマスタが受信したSafeDataをそのまま出力することを前提に作成されています。

図7-59に、SafeData仕様を示します。

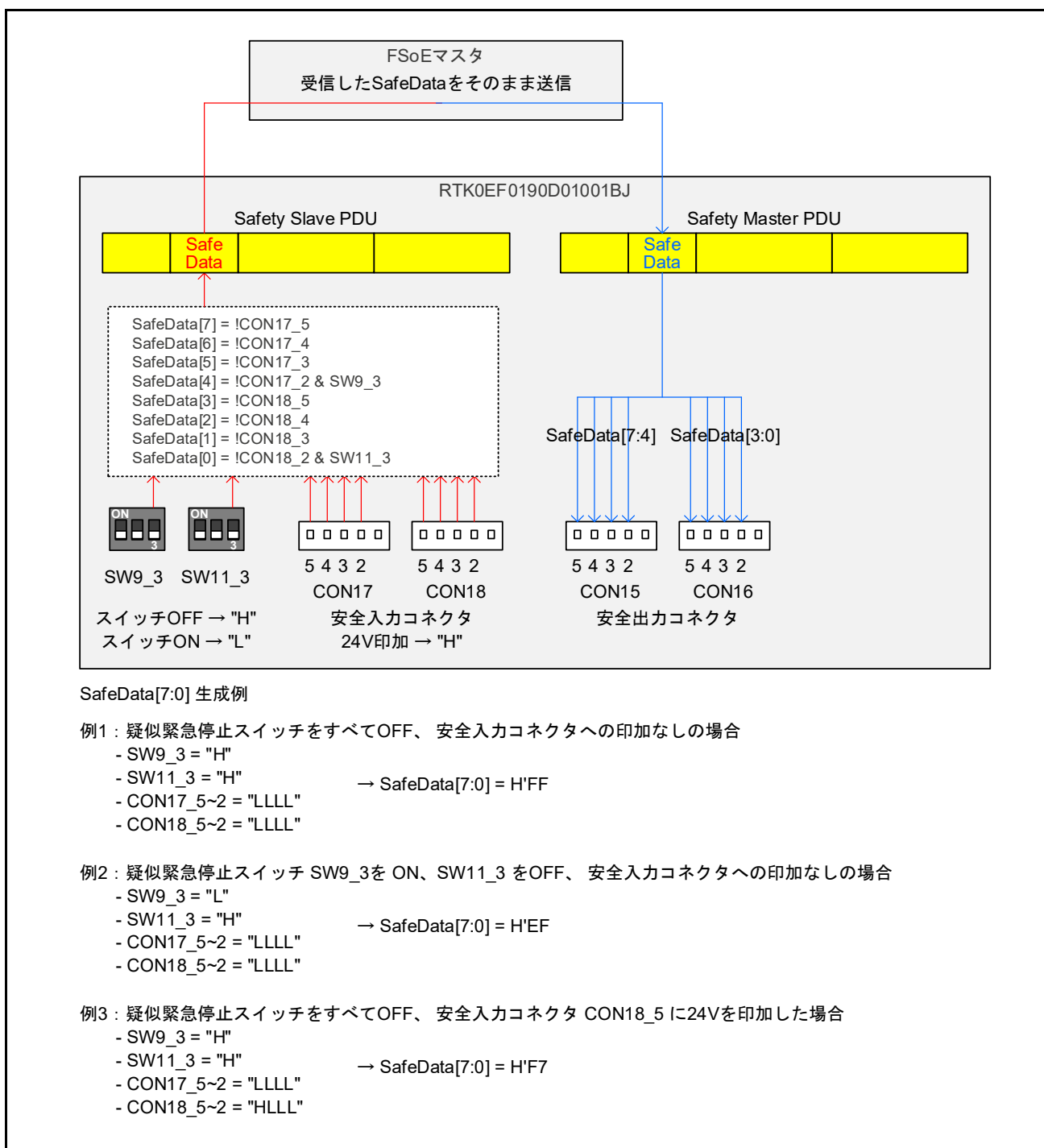


図 7-59 SafeData 仕様

7.4.8 ステータスLED仕様

RZ/T2L機能安全制御用ソフトウェアの動作状態は、リファレンスボード "RTK0EF0190D01001BJ" のLEDで確認できます。

図7-60に、ステータスLED仕様を示します。

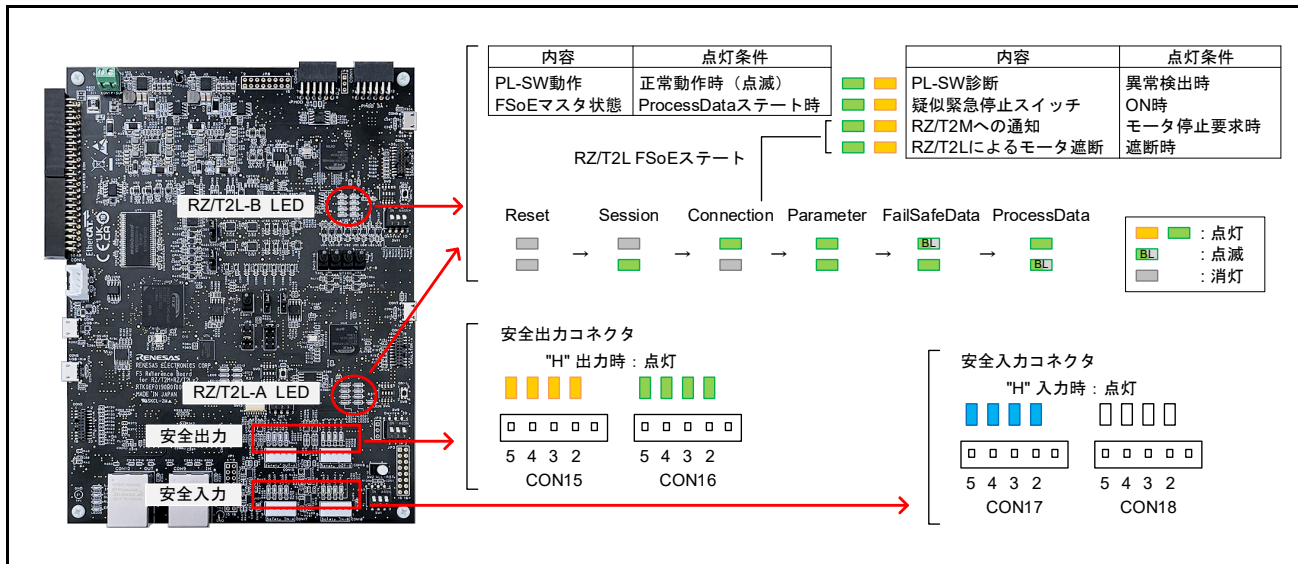


図 7-60 ステータス LED 仕様

安全状態を解除するシステム起動スイッチ押下のタイミングはステータスLEDにより確認できます。

図7-61に、安全状態解除時のLED挙動を示します。

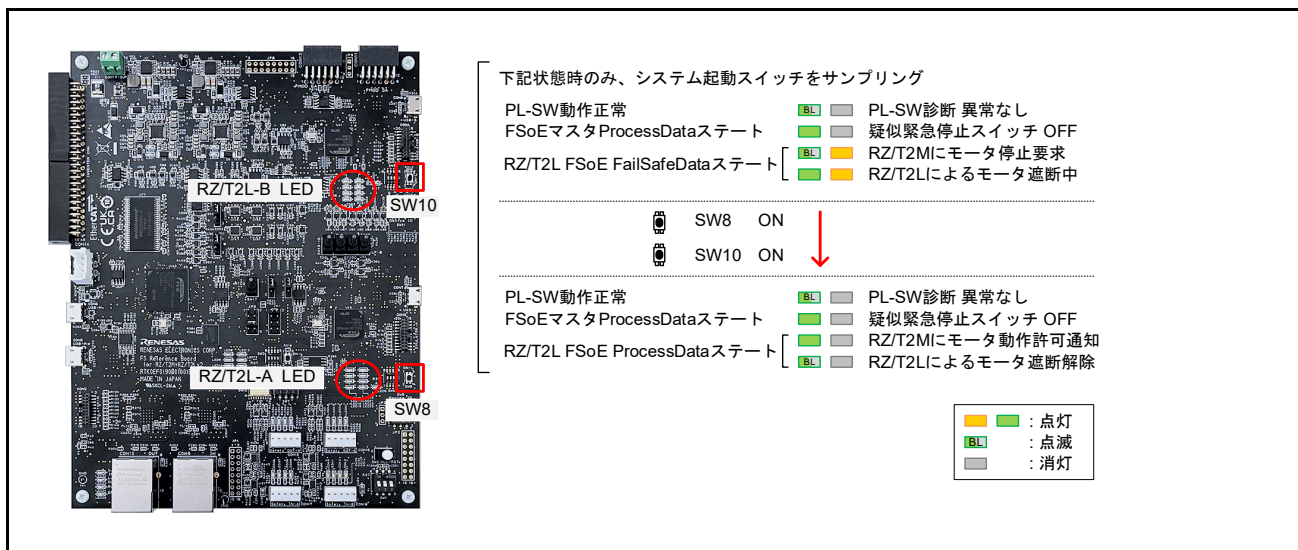


図 7-61 安全状態解除時のLED挙動

## 8. Appendix

### 8.1 統合開発環境インストール

#### 8.1.1 EWARM

IAR システムズ株式会社の web からダウンロードしてインストールしてください。

<https://www.iar.com/products#/search?architecture=Arm>

#### 8.1.2 e<sup>2</sup> studio

下記から最新の e<sup>2</sup> studio をダウンロードしてインストールしてください。

ただし、既に最新の e<sup>2</sup> studio をインストールしている場合は、別フォルダを指定してインストールしてください。

<https://github.com/renesas/rzt-fsp>

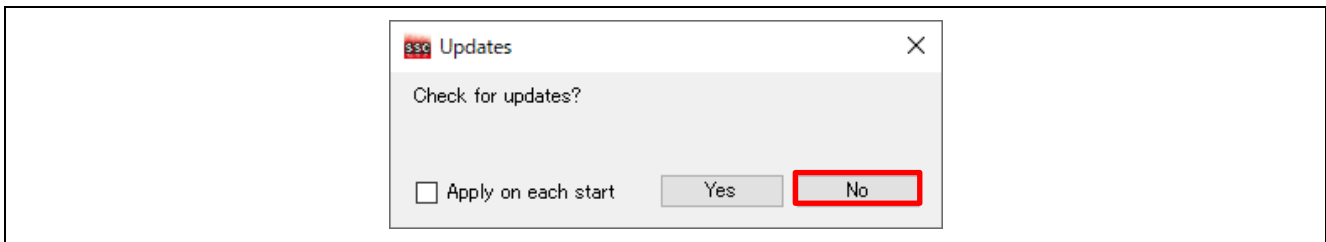
## 8.2 付録 : SSC Tool 使用時の注意点

### ※ SSC Tool を起動する際の注意点

1. SSC Tool は必ず「管理者権限」で起動してください。

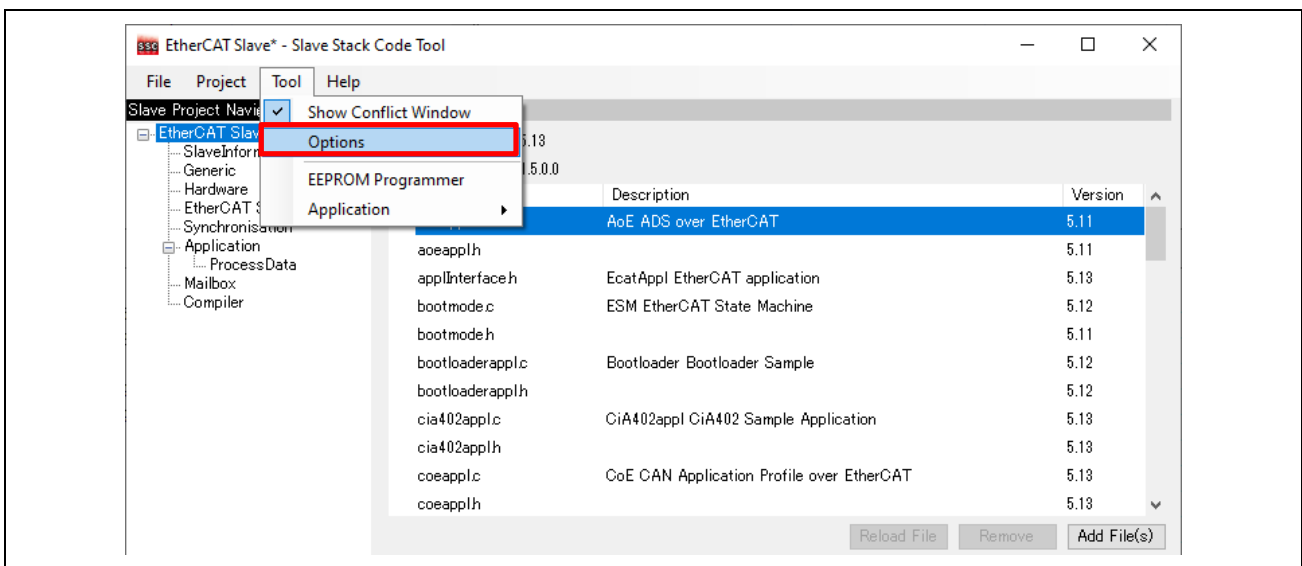
管理者権限で起動しない場合、SSC コードの生成に失敗する可能性があります。

2. 初めて SSC Tool を起動する際、以下のウィンドウが表示される場合があります。  
その場合は、更新の確認を行わないように **[No]** を選択してください。



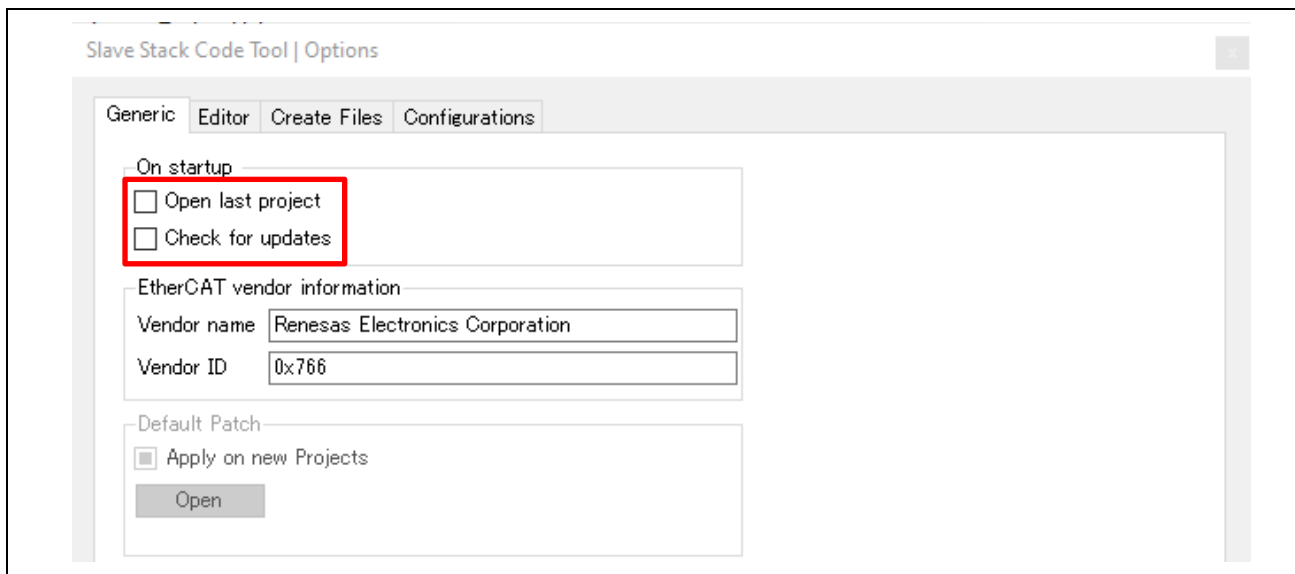
### ※ SSC Tool の設定について

3. メニューから **[Tool]** → **[Options]** を開き、SSC Tool の設定を確認してください。

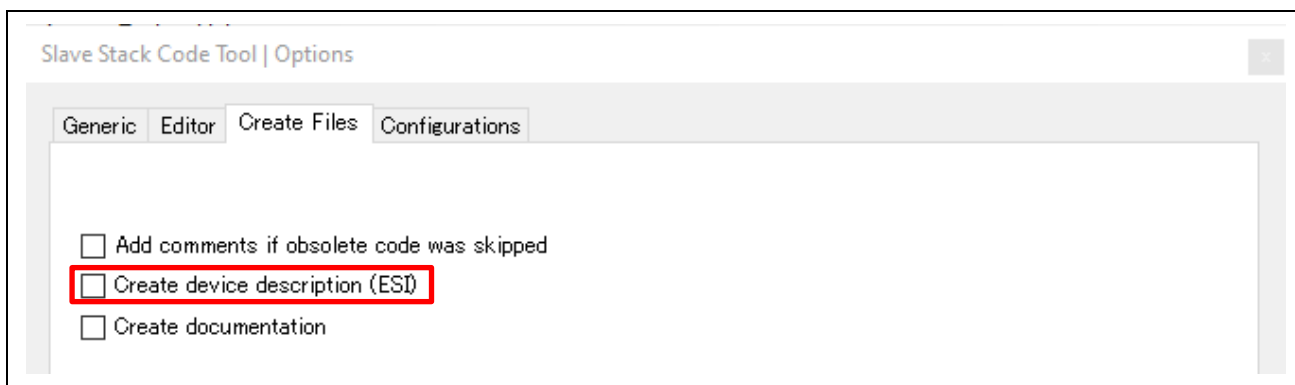


4. **[Generic]** タブ

- **[Open last project]** のチェックを外してください。
- **[Check for updates]** のチェックを外してください。

5. **[Create Files]** タブ

- **[Create device description (ESI)]** のチェックを外してください。



6. 設定を変更したら、**[OK]** をクリックして適用します。  
これで、本サンプルプログラム用の **SSC Tool** のセットアップは完了です。

### 8.3 付録 : patch のインストール方法

patch をインストールする方法は、以下の 2 通りです。

1. Git for Windows (64bit) を使用する方法
2. MinGW Installation Manager を使用する方法

#### 8.3.1 Git for Windows (64bit) を使用する方法

1. Git for Windows の公式サイトからインストーラ (例 : `Git-x.xx.x-64-bit.exe`) をダウンロードします。

[Git for Windows](#)

2. ダウンロードしたインストーラを実行し、セットアップ手順に従ってインストールします。特別な要件がない場合は、デフォルト設定を使用してください。

3. インストール後、`patch.exe` のパスをシステム環境変数に追加します。

- デフォルトのインストールパスは下記の通りです：  
`C:¥Users¥<ユーザー名>¥AppData¥Local¥Programs¥Git¥usr¥bin`

環境変数を更新した後、PC を再起動して変更を適用します。

4. コマンドプロンプトを起動し、`where patch` と入力します。 `patch.exe` のパスが表示されれば、インストール成功です。

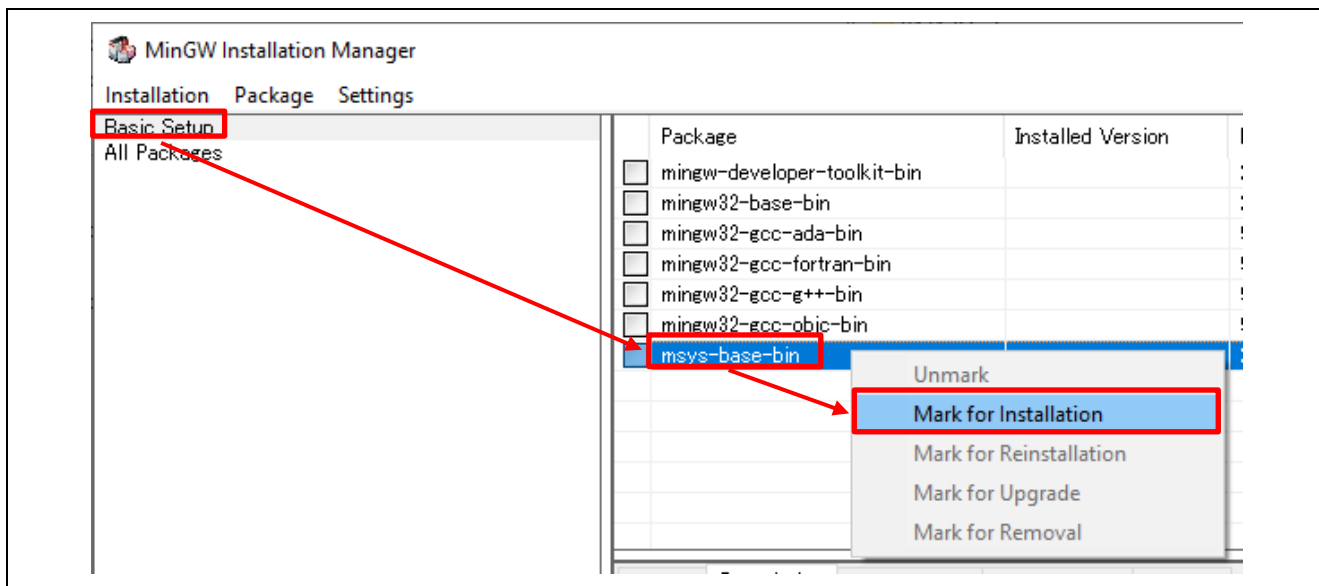
```
C:¥>where patch
C:¥Users¥          ¥AppData¥Local¥Programs¥Git¥usr¥bin¥patch.exe
```

### 8.3.2 MinGW Installation Manager を使用する方法

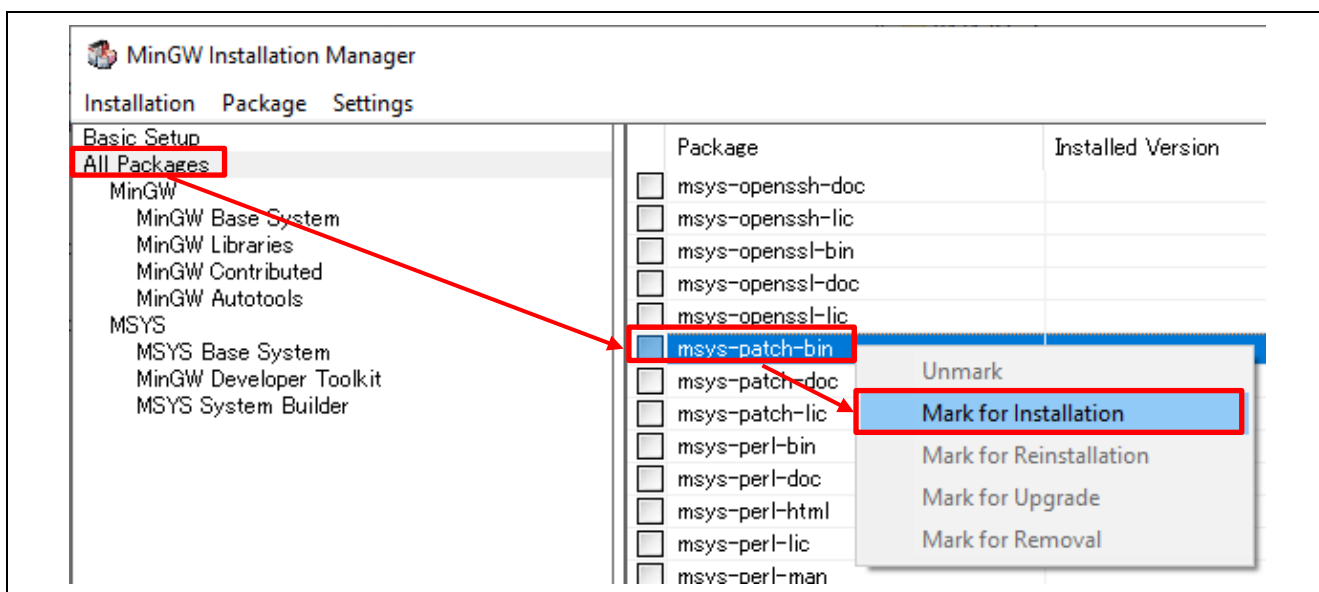
1. 以下の URL から mingw-get-setup.exe をダウンロードします：

<https://sourceforge.net/projects/mingw/>

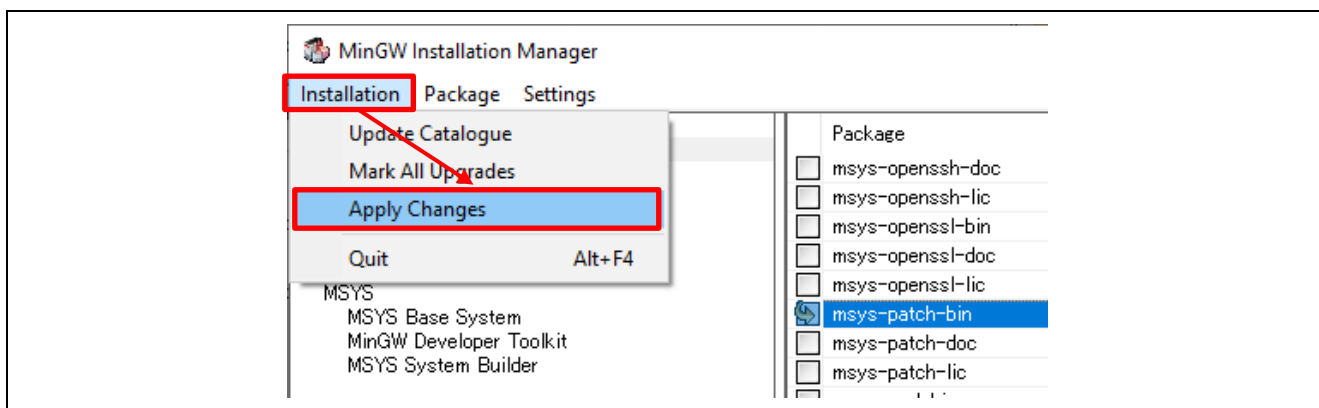
2. mingw-get-setup.exe を実行し、ダイアログに従って「Mingw-installation-manager」をインストールします。
3. インストール完了後、Mingw-installation-manager ウィンドウが表示されたら、左画面で **Basic Setup** を選択し、右画面で **msys-base-bin** を右クリックして「Mark for Installation」を選択します。



4. 次に、左画面で **All Packages** を選択し、右画面で **msys-patch-bin** を右クリックして「Mark for Installation」を選択します。



5. メニューバーの「Installation」から「Apply Changes」を選択します。



6. 「Schedule of Pending Actions」ウィンドウが表示されたら、「Apply」ボタンをクリックします。

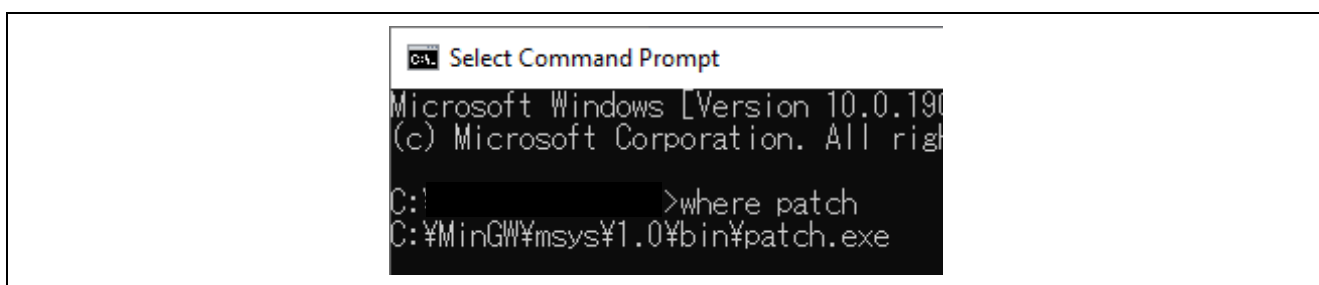
7. 「All changes were applied successfully; you may now close this dialogue.」と表示されれば、patch.exeのインストールは成功です。

8. インストールした patch.exe のパスをシステム環境変数に追加します。

- デフォルトパスの例：  
C:¥MinGW¥msys¥1.0¥bin

環境変数を更新した後、PC を再起動して変更を適用します。

9. コマンドプロンプトを起動し、where patch と入力します。  
インストールした patch.exe のパスが表示されれば問題ありません。



## 8.4 付録：FSoE マスタプログラムの作成手順

本章では、6.2 章で準備する FSoE マスタプログラムと同等のものを作成する手順を示します。

(1) はじめに、TwinSAFE プロジェクト作成手順を示します。

1. TwinCAT3 のシステムマネージャツリーで、[SAFETY] を右クリックします。
2. [Add New Item] を選択し、新規プロジェクトを作成します。
3. [TwinCAT Safety Project Preconfigured ErrAck] を選択します。
4. プロジェクト名を任意で編集します。当ドキュメントでは以降、「safety\_project\_1」をプロジェクト名として説明します。
5. [Add] をクリックします。
6. Target System を「Hardware Safety PLC」に、Programming Language を「Graphical Editor」に設定します。
7. [OK] をクリックします。

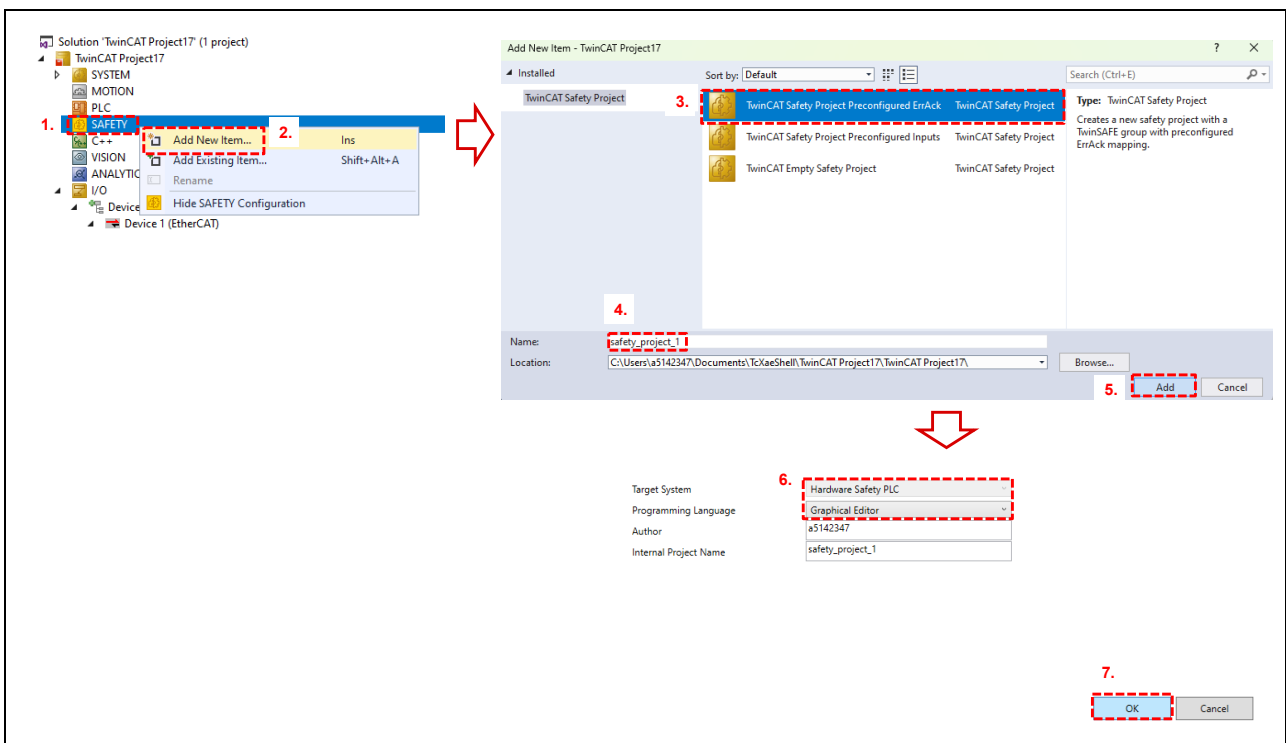


図 8-1. TwinSAFE プロジェクト作成手順

(2) 次に、使用するデバイスのインポートを行います。

1. TwinCAT3 のシステムマネージャツリーで、[SAFETY]→[safety\_project\_1]→[safety\_project\_1 Project]→[TwinSafeGroup1]を順に選択し、その中にある[Alias Devices]を右クリックします。
2. [Import Alias-Device(s) from I/O-configuration]を選択すると、[Select from I/O tree]ウィンドウが表示されます。
3. [Renesas EtherCAT RZ/T2 SafetyMotorSolution]欄のチェックボックスにチェックを入れます。
4. [OK]をクリックします。
5. 正常にインポートできた場合、[Importing of 1 Alias Devices finished]ウィンドウが表示されるので [OK]をクリックします。

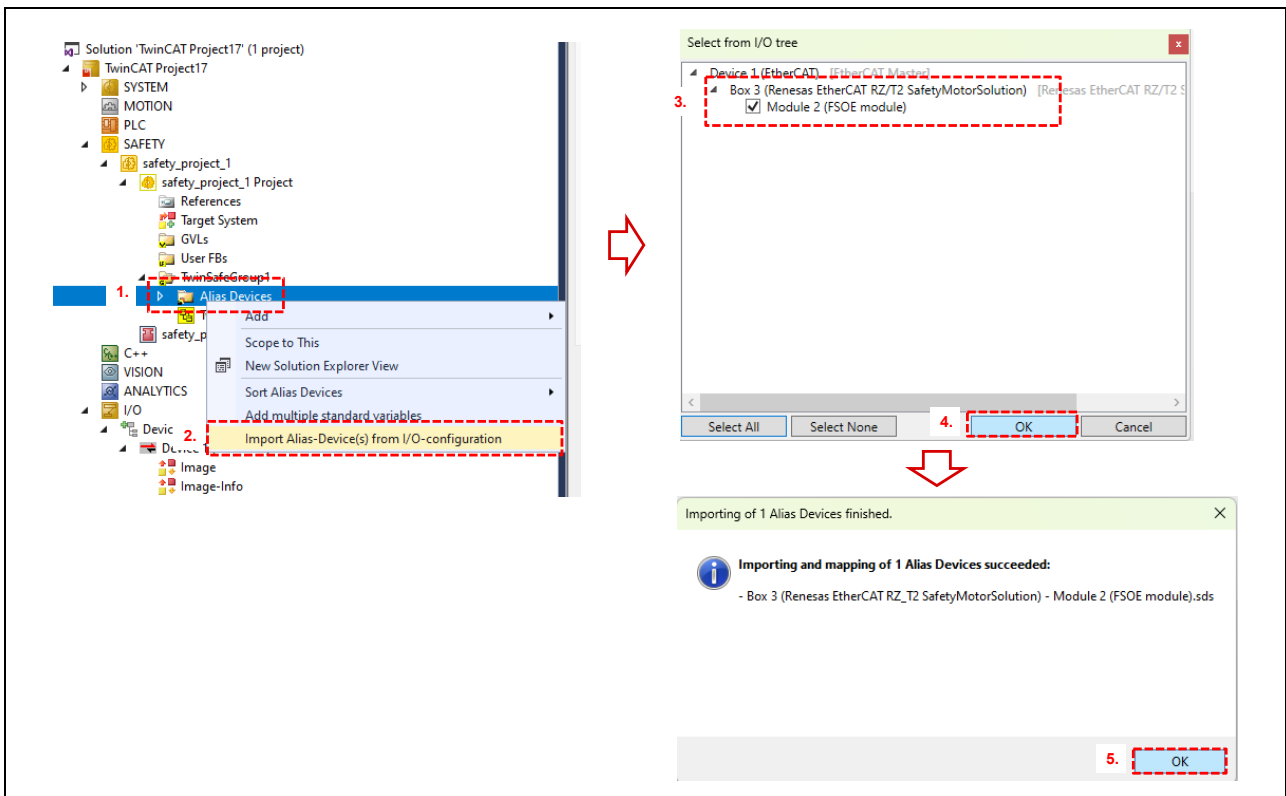


図 8-2. 使用するデバイスのインポート

(3) 次に、TwinSAFE グラフィカルエディタの設定を行います。

1. TwinCAT3 のシステムマネージャツリーで、[SAFETY]→[safety\_project\_1]→[safety\_project\_1 Project]→[TwinSafeGroup1]を順に選択します。
2. [TwinSafeGroup1.sal]を右クリックします。
3. [Open]を選択すると、TwinSAFE グラフィカルエディタが開きます。
4. [Properties Window]ボタンを押し、[Properties]ウィンドウを開きます。
5. [Properties]ウィンドウにおいて、[Info Data]の[Map Diag]と[Map State]の設定値を[True]に変更します。

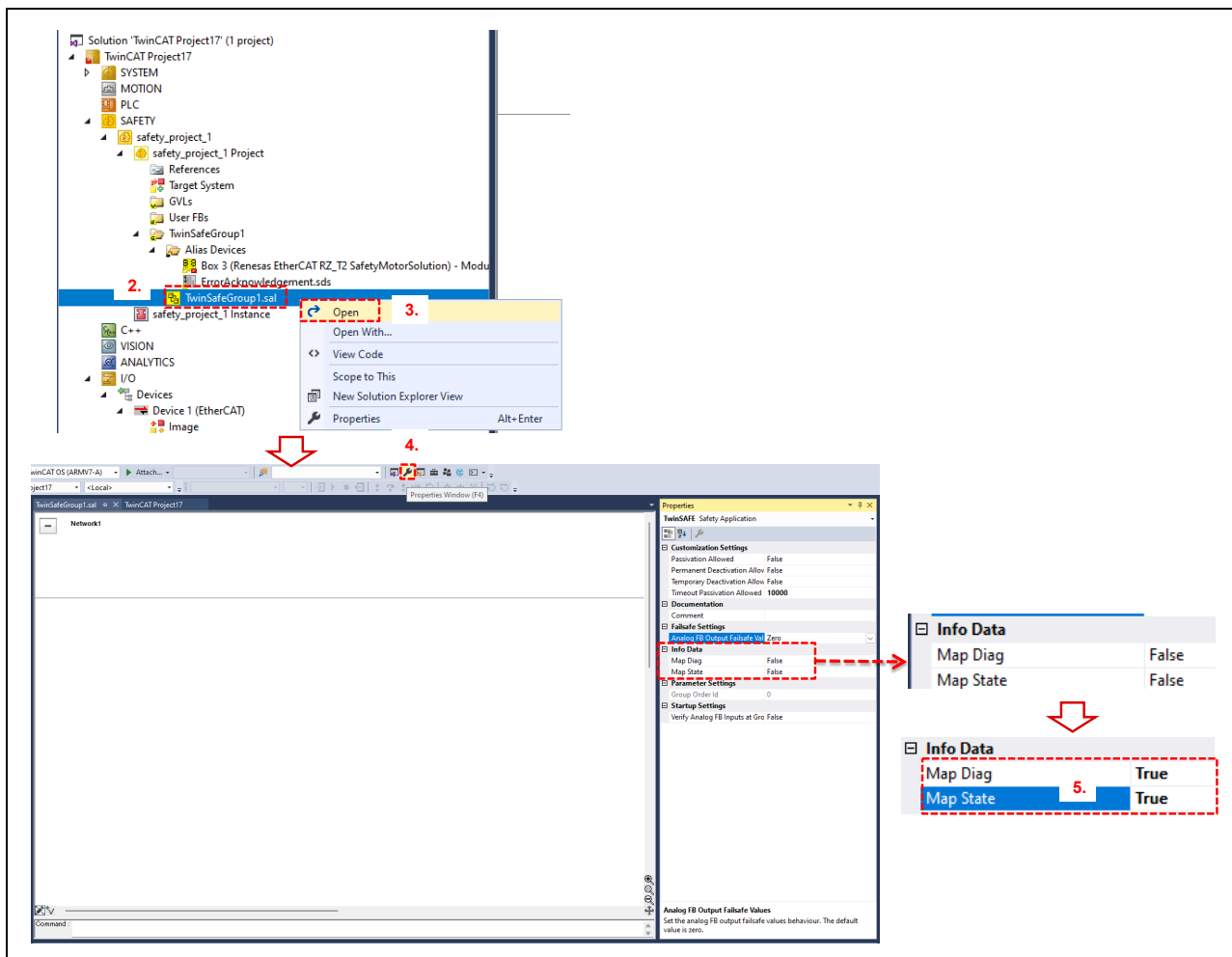


図 8-3. TwinSAFE グラフィカルエディタの設定

(4) 次に、ファンクションブロックの作成を行います。

1. [VIEW]を選択します。
2. [Toolbox]を選択すると、Toolbox ウィンドウが表示されます。

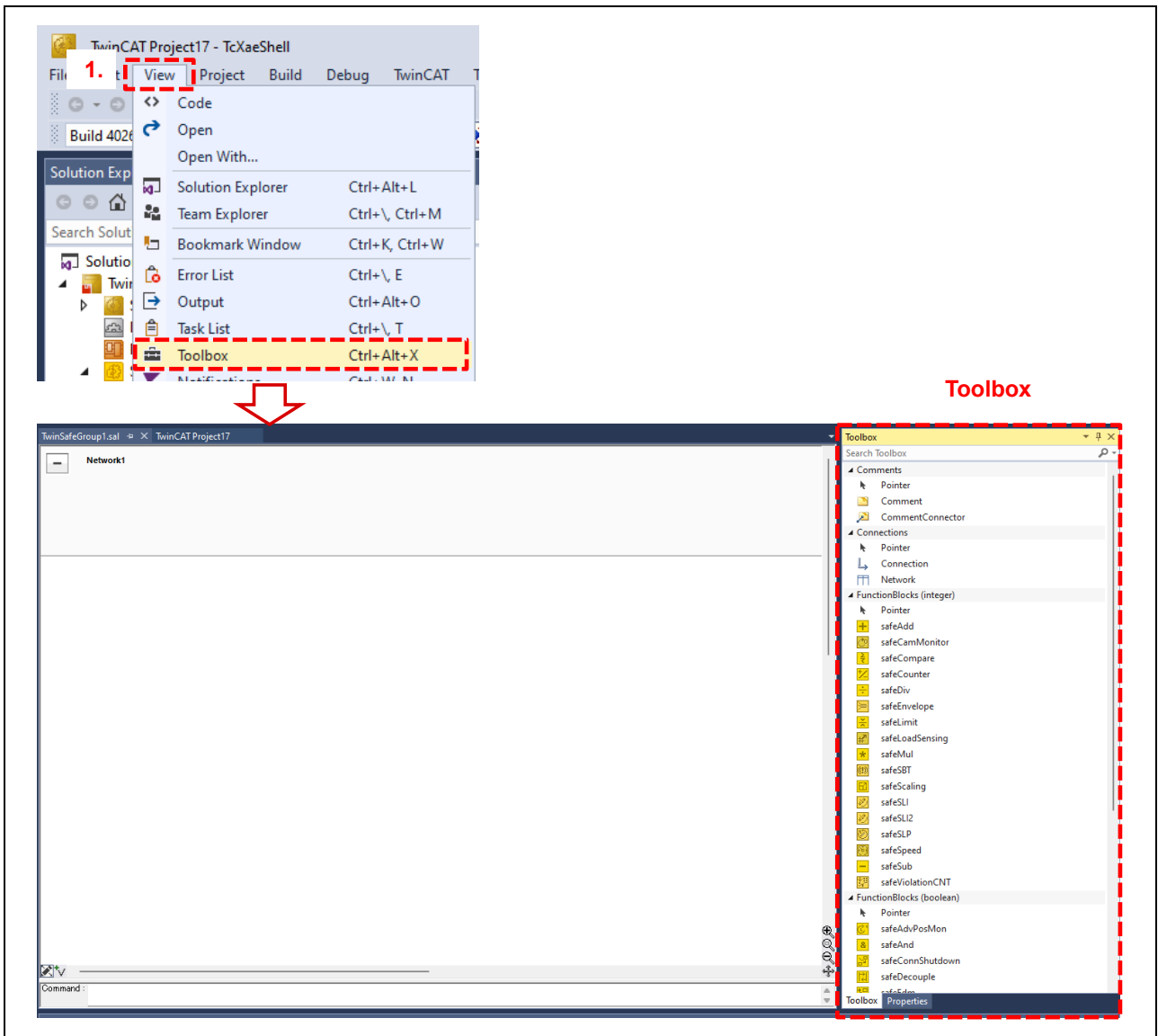


図 8-4. ファンクションブロックの作成 1

3. [FunctionBlocks(boolean)]→[safeDecouple]を選択します。
4. [safeDecouple]を TwinSafeGroup1.sal ウィンドウ内にドラッグ&ドロップすると、ファンクションブロックが生成されます。

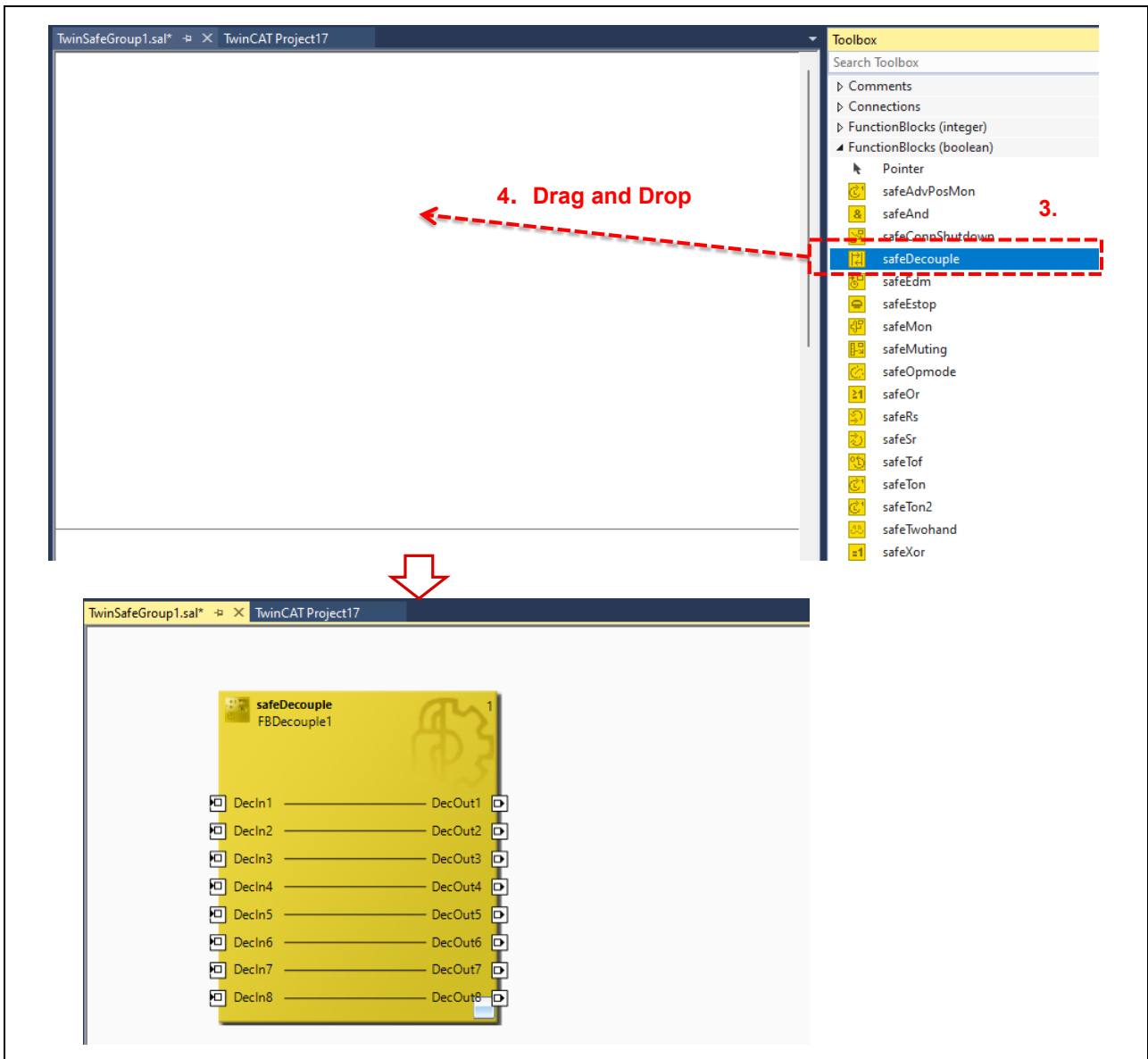


図 8-5. ファンクションブロックの作成 2

(5) 次に、ファンクションブロックの入出力に変数名の割り当てを行います。

1. ファンクションブロック「FBDecouple1」の[Decln1]上で右クリックします。
2. [Properties]を選択します。
3. [Parameter Settings]→[Assigned Variable Name]を[INPUT1]と設定します
4. 同様に[Decln2]には[INPUT2]と設定します。

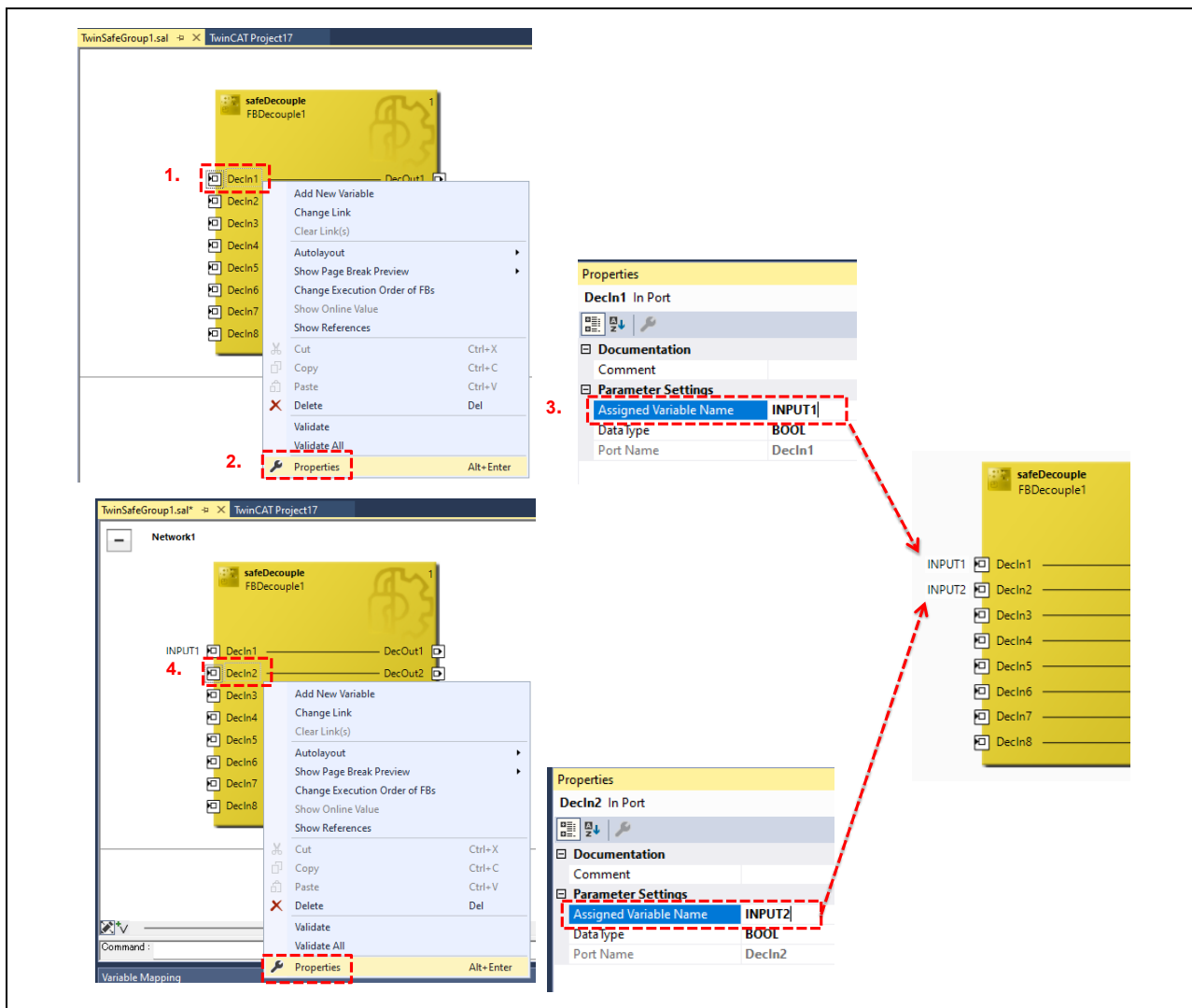


図 8-6. ファンクションブロックの入出力に変数名の割り当て 1

5. 1~4 と同様の方法で、全ての端子に入出力端子名を設定します。

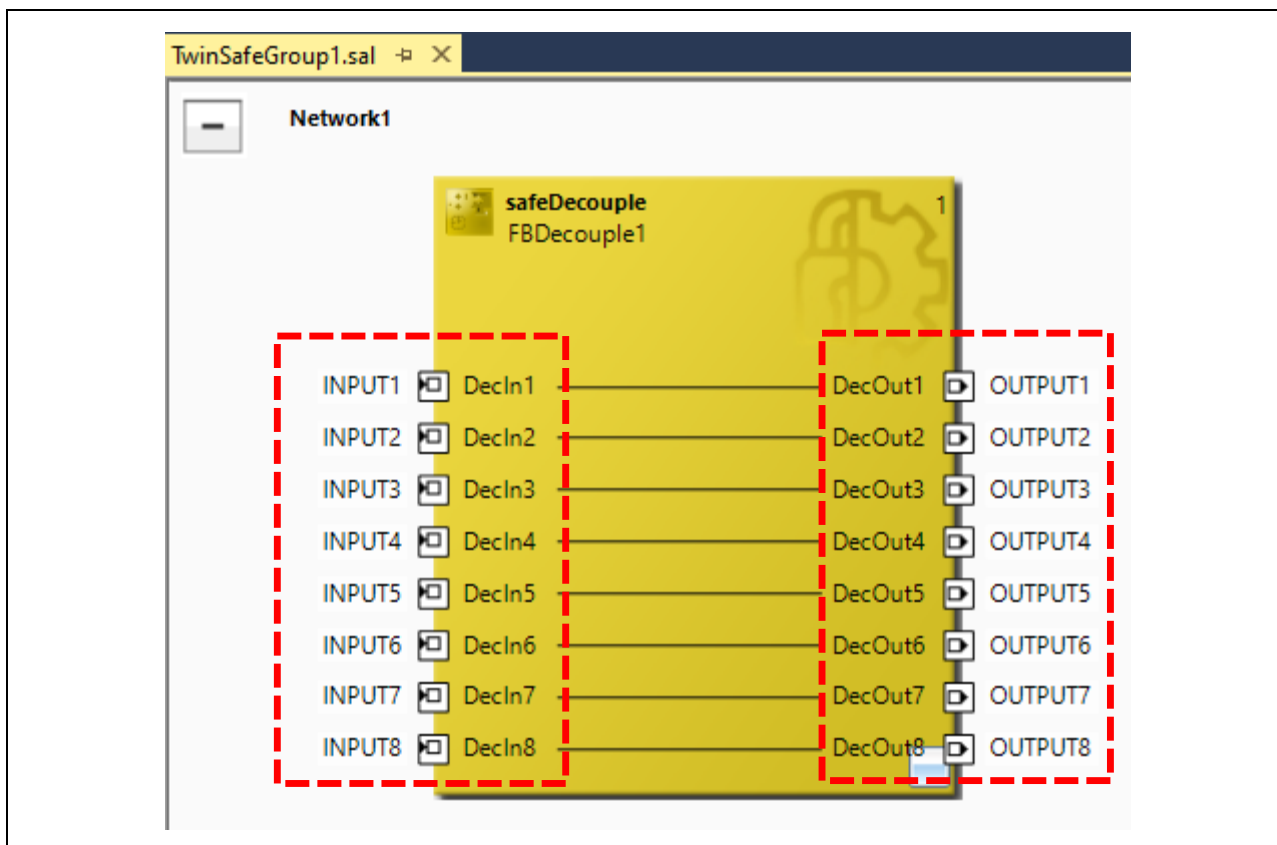


図 8-7. ファンクションブロックの入出力に変数名の割り当て

(6) 次に、ファンクションブロックの変数の割り当てを行います。

1. TwinSAFE グラフィカルエディタの下方にある[Variable Mapping]タブを選択します。
2. [Variables]タブを選択します。
3. INPUT1 欄にある「…」を選択すると、[Map to]ウィンドウが表示されます。
4. [SubIndex 001]を選択します。
5. [OK]をクリックします。
6. 同様に、他の変数についても割り当てを行います。

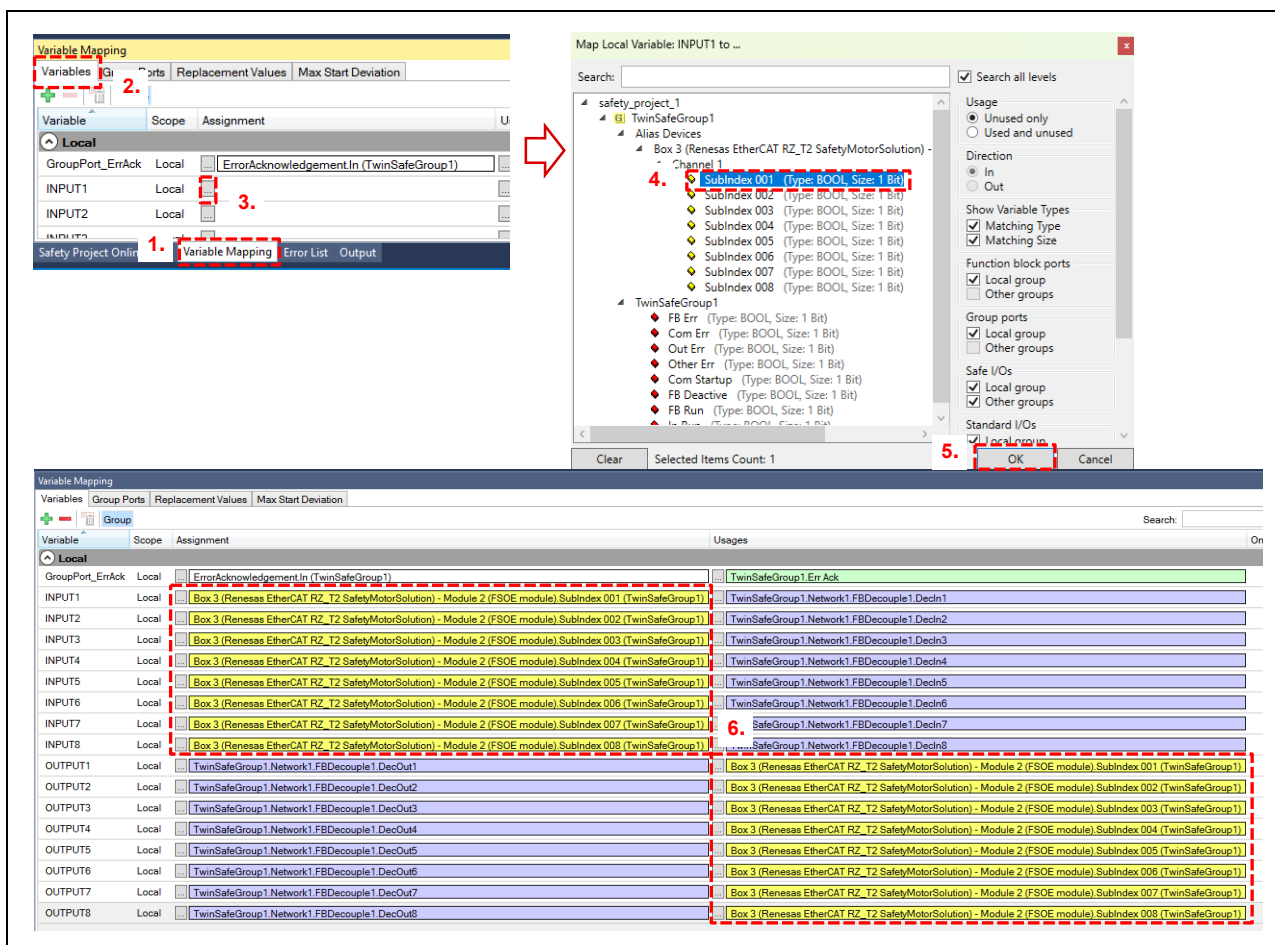


図 8-8. ファンクションブロックの変数の割り当て

(7) 次に、PLC のプロジェクトを作成します。

1. TwinCAT3 のシステムマネージャツリーで、[PLC]を右クリックします。
2. [Add New Item...]を選択し、新規プロジェクトを作成します。
3. [Standard PLC Project]を選択してください。
4. 任意プロジェクト名を記載してください。本資料では、safety\_PLC\_project\_1 というプロジェクト名で以降の説明を行います。
5. [Add]をクリックしてください。

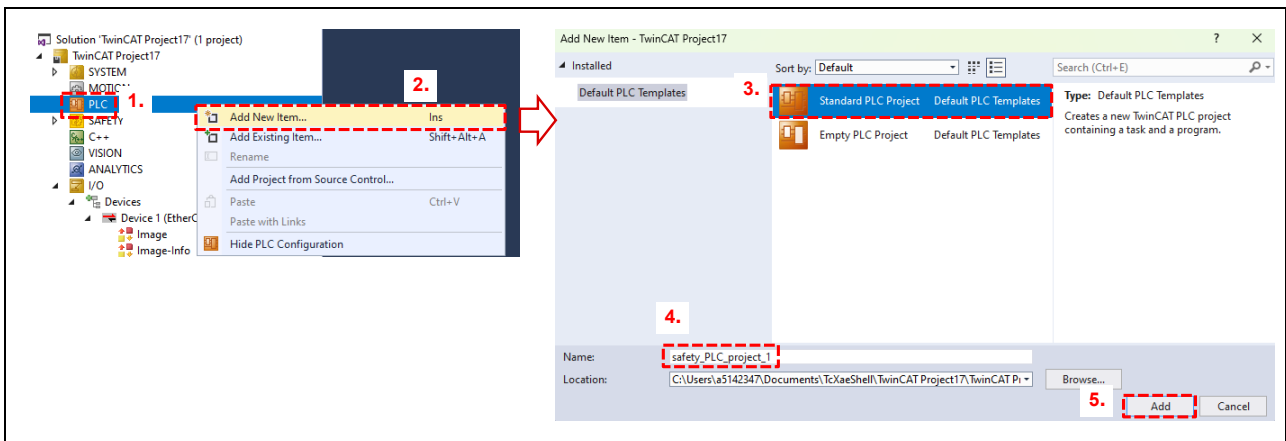


図 8-9. PLC プロジェクトの作成

(8) 次に、PLC プログラムの記述を行います。

1. TwinCAT3 のシステムマネージャツリーで、[PLC]→[safety\_PLC\_project\_1]→[safety\_PLC\_project\_1 project]→[POUs]にある[MAIN(PRG)]を右クリックします。
2. [Open]を選択し、PLC プロジェクトを開きます。
3. 図のプログラムを追記してください。

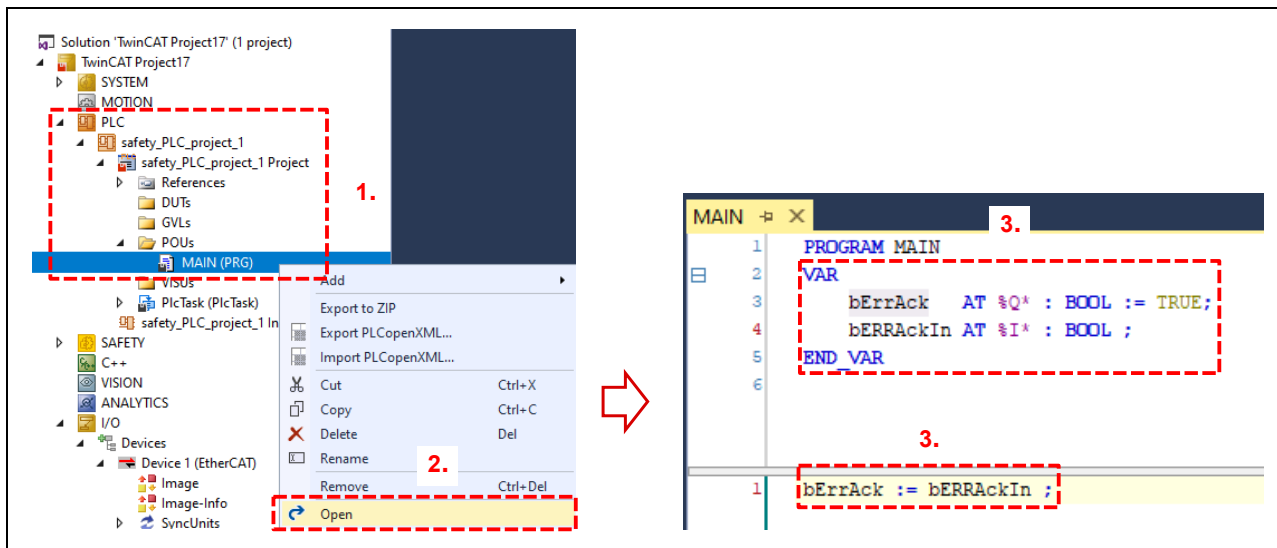


図 8-10. PLC プログラムの記述

(9) 次に、PLC プログラムのビルドを行います。

1. TwinCAT3 のシステムマネージャツリーで、[safety\_PLC\_project\_1 project]を右クリックしてください。
2. [Rebuild]を選択すると、ビルドが実施されます。
3. ビルドが成功すれば[Rebuild All succeeded]と表示され、VAR で宣言した 2 つの変数が Instance 内に生成されます。

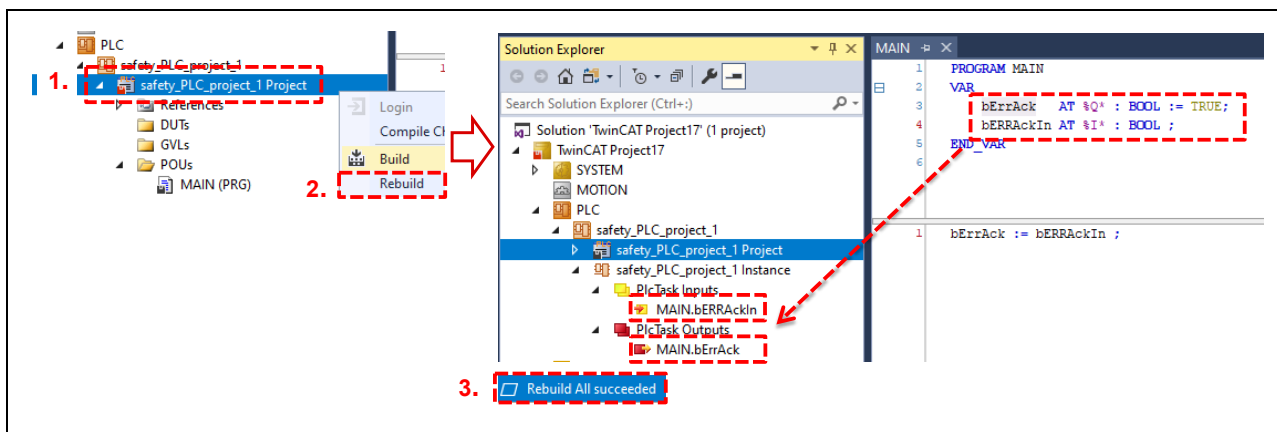


図 8-11. PLC プログラムのビルド

(10) 次に、FSoE マスタ機器の登録設定を行います。

1. TwinCAT3 のシステムマネージャツリーで、[SAFETY]→[safety\_project\_1]を順に選択し、[safety\_project\_1Project]を右クリックします。
2. [Properties]を選択します。
3. [Target System]タブにある[Physical Device]のアイコンをクリックし、Choose physical terminal for mapping ウィンドウを開きます。
4. 接続する FSoE マスタ機器をクリックします。(図では "EL6900" を選択)
5. [OK]をクリックします。
6. 設定が完了するとデバイス情報が更新されます。デバイス情報の「Serial Number」は「」章にて必要となります。
7. [Safe Address]と[Hardware Address]が一致していることを確認します。[Safe Address]と[HardwareAddress]が異なる場合、(a)→(b)の順でアイコンをクリックし、アドレスを一致させてください。

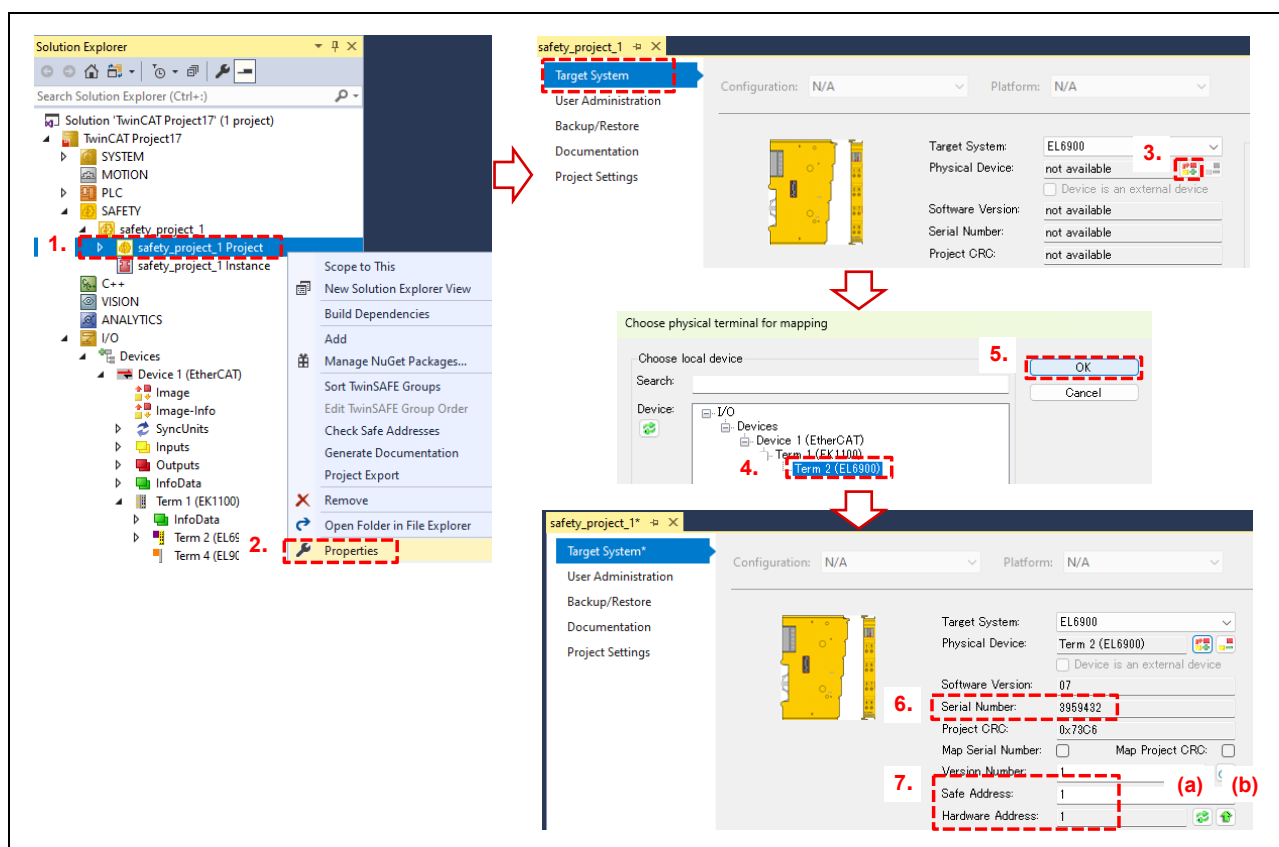


図 8-12. FSoE マスタ機器の登録設定

(11) 次に、FSoE スレーブ機器（Safety Motor Control Reference Kit）の確認を行います。

1. TwinCAT3 のシステムマネージャツリーで、[SAFETY]→[safety\_project\_1]→[safety\_project\_1 project]→[TwinsafeGroup1]→[Alias Devices]を順に選択し、その下にある[Box X(Renesas EtherCAT RZ\_T2 SafetyMotorSolution) - Module X (FSOE module)].sds をダブルクリックしてデバイス情報を確認します。
2. [FSoE Address]と[Dip Switch]の数値がともにスレーブ機器の FSoE アドレスになっていることを確認してください。異なる場合は更新アイコンをクリックして更新してください。（注1）
3. [Full Name]および[Linked to]にデバイス情報が反映されていることを確認してください。[Full Name]および[Linked to]の表示が"not available"となっている場合は、更新アイコンをクリックして更新してください。（注2）

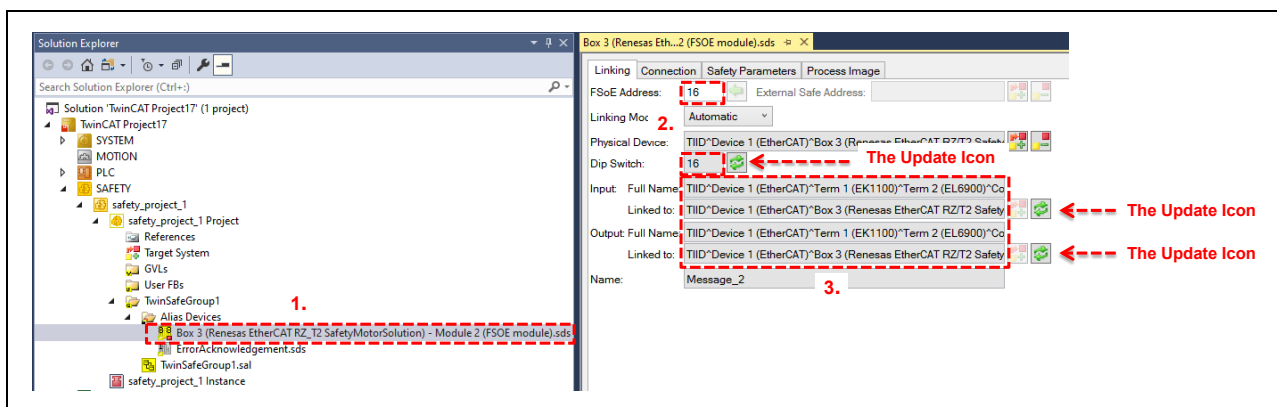


図 8-13. FSoE スレーブ機器（Safety Motor Control Reference Kit）の確認

注1： [Dip Switch]の表示が n.a となっており、更新アイコンをクリックしても表示が更新されない場合、TwinCAT3 側の何らかの原因でスレーブ機器との通信が切断された可能性があります。その場合は、再度 6.1 章からの設定を実施してください。

注2： [Full Name]および[Linked to]の表示が"not available"から更新されない場合は、再度 6.1 章からの設定を実施してください。

(12) 次に、ErrAck 信号の選択を行います。

1. TwinCAT3 のシステムマネージャツリーで、[SAFETY]→[safety\_project\_1]→[safety\_project\_1 project]→[TwinsafeGroup1]→[Alias Devices]を順に選択し、その下にある [ErrorAcknowledgement.sds] をダブルクリックします。
2. [Full Name]横のアイコンをクリックします。
3. MAIN.bErrAck を選択して[OK]をクリックします。

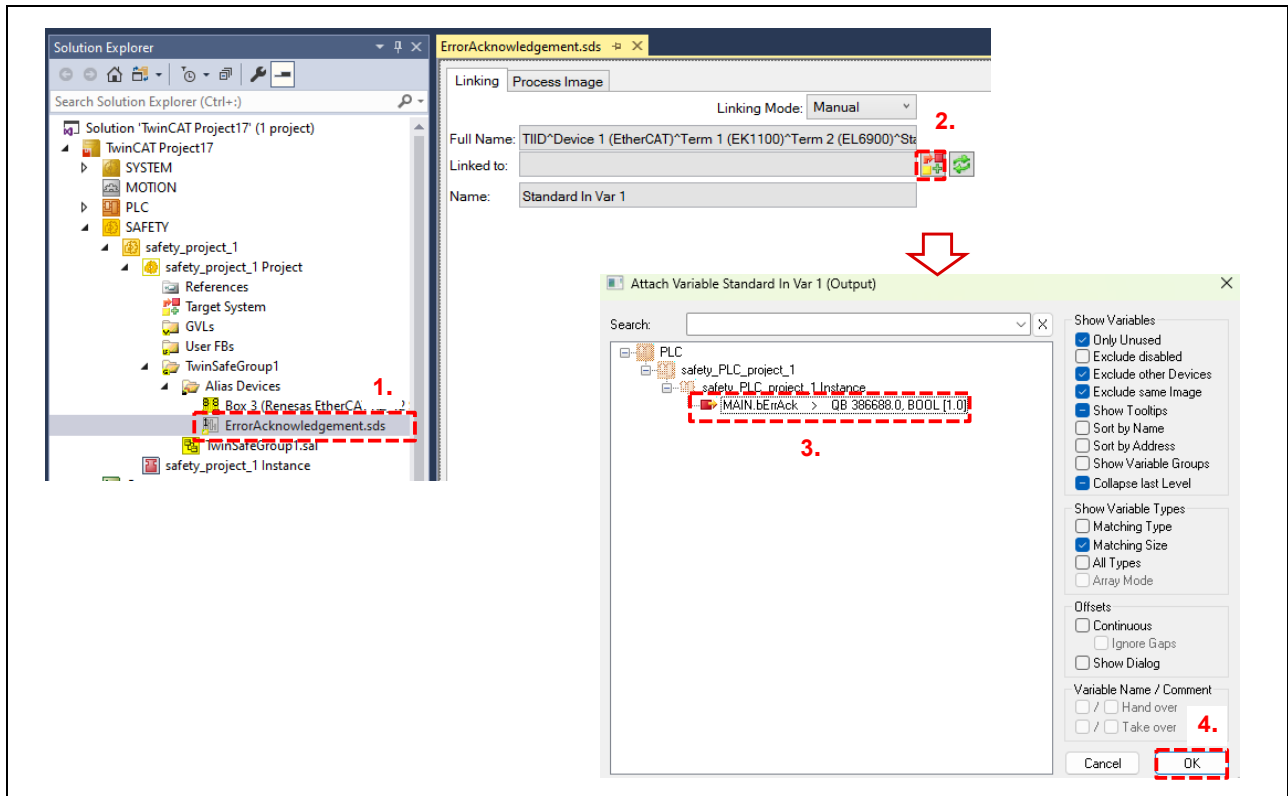
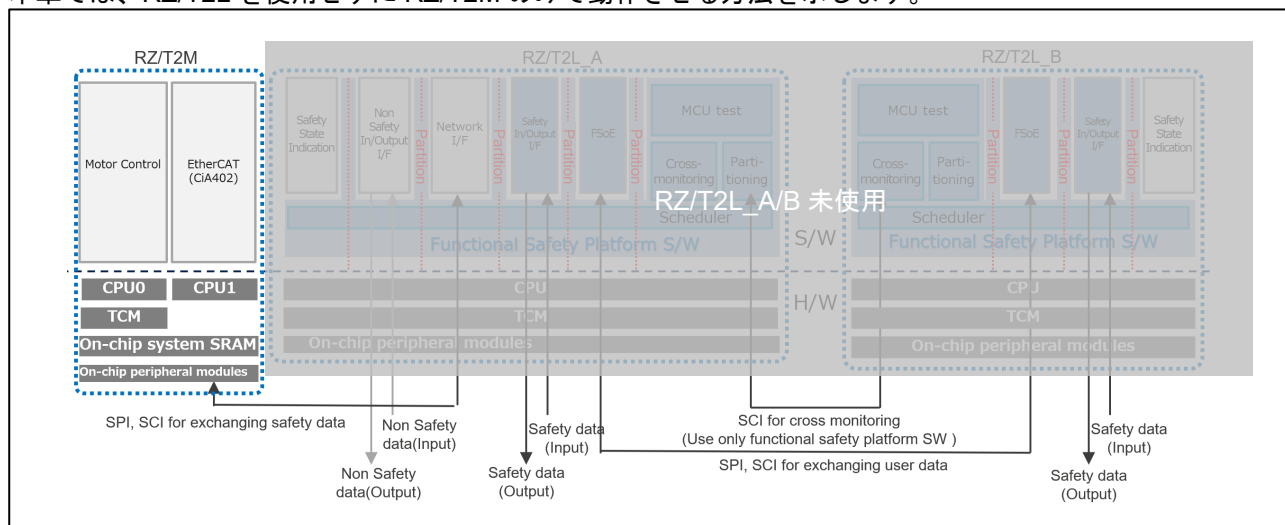


図 8-14. ErrAck 信号の選択

### 8.5 付録：RZ/T2M のみの動作方法

本章では、RZ/T2L を使用せずに RZ/T2M のみで動作させる方法を示します。



#### 8.5.1 接続環境

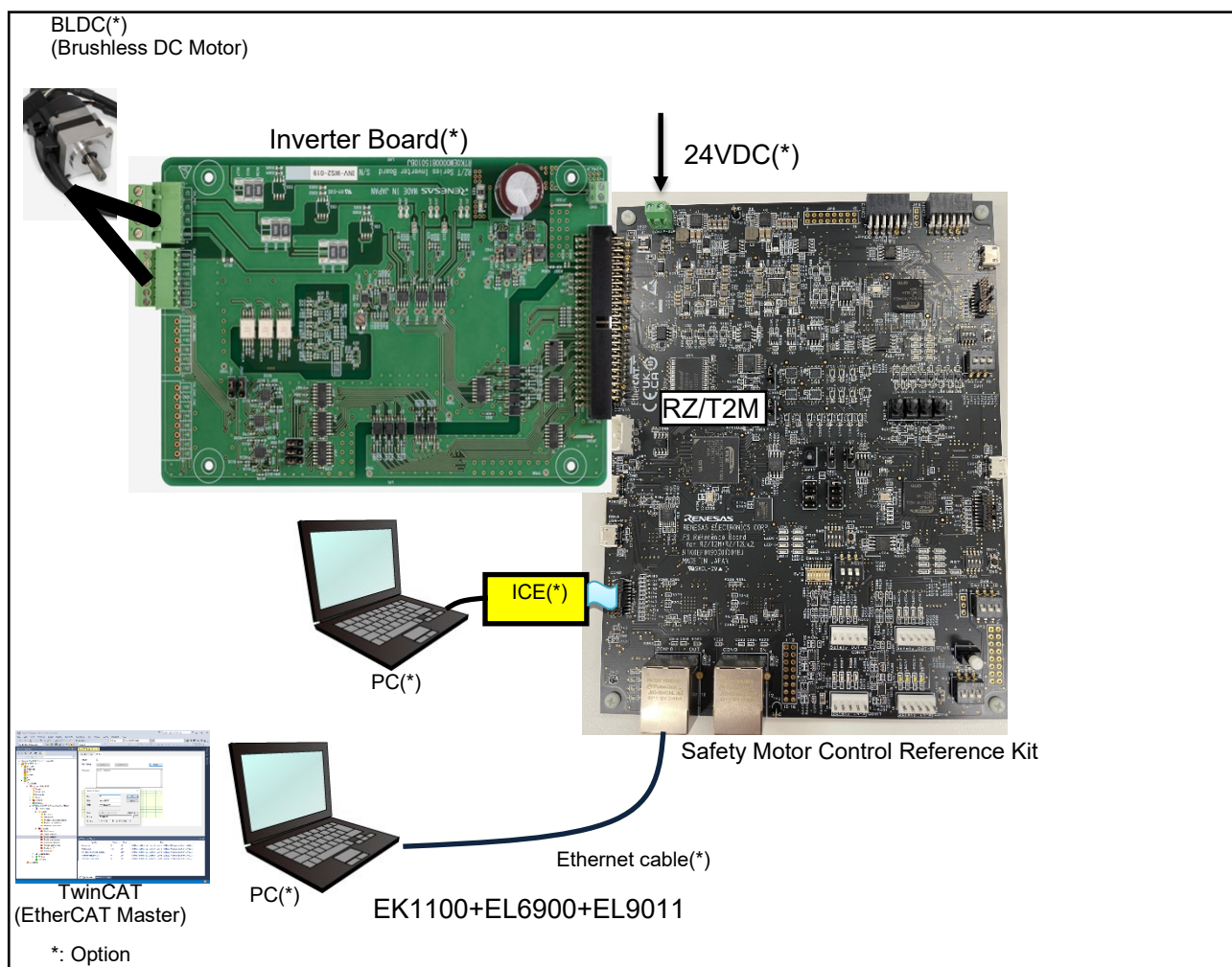


図 8-15 接続環境

## 8.5.2 ボードの設定

### 8.5.2.1 セーフティモータ制御ボードの設定

#### (1) Switches

SW1

Pin	1	2	3
	OFF	ON	ON

セーフティモータ制御ボードのスイッチの詳細は、「RTK0EF0190D01001BJ ユーザーズマニュアル (R30UZ0214)」を参照してください。

#### (2) Jumpers

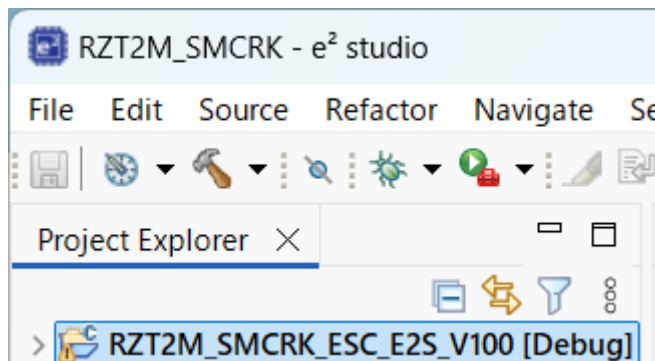
No	JP	設定
5	JP10	2-3 short
6	JP11	2-3 short

セーフティモータ制御ボードのジャンパの詳細は、「RTK0EF0190D01001BJ ユーザーズマニュアル (R30UZ0214)」を参照してください。

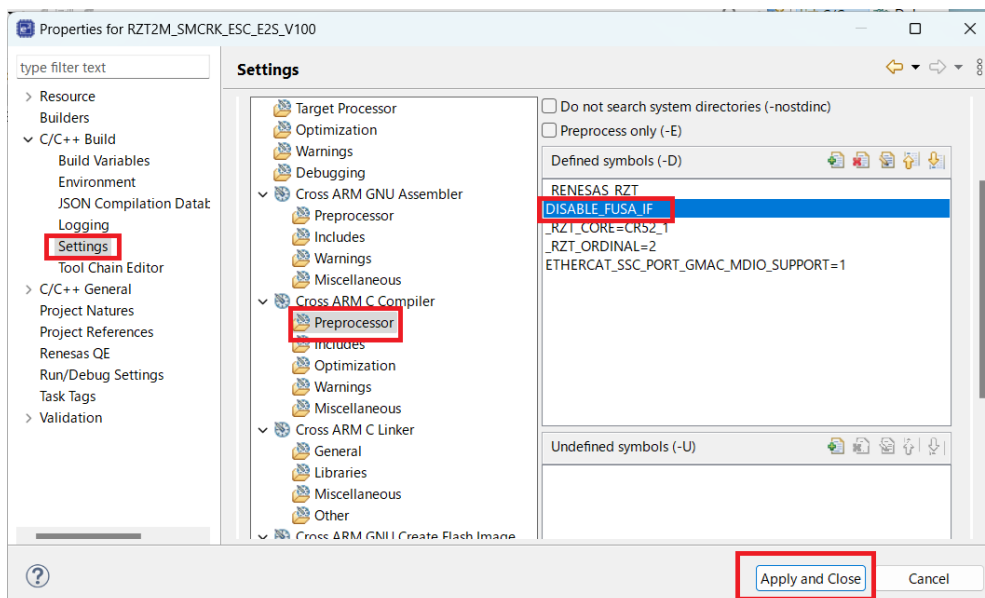
### 8.5.3 RZ/T2M プログラム書き込み

RZ/T2M の CPU1 プロジェクト (RZT2M\_SMCRK\_ESC\_E2S\_V100) をビルドする前にプリプロセッサを変更します。

- ① Project Explorer の "RZT2M\_SMCRK\_ESC\_E2S\_V100" を右クリック -> "property" を選択します。



- ② "Settings" をクリック、"Cross ARM C Compiler" の "Preprocessor" をクリックして、"Defined symbols" に "DISABLE\_FUSA\_IF" を追加します。"Apply and Close" ボタンを押下します。



以上です。

## 改訂記録

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2025.12.19	-	初版作成

## 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。

### 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、 $V_{IL}$  (Max.) から  $V_{IH}$  (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、 $V_{IL}$  (Max.) から  $V_{IH}$  (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えば、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.

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

○EtherCAT® および TwinCAT®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。

○OFA-CODER は、多摩川精機株式会社の商標です

○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。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、お客様の責任において、お客様の機器・システムを設計ください。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. 当社製品または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. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、変更、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、変更、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。
6. 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。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア/ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限られません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因したまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア/ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
  8. 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。
  9. 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。
  10. 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。
  11. 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。
  12. 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いたします。
  13. 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。
  14. 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.5.0-1 2020.10)

## 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](http://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/](http://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。