

# R-IN32M4-CL3

## ユーザーズ・マニュアル

### CC-Link IE TSN 編

本資料に記載の全ての情報は本資料発行時点のものであり、ルネサス エレクトロニクスは、予告なしに、本資料に記載した製品または仕様を変更することがあります。  
ルネサス エレクトロニクスのホームページなどにより公開される最新情報をご確認ください。

## ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
  2. 当社製品、本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
  3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
  4. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
  5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。  
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等  
高品質水準： 輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等  
当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
  6. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
  7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
  8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
  9. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
  10. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものいたします。
  11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
  12. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。
- 注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。
- 注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

(Rev.4.0-1 2017.11)

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

[www.renesas.com](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。

## 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イやマガジンケース、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っていると、フラッシュメモリ、レイアウトパターンなどの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

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

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

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

○CC-Link IE Field、CC-Link IE TSN 及び、SLMP は三菱電機株式会社の登録商標です。

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

# このマニュアルの使い方

## 1. 目的と対象者

このマニュアルは「R-IN32M4-CL3」の CC-Link IE TSN のリモートデバイス局通信機能をユーザに理解していただくためのマニュアルです。このマニュアルを使用するには、電気回路、論理回路、マイクロコンピュータに関する基本的な知識が必要です。

本マイコンは、注意事項を十分確認の上、使用してください。注意事項は、各章の本文中、各章の最後、注意事項の章に記載しています。

改訂記録は旧版の記載内容に対して訂正または追加した主な箇所をまとめたものです。改訂内容すべてを記録したものではありません。詳細は、このマニュアルの本文でご確認ください。

**関連資料** 関連資料は暫定版の場合がありますが、この資料では「暫定」の表示をしておりません。あらかじめご了承ください。また各コアの開発・企画段階で資料を作成しているため、関連資料は個別のお客様向け資料の場合があります。当社ホームページより最新版をダウンロードして参照ください。

### R-IN32M4-CL3 に関する資料

資料名	資料番号
R-IN32M4-CL3 ユーザーズ・マニュアル ハードウェア編	R18UZ0073JJ****
R-IN32M4-CL3 ユーザーズ・マニュアル Gigabit Ethernet PHY 編	R18UZ0075JJ****
R-IN32M4-CL3 ユーザーズ・マニュアル ボード設計編	R18UZ0074JJ****
R-IN32M4-CL3 ユーザーズ・マニュアル CC-Link IE TSN 編	本マニュアル
R-IN32M4-CL3 ユーザーズ・マニュアル CC-Link IE Field 編	R18UZ0071JJ****
R-IN32M4-CL3 プログラミング・マニュアル ドライバ編	R18UZ0076JJ****
R-IN32M4-CL3 プログラミング・マニュアル OS 編	R18UZ0072JJ****

## 2. 数や記号の表記

データ表記の重み：左が上位桁、右が下位桁

アクティブ・ローの表記：

xxxZ (端子、信号名称のあとに Z)

または xxx\_N (端子、信号名称のあとに \_N)

または xxnx (端子、信号名称に n を含む)

注：

本文中につけた注の説明

注意：

気をつけて読んでいただきたい内容

備考：

本文の補足説明

数の表記：

2 進数 … xxxx, xxxxB または n'bxxxx(n ビット)

10 進数 … xxxx

16 進数 … xxxxH または n'hxxxx(n ビット)

2 のべき数を示す接頭語 (アドレス空間、メモリ容量)：

K (キロ) …  $2^{10} = 1024$

M (メガ) …  $2^{20} = 1024^2$

G (ギガ) …  $2^{30} = 1024^3$

データ・タイプ：

ワード … 32 ビット

ハーフワード … 16 ビット

バイト … 8 ビット

### 【備考】

- 通信速度を表す場合は  $10^6$  を示します。

10Mbps =  $10 \times 10^6$ bps

100Mbps =  $100 \times 10^6$ bps

# 目次

1.	はじめに.....	1
1.1	用語.....	1
1.2	総称/略称.....	3
1.3	関連マニュアル.....	4
1.4	CC-Link 協会のご案内.....	4
2.	概要.....	6
2.1	開発の特長.....	6
2.2	R-IN32M4-CL3 の主な仕様.....	6
2.3	R-IN32M4-CL3 適用製品の通信仕様.....	7
2.3.1	CC-Link IE TSN Class A の動作時の注意事項.....	8
2.3.2	サンプルコードのバージョンによる機能の制約.....	8
2.4	サンプルコードのフォルダ構成.....	9
2.5	サンプルコードの概要.....	10
2.6	サンプル CSP+ファイルの概要.....	11
2.7	システム構成.....	12
2.8	R-IN32M4-CL3 適用製品が対応しているプロトコル.....	14
3.	開発前の仕様検討と準備.....	15
3.1	MAC アドレスの取得.....	16
3.2	ベンダーコードの取得と機種タイプの選択.....	16
3.3	H/W スイッチに接続する端子.....	16
3.4	S/W 開発の準備.....	17
3.4.1	S/W 開発の手順.....	17
3.4.2	開発環境.....	18
3.4.3	フラッシュローダのプログラム変更.....	19
3.4.4	コンパイルスイッチ.....	20
3.5	IP アドレスの設定方法の検討.....	21
3.6	異常時のサイクリック出力保持/クリア機能の対応検討.....	23
3.7	エンジニアリングツール各種機能への対応検討.....	23
3.7.1	デバイス局パラメータ処理・コマンド実行.....	23
3.8	CSP+および関連ファイルの作成準備.....	24
3.9	コンFORMANCEテスト受験の準備.....	26
3.9.1	1000BASE-T コンプライアンステスト.....	26
4.	R-IN32M4-CL3 適用製品のできること.....	27
4.1	サイクリック伝送機能.....	28
4.2	トランジェント伝送機能.....	29
4.2.1	トランジェント伝送のクライアント・サーバ機能.....	29
4.2.2	SLMP の伝送フレームについて.....	30
4.2.3	SLMP のフレーム種別について.....	30
4.2.4	SLMP のコマンドについて.....	30
4.3	状態表示機能.....	33
4.3.1	LED による状態表示.....	33
4.3.2	SD/SDRD1、RD/SDRD2 LED の点灯モード設定.....	34
4.3.3	LED の制御.....	34
4.3.4	USER LED の制御.....	35
4.3.5	L ERR. LED の制御.....	35
4.4	CC-Link IE TSN 診断.....	36
4.5	ネットワーク同期通信.....	37

4.5.1	ネットワーク同期通信時の同期信号出力.....	40
4.6	CANopen 通信 .....	41
4.6.1	CANopen 通信時の増設ユニット .....	43
4.7	安全通信時の安全 PDU 送受信 .....	44
4.8	SLMP 経由の通信速度・CC-Link IE TSN Class 設定 .....	46
4.9	デバイス局の IP アドレス設定 .....	47
4.10	デバイス局のパラメータ設定・コマンド実行 .....	48
4.11	SNMP コミュニティ名設定 .....	51
5.	ユーザプログラムの作成 .....	53
5.1	ユーザプログラム一覧 .....	53
5.2	ユーザプログラムのタスク .....	57
5.2.1	イニシャルタスク .....	60
5.2.2	アイドルタスク .....	61
5.2.3	定周期処理タスク .....	63
5.2.4	エラー管理タスク .....	65
5.2.5	同期タイミング処理タスク .....	66
5.2.6	同期解列時処理タスク .....	68
5.2.7	スケジューラタスク .....	70
5.2.8	GPIO 通信送信タスク .....	71
5.2.9	GPIO 通信応答受信タスク .....	72
5.2.10	GPIO 通信受信タスク .....	73
5.2.11	RT DMA 転送完了タスク .....	74
5.2.12	サイクリックフレーム送信タスク(CC-Link IE TSN Class A) .....	75
5.3	ユーザプログラム詳細(イニシャル関連) .....	76
5.3.1	初期化处理 .....	76
5.3.2	通信開始処理 .....	79
5.3.3	SLMP 受信初期化处理 .....	80
5.3.4	SLMP 要求フレーム作成初期化处理 .....	81
5.3.5	パラメータオペレーション初期化处理 .....	81
5.4	ユーザプログラム詳細(サイクリック伝送関連) .....	83
5.4.1	サイクリック受信処理 .....	83
5.4.2	自局状態送信処理 .....	87
5.4.3	サイクリック送信処理 .....	88
5.4.4	通信状態更新処理 .....	92
5.4.5	サイクリック伝送状態更新処理 .....	93
5.5	ユーザプログラム詳細(状態管理およびトランジェント伝送関連) .....	94
5.5.1	自局エラー処理 .....	94
5.5.2	イベント処理 .....	94
5.5.3	LED 更新処理 .....	95
5.5.4	MIB(統計)情報取得処理 .....	96
5.5.5	SLMP 受信処理 .....	97
5.5.6	SLMP 送信処理 .....	99
5.5.7	SLMP 要求フレーム作成処理 .....	99
5.5.8	SLMP ST(3E)応答フレーム受信処理 .....	100
5.5.9	R-IN32M4-CL3 ドライバ内検出 Fatal エラー確認処理 .....	101
5.6	ユーザプログラム詳細(SLMP コマンド実行関連) .....	103
5.6.1	SLMP メモリ読み出し要求コマンド受信処理 .....	103
5.6.2	SLMP メモリ書き込み要求コマンド受信処理 .....	104
5.6.3	SLMP メモリ読み出し要求コマンド作成処理 .....	105
5.6.4	SLMP メモリ読み出し応答コマンド受信処理 .....	106
5.6.5	SLMP リモトリセット要求コマンド受信処理 .....	107
5.6.6	SLMP インディケータ表示要求コマンド受信処理 .....	108
5.6.7	SLMP IP アドレス変更要求コマンド受信処理 .....	109

5.6.8	SLMP エラー履歴クリア要求コマンド受信処理 .....	113
5.6.9	SLMP ネットワーク時刻オフセット配信コマンド受信処理 .....	114
5.6.10	SLMP ネットワーク時刻配信コマンド受信処理 .....	115
5.6.11	SLMP ウォッチドッグカウンタ情報設定要求コマンド受信処理 .....	116
5.6.12	SLMP SetCommunityName 要求コマンド受信処理 .....	117
5.7	ユーザプログラム詳細(デバイス局パラメータ自動設定関連) .....	120
5.7.1	SLMP 通信設定取得要求コマンド受信処理 .....	120
5.7.2	SLMP パラメータ配信可否チェック要求コマンド受信処理 .....	121
5.7.3	SLMP パラメータ配信可否チェック処理 .....	122
5.7.4	SLMP リストア開始通知要求コマンド受信処理 .....	123
5.7.5	SLMP リストア終了通知要求コマンド受信処理 .....	124
5.7.6	SLMP パラメータデータ書き込み要求コマンド受信処理 .....	125
5.7.7	SLMP パラメータデータ書き込み処理 .....	126
5.8	ユーザプログラム詳細(SLMP 経由の通信速度・CC-Link IE TSN Class 設定関連) .....	127
5.8.1	SLMP 接続機器検出(拡張)要求コマンド受信処理 .....	128
5.8.2	SLMP 機能設定(サポート情報取得)要求コマンド受信処理 .....	130
5.8.3	SLMP 機能設定読出し(通信速度)要求コマンド受信処理 .....	132
5.8.4	SLMP 機能設定書き込み(通信速度)要求コマンド受信処理 .....	133
5.8.5	SLMP 機能設定読出し(CC-Link IE TSN Class)要求コマンド受信処理 .....	134
5.8.6	SLMP 機能設定書き込み(CC-Link IE TSN Class)要求コマンド受信処理 .....	135
5.9	ユーザプログラム詳細(CANopen 通信関連) .....	136
5.9.1	CANopen 通信機能の初期化处理 .....	139
5.9.2	サイクリック受信処理(RPDO 更新) .....	142
5.9.3	サイクリック送信処理(TPDO 更新) .....	143
5.9.4	SLMP ReadObject 要求コマンド受信処理 .....	144
5.9.5	SLMP WriteObject 要求コマンド受信処理 .....	145
5.9.6	SLMP ObjectSubIDReadBlock 要求コマンド受信処理 .....	146
5.9.7	SLMP ObjectSubIDWriteBlock 要求コマンド受信処理 .....	147
5.9.8	SLMP NMTStateUpload 要求コマンド受信処理 .....	148
5.9.9	SLMP NMTStateDownload 要求コマンド受信処理 .....	149
5.9.10	SLMP 終了コード取得処理 .....	150
5.9.11	CANopen パラメータ設定処理 .....	152
5.9.12	Index1010 書き込み処理 .....	153
5.9.13	Index1011 書き込み処理 .....	154
5.10	ユーザプログラム詳細(MCU 間 I/F 関連) .....	155
5.10.1	MCU 間 I/F 初期化处理 .....	161
5.10.2	GPIO 通信受付処理 .....	162
5.10.3	GPIO 通信受付クリア処理 .....	162
5.10.4	GPIO 通信イベント種別/コード設定処理 .....	163
5.10.5	安全 PDU 転送処理(MCU 内部→外部) .....	163
5.10.6	安全 PDU 転送処理(MCU 内部←外部) .....	164
5.10.7	安全 PDU 転送準備完了(MCU 内部→外部)受付受信処理 .....	164
5.10.8	安全 PDU 転送準備完了(MCU 内部←外部)受付受信処理 .....	165
5.10.9	安全 PDU 転送完了処理(MCU 内部←外部) .....	166
5.11	ユーザプログラム詳細(H/W テスト関連) .....	167
5.11.1	H/W テスト処理(IEEE802.3ab コンプライアンステスト) .....	167
5.11.2	H/W テスト処理(折り返し通信テスト) .....	169
5.11.3	折り返し通信テスト準備処理 .....	170
5.11.4	折り返し通信テストリンク状態判定処理 .....	171
5.11.5	折り返し通信テストデータ送信処理 .....	172
5.11.6	折り返し通信テスト受信結果判定処理 .....	173
5.11.7	折り返し通信テスト完了処理 .....	174



6.	R-IN32M4-CL3 ドライバの関数仕様 .....	175
6.1	各関数の概要 .....	175
6.2	R-IN32M4-CL3 ドライバインタフェース関数一覧 .....	177
6.3	R-IN32M4-CL3 ドライバのタスク .....	181
6.4	R-IN32M4-CL3 ドライバインタフェース関数詳細 .....	183
6.4.1	初期設定 .....	183
6.4.2	ウォッチドッグ・タイマ .....	201
6.4.3	イベント .....	203
6.4.4	サイクリック伝送 .....	204
6.4.5	自局状態設定 .....	216
6.4.6	自局状態取得 .....	217
6.4.7	LED 制御 .....	222
6.4.8	ネットワーク時刻 .....	228
6.4.9	MDIO アクセス .....	234
6.4.10	SLMP 送受信 .....	236
6.4.11	SLMP コマンド実行 .....	244
6.4.12	エラー履歴 .....	247
6.4.13	H/W テスト .....	249
6.4.14	汎用共通 .....	251
6.4.15	ネットワーク同期通信 .....	255
6.4.16	CANopen 通信 .....	260
6.4.17	MCU 間 I/F .....	269
6.5	R-IN32M4-CL3 ドライバコールバック関数一覧 .....	274
6.6	R-IN32M4-CL3 ドライバコールバック関数詳細 .....	275

# 1. はじめに

## 1.1 用語

本書では、特に明記する場合を除き、下記の用語を使用して説明します。

用語	内容
CANopen	CAN をベースとした上位層プロトコルです。CC-Link IE TSN における CANopen は、CANopen プロトコルのアプリケーション層を CC-Link IE TSN へ適用し、コミュニケーションオブジェクト(SDO, PDO)の送受信を行います。
CC-Link IE TSN Class	CC-Link 協会による、CC-Link IE TSN に対応したユニットおよび産業用スイッチの機能・性能によるランク分けです。CC-Link IE TSN Class A および CC-Link IE TSN Class B があり、R-IN32M4-CL3 は CC-Link IE TSN Class B/A の機器開発に使用できます。
GX Works3	MELSEC シーケンサソフトウェアパッケージの製品名です。
IEEE1588	Ethernet 上のノード間でクロック同期するための標準プロトコルです。
IEEE802.1AS	AVB (Audio/Video Bridging) ネットワークにおける正確なタイミングと同期の転送の規格です。
Management I/F	R-IN32M4-CL3 から PHY のレジスタにアクセスするためのインタフェースです。MDIO と MDC で構成します。
PDO	複数の CANopen ノード間で周期的に転送されるアプリケーションオブジェクトの集合体です。TPDO(Transmit PDO)でデータを送信し、RPDO(Receive PDO)で任意の TPDO からデータを受信します。
R-IN32M4-CL2	ルネサスエレクトロニクス株式会社の産業用 Ethernet 通信 LSI です。
R-IN32M4-CL3	CC-Link IE TSN リモート局用 GbE-PHY 内蔵通信 LSI です。
R-IN32M4-CL3 適用回路	R-IN32M4-CL3 および周辺部品で構成した CC-Link IE TSN の通信用回路です。
R-IN32M4-CL3 適用製品	本書を参考に製作した、CC-Link IE TSN 接続対応製品です。
RWr	リンクデバイスのリモートレジスタです。デバイス局からマスタ局に 16 ビット(1 ワード)単位で入力される情報です。
RWw	リンクデバイスのリモートレジスタです。マスタ局からデバイス局に 16 ビット(1 ワード)単位で出力される情報です。
RX	リンクデバイスのリモート入力です。デバイス局からマスタ局にビット単位で入力される情報です。
RY	リンクデバイスのリモート出力です。マスタ局からデバイス局にビット単位で出力される情報です。
SDO	CANopen ノードのオブジェクトディクショナリ内のオブジェクトエントリにアクセスするためのメッセージです。SLMP のトランジェント伝送フレームに配置され、通信周期に関係なく他局と送受信します。
エンドユーザ	ユーザが開発した CC-Link ファミリー接続対応製品の購入・使用者です。
グランドマスタ	PTP を使用した時刻同期における、時刻同期元となる機器です。
サイクリック伝送	リンクデバイスまたは CANopen の PDO を使用して、同一ネットワークの局間で定期的にデータ交信する機能です。
デバイス	シーケンサ CPU が内部に持っている各種メモリ(X/Y/M/D など)です。 または、ユーザアプリケーションが内部に持っている、R-IN32M4-CL3 と授受するデータを格納するメモリです。

用語	内容
デバイス局	マスタ局以外の局(ローカル局, リモート局)です。
トランジェント伝送	ユーザアプリケーションからの要求時に、他局との交信を行う機能です。
ノード番号	IP アドレスを設定するために使用する、内部の識別子です。
バッファメモリ	ユーザアプリケーションが持っている、データ(設定値、モニタ値など)を格納するメモリです。
マスタ局	CC-Link IE TSN を制御する局で、すべての局とサイクリック伝送およびトランジェント伝送が可能です。
マルチキャストフィルタ	自局が受信したマルチキャストのサイクリックデータを、後続の局に流すか流さないかを選択できるフィルタ機能です。マスタ局がシステム構成に応じて自動で設定するため、パラメータの設定は不要です。
ユーザ	本書をもとに CC-Link ファミリー接続対応製品を開発・販売するメーカです。ベンダー、パートナーメーカと同義です。
リモート局	マスタ局とサイクリック伝送およびトランジェント伝送する局です。
リンクデバイス	ネットワークユニットが内部に持っているデバイス(RX,RY,RWr,RWw)です。
ローカル局	マスタ局および他ローカル局と、サイクリック伝送およびトランジェント伝送する局です。
安全 PDU	安全通信に用いるデータです。
安全プロトコルスタック	CC-Link IE 安全通信を実現するための専用ライブラリソフトウェアです。安全プロトコルスタックを機器に組み込むことで容易に CC-Link IE 安全通信が行えます。
安全出力	安全機能を実現するために安全局から出力される信号を示します。
安全通信	同一ネットワークの安全局間で、安全 PDU の交信を行う機能です。
安全入力	安全機能を実現するために安全局に入力される信号を示します。
一般通信	安全通信以外の通信(CC-Link IE TSN のサイクリック伝送とトランジェント伝送など)です。
解列	データリンク異常時に、データリンクを停止する処理です。
局	ネットワークを構成し、データの送信、受信、転送を行う要素を指します。
局番	ネットワーク内で局を一意に識別するための識別子です。
自局	本書を基に開発するリモート局です。
他局	自局以外の局です。
復列	異常局が正常になったときに、データリンクを再開する処理です。
予約局	実際には接続せずに、将来接続する局として、ネットワークの台数に含めておく局です。

## 1.2 総称/略称

本書では、特に明記する場合を除き、下記の総称/略称を使用して説明します。

総称/略称	内容
CAN	Controller Area Network の略称です。
CSP+	Control & Communication システムプロファイルの略称です。 CC-Link ファミリー対応機器の立上げ、運用・保守のために必要な情報を記述するための仕様です。
GbE-PHY	Gigabit Ethernet PHY の略称です。本書は GMII を持ち、1000BASE-T に対応したものを指します。
GMII	Gigabit Media Independent Interface の略称です。 R-IN32M4-CL3 の MAC ポート(MAC 層)と PHY(物理層)間でデータを交信するためのインタフェースです。
MDC	Management Data Clock の略称です。GMII に規定されている MDIO のクロックです。 MDIO と一緒に Management I/F を構成します。
MDI	Medium Dependent Interface の略称です。R-IN32M4-CL3 とパルストランス間、およびパルストランスと RJ-45 コネクタ間で、データを交信するためのインタフェースです。
MDIO	Management Data Input/Output の略称です。GMII に規定されている PHY レジスタにアクセスするためのデータ入出力バスです。MDC と一緒に Management I/F を構成します。
MIB	Management Information Base の略称です。R-IN32M4-CL3 の通信状態を保存するためのマネジメント情報ベースです。
PHY	Physical layer の略称です。本書は Ethernet などのインタフェースにおいて、論理信号を実際の電気的な信号に変換する、R-IN32M4-CL3 の機能の一部を指します。
PTP	Precision Time Protocol の略称です。ネットワーク内の機器間で時刻を同期させるために使用するプロトコルです。
RSPDU	Received Safety Protocol Data Unit の略です。自局が受信する安全 PDU を示します。
SLMP	Seamless Message Protocol の略称です。外部機器から SLMP 対応機器、および SLMP 対応機器に接続されたシーケンサにアクセスするためのプロトコルです。
SSPDU	Send Safety Protocol Data Unit の略です。自局が送信する安全 PDU を示します。
UTC	Coordinated Universal Time の略称です。GMT(グリニッジ標準時)とのずれを調整するために閏秒を追加した時間です。
WDC	ウォッチドッグカウンタの略称です。
データリンク	サイクリック伝送、トランジェント伝送の総称です。
安全局	安全通信および一般通信を行う局の総称です。

### 1.3 関連マニュアル

CC-Link IE TSN の用語、機能などの詳細については記述していません。必要に応じて関連マニュアルを、下記三菱電機 FA サイトからダウンロードして参照してください。

三菱電機 FA サイト (<http://www.MitsubishiElectric.co.jp/fa>)

マニュアル名称 [マニュアル番号]	内容
MELSEC iQ-R CC-Link IE TSN ユーザーズマニュアル (スタートアップ編) [SH-082126]	CC-Link IE TSN の仕様、運転までの手順、システム構成、配線、交信例について記載しています。
MELSEC iQ-R CC-Link IE TSN ユーザーズマニュアル (応用編) [SH-082128]	CC-Link IE TSN の機能、パラメータ設定、プログラミング、トラブルシューティング、入出力信号およびバッファメモリなどの説明を記載しています。
SLMP リファレンスマニュアル [SH-080931]	外部機器から SLMP 対応機器に対して、データの読出し、書込みを行うプロトコル(SLMP)について記載しています。
MELSEC iQ-R ユニット間同期機能リファレンスマニュアル [SH-081400]	ユニット間で同期制御を行うユニット間同期機能について記載しています。

### 1.4 CC-Link 協会のご案内

#### (1) 仕様書について

本書に関連する資料には、CC-Link 協会（CLPA）発行の下記仕様書があります。

CC-Link IE TSN および SLMP の詳細については、CC-Link 協会のホームページから下記資料をダウンロードの上、参照してください。

資料名称	資料番号
CC-Link IE TSN 仕様書 (概要編)	BAP-C2011-001
CC-Link IE TSN 仕様書 (物理層・データリンク層編)	BAP-C2011-002
CC-Link IE TSN 仕様書 (アプリケーション層サービス編)	BAP-C2011-003
CC-Link IE TSN 仕様書 (アプリケーション層プロトコル編)	BAP-C2011-004
CC-Link IE TSN 仕様書 (通信プロファイル編)	BAP-C2011-005
CC-Link IE TSN 仕様書 (実装規約編)	BAP-C2011-006
SLMP 仕様書 (概要編)	BAP-C2006-001
SLMP 仕様書 (SLMP 仕様サービス編)	BAP-C2006-002
SLMP 仕様書 (SLMP 仕様プロトコル編)	BAP-C2006-003
CC-Link IE 安全通信機能仕様書 (概要編)	BAP-C2007-001
CC-Link IE 安全通信機能仕様書 (アプリケーション層サービス・プロトコル編)	BAP-C2007-002
CC-Link IE 安全通信機能仕様書 (通信プロファイル編)	BAP-C2007-003
CC-Link IE 安全通信機能仕様書 (実装規約編)	BAP-C2007-004
CC-Link IE 安全通信機能仕様書 (データフォーマット編)	BAP-C2007-005

#### (2) コンフォーマンステストについて

本書に記載している情報に基づき製品開発を行った場合、CC-Link 協会が実施するコンフォーマンステストを受験していただく必要があります。コンフォーマンステストの詳細は、CC-Link 協会のホームページから下記資料をダウンロードの上、参照してください。

資料名称	資料番号
CC-Link IE TSN コンフォーマンステスト仕様書 リモート局編 (ツイストペアケーブル)	BAP-C0401-049
CC-Link IE 安全通信機能および CC-Link IE TSN コンフォーマンステスト依頼書	BAP-C0401-065

### (3) Control & Communication システムプロファイル(CSP+)の作成

コンフォーマンステストでは CSP+の確認があります。あらかじめ CSP+ファイルを作成していただく必要があります。CSP+の詳細については、CC-Link 協会のホームページから下記仕様書をダウンロードの上、参照してください。また、CSP+ファイル作成の補助として下記の関連資料・ツールなどがありますので、同ホームページからダウンロードの上、ご活用ください。

資料名称/関連資料・ツール	資料番号
Control & Communication システムプロファイル仕様書	BAP-C2008-001
Control & Communication システムプロファイル作成ガイドライン	-
CSP+作成支援ツール	-
Sample CSP+ ファイル	-
CSP+テンプレート	-

### (4) お問い合わせ先について

CC-Link 協会発行仕様書の請求およびコンフォーマンステストの詳細は、以下の連絡先までお問い合わせ願います。

CC-Link 協会	TEL : 052-919-1588
	FAX : 052-916-8655
	E-mail : <a href="mailto:info@cc-link.org">info@cc-link.org</a>
	Web : <a href="https://www.cc-link.org/">https://www.cc-link.org/</a>

## 2. 概要

本書は、CC-Link IE TSN リモート局用通信 LSI R-IN32M4-CL3 を使用して、CC-Link IE TSN のリモート局を開発する方法を示したものです。主な記載内容は以下のとおりです。

- R-IN32M4-CL3 の仕様
- R-IN32M4-CL3 適用回路の設計
- ユーザプログラムの設計
- R-IN32M4-CL3 ドライバの仕様

### 2.1 開発の特長

R-IN32M4-CL3 は、CC-Link IE TSN 用通信 IP コア、CPU、および GbE-PHY を一体化した LSI です。一体化した LSI により CPU と GbE-PHY に関連する開発のコスト・工数を削減することができます。R-IN32M4-CL3 を使用した開発の特長を以下に示します。

- (1) プロトコルを意識することなく、CC-Link IE TSN のリモート局を開発できます。
- (2) GbE-PHY を一体化していますので、通信 IP コアと GbE-PHY 間のパターン設計が不要です。そのため、CC-Link IE TSN の通信回路のパターン設計が容易になります。また、CPU や GbE-PHY 周辺の部品・回路が少ないため、開発する基板をコンパクトにすることができます。
- (3) ユーザのハードウェア仕様やアプリケーションに応じて、カスタマイズできるサンプルコードを提供します。
- (4) H/W-RTOS を搭載していますので、CPU 負荷低減することができ、開発機器の低消費電力を実現します。

### 2.2 R-IN32M4-CL3 の主な仕様

R-IN32M4-CL3 の H/W に関する主な仕様を以下に示します。

表 2.1 R-IN32M4-CL3 の主な仕様

項目		内容
外観	ピン数	BGA 484 ピン
	サイズ	23mm x 23mm
電源電圧		3.3±0.165V、2.5±0.125V、1.15±0.06V
動作周囲温度		-40～85℃
CPU		Arm Cortex -M4 Processor (100MHz) 内蔵
命令 RAM		768K バイト内蔵 (ECC 対応)
データ RAM		512K バイト (ECC 対応)
バッファ RAM		64K バイト (ECC 対応)
I/O ポート		CMOS 入出力 : 最大 106 本
Ethernet PHY		100BASE-TX、1000BASE-T GbE-PHY 内蔵 x 2 PORT

## 2.3 R-IN32M4-CL3 適用製品の通信仕様

R-IN32M4-CL3 適用製品の通信に関する主な仕様を以下に示します。

表 2.2 R-IN32M4-CL3 適用製品の通信仕様(CC-Link IE TSN)

項目	内容
局種別	リモート局
局番	1～254
通信速度	1Gbps、100Mbps
CC-Link IE TSN プロトコルバージョン	2.0 ※1
CC-Link IE TSN Class	CC-Link IE TSN Class B/A の機器開発が可能
通信方式	CC-Link IE TSN Class A : タイムマネージド・ポーリング方式 CC-Link IE TSN Class B : 時分割方式
伝送路形式	ライン型、スター型(ライン型とスター型の混在も可能)、リング型
通信ケーブル	1Gbps : 1000BASE-T の規格を満たす Ethernet ケーブル (カテゴリ 5e 以上、シールド付・STP) 100Mbps : 100BASE-TX の規格を満たす Ethernet ケーブル (カテゴリ 5 以上、シールド付・STP)
最大局間距離	100m
マルチキャストフィルタ	対応
サイクリック伝送機能	送信最大サイズ : RX、RWr および SSPDU の合計サイズが 2400 バイト以内 受信最大サイズ : RY、RWw および RSPDU の合計サイズが 2400 バイト以内
トランジェント伝送機能	クライアント機能 : あり サーバ機能 : あり
CANopen 通信	PDO 送受信 : あり SDO 送受信 : あり 最大オブジェクトディクショナリ数(最大制御軸数) : 8
安全 PDU 送受信	RSPDU 最大サイズ : 80 バイト SSPDU 最大サイズ : 80 バイト

【注】 ※1 : R-IN32M4-CL3 サンプルコード Ver1.06 から対応しています。

表 2.3 R-IN32M4-CL3 適用製品の通信仕様(Ethernet)

項目	内容
通信速度	1Gbps、100Mbps
通信モード	1000BASE-T (全二重)、100BASE-TX (全二重)
インタフェース	RJ-45 コネクタ(AUTO MDI/MDI-X)
最大フレームサイズ	1518 バイト (ジャンボフレーム不可)
最大セグメント長	産業用スイッチと局の距離 : 100m 産業用スイッチと産業用スイッチ間の距離 : 使用する産業用スイッチのメーカーに確認してください。
カスケード接続段数	使用する産業用スイッチのメーカーに確認してください。
IP バージョン	IPv4 に対応※1

【注】 ※1 : IP アドレスは、0.0.0.1～223.255.255.254 の範囲で設定可能です。

設定方法は「3.5 IP アドレスの設定方法の検討」を参照してください。



### 2.3.1 CC-Link IE TSN Class A の動作時の注意事項

R-IN32M4-CL3 適用製品を CC-Link IE TSN Class A で動作させる場合の注意事項を示します。

#### (1) マスタ局のバージョン

CC-Link IE TSN プロトコルバージョン 2.0 に対応したマスタ局が必要です。

マスタ局に RJ71GN11-T2 を使用する場合、ファームウェアバージョン "15" 以降のユニットを使用してください。

マスタ局に RD78G(H)を使用する場合、ファームウェアバージョン "24" 以降のユニットを使用してください。

#### (2) サイクリック伝送

マスタ局に RJ71GN11-T2 または RD78G(H)を使用する場合、CC-Link IE TSN Class A で動作するすべてのデバイス局のサイクリックデータサイズ合計が 2K バイトを超える接続はできません。サイクリックデータサイズ合計を 2K バイト以内にしてください。

上記以外のマスタ局を使用する場合、使用するマスタ局の仕様にしてください。

#### (3) トランジェント伝送 (SLMP)

CC-Link IE TSN Class B/A 混在時のシステムでは、時分割制御しない局(CC-Link IE TSN Class A)から時分割制御する局(CC-Link IE TSN Class B)にフレームが中継される際、接続台数(使用環境やフレームサイズにより変動)によっては中継できずにフレームが消失する場合があります。

### 2.3.2 サンプルコードのバージョンによる機能の制約

サンプルコードのバージョンにより、使用できる機能に制約があります。制約内容と対応するバージョンについて下記に示します。

#### (1) R-IN32M4-CL3 適用製品が接続可能なシステムの最大接続局数

サンプルコードのバージョンによって、R-IN32M4-CL3 適用製品が接続可能なシステムの最大接続局数は異なります。

R-IN32M4-CL3 適用製品をマスタ局 1 ポートに接続するデバイス局数が 120 局を超えるシステムに接続する場合、R-IN32M4-CL3 用サンプルコード Ver.1.09K 以降を使用してください。

表 2.4 R-IN32M4-CL3 適用製品が接続可能なシステムの最大接続局数

サンプルコードのバージョン	R-IN32M4-CL3 適用製品が接続可能なシステムの最大接続局数
Ver.1.08J 以前	マスタ局 1 ポートあたり 120 局以内
Ver.1.09K 以降	マスタ局 1 ポートあたり 253 局以内

## 2.4 サンプルコードのフォルダ構成

サンプルコードのフォルダ構成およびファイル概要について示します。

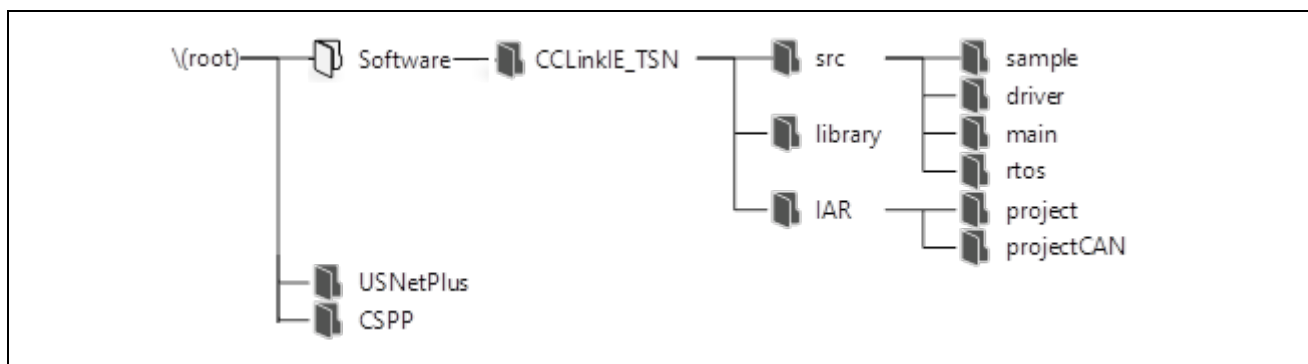


図 2.1 フォルダ構成図

表 2.5 ファイル概要

フォルダ名			ファイル内容
Software	src	sample	CC-Link IE TSN リモート局用サンプルコード ユーザプログラムのソースファイルを格納するフォルダ
		driver	CC-Link IE TSN リモート局用サンプルコード R-IN32M4-CL3 ドライバのソースファイルを格納するフォルダ
		main	タスク関連のフォルダ
		rtos	RTOS のコンフィグレーションファイル関連のフォルダ
	library		ライブラリ関連(GbE-PHY 制御用、HW-RTOS 制御用)のフォルダ
	IAR	project	IAR 開発環境関連(プロジェクト、生成物など)のフォルダ
projectCAN		サイクリック伝送において、リンクデバイスを使用する場合と CANopen の PDO を使用する場合の 2 種類のプロジェクトを格納	
USNetPlus	usnet_user_manual.pdf		USNetPlus のユーザーズ・マニュアル(英語) (本書の 1.8 節を参照してください)
CSPP			サンプル CSP+ファイル, およびアイコン・画像などを圧縮したファイル (2.6 節、3.8 節を参照してください)

## 2.5 サンプルコードの概要

サンプルコードは、R-IN32M4-CL3 を使って CC-Link IE TSN リモート局を開発する場合に使用します。ユーザプログラム、R-IN32M4-CL3 ドライバインタフェース関数、R-IN32M4-CL3 ドライバコールバック関数、および R-IN32M4-CL3 ドライバ本体で構成します。CC-Link IE TSN に関して(通信機能)のみ記述しています。

### (1) ユーザプログラム

ユーザプログラムは、ユーザが作成するアプリケーションプログラムです。リモート局のロジックを確認するための参考用です。ユーザの要求仕様に応じてカスタマイズしてください。

詳細は「5 ユーザプログラムの作成」を参照してください。

### (2) R-IN32M4-CL3 ドライバインタフェース関数

R-IN32M4-CL3 ドライバインタフェース関数は、R-IN32M4-CL3 ドライバの機能をユーザプログラムから使用する場合に呼び出す関数です。カスタマイズは不要です。

詳細は「6 R-IN32M4-CL3 ドライバの関数仕様」を参照してください。

### (3) R-IN32M4-CL3 ドライバコールバック関数

R-IN32M4-CL3 ドライバコールバック関数は、R-IN32M4-CL3 ドライバ側で発生するイベントに対するユーザプログラム側の処理例を記述しています。ユーザの要求仕様に合わせてカスタマイズしてください。

詳細は「6 R-IN32M4-CL3 ドライバの関数仕様」を参照してください。

### (4) R-IN32M4-CL3 ドライバ本体

R-IN32M4-CL3 ドライバ本体は、R-IN32M4-CL3 ドライバインタフェース関数に呼び出され、R-IN32M4-CL3 を制御するドライバ部の本体です。

## 2.6 サンプル CSP+ファイルの概要

付属するサンプル CSP+ファイルは、R-IN32M4-CL3 を使って CC-Link IE TSN リモート局を開発する場合に参考にしてください。サンプル CSP+ファイルの一覧と用途を以下に示します。

表 2.6 ファイル概要

ファイル名	用途
0x1234_RemoteSample_1_en.CSPP.zip	サイクリック伝送でリンクデバイスを使用する場合に使用します。
0x1234_RemoteSample_CAN_1_en.CSPP.zip	サイクリック伝送で CANopen の PDO を使用する場合に使用します。
0x1234_RemoteSample_CAN_Base_1_en.CSPP.zip	(サイクリック伝送で CANopen の PDO を使用する場合) 基本ユニットおよび増設ユニットで構成する機器のうち、基本ユニットで使用します。
0x1234_RemoteSample_CAN_Ext_1_en.CSPP.zip	(サイクリック伝送で CANopen の PDO を使用する場合) 基本ユニットおよび増設ユニットで構成する機器のうち、増設ユニットで使用します。
0x1234_RemoteSample_Safe_1_en.CSPP.zip	安全 PDU を送受信する場合に使用します。

サンプル CSP+ファイルは CC\_IE\_TSN 構成ウィンドウに R-IN32M4-CL3 適用製品を表示させ、ならびに「デバイス局パラメータ処理・コマンド実行」※1を確認するための参考用です。ユーザの要求仕様に応じてカスタマイズしてください。CSP+を作成する際は「3.8 CSP+および関連ファイルの作成準備」を参照してください。

※1:「パラメーター括読出し」「パラメーター括書込み」および「パラメータ自動設定」に必要な情報を記述しています。詳細は「3.7 エンジニアリングツール各種機能への対応検討」を参照してください。

## 2.7 システム構成

### (1) S/W 構成

R-IN32M4-CL3 適用製品の S/W 構成例を以下に示します。R-IN32M4-CL3 ドライバの提供する各種関数と、HW-RTOS のライブラリを使用することで、ユーザプログラムが R-IN32M4-CL3 のサイクリック伝送やトランジェント伝送など各機能を使用することができます。

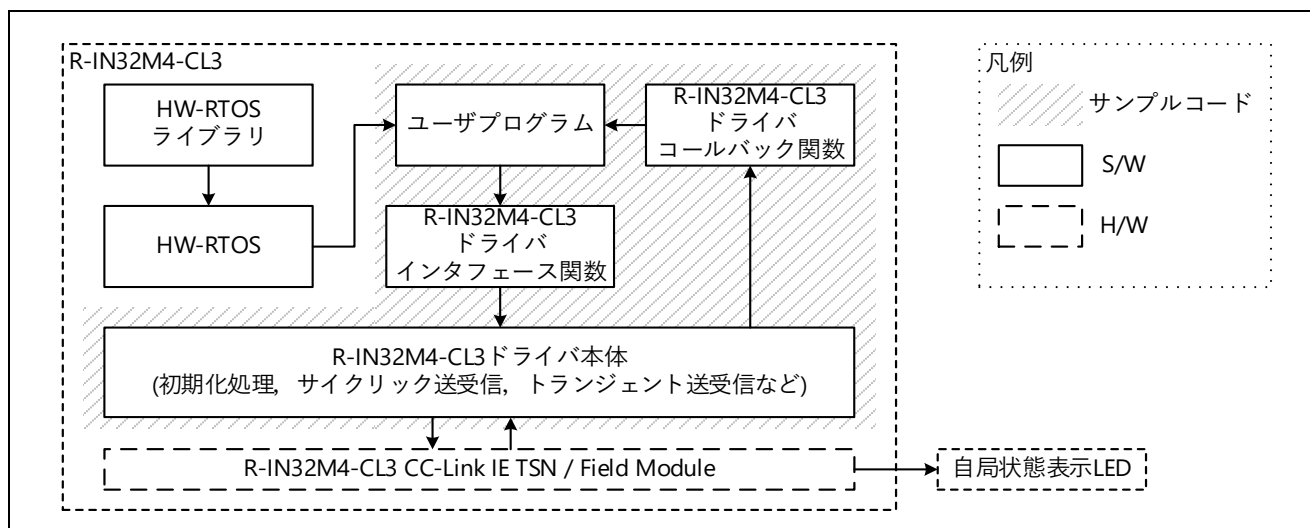


図 2.2 S/W 構成概略図

## (2) H/W 構成

R-IN32M4-CL3 適用製品の H/W 構成例を以下に示します。R-IN32M4-CL3、周辺部品、および 2 つの Ethernet PORT で構成します。

なお、以降の章で記載する「CPU」は、R-IN32M4-CL3 の GbE-PHY 部を除いた箇所を指します。

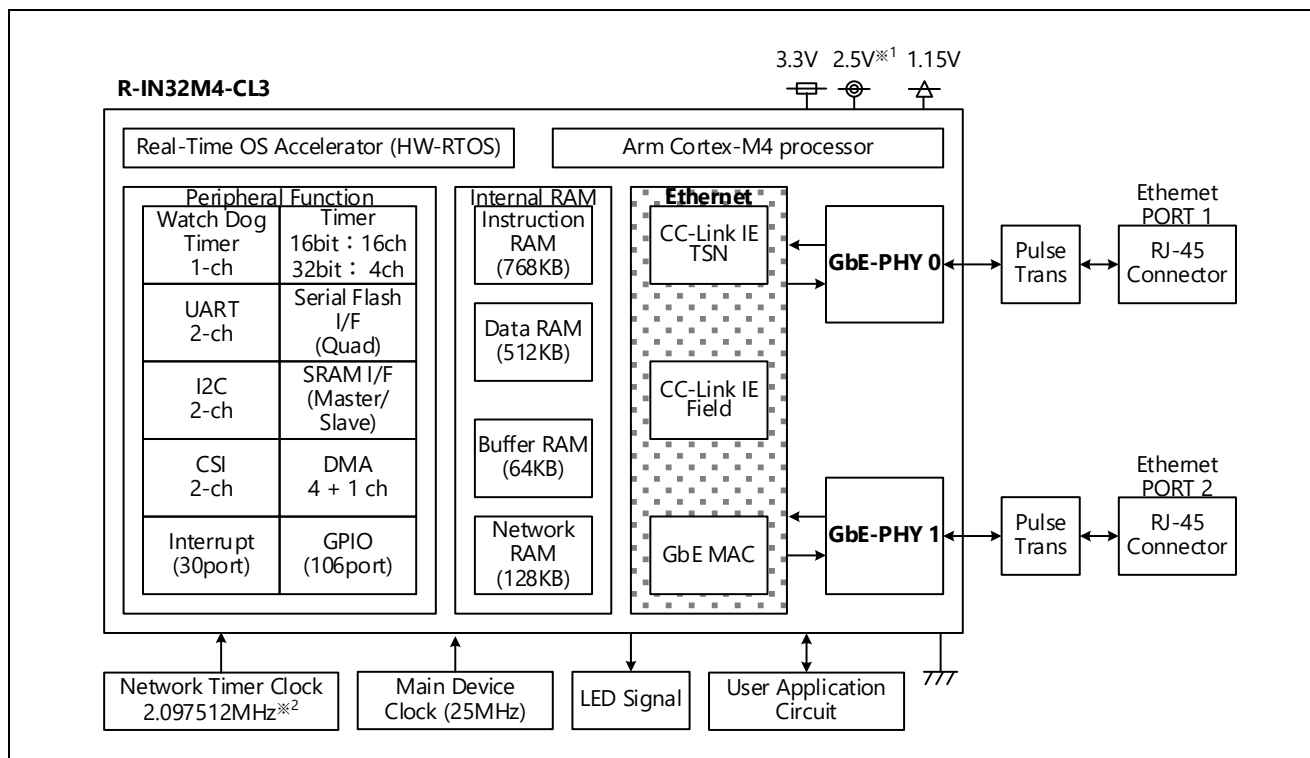


図 2.3 H/W 構成概略図

【注】※1 R-IN32M4-CL3 への+2.5V 電源入力、外部端子(REG\_EN)の設定により、R-IN32M4-CL3 内部で生成した+2.5V 電源入力、R-IN32M4-CL3 外部で生成した+2.5V 電源入力の選択が可能です。

※2 CC-Link IE Field 動作時における時計用クロック入力(Network Timer Clock)は、外部端子(CLK2MSEL)の設定により、R-IN32M4-CL3 内部の PLL、R-IN32M4-CL3 外部の発振器出力の選択が可能です。

## 2.8 R-IN32M4-CL3 適用製品が対応しているプロトコル

R-IN32M4-CL3 適用製品で対応しているプロトコルは以下のとおりです。

表 2.7 対応しているプロトコル一覧

プロトコル	内容
UDP	トランジェント伝送に使用
IPv4	・ UDP/IPv4 を下層のプロトコルとした SLMP 通信に使用
ICMP	PING の応答などに使用
ARP	MAC アドレス取得に使用
RARP	IP アドレス問い合わせに使用
PING	通信の疎通確認に使用
SNMPv2	診断情報収集機能に使用
GARP	IP アドレス設定重複検出に使用
TCP	ユーザ拡張用(ユーザアプリケーションの実装次第で TCP/IP も使用可)

なお、上表のプロトコルは、GARP を除いて株式会社日新システムズの TCP/IP スタック「USNetPlus®」を使用しています。

R-IN32M4-CL3 適用製品がサイクリック伝送・トランジェント伝送を行うだけであれば、サンプルコードのユーザプログラムが TCP/IP 通信を直接制御することがないため、ユーザが USNetPlus を意識する必要がありません。

そのうえで、TCP/IP 通信機能を自由に実装したいなどの場合には、同梱の USNetPlus ユーザズ・マニュアル「usnet\_user\_manual.pdf」を参照して USNetPlus の API 仕様などをご確認ください。

ただし、サンプルコードには、汎用的な TCP/IP 通信に関するサンプル処理を記述していません。

### 3. 開発前の仕様検討と準備

本章は、R-IN32M4-CL3 適用製品を開発するにあたり、あらかじめ仕様検討や準備が必要な事項について示します。

以下にユーザの開発工程を例示します。

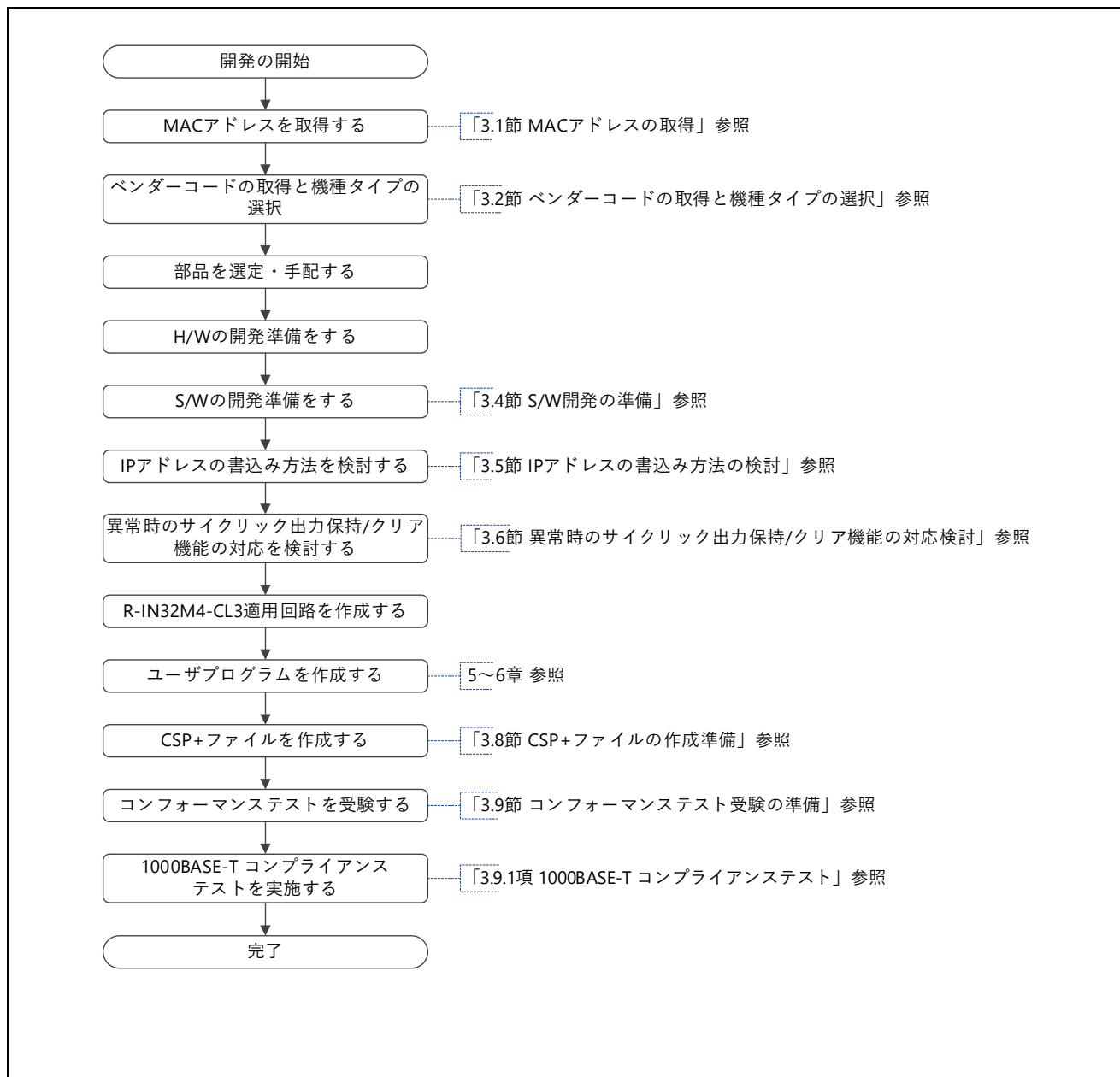


図 3.1 開発の工程例



### 3.1 MAC アドレスの取得

R-IN32M4-CL3 適用製品は、Ethernet(IEEE 802.3ab)に準拠しているため、開発機器固有の MAC アドレスの MA-L(MAC Address Block Large)を取得してください。

MAC アドレスの取得については、以下の米国の管理機関(学会)にお問い合わせください。

The IEEE Registration Authority

Web : <https://standards.ieee.org/products-programs/regauth/> (2023 年 12 月現在)

### 3.2 ベンダーコードの取得と機種タイプの選択

R-IN32M4-CL3 適用製品は、ベンダーコードと機種タイプの登録が必要です。ベンダーコードと機種タイプの割り当てや管理は CC-Link 協会が行っています。ご不明な点があれば CC-Link 協会までお問い合わせください。

表 3.1 ベンダーコードと機種タイプ

項目	内容
ベンダーコード (vendorCode)	CC-Link 協会入会時に発行される ID 番号(5 桁目～8 桁目) 例えば、ID 番号が 123-456-7890 の場合、ベンダーコードは 5678 となります。
機種タイプ (deviceType)	CC-Link 協会 HP に掲載している機種タイプから、該当する機種タイプを選択してください。 該当する機種タイプがない場合は、CC-Link 協会にご相談ください。

### 3.3 H/W スイッチに接続する端子

R-IN32M4-CL3 適用回路では、下表のとおり H/W スイッチを使用して各種設定を切り替えます。

下表に示す端子はユーザにて変更することができますので、端子を H/W スイッチに接続するか検討してください。

H/W スイッチを使用しない場合、R-IN32M4-CL3 適用製品の周辺機器などから各種設定値を書き込む処理を追加してください。

表 3.2 H/W スイッチに接続する端子 (CC-Link IE TSN)

符号	スイッチ	端子
SW5	IP アドレス第 4 オクテット設定スイッチ (x1)	RP10、RP11、RP12、RP13
SW3	IP アドレス第 4 オクテット設定スイッチ (x10)	RP14、RP15、RP16、RP17

## 3.4 S/W 開発の準備

### 3.4.1 S/W 開発の手順

R-IN32M4-CL3 適用製品の S/W 開発の手順例を示します。

(1) 「4 章」を参考に、実装する機能・仕様を検討する。

(2) ユーザプログラムの作成

「5 ユーザプログラムの作成」を参考にユーザプログラムを作成してください。

(3) ユーザプログラムと R-IN32M4-CL3 ドライバコールバック関数のコンパイル

カスタマイズしたユーザプログラムと R-IN32M4-CL3 ドライバ コールバック関数をコンパイルしてください。

(4) オブジェクトモジュールファイルとライブラリファイルの結合

コンパイルしたファイル(オブジェクトモジュールファイル)、OS ドライバ・ライブラリファイル、および R-IN32M4-CL3 ドライバのライブラリファイルを結合し、ロードモジュールファイルを作成してください。

(5) 実行ファイルの書き込み

デバッガや ICE などを使用して、ロードモジュールファイルを R-IN32M4-CL3 適用製品(ターゲット)にロードしてください。

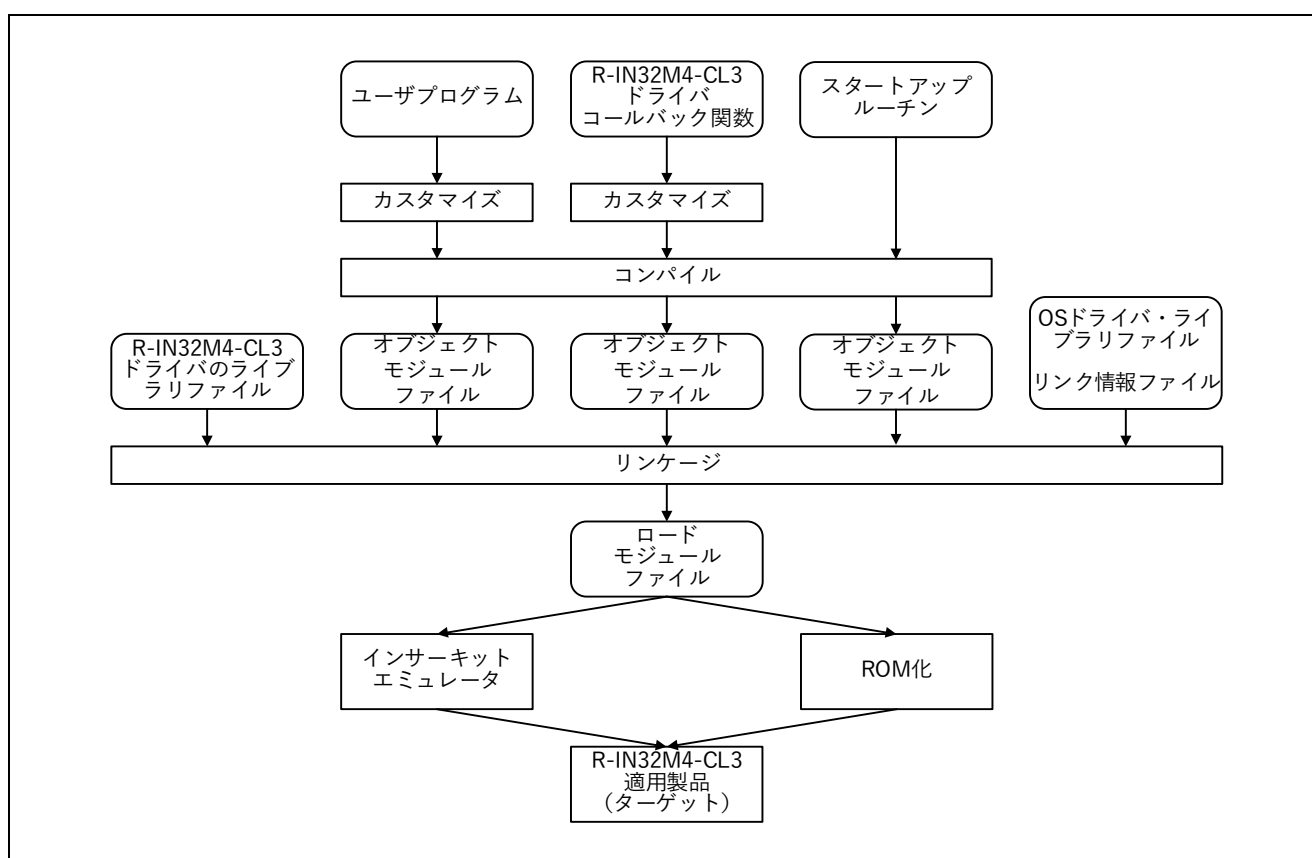


図 3.2 S/W 開発の手順例

### 3.4.2 開発環境

R-IN32M4-CL3 に内蔵されているマイコン(Cortex-M4)のソフトウェア開発環境として以下の開発環境を使用します。

表 3.3 ソフトウェア開発環境

ツールチェーン	コンパイラ	デバッガ	エミュレータ
Embedded Workbench for Arm V9.32.1～最新版 (最新版をお使いください) (IAR Systems 社)			i-Jet JTAGjet-Trace-CM (IAR Systems 社)

#### 3.4.2.1 実行手順

(a) ビルド手順

「CCLinkIE\_TSN」のプロジェクトフォルダにある IAR プロジェクト「ProjectIAR.eww」を起動し、「プロジェクト」→「すべてを再ビルド」を選択してください。ワークスペースに登録されたソースコードが再ビルドされます。

(b) F/W 書込み手順

「CCLinkIE\_TSN」のプロジェクトフォルダにある IAR プロジェクト「ProjectIAR.eww」を起動し、「プロジェクト」→「ダウンロード」→「ファイルのダウンロード」にて、「main.out」ファイルを選択してください。プログラムがフラッシュメモリにダウンロードされます。

### 3.4.3 フラッシュローダのプログラム変更

デバッグに「IAR Embedded Workbench for Arm (以降 EWARM と記述します)」を使用する場合、実行ファイルをフラッシュメモリに書き込む際に使用するフラッシュローダのプログラム変更が必要となる場合があります。

本書は、Winbond®社製シリアルフラッシュ ROM (W25Q64JV)を使用する場合の、フラッシュローダのプログラム変更について示します。なお、フラッシュローダに関する詳細は「フラッシュローダ開発ガイド IAR Embedded Workbench 用 (UFLX-4) IAR SYSTEMS」を参照ください。

#### 3.4.3.1 フラッシュローダの開発環境

EWARM インストール時にはフラッシュローダの開発環境も同時にインストールされます。この開発環境を使用してフラッシュローダのプログラムを変更します。Winbond®社製シリアルフラッシュ ROM (W25Q64JV)に対応するフラッシュローダの開発環境は以下に格納されています。(開発環境のインストール先によって、格納場所が異なります。)

C:\Program Files (x86)\IAR Systems\Embedded Workbench 8.4\arm\src\flashloader\Renesas  
¥FlashRIN32M4\_SerialFlash

フラッシュローダのプログラム(FlashRIN32M4\_SerialFlash.c)には、フラッシュロード可能なフラッシュ ROM の ID など、情報がフラッシュ ROM テーブル(flashType[])として管理されています。使用しているフラッシュ ROM の情報がフラッシュ ROM テーブル(flashType[])に定義されていない場合は、追加する必要があります。

#### 3.4.3.2 フラッシュ ROM テーブルの設定

Winbond®社製シリアルフラッシュ ROM (W25Q64JV)をフラッシュ ROM テーブル(flashType[])に追加する際の設定値を下表に示します。フラッシュローダのプログラム(FlashRIN32M4\_SerialFlash.c)に定義されているフラッシュ ROM テーブル(flashType[])に、下表の設定値を追加します。

表 3.4 フラッシュ ROM 情報)

テーブル情報	W25Q64JV 設定値	概要
ID	1740EFH	W25Q64JV が持つ JEDEC ID を設定します。
ブロックサイズ	16	セクタ当たりのブロックサイズ(キロバイト)を 2 べき乗数で設定します。 W25Q64JV の場合、64KByte になりますので 16 を設定します。
ページサイズ	8	ページプログラムを行う場合のサイズ(バイト)を 2 べき乗数で設定します。 W25Q64JV の場合、256 バイトになりますので 8 を設定します。
デバイスサイズ	23	フラッシュメモリのサイズ(バイト)を 2 のべき乗数で設定します。 W25Q64JV の場合、8Mbyte になりますので 23 を設定します。 2 の 23 乗で 8,388,608Byte となります。

### 3.4.3.3 フラッシュローダのプログラム変更

フラッシュローダのプログラム変更前/変更後を以下に示します。

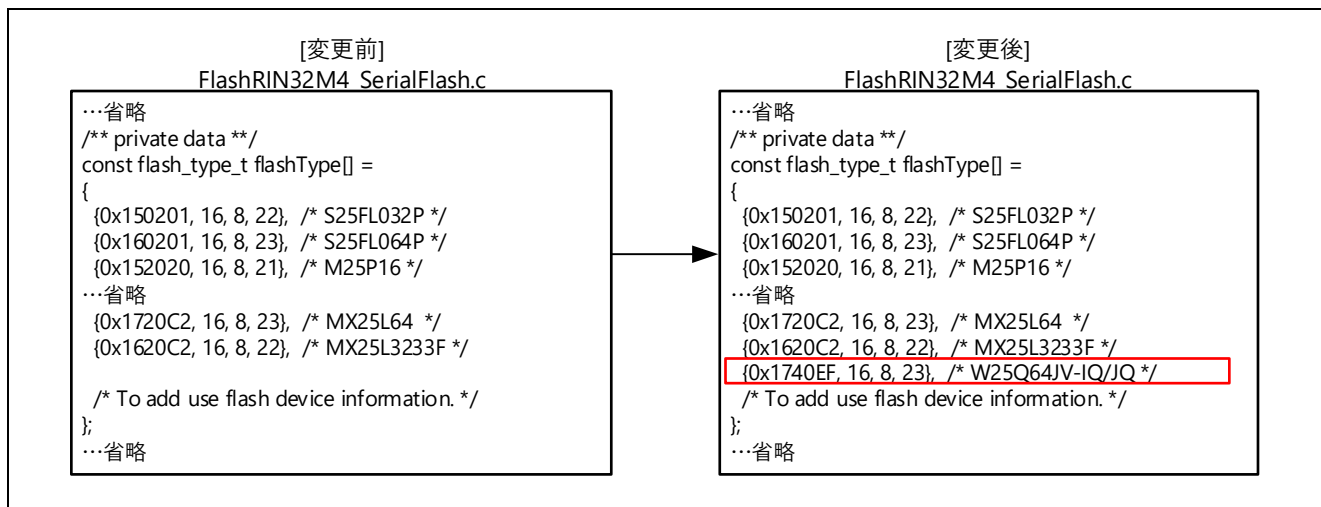


図 3.3 フラッシュローダのプログラム変更例

### 3.4.4 コンパイルスイッチ

コンパイルスイッチとして使用するマクロ定義名を以下に示します。必要に応じて切り替えてください。

表 3.5 コンパイルスイッチ用マクロ定義

マクロ名	マクロ定義時の動作	Default 定義 (コンパイルオプション)	備考
FOR_DEBUG	WDT の無効化※1	定義あり	
NOUSE_WDT	WDT の無効化※2	定義あり	
TSN_CAN_ENABLE	CANopen 通信の有効化	projectCAN のみ定義あり	CANopen 通信時
TSN_CAN_MULTIAxis_ENABLE	CANopen 通信時の増設ユニットを有効化※3	projectCAN のみ定義あり	
USE_LOOPBACKTEST	折り返し通信テストの有効化	定義なし	
USE_COMPLIANCE_TEST	IEEE802.3ab コンプライアンステストの有効化	定義なし	
SLMP_MULTI_DROP_DISABLE	SLMP フレームの「要求先局プロセッササブ番号」の無効化(0 設定)	定義なし	
SAFETY_PDU_ENABLE	安全 PDU 送受信の有効化	定義なし	安全通信時
MCUIF_ENABLE	安全 PDU 送受信の MCU 間 I/F 機能を有効化※4	定義なし	
CURERR_OPTIONINFO_ENABLE	MIB 現在エラー情報のオプション情報に関する処理を有効化	定義なし	増設ユニット装着時
ACCUMULATE_STATISTICS_INFORMATION	統計情報累積格納数保持を有効化	定義なし	-
SNMP_COMMUNITY_NAME_SETTING_ENABLE	SNMP コミュニティ名設定の有効化	定義あり	-

【注】 ※1 デバッグ中に WDT エラーが発生することを防止します。

※2 CC-Link IE TSN で使わない(CC-Link IE Field で使う)WDT を無効化します。

※3 「TSN\_CAN\_ENABLE」の定義が必要です。

※4 「SAFETY\_PDU\_ENABLE」の定義が必要です。

### 3.5 IP アドレスの設定方法の検討

自局をデータリンクさせるためには、IP アドレス(第 4 オクテット)を R-IN32M4-CL3 へ書き込む必要があります。そのため、R-IN32M4-CL3 適用製品の仕様に応じた IP アドレスの書き込み方法をあらかじめ検討してください。

参考として、ハードウェアで設定する方法とソフトウェアで設定する方法を以下に例示します。

表 3.6 IP アドレスの設定例

No	区分	設定方法	概要
1	ハードウェア設定	H/W スイッチ	H/W スイッチを実装し、H/W スイッチで設定する
2	ソフトウェア設定	(a) GX Works3	不揮発性メモリを実装し、GX Works3 で設定する
		(b) ユーティリティソフト	不揮発性メモリを実装し、ユーザ作成のユーティリティソフトで設定する

#### (1) ハードウェア設定

R-IN32M4-CL3 適用製品に H/W スイッチを実装し、H/W スイッチで IP アドレスを設定します。

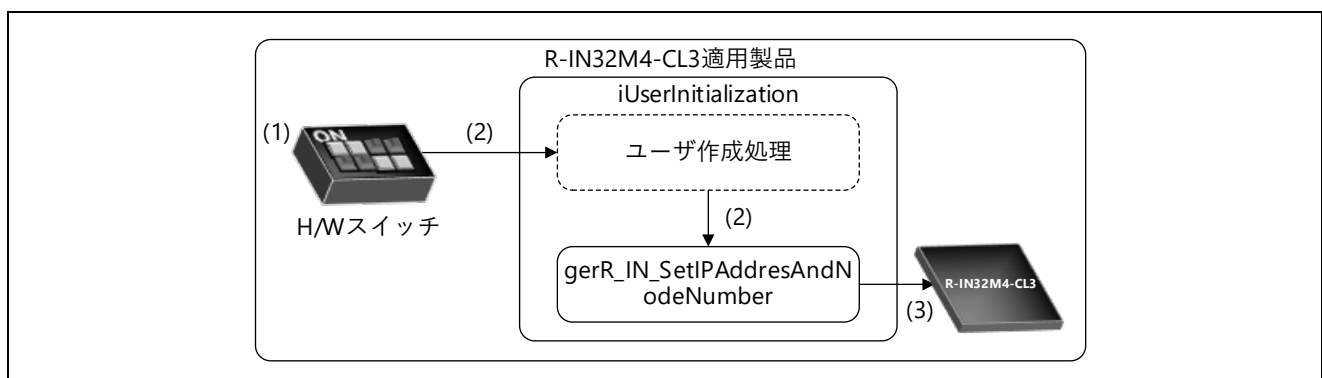


図 3.4 IP アドレス設定イメージ (H/W スイッチ)

手順	内容
1	H/W スイッチを使用して IP アドレスを設定します。
2	次の電源 OFF→ON 時にユーザ作成の処理が H/W スイッチの値を読み出して、gerR_IN_SetIPAddressAndNodeNumber の引数へセットします。
3	gerR_IN_SetIPAddressAndNodeNumber が引数の値を R-IN32M4-CL3 に設定します。

iUserInitialization は「5.3.1 初期化処理」を参照してください。

gerR\_IN\_SetIPAddressAndNodeNumber は「6.4.1(4) gerR\_IN\_SetIPAddressAndNodeNumber」を参照してください。

## (2) ソフトウェア設定

## (a) GX Works3

GX Works3 を使って R-IN32M4-CL3 適用製品の IP アドレスを設定します。

詳細は「4.9 デバイス局の IP アドレス設定」を参照してください。

## (b) ユーティリティソフト

R-IN32M4-CL3 適用製品に不揮発性メモリを実装し、不揮発性メモリから IP アドレスを設定します。不揮発性メモリに IP アドレスを書きこむ際は、ユーザ作成のユーティリティソフトを使用します。

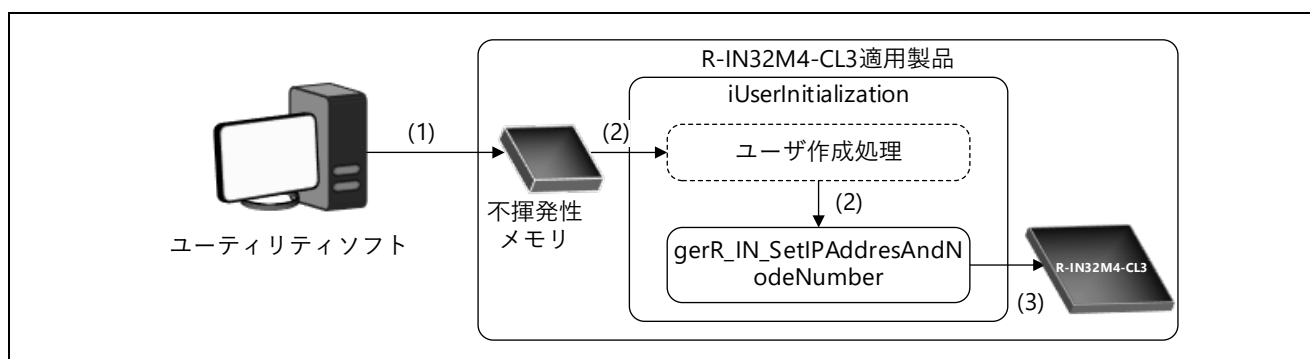


図 3.5 IP アドレス設定イメージ（ユーティリティソフト）

手順	内容
1	ユーティリティソフトを使用し、任意の転送方法で IP アドレス設定値を不揮発性メモリへ書き込みます。
2	次回の電源 OFF→ON 時に、ユーザ作成の処理が不揮発性メモリに保存している IP アドレス設定値を読み出して、gerR_IN_SetIPAddressAndNodeNumber の引数へセットします。
3	gerR_IN_SetIPAddressAndNodeNumber が引数の値を R-IN32M4-CL3 に設定します。

iUserInitialization は「5.3.1 初期化処理」を参照してください。

gerR\_IN\_SetIPAddressAndNodeNumber は「6.4.1(4) gerR\_IN\_SetIPAddressAndNodeNumber」を参照してください。

### 3.6 異常時のサイクリック出力保持/クリア機能の対応検討

異常時のサイクリック出力保持/クリア機能は、マスタ局アプリケーションが停止/異常のとき、または自局がデータリンクから解列したとき、受信したサイクリックデータ(RY および RWw)の自局外部への出力を続行する(Hold)、または停止する(Clear)機能です。

マスタ局の停止/異常、またはデータリンクからの解列に対してのフェールセーフとして、本機能に対応するかをあらかじめ検討してください。なお、本機能に対応する場合、サンプルコードに保持/クリアに関する処理(Hold/Clear 処理)を追加する必要があります。

マスタ局アプリケーションが停止/異常したときの Hold/Clear 処理は、「5.4.1 サイクリック受信処理」を参照してください。データリンクから解列したときの Hold/Clear 処理は、「5.4.4 通信状態更新処理」を参照してください。

### 3.7 エンジニアリングツール各種機能への対応検討

マスタ局のシーケンサ CPU に接続したエンジニアリングツール※1 を使うことで、以下の機能を実行することができます。R-IN32M4-CL3 適用製品の仕様として、各機能に対応するかをあらかじめ検討してください。

【注】 ※1 三菱電機のシーケンサの場合は GX Works3 を指します。

表 3.7 エンジニアリングツールの機能

項目		R-IN32M4-CL3 適用製品に必要な事項
デバイス局パラメータ処理・コマンド実行		CSP+ファイルを「図 3.6」の範囲③まで記述する。
a	パラメーター一括読出し/一括書込み	SLMP の要求受信・応答送信処理 (SLMP コマンド：一括読出し(0613H)、一括書込み(1613H))
b	パラメータ自動設定	SLMP の要求受信・応答送信処理 (SLMP コマンド：デバイス局パラメータ自動設定関連(0EB0H など))

#### 3.7.1 デバイス局パラメータ処理・コマンド実行

本機能により、R-IN32M4-CL3 適用製品のパラメータ設定とコマンド実行を、プログラミングすることなく実施できます。R-IN32M4-CL3 適用製品のエンドユーザに対してパラメータ設定とコマンド実行に対するプログラミングを軽減することができます。

なお、本機能は、CSP+ファイルを作成する必要があります。そのうえで、CSP+ファイルに記述した SLMP コマンド(マスタ局からの SLMP 要求)に応答する必要があります。

CSP+ファイルは、「3.8 CSP+および関連ファイルの作成準備」を参照してください。

SLMP の要求受信・応答送信処理については「5.6 ユーザプログラム詳細(SLMP コマンド実行関連)」および「5.7 ユーザプログラム詳細(デバイス局パラメータ自動設定関連)」を参照してください。(サンプルコードのユーザプログラムに SLMP の各コマンドの送受信処理を記述していますのでご活用ください。)

##### (1) パラメーター一括読出し/一括書込み

デバイス局の立上げに必要なパラメータの読出しと書込みができます。

##### (2) パラメータ自動設定

マスタ局が事前に保存したデバイス局パラメータと実際のデバイス局パラメータを比較します。不一致の場合、マスタ局がデバイス局にパラメータを自動で書き込みます。



### 3.8 CSP+および関連ファイルの作成準備

#### (1) 概要

CSP+は、CC-Link ファミリー接続対応製品の立上げ、運用・保守のために必要な情報を記述するための仕様です。CSP+ファイルおよび関連ファイル(画像ファイル、アイコンファイル、オブジェクトディクショナリファイル)を R-IN32M4-CL3 適用製品のエンドユーザに提供することで、CC-Link IE TSN の全局を 1 つのエンジニアリングツールから管理することができます。

CSP+の詳細は、「Control & Communication システムプロファイル仕様書」を参照してください。

CSP+ファイルおよび関連ファイルの作成の際は、「Control & Communication システムプロファイル作成ガイドライン」および「CSP+作成支援ツール」をご利用ください。併せて、付属のサンプル CSP+をご活用ください。(2.6 節参照)

#### (2) 作成範囲

リモート局として CSP+ファイルおよび関連ファイルを作成する範囲を示します。

コンフォーマンステストにおいて CSP+の確認がありますので、作成範囲①の CSP+ファイル、ならびに⑤オブジェクトディクショナリファイル※1 は必ず作成してください。そのうえで、R-IN32M4-CL3 適用製品の仕様として、どの機能(作成範囲②、③)に対応するかあらかじめ検討してください。

【注】 ※1 CANopen 通信を行う場合、作成が必須です。

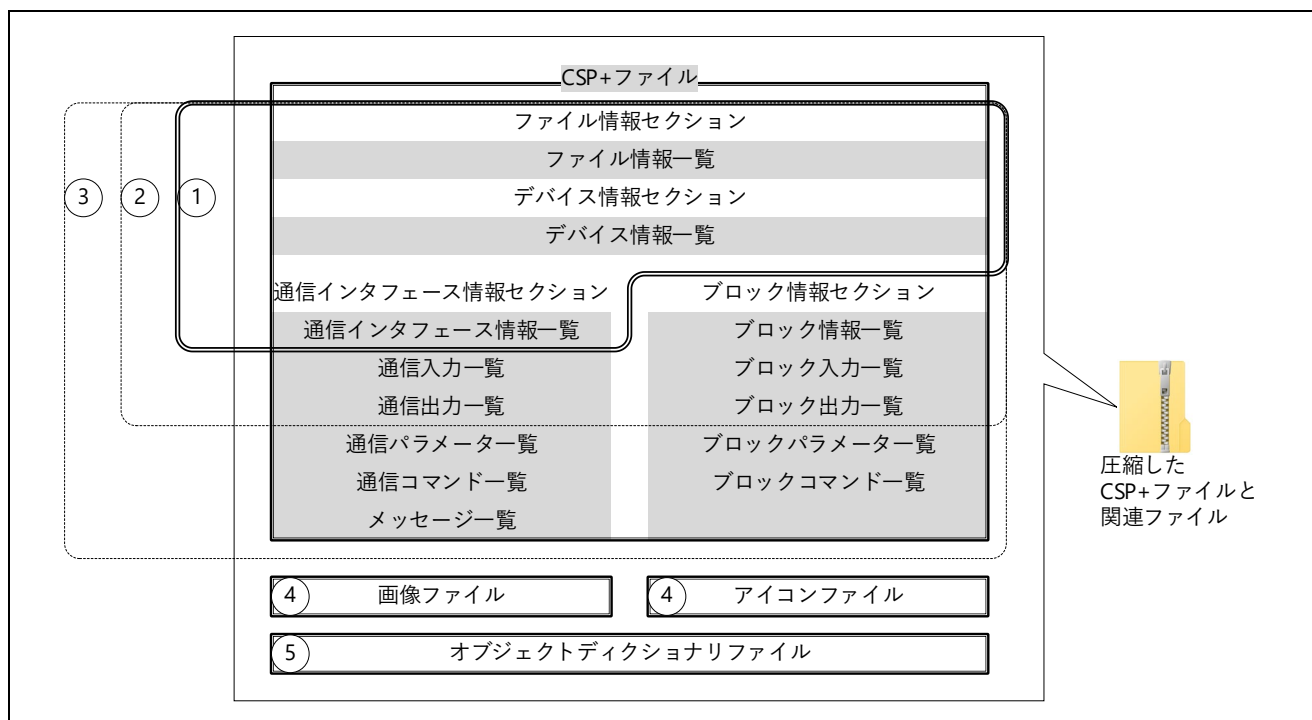


図 3.6 CSP+ファイルのセクション構成および関連ファイル

範囲	内容	作成要否
①	CC-Link 協会のコンFORMANCEテストにおいて、必須項目確認のために必要な情報(GX Works3 の場合) CC_IE_TSN 構成ウィンドウに R-IN32M4-CL3 適用製品が表示され、ネットワーク構成を簡単に作成することができます。	必須
②	デバイス局のリンクデバイスとマスタ局のデバイスとの割付けを表示するために必要な情報	任意
③	デバイス局パラメータ処理・コマンド実行をするために必要な情報(GX Works3 の場合) CC_IE_TSN 構成ウィンドウから CC-Link IE TSN 接続対応製品のパラメータを簡単に設定できます。	任意
④	エンジニアリングツール上で R-IN32M4-CL3 適用製品を画像またはアイコンで表示するために必要なファイル(GX Works3 の場合) CC_IE_TSN 構成ウィンドウにおいて、ネットワーク構成図に画像、ユニット一覧にアイコンが表示されます。	任意
⑤	R-IN32M4-CL3 適用製品の PDO マッピング設定に必要なファイル(GX Works3 の場合) CC_IE_TSN 構成ウィンドウから CC-Link IE TSN 接続対応製品の PDO マッピング設定ができます。	※1

【注】 ※1 CANopen 通信を行う場合、作成が必須です。

PDO マッピング設定の補足
CC-Link IE TSN における CANopen 通信では、オブジェクトディクショナリ(Object Dictionary)を CSV ファイルで定義します。オブジェクトディクショナリファイルにより CANopen 通信に対応した局に対して、RWr, RWw と PDO オブジェクトの紐づけができるようになります。

### (3) CANopen 通信時の増設ユニット

CANopen 通信時において、R-IN32M4-CL3 適用製品が例えば多軸一体型サーボアンプのように、通信を行う「基本ユニット(軸 1)」と通信を行わない「増設ユニット(軸 2～)」で構成する機器の場合、基本ユニットと増設ユニットの 2 種類の CSP+ファイルを作成する必要があります。

下記の情報は増設ユニットの CSP+ファイルに記述できませんので、基本ユニットの CSP+ファイルに記述してください。

- ・通信入カー一覧
- ・通信パラメーター一覧
- ・ブロック入カー一覧
- ・ブロックパラメーター一覧
- ・通信出カー一覧
- ・通信コマンド一覧
- ・ブロック出カー一覧
- ・ブロックコマンド一覧
- ・メッセージ一覧

### 3.9 コンフォーマンステスト受験の準備

コンフォーマンステストは、CC-Link IE TSN 接続対応製品の通信において高い信頼性を確保するために、機種ごとに実施していただく試験です。ユーザの開発した製品が CC-Link IE TSN の通信仕様を満たし、ネットワークに接続できることを確認します。

開発の準備段階からコンフォーマンステスト仕様書入手し、試験の要求仕様を満たすように設計してください。

コンフォーマンステストに合格した CC-Link IE TSN 接続対応製品は、認定品として「CC-Link パートナー製品カタログ」などに掲載することができます。

ポイント
------

開発いただく時期により、機能が未サポートの場合もございますので、コンフォーマンステストを行う際には、CC-Link 協会までお問い合わせ願います。
---

#### 3.9.1 1000BASE-T コンプライアンステスト

CC-Link IE TSN は 1000BASE-T に準拠していますので、R-IN32M4-CL3 適用製品は IEEE802.3ab の仕様に基づいた 1000BASE-T コンプライアンステストを実施することを推奨します。

1000BASE-T コンプライアンステストでは、伝送路の波形確認として Ethernet PORT から 4 つのテスト用波形を測定します。サンプルコードの「UserIEEETest」に波形を出力する処理を記述していますので、ご活用ください。

UserIEEETest の詳細は「5.11.1 H/W テスト処理(IEEE802.3ab コンプライアンステスト)」を参照してください。

コンプライアンステストに必要な事項
-------------------

- |   |
|---|
| (1) UserIEEETest はコンパイルスイッチの定義” USE_COMPLIANCETEST” が有効のとき、アイドルタスクから呼び出されます。このとき、R-IN32M4-CL3 適用製品はデータリンクを開始せず、コンプライアンステストだけを実行します。<br>そのため、例えばデータリンクせずにテストのみ行う専用 F/W を作成するなど、任意のタイミングで UserIEEETest を呼び出せる方法を検討してください。 |
| (2) UserIEEETest はテスト用波形を出力する「テストモード(1~4)」を引数として「gerR_IN_IEEETest」を実行し、PHY レジスタにテスト用波形データを書き込みます。ただし、「テストモード(1~4)」を任意に切り替える処理がありません。<br>そのため、例えば H/W スイッチを使用して「テストモード(1~4)」を指定するなど、4 つのテスト用波形を任意に切り替えられる方法を検討してください。    |

## 4. R-IN32M4-CL3 適用製品のできること

本章は、R-IN32M4-CL3 適用製品(CC-Link IE TSN)で、使用できる機能の概略を示します。

表 4.1 R-IN32M4-CL3 適用製品の機能

No.	機能	機能の実装区分	処理の区分※1	参照
1	サイクリック伝送 (CyclicMs、CyclicSs)	必須	ユーザプログラム	4.1
2	トランジェント伝送 (NRSV-Transient)	—	-	4.2
	a. ユーザ用のトランジェント伝送	任意	ユーザプログラム	4.2.4(1)
	b. 管理用のトランジェント伝送	必須	R-IN32M4-CL3 ドライバ	4.2.4(2)
3	LED による状態表示	任意	ユーザプログラムまたは R-IN32M4-CL3 ドライバ	4.3
4	CC-Link IE TSN 診断	任意	R-IN32M4-CL3 ドライバ	4.4
5	予約局設定	任意	R-IN32M4-CL3 ドライバ	-
6	IEEE1588 や IEEE 802.1AS による時刻同期	必須※2	R-IN32M4-CL3 ドライバ	-
7	ネットワーク同期通信	任意※2 ※3	ユーザプログラム	4.5
8	ネットワーク同期通信時の同期信号出力	任意※2 ※3	-(ハードウェア)	4.5.1
9	CANopen 通信	任意※3	ユーザプログラム	4.6
10	CANopen 通信時の増設ユニット	任意	ユーザプログラム	4.6.1
11	安全通信時の安全 PDU 送受信	任意※3	ユーザプログラム	4.7
12	SLMP 経由の通信速度・CC-Link IE TSN Class 設定	任意	ユーザプログラム	4.8
13	デバイス局の IP アドレス設定	任意	ユーザプログラム	4.9
14	デバイス局のパラメータ設定・コマンド実行	任意	ユーザプログラム	4.10
15	SNMP コミュニティ名設定	任意	ユーザプログラム	4.11

【注】 ※1 「ユーザプログラム」は「5 ユーザプログラムの作成」に示す処理を実装する必要があります。

「R-IN32M4-CL3 ドライバ」は R-IN32M4-CL3 ドライバに実装済みですので、ユーザは特に意識する必要はありません。

※2 CC-Link IE TSN Class A の場合は不要です。

※3 ネットワーク同期通信時は、安全 PDU 送受信を使用できません。また、CC-Link IE TSN Class A 動作時は使用できません。

CANopen 通信時は、ネットワーク同期通信および安全 PDU 送受信を使用できません。

安全 PDU 送受信時は、ネットワーク同期通信および CANopen 通信を使用できません。

### <ポイント>

同じネットワークに接続されている他のデバイス局と自局の動作タイミングを合わせたい場合、ネットワーク同期通信機能(上表 No.7)の実装が必要になります。

例えば、R-IN32M4-CL3 適用製品とサーボアンプで動作タイミングを合わせたい、R-IN32M4-CL3 適用製品同士で動作タイミングを合わせたい等の場合は、本機能の実装を検討してください。

## 4.1 サイクリック伝送機能

サイクリック伝送は、リンクデバイスを使用してマスタ局と定期的にデータ交信する機能です。R-IN32M4-CL3 適用製品は、CyclicMs を受信し、CyclicSs を送信します。サイクリックデータの流れを以下に例示します。

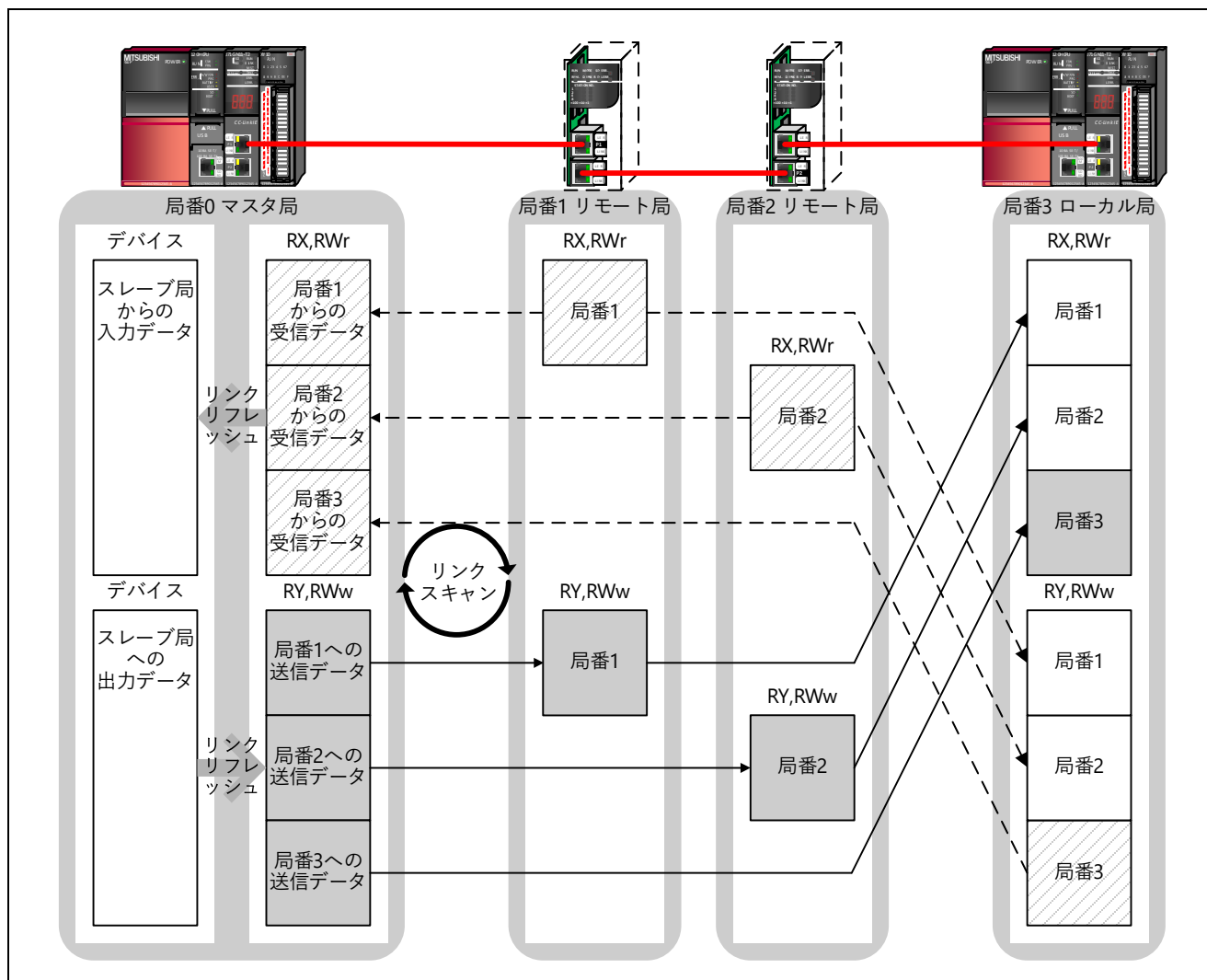


図 4.1 サイクリックデータの流れ

マスタ局のリンクデバイス(RY および RWw)の状態が、リンクスキャンにより自局のリンクデバイス(RY および RWw)に格納されます。

自局のリンクデバイス(RX および RWr)の状態が、リンクスキャンによりマスタ局のリンクデバイス(RX および RWr)に格納されます。

サイクリック伝送は、ユーザプログラムの「5.4.1 サイクリック受信処理」および「5.4.3 サイクリック送信処理」に記載する処理が必要です。

## 4.2 トランジェント伝送機能

トランジェント伝送は、他局からの交信要求があったとき、または自局が交信要求したときに、データ交信を行う機能です。他局のデバイス/バッファメモリへ直接アクセスしてデータ交信します。

R-IN32M4-CL3 適用製品が使用できるトランジェント伝送のうち、ユーザが任意のデータを送受信できる方式は、以下 2 種類あります。いずれの方式も、R-IN32M4-CL3 適用製品は NRSV-Transient 型で送受信します。

- SLMP を使った通信
- 汎用 TCP/IP 通信

SLMP メモリ読出し命令の場合で、トランジェントデータの流れを以下に例示します。

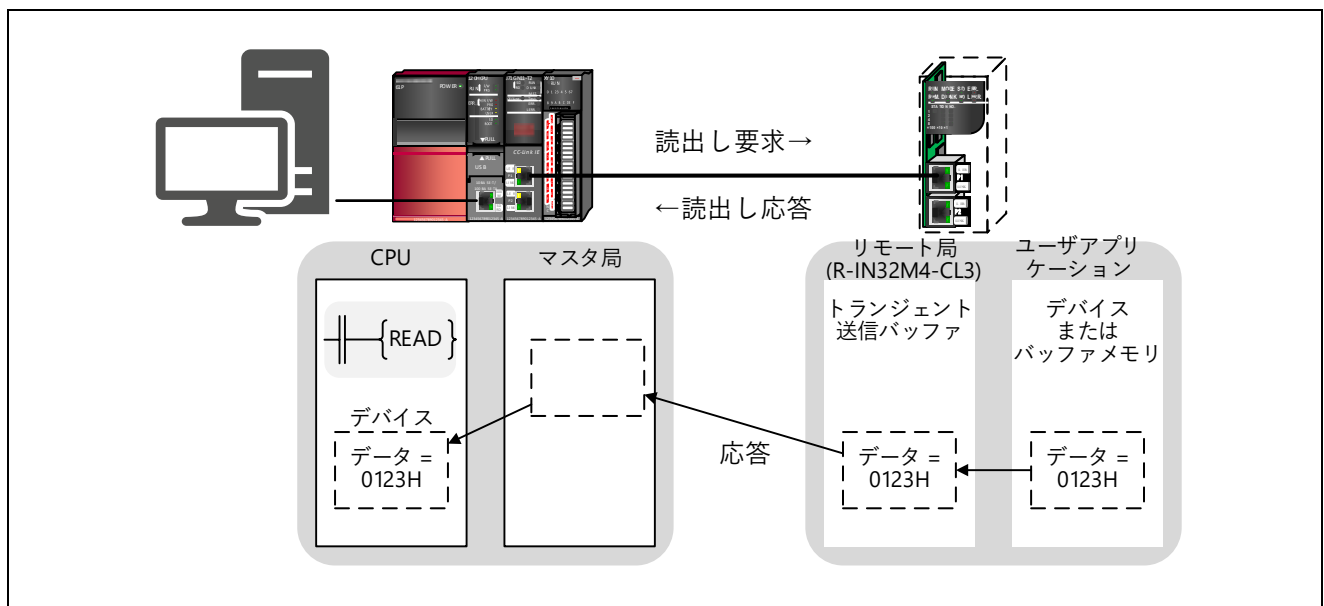


図 4.2 トランジェントデータの流れ

トランジェント伝送は、ユーザプログラムの「5.5 ユーザプログラム詳細(状態管理およびトランジェント伝送関連)」に記載する処理が必要です。なお、以降のトランジェント伝送は、「SLMP を使った通信」について記載します。

### 4.2.1 トランジェント伝送のクライアント・サーバ機能

トランジェント伝送には、クライアント機能とサーバ機能があります。

クライアント機能は、サーバ機能を持つ局に対しトランジェント要求を送信する機能です。

サーバ機能は、クライアント機能を持つ局からのトランジェント要求に対してトランジェント応答を送信する機能です。

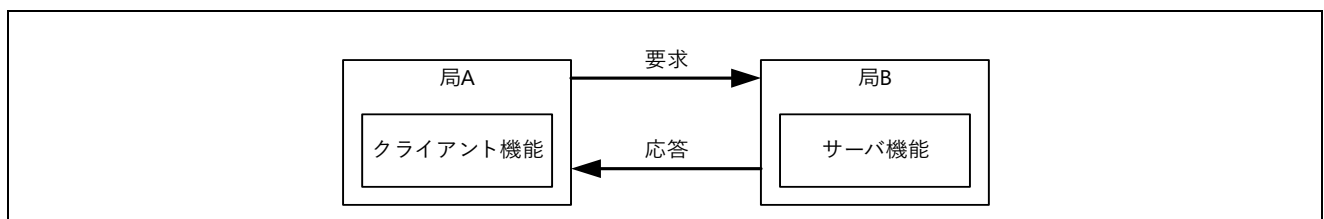


図 4.3 トランジェント クライアント/サーバ機能

### 4.2.2 SLMP の伝送フレームについて

R-IN32M4-CL3 適用製品は IP フレーム(Ether Type : 0800H)、または CC-Link IE TSN フレーム(Ether Type : 890FH)の AcyclicData を使って SLMP を伝送します。

#### (1) IP フレーム

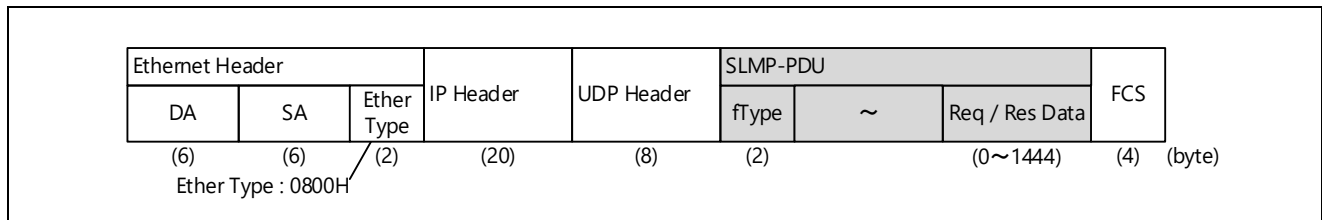


図 4.4 IP フレームの SLMP 構造

#### (2) CC-Link IE TSN フレーム(AcyclicData)

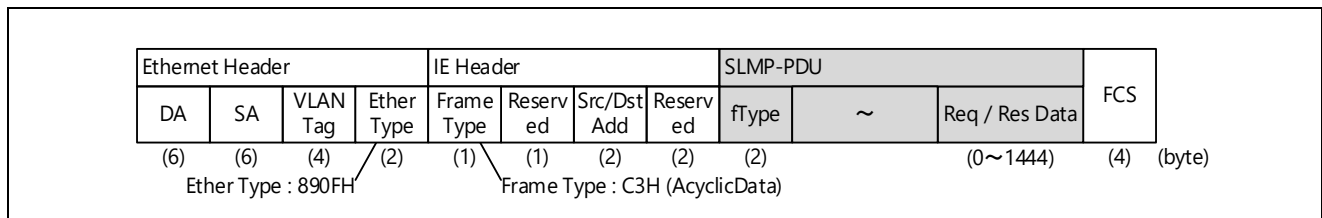


図 4.5 CC-Link IE TSN フレーム(AcyclicData)の SLMP 構造

### 4.2.3 SLMP のフレーム種別について

SLMP には 3 つのフレーム種別(fType)があります。

R-IN32M4-CL3 適用製品はすべての SLMP フレームを送受信できます。ただし、バイナリモードのみ対応します。

各種別の詳細は、CC-Link 協会の「SLMP 仕様書」または関連マニュアル「SLMP リファレンスマニュアル」を参照してください。

表 4.2 送受信できる SLMP フレームの一覧

No.	種別	種別値 (fType)	略号
1	LMT (Large-Node Number-Multi-Transmission type)	要求 0068H, 応答 00E8H	6E※1
2	MT (Multi-Transmission type)	要求 0054H, 応答 00D4H	4E※1
3	ST (Single-Transmission type)	要求 0050H, 応答 00D0H	3E※1

【注】※1 : 6E, 4E, 3E の表記は、関連マニュアル「SLMP リファレンスマニュアル」やサンプルコードなどで一部使用しています。

### 4.2.4 SLMP のコマンドについて

SLMP には用途に応じた種々のコマンドがあります。R-IN32M4-CL3 適用製品のユーザプログラム側と R-IN32M4-CL3 ドライバ側でそれぞれ種々のコマンドを送受信します。

サンプルコードのデフォルトで送受信できる SLMP コマンドを次表に示します。

表に記載していないコマンドを送受信する場合は、CC-Link 協会「SLMP 仕様書」または関連マニュアル「SLMP リファレンスマニュアル」を参照して、該当コマンドの送受信処理を追加してください。

## (1) ユーザプログラムが送受信する SLMP コマンド

下記コマンドは、IP フレーム(ポート番号 45239)または AcyclicData フレームを使用して送受信します。送受信処理はユーザプログラムが行います。送受信処理の詳細は、表中の参照先を確認してください。

表 4.3 ユーザプログラムが送受信する SLMP コマンド

No	種別	操作	コマンド	サブ コマンド	対応 要否	伝送 フレーム	参照
自局がクライアントのとき(要求コマンドを送信し、応答コマンドを受信するとき)							
1	デュアルポートメモリ	一括読出し	0613H	0000H	任意	IP	5.6.3 5.6.4
自局がサーバのとき (要求コマンドを受信し、応答コマンドを送信するとき)							
2	デュアルポートメモリ	一括読出し	0613H	0000H	任意	IP	5.6.1
3		一括書込み	1613H	0000H	任意	IP	5.6.2
4	リモート制御	リモートリセット	1006H	0000H	任意	IP	5.6.5
5		インディケータ表示	3070H	0000H	任意	IP AcyclicData	5.6.6
6	機器接続	接続機器の検出	0E30H	0001H	任意	IP	5.8.1
7		接続機器の IP アドレス設定	0E31H	0000H	任意	IP AcyclicData	5.6.7
8	エラー履歴	エラー履歴クリア	1619H	0000H	任意	IP	5.6.8
9	デバイス局パラメータ 自動設定	通信設定取得	0EB0H	0001H	任意	IP	5.7.1
10		パラメータ配信要否チェック	0EBEH	0001H	任意	IP	5.7.2
11		リストア開始通知	0EB8H	0001H	任意	IP	5.7.4
12		リストア終了通知	0EB9H	0001H	任意	IP	5.7.5
13		パラメータデータ書込み	0EBAH	0001H	任意	IP	5.7.6
14	イベント履歴	ネットワーク時刻のオフセット配信	3062H	0000H	任意	IP	5.6.9
15		ネットワーク時刻の配信	3063H	0000H	任意	IP	5.6.10
16	ウォッチドッグカウンタ	ウォッチドッグカウンタ設定	3210H	0000H	※1	IP	5.6.11
17	CAN アプリケーション オブジェクトへの アクセス	オブジェクト読出し	4020H	0001H	※2	IP	5.9.4
18		オブジェクト書込み	4020H	0002H	※2	IP	5.9.5
19		オブジェクト SubIndex 連続読出し	4020H	0005H	※2	IP	5.9.6
20		オブジェクト SubIndex 連続書込み	4020H	0006H	※2	IP	5.9.7
21		NMT 状態の取得	4020H	0007H	※2	IP	5.9.8
22		NMT 状態の設定	4020H	0008H	※2	IP	5.9.9
23	CC-Link IE TSN 機器の パラメータ設定	機能設定のサポート情報取得	3080H	0000H	任意	IP	5.8.2
24		機能設定の読出し(通信速度)	3081H	0000H	任意	IP	5.8.3
25		機能設定の読出し(CC-Link IE TSN Class)	3081H	0001H	任意	IP	5.8.5
26		機能設定の書込み(通信速度)	3082H	0000H	任意	IP	5.8.4
27		機能設定の書込み(CC-Link IE TSN Class)	3082H	0001H	任意	IP	5.8.6
28	SNMP コミュニティ名設定	SNMP コミュニティ名の設定	3084H	0000H	任意	IP	5.6.12

※1：ネットワーク同期通信を行う場合に必要です。

※2：CANopen 通信を行う場合に必要です。



## 【各 SLMP コマンドの用途】

- デュアルポートメモリ：対象局のメモリ(バッファメモリなど)に対して、汎用データの読出しまたは書込みを行います。
- リモート制御：ネットワーク経由で、対象局を操作します。
- エラー履歴：エラー情報を管理します。
- デバイス局パラメータ自動設定：マスタ局がデバイス局のパラメータを自動で配信します。
- イベント履歴：マスタ局の時刻に同期できる補正データを配信します。
- ウォッチドッグカウンタ：マスタ局と同期しているか否かを監視するための、カウンタを管理します。
- CAN アプリケーションオブジェクトへのアクセス：SDO、および NMT の送受信を行います。
- CC-Link IE TSN 機器のパラメータ設定：通信速度や CC-Link IE TSN Class を SLMP 経由で変更します。
- SNMP コミュニティ名設定：SNMP のコミュニティ名を SLMP 経由で変更します。

## (2) ドライバが送受信する SLMP コマンド

下記コマンドは IP フレーム(ポート番号 45238)を使用して送受信します。送受信処理は R-IN32M4-CL3 ドライバが行いますのでユーザは特に意識する必要ありません。

参考として、R-IN32M4-CL3 ドライバが送受信する SLMP コマンドを以下に示します。

表 4.4 R-IN32M4-CL3 ドライバが送受信する SLMP コマンド

No.	種別	操作	コマンド	サブ コマンド	伝送 フレーム	処理の区分
1	ネットワーク 管理	ネットワーク設定(メイン)	0E90H	0000H	IP	サーバ
2		ネットワーク設定(タイムスロット情報)	0E90H	0001H	IP	サーバ
3		各種通知	0E94H	0000H	IP	サーバ
4		各種通知	0E94H	0000H	IP	クライアント
5	パラメータ 管理	デバイス局設定	0E92H	0000H	IP	サーバ
6		サイクリック伝送設定(メイン)	0E93H	0000H	IP	サーバ
7		サイクリック伝送設定(送信サブペイロード情報)	0E93H	0001H	IP	サーバ
8		サイクリック伝送設定(受信サブペイロード情報)	0E93H	0002H	IP	サーバ
9		サイクリック伝送設定(受信対象アドレス)	0E93H	0003H	IP	サーバ

### 4.3 状態表示機能

R-IN32M4-CL3 は自局の状態と Ethernet PORT の状態を、LED を使用して表示することができます。

#### 4.3.1 LED による状態表示

エンドユーザの使いやすさの観点から、下表の USER LED 以外のすべての LED を実装することを推奨します。

USER LED は用途に応じて実装してください。

なお、R-IN32M4-CL3 適用製品の筐体から LED の点灯が目視できるように実装してください。

LED の色・形状は規定がありませんので、ユーザの仕様に応じて LED を選定してください。

表 4.5 LED 状態表示一覧

LED 種別	LED 名称	内容
自局状態 表示	RUN	運転状態を表示します。
	点灯	正常運転中
	消灯	ハードウェア異常または WDT エラーが発生
	SD/SDRD1※1	SD モード
		データの送信状態を表示します。
	点灯	データ送信中
	消灯	データ未送信
	RD/SDRD2※1	RD モード
		データの受信状態を表示します。
	点灯	データ受信中
	消灯	データ未受信
	D LINK	データリンクの状態を表示します。
	点灯	データリンク中(サイクリック伝送中)
	消灯	データリンク未実施(解列中)
	点滅	データリンク中(サイクリック伝送停止中)
	ERR.	R-IN32M4-CL3 のエラー状態を表示します。
	点灯	自局でエラーが発生
	消灯	正常動作中
	L ERR.	受信データおよび回線のエラー状態を表示します。 本 LED が点灯したとき、L ER LED でエラーを検出したポートを確認できます。
	点灯	異常なデータを受信、またはループバック実施中
	消灯	正常なデータを受信、またはループバック未実施
	USER LED1	ユーザが定義した状態を表示します。
	USER LED2	ユーザが定義した状態を表示します。
Ethernet PORT1 状態表示	LINK	点灯 リンクアップ中 消灯 リンクダウン中
	L ER	点灯 異常なデータを受信、またはループバック実施中 消灯 正常なデータを受信、またはループバック未実施
	LINK	点灯 リンクアップ中 消灯 リンクダウン中
	L ER	点灯 異常なデータを受信、またはループバック実施中 消灯 正常なデータを受信、またはループバック未実施

【注】 ※1 SD/SDRD1 LED と RD/SDRD2 LED は、点灯モードを「SD/RD」、「SDRD1/SDRD2」に切り替えることができます。詳細は「4.3.2 SD/SDRD1、RD/SDRD2 LED の点灯モード設定」を参照してください。

### 4.3.2 SD/SDRD1、RD/SDRD2 LED の点灯モード設定

SD/SDRD1 LED および RD/SDRD2 LED は、それぞれ 2 つの点灯モードがあり、ユーザの仕様に応じてモードを指定してください。点灯モードは R-IN32M4-CL3 ドライバインタフェース関数「gerR\_IN\_SetSDRDLEDMode」(6.4.7(10))にて設定することができます。

表 4.6 点灯モード

点灯モード	内容
SD モード RD モード	送信状態と受信状態を別々の LED で表示する。 ただし、Ethernet PORT1 と Ethernet PORT2 を区別しない。
SDRD1 モード SDRD2 モード	Ethernet PORT1 の送受信状態と Ethernet PORT2 の送受信状態を別々の LED で表示する。 ただし、送信状態と受信状態を区別しない。

### 4.3.3 LED の制御

LED の制御は、H/W で制御する LED と S/W で制御する LED の 2 種類あります。

H/W で制御する LED は、R-IN32M4-CL3 が自局の状態に応じて該当する LED を点灯/消灯するため、S/W による制御が不要です。

S/W で制御する LED は、R-IN32M4-CL3 ドライバインタフェース関数を使用して、自局の状態に応じて該当する LED を点灯/消灯します。R-IN32M4-CL3 ドライバインタフェース関数の詳細は、「6.4.7 LED 制御」を参照してください。

以下に各 LED の制御とリセット/エラー時の点灯状態を示します。

表 4.7 LED 制御とリセット/エラー時の点灯状態

種別	LED 名称	R-IN32M4-CL3 出力 信号名	制御区分	パワーオン リセット	システム リセット	内部 WDT エラー時 外部 WDT エラー時
自局状態 表示	RUN	CCI_RUNLEDZ	H/W または S/W 制御	消灯	消灯	消灯
	SD/SDRD1	CCI_SDLEDZ	H/W 制御	消灯	-	-
	RD/RDSD2	CCI_RDLEDZ	H/W 制御	消灯	-	-
	D LINK	CCI_DLINKLEDZ	H/W または S/W 制御	消灯	消灯	消灯
	ERR.	CCI_ERRLEDZ	H/W または S/W 制御	消灯	消灯	点灯
	L ERR.	-	H/W または S/W 制御	-	-	-
	USER LED1	RP20	S/W 制御	消灯	消灯	消灯
	USER LED2	RP21	S/W 制御	消灯	消灯	消灯
Ethernet PORT 1 状態表示	LINK	PHY0_LED0	H/W 制御	消灯	消灯	-
	L ER	CCI_LERR1LEDZ	H/W または S/W 制御	消灯	-	-
Ethernet PORT 2 状態表示	LINK	PHY1_LED0	H/W 制御	消灯	消灯	-
	L ER	CCI_LERR2LEDZ	H/W または S/W 制御	消灯	-	-

#### 4.3.4 USER LED の制御

USER LED は、R-IN32M4-CL3 適用製品の仕様に合わせて自由に定義できる LED です。

例として、以下の状態を表示する場合に USER LED を点灯・消灯・点滅してください。

- 自局のオンラインモード/オフラインモード(ハードウェアテストモード)状態を表示する。
- ハードウェアテストなど各種テストにおける、正常/異常状態を表示する。

USER LED の制御は、R-IN32M4-CL3 ドライバインタフェース関数「gerR\_IN\_SetUSER1LED」、および「gerR\_IN\_SetUSER2LED」を使用して、点灯制御してください。R-IN32M4-CL3 ドライバインタフェース関数の詳細は「6.4.7 LED 制御」を参照してください。

#### 4.3.5 L ERR. LED の制御

L ERR. LED は CCI\_LERR1LEDZ および CCI\_LERR2LEDZ の論理和回路によって点灯・消灯します。

また、R-IN32M4-CL3 のポート端子からの信号と論理和をとることで、ユーザプログラムからでも制御することができます。R-IN32M4-CL3 適用回路中に OR 回路を実装して、信号を結線してください。

なお、本書はポート端子 P57 を使用した場合で例示します。必要に応じてポート端子を変更してください。

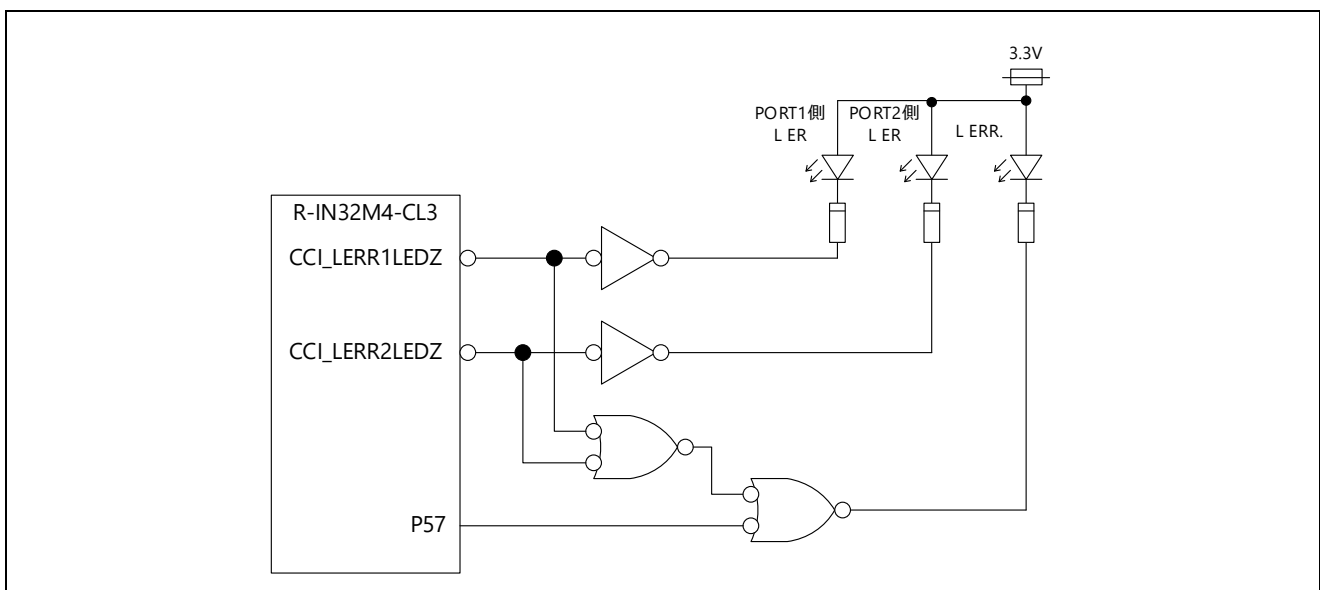


図 4.6 L ERR.点灯回路 (P57 使用例)

P57 端子の出力と L ERR. LED の状態は、下表の関係となります。

表 4.8 L ERR. LED の点灯条件

CCI_LERR1LEDZ	CCI_LERR2LEDZ	P57	L ERR. LED
X	X	Low	点灯(異常)
Low (異常)	X	High	点灯(異常)
X	Low (異常)	High	点灯(異常)
High (正常)	High (正常)	High	消灯(正常)

X : High or Low

起動時、P57 端子の状態は次のようになっています。

- ・リセット中：Hi-Z(High)
- ・I/O スタンバイ時：Hi-Z(High)

また、P57 の使用にあたり、下表のレジスタ設定が必要となります。

表 4.9 P57 端子を出力端子にするためのレジスタ設定

レジスタ名	機能	初期値	設定値
PM5B	入力/出力の切替え	L FFH (入力モード)	7FH (P57 が出力ポートになる)

表 4.10 (参考)P57 端子に関連するレジスタ設定

レジスタ名	機能	初期値	設定値
PMC5B	ポート兼用機能の切替え	00H (ポートモード)	00H (ポートモード)
DRCTLP5H	バッファ機能の切替え	0000 9000H (Pull-up 抵抗)	0000 9000H (Pull-up 抵抗)

## 4.4 CC-Link IE TSN 診断

CC-Link IE TSN 診断機能は、CC-Link IE TSN 全体を診断して、設定異常や通信異常を検出する機能です。

本機能は、マスタ局からの診断要求に応答することで、マスタ局に各局の状態が収集されます。

応答の処理は、R-IN32M4-CL3 ドライバが自動で行います。ユーザプログラムに特定の処理を実装する必要はありません。GX Work3 を使用した場合の診断画面を以下に示します。

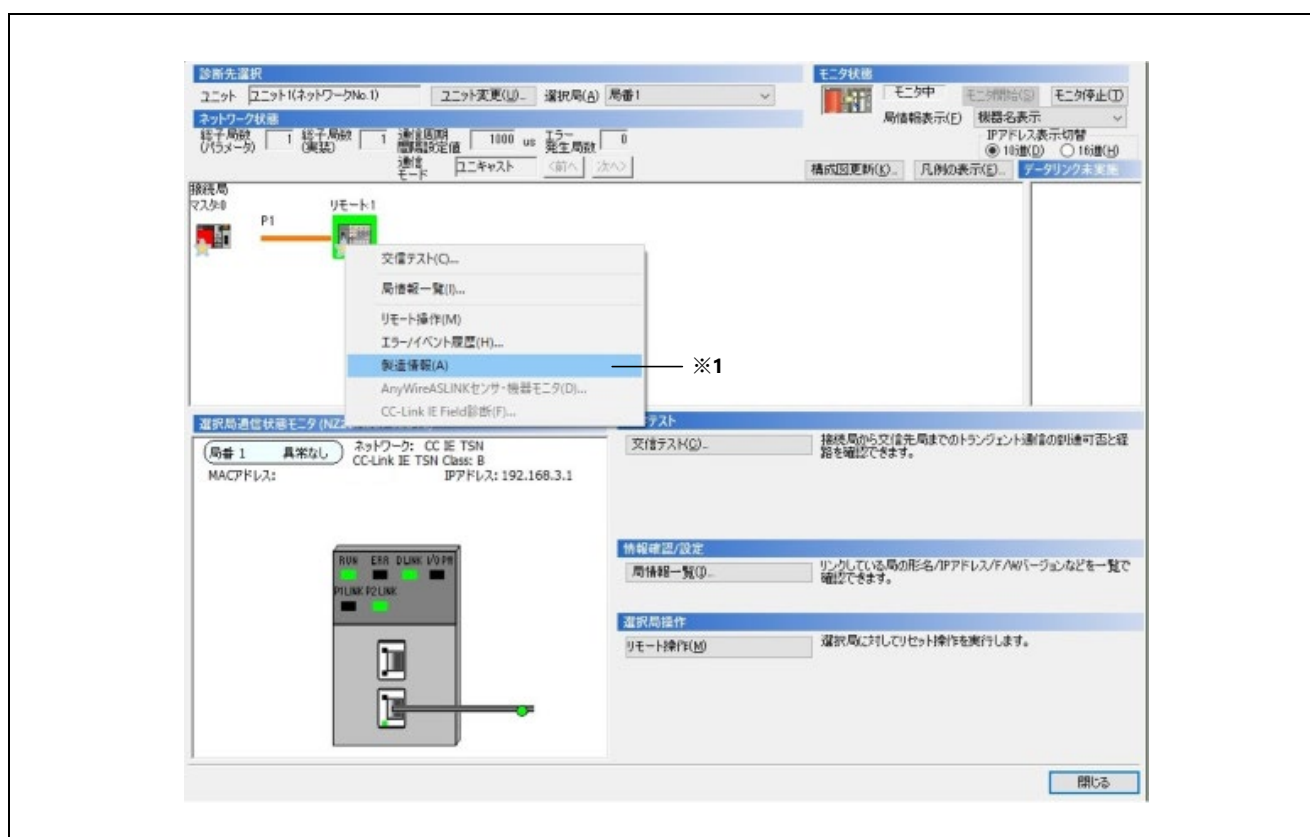


図 4.7 診断画面 (GX works3 の場合)

※1：R-IN32M4-CL3 サンプルコードが「製造情報」に未対応のため、「製造情報」を実行すると、GX Works3 のエラーメッセージが表示されます。

## 4.5 ネットワーク同期通信

ネットワーク同期通信は、マスタ局の同期周期でデバイス局の内部処理を行う機能です。

本機能により、同じネットワークに接続されている他のデバイス局と内部処理の動作タイミングを揃えることができます。

本機能では CC-Link IE TSN 局間の正常なサイクリック伝送を保証するためにウォッチドッグカウンタを使用します。ウォッチドッグカウンタは、マスタ局とデバイス局の受信データが毎通信周期更新されていることをマスタ局とデバイス局の相互で監視するためのカウンタです。

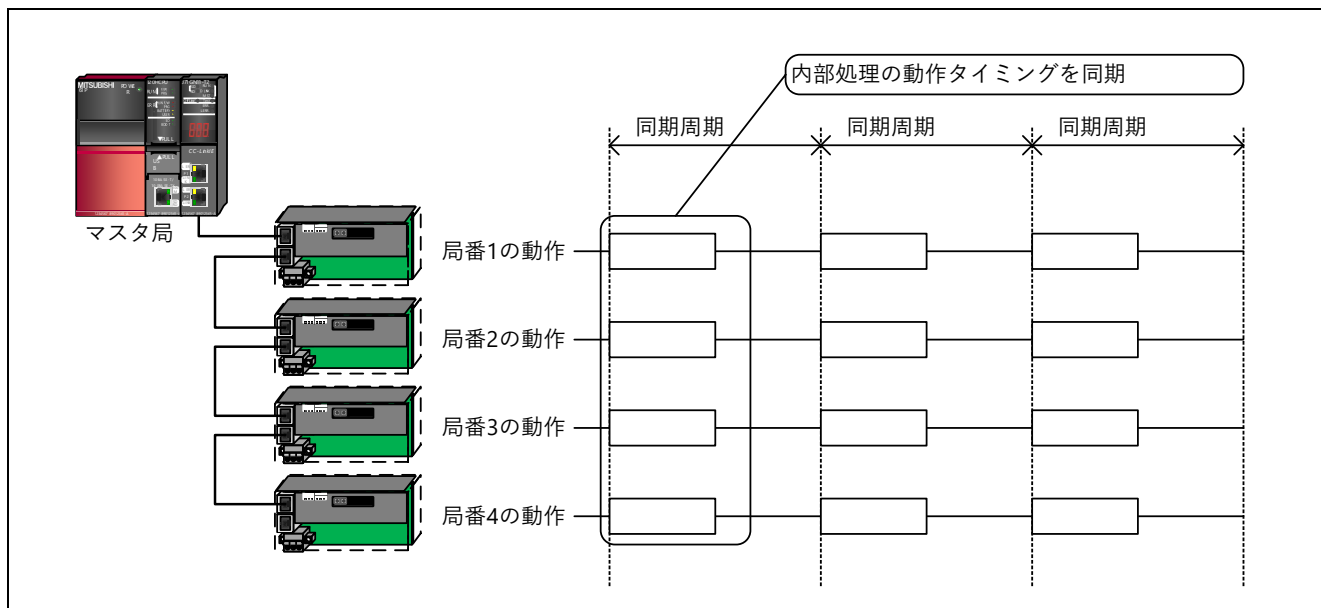


図 4.8 ネットワーク同期通信のイメージ

ネットワーク同期通信を使用するためには、以下の事項が必要です。

No.	必要事項	参照
1	ネットワーク同期通信関連のタスクと、タスクが呼び出す処理を実装する。	表 4.11
2	ウォッチドッグカウンタ情報(stWdcInfo)を設定する。	表 6.14
3	ウォッチドッグカウンタ関連のユーザプログラムを実装する。	5.6.11

表 4.11 ネットワーク同期通信関連のタスク

No	タスク	タスク ID	関連処理 (参照)	参照
1	低優先割込みタスク	TSKID_NX_LOW_INT	CC-Link IE TSN の低優先割込み発生時に起床し、割込み要因が「サイクリック伝送終了」(TS2 開始割込み)である場合に、「同期サイクリック通信処理」をコールバックします。	6.6(9)
2	同期タイミング処理タスク	TSKID_SYNCPROC	同期信号出力のタイミング <sup>※1</sup> で起床し、「同期タイミング処理」を実行します。	5.2.5
3	同期解列時処理タスク	TSKID_PERIO_DLERR	ネットワーク同期通信機能が有効な場合に定周期で起床し「同期解列時処理」を実行します。	5.2.6

※1：RJ71GN11-T2 をマスタ局とする場合は、TS1 開始が同期タイミング処理タスクの起床タイミングとなります。

その他、ネットワーク同期通信に対応したマスタ局が必要です。

マスタ局に RJ71GN11-T2 を使用する場合は、ファームウェアバージョン” 11” 以降のユニットを使用してください。

設定方法と機能の詳細は、以下関連マニュアルを参照してください。

- ・ MELSEC iQ-R CC-Link IE TSN ユーザズマニュアル(応用編)
- ・ MELSEC iQ-R ユニット間同期機能リファレンスマニュアル

#### <注意事項>

CC-Link IE TSN Class A で動作する場合、本機能を使用できません。

CC-Link IE TSN Class B で動作する場合、安全通信においては本機能を使用できません。

3つのタスクから呼び出される処理のタイミングチャートを示します。

図中の①～③は、「表 4.11」における関連処理です。

#### (1) ネットワーク同期通信のタスク/処理タイミングチャート (データリンク中)

「①同期サイクリック通信処理」がコールバックされ、受信データの読出しと送信データの書き込みを行います。「②同期タイミング処理」が同期させたいユーザアプリケーションの処理を実行します。

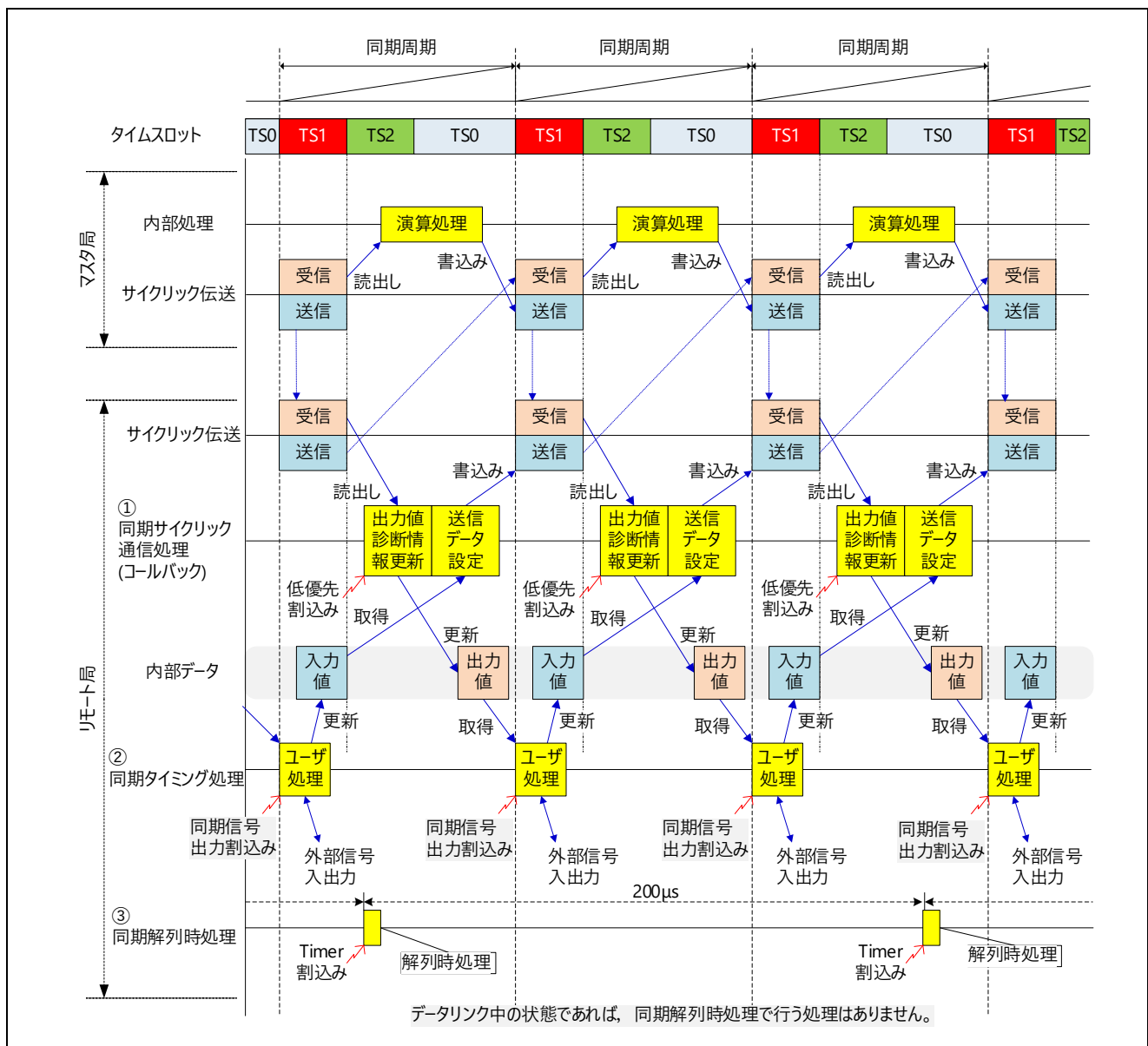


図 4.9 同期通信のタスク/処理タイミングチャート(データリンク中)

## (2) ネットワーク同期通信のタスク/処理タイミングチャート(データリンク未実施(解列状態))

解列状態では「①同期サイクリック通信処理」がコールバックされず、「②同期タイミング処理」も実行できません。「③同期解列時処理」で解列時に必要な処理を実行します。

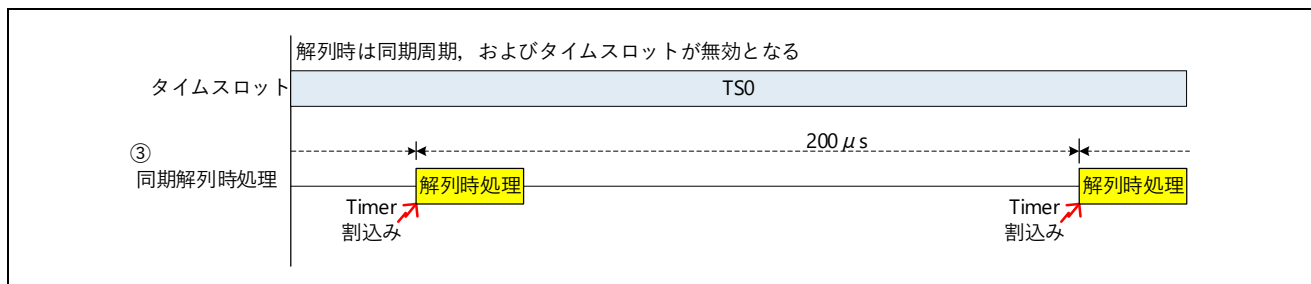


図 4.10 同期通信のタスク/処理タイミングチャート(データリンク未実施(解列状態))



4.5.1 ネットワーク同期通信時の同期信号出力

同期信号出力は、ネットワーク同期通信有効時に、同期周期の開始タイミングにてパルス信号を外部出力する機能です。ハードウェア (R-IN32M4-CL3)が外部出力処理を行うため、ソフトウェア (同期タイミング処理タスク)で行う外部信号出力よりも正確に、同期周期の開始タイミングを外部 MCU や FPGA などへ通知することができます。

本機能を使用するには、以下の事項が必要となります。

No.	必要事項	参照
1	R-IN32M4-CL3 の P51 端子を兼用機能 3「CCI_SYNC」に設定する。	
2	ユーザプログラムの R-IN32M4-CL3 の初期化処理にて、同期信号出力パルス幅(※1)を設定する。	表 6.17

※1：16ns～30,000ns(30us)の範囲で設定可能です。本機能を使用しない場合は 0 を設定してください。

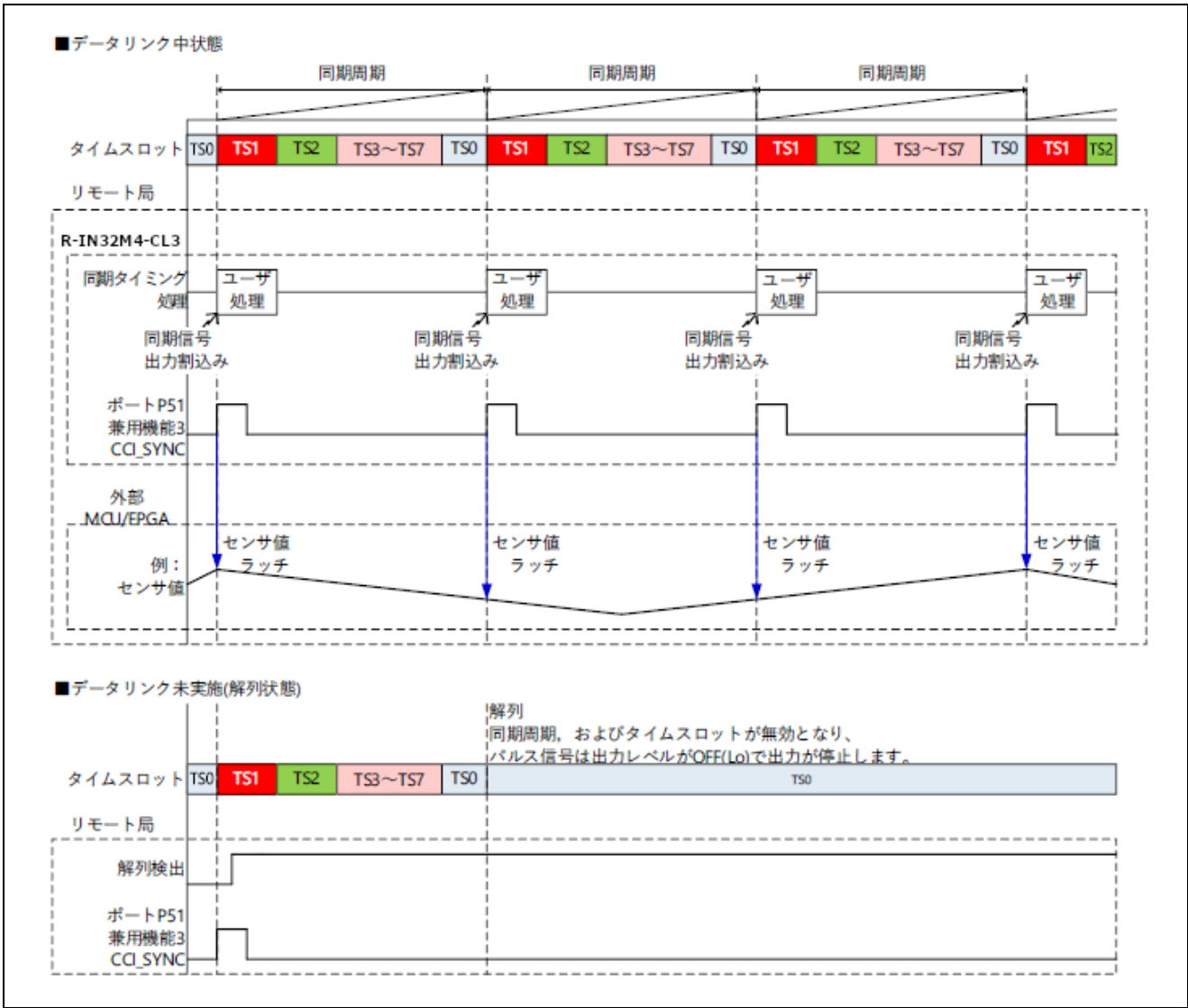


図 4.11 CCI\_SYNC パルス信号外部出力タイミングのイメージ

## 4.6 CANopen 通信

CANopen 通信は CANopen プロファイルに対応した機器を制御する機能です。

サイクリック伝送で送受信する PDO やトランジェント伝送(SLMP)で送受信する SDO を利用して、リンクデバイスと同等の操作で対象機器を制御します。

なお、本書は主に CC-Link IE TSN 上で SDO, PDO を送受信する方法について記載します。CANopen の詳細な仕様は、CANopen 規格書を参照してください。

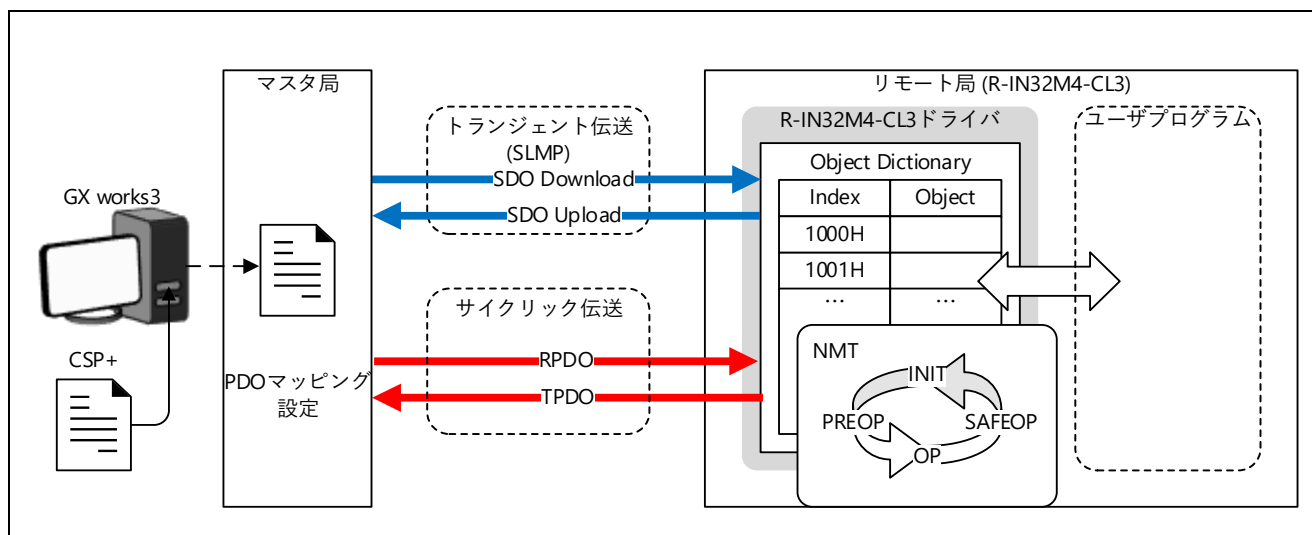


図 4.12 CC-Link IE TSN における CANopen 通信のイメージ

表 4.12 CC-Link IE TSN における CANopen 通信の概要

No	項目	概要
1	Object Dictionary (OD)	R-IN32M4-CL3 適用製品が保持する制御パラメータ、指令値などの各種データの集合体をオブジェクトディクショナリ (Object Dictionary) と呼びます。 Object Dictionary の各エントリは Index(16bit) と SubIndex(8bit) により識別します。
2	Network Management (NMT)	Network Management (NMT) はリモート局の通信状態を管理します。 通信状態は、マスター局とリモート局間のイニシャル処理や SLMP (NMTState Download) により遷移します。
3	Service Data Object (SDO)	Service data object (SDO) はマスター局がリモート局の Object Dictionary へアクセスするためのメッセージです。 SDO はトランジェント伝送(SLMP)を使用し、マスター局とリモート局間で非周期的に送受信されます。
4	Process Data Object (PDO)	Process data object(PDO) は、機器の制御や監視のために局間で周期的に転送されるアプリケーションオブジェクトの集合体です。 PDO はサイクリック伝送を使用し、マスター局とリモート局間で周期的に送受信されます。 PDO に格納されるデータは PDO マッピング設定により決まり、サイクリック送信データに格納される PDO を Transmit PDO (TPDO) と呼び、サイクリック受信データに格納される PDO のデータを Receive PDO(RPDO) と呼びます。

PDO のデータ内容は PDO マッピングで定義し、(GX works3 の場合) ”PDO マッピングの一括設定” または ”PDO マッピング設定” で設定します。

CANopen 通信を使用するためには、以下の事項が必要です。

No	必要事項	参照
1	CANopen 通信用の開発環境を使用する。	2.4
2	コンパイルスイッチ「TSN_CAN_ENABLE」を有効に設定する。	3.4.4
3	CANopen 通信関連のユーザプログラムを実装する	5.9
4	CANopen 通信用の CSP+ファイルを作成する。	2.6

その他、CANopen 通信に対応したマスタ局が必要です。

マスタ局に RJ71GN11-T2 を使用する場合は、ファームウェアバージョン” 12” 以降のユニットを使用してください。

設定方法と機能の詳細は、以下関連マニュアルを参照してください。

・ MELSEC iQ-R CC-Link IE TSN ユーザーズマニュアル(応用編)

### 4.6.1 CANopen 通信時の増設ユニット

CANopen 通信時において、R-IN32M4-CL3 適用製品の Object Dictionary 数を複数定義することで、例えばサーボアンプのように軸数などを拡張することができます。

Object Dictionary 数は、R-IN32M4-CL3 サンプルコードのマクロ定義「R\_IN\_CAN\_MAX\_ODTABLE\_NUM」(R\_IN32M4\_CL3CanConst.h)を設定することで変更できます。

以下に、通信を行う「基本ユニット(軸 1)」と通信を行わない「増設ユニット(軸 2～3)」で構成する多軸一体型サーボアンプを例に、増設ユニットのイメージを示します。

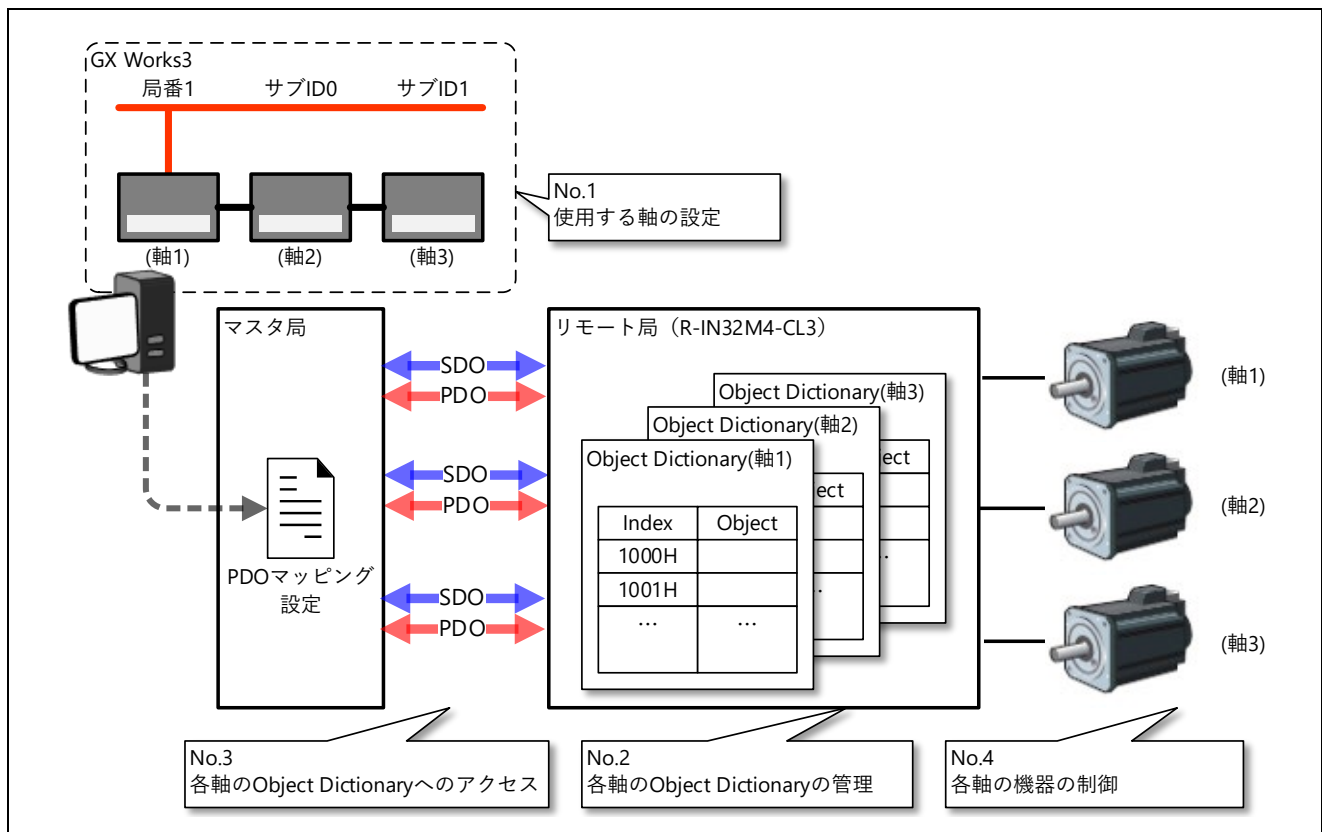


図 4.13 CANopen 通信における増設ユニットのイメージ

No	項目	概要
1	使用する軸の設定	エンジニアリングツールのネットワーク構成図にて、基本ユニットに増設ユニットを接合することで使用する軸を任意に設定することができます。 (例：最大 3 軸対応のユニットで 1～2 軸を使用する設定など)
2	各軸の Object Dictionary の管理	リモート局(R-IN32M4-CL3)の Object Dictionary は、軸ごとに定義します。 最大 3 軸構成で使用する場合は、基本ユニット(軸 1)と増設ユニット(軸 2～3)の計 3 つの Object Dictionary を定義し、「gerR_IN_CanInit」(6.4.16(1) CANopen 通信機能の初期化)の引数に設定してください。
3	各軸の Object Dictionary へのアクセス	マスター局からリモート局(R-IN32M4-CL3)の各軸の Object Dictionary へのアクセスには SDO (SLMP)を使用します。SLMP フレームの「要求先局プロセッササブ番号」に「軸番号(Object Dictionary 番号)」が格納されます。
4	各軸の機器の制御	各軸の Object Dictionary のデータを使用して、ユーザアプリケーションから各軸の制御を行います。

CANopen 通信の増設ユニットを使用するためには、以下の事項が必要です。

No	必要事項	参照
1	コンパイルスイッチ「TSN_CAN_MULTIAxis_ENABLE」を有効に設定する。	3.4.4
2	増設ユニットの装着数に応じて各種設定を変更する。	表 5.31
3	基本ユニット用、増設ユニット用の CSP+ファイルを作成する。	2.6

#### 4.7 安全通信時の安全 PDU 送受信

安全 PDU 送受信は、安全通信時においてマスタ局(安全局)からサイクリック伝送で受信した安全 PDU をユーザプログラムへ通知、またはユーザプログラムで作成した安全 PDU をサイクリック伝送経由でマスタ局(安全局)へ送信する機能です。

本機能により、サイクリック伝送におけるサブペイロードの構成を意識せずに、安全 PDU を取得または設定することができます。なお本機能は、安全 PDU と RX/Ry/RWr/RWw の混在に対応します。

安全通信のイメージを示します。

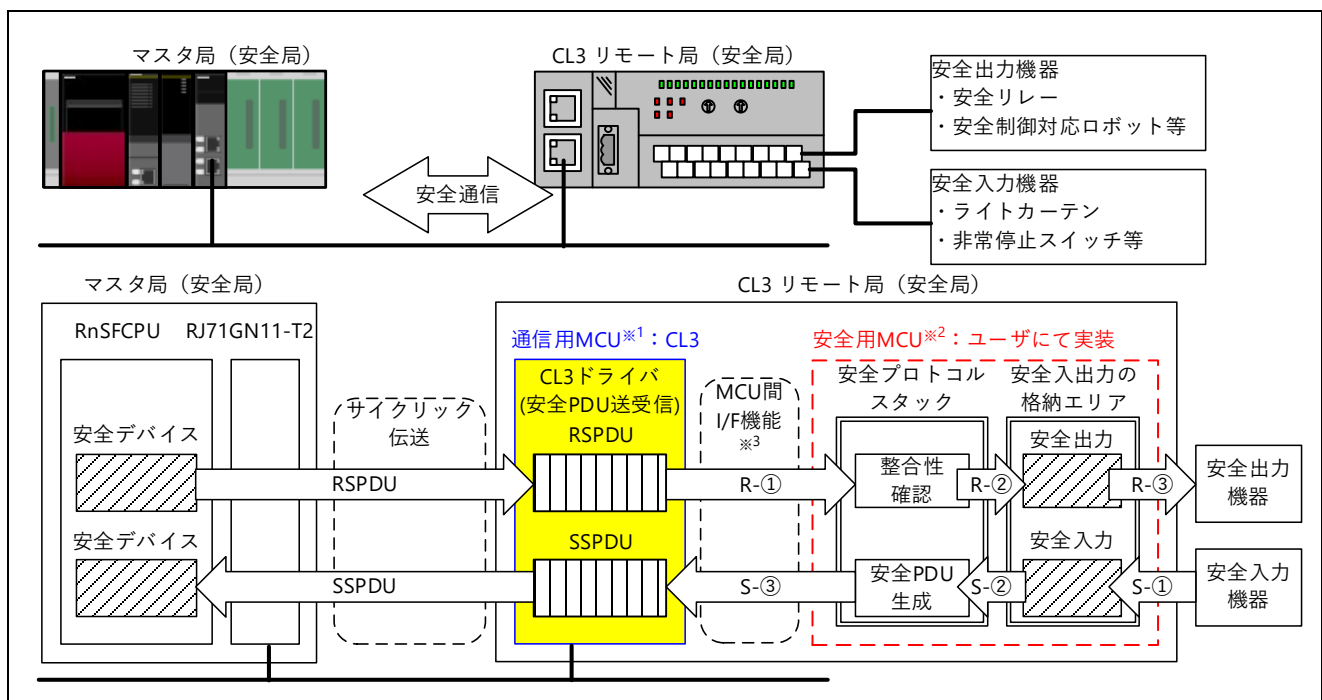


図 4.14 安全通信イメージ

区分	実装箇所	ユーザが実装する処理の概要
受信時	R-①	R-IN32M4-CL3 ドライバから RSPDU を読み出し、安全プロトコルスタックにデータを渡します。
	R-②	安全プロトコルスタックが抽出した安全出力データを取得し、安全出力エリアに格納します。
	R-③	安全出力エリアの出力値をもとに安全出力機器を制御します。
送信時	S-①	安全入力機器からの入力値を安全入力エリアに設定します。
	S-②	安全入力データを安全プロトコルスタックに渡し、安全 PDU を生成します。
	S-③	生成した安全 PDU を R-IN32M4-CL3 ドライバへ設定します。

※1：マスタ局(安全局)との通信制御(安全用 MCU との安全通信の仲介を含む)に使用する MCU です。

※2：安全通信(安全 PDU の生成/整合性確認)、安全入出力の制御など安全にかかわる制御に使用する MCU です。

※3：通信用 MCU と安全用 MCU 間で安全 PDU を交信します。詳細は「5.10 ユーザプログラム詳細(MCU 間 I/F 関連)」を参照してください。

安全通信を使用するためには、以下の事項が必要です。

No	必要事項	参照
1	安全用 MCU (外部マイコン)を選定し、R-IN32M4-CL3 に接続する。	-
2	コンパイルスイッチ「SAFETY_PDU_ENABLE」および「MCUIF_ENABLE」を有効に設定する。	3.4.4
3	MCU 間 I/F 関連のタスクを実装する。	5.2
4	MCU 間 I/F 関連のユーザプログラムを実装する。	5.10
5	安全通信用の CSP+ファイルを作成する。	2.6

その他、安全通信に対応したマスタ局が必要です。

マスタ局に RJ71GN11-T2 を使用する場合は、下記のユニットを使用してください。

- ・ファームウェアバージョンが” 20” 以降のシーケンサ安全 CPU (RnSFCPU)、および安全機能ユニット(R6SFM)
- ・ファームウェアバージョンが” 10” 以降の RJ71GN11-T2

設定方法と機能の詳細は、以下関連マニュアルを参照してください。

- ・MELSEC iQ-R CC-Link IE TSN ユーザーズマニュアル(応用編)

#### 4.8 SLMP 経由の通信速度・CC-Link IE TSN Class 設定

R-IN32M4-CL3 適用製品の通信速度/CC-Link IE TSN Class を設定ツールで設定する機能です。

設定ツールが SLMP を使用して通信速度/CC-Link IE TSN Class 設定データを R-IN32M4-CL3 適用製品に送信します。R-IN32M4-CL3 適用製品は受信した設定データを不揮発性メモリに書き込み、次回電源 OFF→ON 時に設定を反映します。

なお、設定ツール(SLMP のクライアント)には、CC-Link 協会の CC-Link IE TSN 設定ツール(CC-Link IE TSN Configurator)を使用します。あらかじめ CC-Link 協会のホームページからダウンロードしてください。



図 4.15 通信速度・CC-Link IE TSN Class 設定のイメージ

SLMP 経由で通信速度および CC-Link IE TSN Class を設定するためには、以下の事項が必要です。

No	必要事項	参照
1	受信した通信速度・CC-Link IE TSN Class を保存するための不揮発性メモリを用意する。	-
2	通信速度設定および CC-Link IE TSN Class 設定に関連する SLMP を受信できるように、受信処理を実装する。	5.8

本機能は、システムの立上げ前(R-IN32M4-CL3 適用製品をシステムに組み込む前)に実施することを推奨します。

##### <注意事項>

(R-IN32M4-CL3 適用製品が CC-Link IE TSN Class A で動作している場合)

本機能を実行する際、CC-Link IE TSN 設定ツールは設定(通信速度・CC-Link IE TSN Class)対象を検出するために、応答を伴うブロードキャストの SLMP コマンド(接続機器の検出(03E0H))を送受信します。接続台数(使用環境やフレームサイズにより変動)によっては、R-IN32M4-CL3 適用製品が送信した応答フレームが中継されず消失し、CC-Link IE TSN 設定ツールが R-IN32M4-CL3 適用製品を検出できない場合があります。

これを回避するために、本機能はシステム全体を解列させた状態で実施してください。

## 4.9 デバイス局の IP アドレス設定

マスタ局に接続されているデバイス局の IP アドレスを GX Works3 を使用して設定する機能です。

GX Works3 で入力した IP アドレスは、マスタ局が SLMP (0E31H : IP アドレス設定) でリモート局(R-IN32M4-CL3)に送信します。

本機能により、IP アドレス設定用のハードウェアスイッチが無い機器に対して、IP アドレスをソフトウェアで設定することが可能になります。

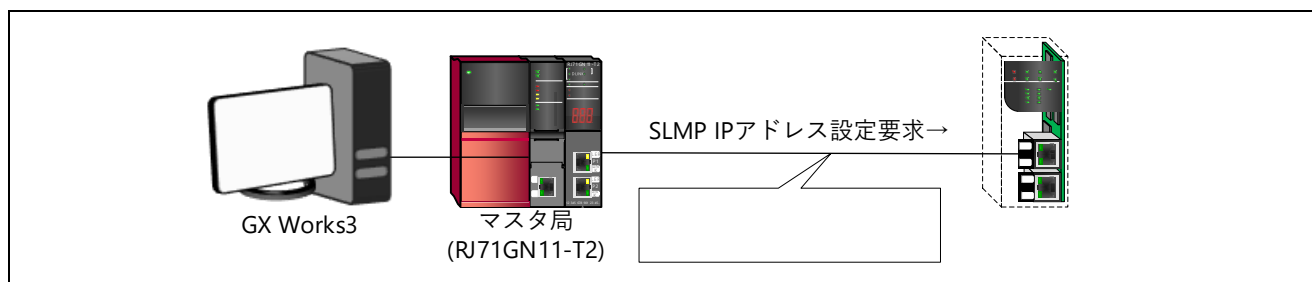


図 4.16 デバイス局の IP アドレス設定イメージ

AcyclicData フレームを使用することで、IP アドレス重複が発生しているデバイス局に対して SLMP IP アドレス設定要求を発行できます。

デバイス局の IP アドレスを使用するためには、以下の事項が必要です。

No	必要事項	参照
1	受信した IP アドレスを保存するための不揮発性メモリを用意する。	-
2	デバイス局の IP アドレス設定を実行できるように、初期化処理でノード番号(00H)を設定する。	5.3.1
3	SLMP IP アドレス変更要求コマンドを受信できるように、受信処理を実装する。	5.6.7

その他、デバイス局の IP アドレス設定に対応したマスタ局が必要です。

マスタ局に RJ71GN11-T2 を使用する場合は、下記バージョン以降の RJ71GN11-T2 および GX Works3 を使用してください。

- ・ RJ71GN11-T2 : ファームウェアバージョン” 18” 以降
- ・ GX Works3 : バージョン “1.100E” 以降

設定方法と機能の詳細は、以下関連マニュアルを参照してください。

- ・ MELSEC iQ-R CC-Link IE TSN ユーザーズマニュアル(応用編)



#### 4.10 デバイス局のパラメータ設定・コマンド実行

マスタ局に接続されているデバイス局のパラメータ設定とコマンド実行を、GX Works3 を使用して実施する機能です。本機能によりデバイス局に対するプログラミングを軽減することができます。デバイス局のパラメータ設定・コマンド実行の概要を下表に示します。

表 4.13 デバイス局のパラメータ設定・コマンド実行の概要

No	区分	名称	主な用途	参照
1	パラメータ	デバイス局のパラメータ自動設定	アプリケーションの初期設定、設定変更等	本節(1)
2		デバイス局のパラメータ設定		本節(2)
3	コマンド	デバイス局のコマンド実行	システム・リセット、エラー履歴クリア等	本節(3)

##### (1) デバイス局パラメータ自動設定

本機能は、デバイス局のパラメータをマスタ局に保存しておき、デバイス局の接続・復列時にパラメータを自動設定します。マスタ局とデバイス局間のパラメータ送受信には SLMP を使用します。

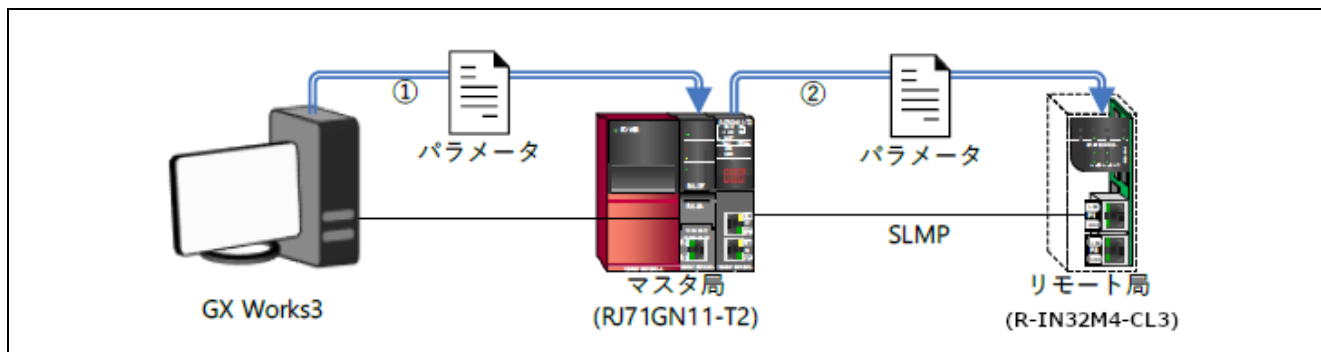


図 4.17 デバイス局パラメータ自動設定のイメージ

- ① エンジニアリングツールで設定した自局のパラメータを、マスタ局のシーケンサ CPU のメモリまたは SD メモリカードに書き込んで保存しておきます。
- ② 自局が電源 ON などにより接続・復列したとき、保存済みパラメータをマスタ局から自動設定します。自局は、パラメータ自動設定後にマスタ局とのデータリンクを開始します。

デバイス局のパラメータ自動設定を使用するためには、以下の事項が必要です。

No	必要事項	参照
1	サンプル CSP+ファイルを参考に自局の CSP+ファイルを作成する。	2.6
2	パラメータ自動設定に関する初期化処理を実装する。	5.3.5
3	パラメータ自動設定関連の SLMP を受信できるように、受信処理を実装する。	5.7

その他、デバイス局パラメータ自動設定に対応したマスタ局が必要です。

マスタ局に RJ71GN11-T2 を使用する場合、設定方法と機能の詳細は、以下の関連マニュアルを参照してください。

・ MELSEC iQ-R CC-Link IE TSN ユーザーズマニュアル(応用編)

## (2) デバイス局のパラメータ設定

本機能は、デバイス局のパラメータを任意のタイミングで設定します。デバイス局から読み出したパラメータは CSV ファイルに保存でき、CSV ファイルからパラメータの設定値を読み出してデバイス局へ書き込むことができます。マスタ局とデバイス局間のパラメータ送受信には SLMP を使用します。

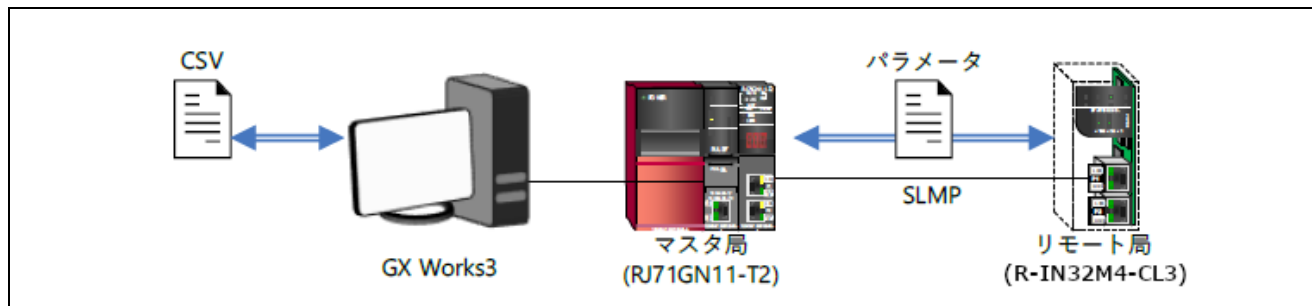


図 4.18 デバイス局のパラメータ設定のイメージ

デバイス局のパラメータ設定を使用するためには、以下の事項が必要です。

No	必要事項	参照
1	サンプル CSP+ファイルを参考に自局の CSP+ファイルを作成する。	2.6
2	作成した CSP+ファイルに記述している SLMP を受信できるように、受信処理を実装する。 <sup>*1</sup>	5.3.1

※1：サンプル CSP+ファイルでは、SLMP の例としてデュアルポートメモリー一括読出し(0613H)、一括書込み(1613H)コマンドを記述しています。SLMP コマンドの受信処理は 5.6.1 項および 5.6.2 項を参照してください。

その他、デバイス局のパラメータ設定に対応したマスタ局が必要です。

マスタ局に RJ71GN11-T2 を使用する場合、設定方法と機能の詳細は、以下関連マニュアルを参照してください。

- ・ MELSEC iQ-R CC-Link IE TSN ユーザズマニュアル(応用編)

(3) デバイス局のコマンド実行

本機能は、デバイス局に対するコマンド（リセット、エラー履歴クリア等の遠隔操作）を GX Works3 を使用して実行します。マスタ局とデバイス局間のコマンド送受信には SLMP を使用します。

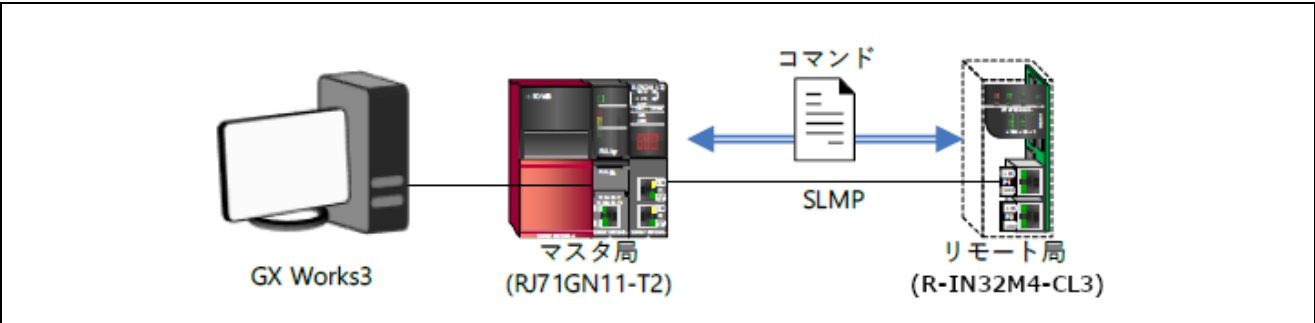


図 4.19 デバイス局のコマンド実行のイメージ

デバイス局のコマンド実行を使用するためには、以下の事項が必要です。

No	必要事項	参照
1	サンプル CSP+ファイルを参考に自局の CSP+ファイルを作成する。	2.6
2	作成した CSP+ファイルに記述している SLMP を受信できるように、受信処理を実装する。	*1

※1：サンプル CSP+ファイルでは、SLMP の例としてエラー履歴クリア(1619H)コマンドを記述しています。SLMP コマンドの受信処理は 5.6.8 項を参照してください。

その他、デバイス局のコマンド実行に対応したマスタ局が必要です。

マスタ局に RJ71GN11-T2 を使用する場合、設定方法と機能の詳細は、以下関連マニュアルを参照してください。

・ MELSEC iQ-R CC-Link IE TSN ユーザーズマニュアル(応用編)

#### 4.11 SNMP コミュニティ名設定

マスタ局に接続されているデバイス局に対して、SNMP のコミュニティ名を設定する機能です。

GX Works3 で入力した SNMP のコミュニティ名は、マスタ局が SLMP(3084H:SetCommunityName)でリモート局(R-IN32M4-CL3)に送信します。SLMP を受信したリモート局(R-IN32M4-CL3)は、コミュニティ名を設定します。

本機能により、デフォルトの SNMP コミュニティ名を変更できるようになります。

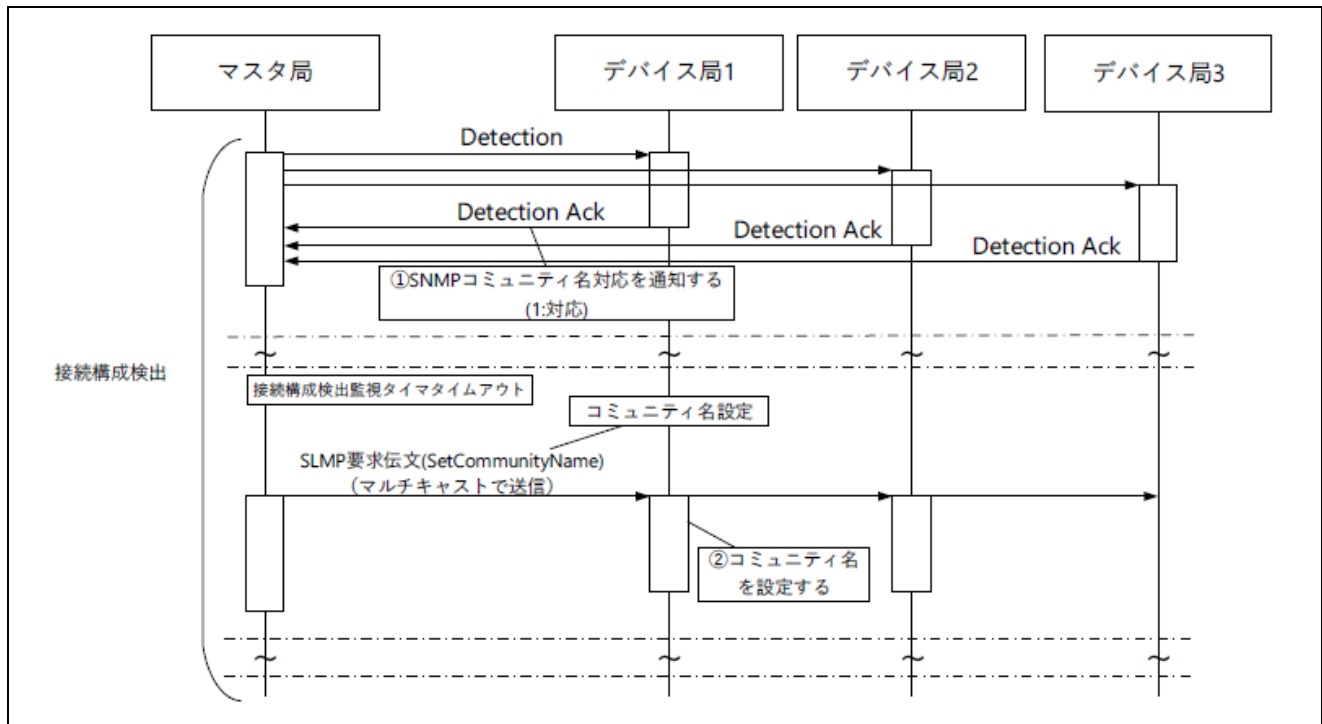


図 4.20 SNMP コミュニティ名設定イメージ

SNMP コミュニティ名設定を使用するためには、上記の①および②を行うために以下の対応が必要です。

No	必要事項	参照
1	コンパイルスイッチ「SNMP_COMMUNITY_NAME_SETTING_ENABLE」を有効に設定してください。	-
2	<p>① SNMP コミュニティ名対応を通知する</p> <p>マスタ局へ DetectionAck「Function3」の「Bit0：SNMP コミュニティ名設定」を「1:対応」で通知するため、R-IN32M4-CL3 ユニット情報「対応機能」の「Bit8：SNMP コミュニティ名設定対応」に「1:対応」を設定してください。</p> <p>※以下の機能を実行する場合、SNMP 通信が発生します。その際に、該当のデバイス局に対して SLMP コマンドにて SNMP コミュニティ名設定対応・未対応の確認が行われます。</p> <p>そのため、SLMP FuncSettingInfoget 要求コマンドにて SNMP コミュニティ名設定機能対応を通知できるよう、受信処理を実装してください。</p> <ul style="list-style-type: none"> <li>・エンジニアリングツールによる「デバイス局のパラメータ」操作</li> <li>・エンジニアリングツールによる「デバイス局のコマンド実行」操作</li> <li>・エンジニアリングツールによる「接続/切断した機器の検出」操作</li> </ul>	-
3	<p>② コミュニティ名を設定する</p> <p>マスタ局から受信した SNMP コミュニティ名を自局に設定するため、SLMP SetCommunityName 要求コマンドを受信できるように、受信処理を実装する。</p>	5.6.12

その他、SNMP コミュニティ名設定に対応したマスタ局が必要です。

マスタ局に RJ71GN11-T2 を使用する場合は、下記バージョン以降の RJ71GN11-T2 および GX Works3 を使用してください。

- ・ RJ71GN11-T2 : ファームウェアバージョン “28” 以降
- ・ GX Works3 : バージョン “1.115V” 以降

設定方法と機能の詳細は、以下関連マニュアルを参照してください。

- ・ MELSEC iQ-R CC-Link IE TSN ユーザズマニュアル(応用編)

#### <注意事項>

(サンプルコード Ver.1.09K 以降の場合)

デフォルトでは SNMP コミュニティ名 “public” で動作します。

(サンプルコード Ver.1.08J 以前の場合)

SNMP コミュニティ名 “public” で動作します。

また、SNMP コミュニティ名設定に未対応のため、SNMP コミュニティ名は変更できません。

#### <制約事項>

本機能は、増設ユニットまたはスライス I/O の接続に対応した R-IN32M4-CL3 適用製品では使用できません。

コンパイルスイッチ「SNMP\_COMMUNITY\_NAME\_SETTING\_ENABLE」を無効に設定してください。

## 5. ユーザプログラムの作成

本章は、サンプルコードのうち、ユーザプログラムの処理概要について示します。

ユーザプログラムは、リモート局の通信処理のロジックを確認するためのサンプル処理です。

### 5.1 ユーザプログラム一覧

ユーザプログラムの一覧、および実装要否を示します。

表 5.1 インイシャル関連のユーザプログラム一覧

No.	関数名	概要	実装要否	参照
1	iUserInitialization	初期化处理	必須	5.3.1
2	iUserStart	通信開始処理	必須	5.3.2
3	UserSmpReceiveInitial	SLMP 受信初期化处理	必須	5.3.3
4	UserSmpMakeRequestInitial	SLMP 要求フレーム作成初期化处理	任意	5.3.4
5	UserInitialParameterOperation	パラメータオペレーション初期化处理	任意	5.3.5

表 5.2 サイクリック伝送関連のユーザプログラム一覧

No.	関数名	概要	実装要否	参照
1	UserReceiveCyclic	サイクリック受信処理	必須	5.4.1
2	UserSendNodeStatus	自局状態送信処理	必須	5.4.2
3	UserSendCyclic	サイクリック送信処理	必須	5.4.3
4	UserUpdateStatus	通信状態更新処理	必須	5.4.4
5	UserGetCyclicStatus	サイクリック伝送状態更新処理	任意	5.4.5

表 5.3 状態管理およびトランジェント伝送関連のユーザプログラム一覧

No.	関数名	概要	実装要否	参照
1	UserForceStop	自局エラー処理	任意	5.5.1
2	iUserExecuteMain	イベント処理	必須	5.5.2
3	UserUpdateLed	LED 更新処理	必須	5.5.3
4	UserGetMIB	MIB(統計)情報取得処理	任意	5.5.4
5	UserSmpReceive	SLMP 受信処理	必須	5.5.5
6	UserSendSmp	SLMP 送信処理	必須	5.5.6
7	UserSmpMakeRequest	SLMP 要求フレーム作成処理	任意	5.5.7
8	UserSmpReceive3EFrame	SLMP ST(3E)応答フレーム受信処理	任意	5.5.8
9	UserConfirmFatalError	R-IN32M4-CL3 ドライバ内検出 Fatal エラー確認処理	任意	5.5.9

表 5.4 SLMP コマンド実行関連のユーザプログラム一覧

No.	関数名	概要	区分	実装 要否	参照
1	erUserSmpReceiveMemoryReadRequest	SLMP メモリ読み出し要求コマンド受信処理	サーバ	任意	5.6.1
2	erUserSmpReceiveMemoryWriteRequest	SLMP メモリ書き込み要求コマンド受信処理	サーバ	任意	5.6.2
3	iUserSmpMakeMemoryReadRequest	SLMP メモリ読み出し要求コマンド作成処理	クライアント	任意	5.6.3
4	UserSmpReceiveMemoryReadResponse	SLMP メモリ読み出し応答コマンド受信処理	クライアント	任意	5.6.4
5	erUserSmpRemoteReset	SLMP リモートリセット要求コマンド受信処理	サーバ	任意	5.6.5
6	erUserSmpNodeIndication	SLMP インディケータ表示要求コマンド受信処理	サーバ	任意	5.6.6
7	erUserSmpSetIpAddress	SLMP IP アドレス変更要求コマンド受信処理	サーバ	任意	5.6.7
8	erUserSmpClearErrorHistory	SLMP エラー履歴クリア要求コマンド受信処理	サーバ	任意	5.6.8
9	erUserSmpClockOffsetDataSend	SLMP ネットワーク時刻オフセット配信コマンド受信処理	サーバ	任意	5.6.9
10	erUserSmpNetworkClockDataSend	SLMP ネットワーク時刻配信コマンド受信処理	サーバ	任意	5.6.10
11	erUserSmpSetCommunityName	SLMP コミュニティ名の設定要求コマンド受信処理	サーバ	任意	5.6.12

表 5.5 デバイス局パラメータ自動設定関連のユーザプログラム一覧

No.	関数名	概要	区分	実装 要否	参照
1	erUserSmpGetCommunicationSet	SLMP 通信設定取得要求コマンド受信処理	サーバ	任意	5.7.1
2	erUserSmpCheckParameterDelivery	SLMP パラメータ配信要否チェック要求コマンド受信処理	サーバ	任意	5.7.2
3	UserSmpCheckParameterDeliveryMain	SLMP パラメータ配信要否チェック処理	サーバ	任意	5.7.3
4	erUserSmpStartRestore	SLMP リストア開始通知要求コマンド受信処理	サーバ	任意	5.7.4
5	erUserSmpEndRestore	SLMP リストア終了通知要求コマンド受信処理	サーバ	任意	5.7.5
6	erUserSmpSetParameter	SLMP パラメータデータ書き込み要求コマンド受信処理	サーバ	任意	5.7.6
7	UserSmpSetParameterMain	SLMP パラメータデータ書き込み要求処理	サーバ	任意	5.7.7

表 5.6 SLMP 経由の通信速度・CC-Link IE TSN Class 設定関連のユーザプログラム一覧

No.	関数名	概要	区分	実装 要否	参照
1	erUserSlmpSearchNodeExtention	SLMP 接続機器検出(拡張)要求コマンド受信処理	サーバ	任意	5.8.1
2	erUserSlmpGetFunctionSettingInfo	SLMP 機能設定(サポート情報取得)要求コマンド受信処理	サーバ	任意	5.8.2
3	erUserSlmpReadLinkSpeed	SLMP 機能設定読出し(通信速度)要求コマンド受信処理	サーバ	任意	5.8.3
4	erUserSlmpWriteLinkSpeed	SLMP 機能設定書込み(通信速度)要求コマンド受信処理	サーバ	任意	5.8.4
5	erUserSlmpReadCCIETSNClass	SLMP 機能設定読出し(CC-Link IE TSN Class)要求コマンド受信処理	サーバ	任意	5.8.5
6	erUserSlmpWriteCCIETSNClass	SLMP 機能設定書込み(CC-Link IE TSN Class)要求コマンド受信処理	サーバ	任意	5.8.6

表 5.7 ネットワーク同期通信関連のユーザプログラム一覧

No.	関数名	概要	実装 要否	参照
1	UserSyncTimingRoutine	同期タイミング処理	※1	5.2.5
2	UserSyncComLinkErrorRoutine	同期解列時処理	※1	5.2.6
3	erUserSlmpSetWatchdogCounterInfo	SLMP ウォッチドッグカウンタ情報設定要求コマンド受信処理	※1	5.6.11

【注】 ※1 ネットワーク同期通信を行う場合、実装が必須です。

表 5.8 CANopen 通信関連のユーザプログラム一覧

No.	関数名	概要	実装 要否	参照
1	iUserCanInitial	CANopen 通信機能の初期化処理	※1	5.9.1
2	UserReceiveCyclic	サイクリック受信処理(RPDO 更新)	※1	5.9.2
3	UserSendCyclic	サイクリック送信処理(TPDO 更新)	※1	5.9.3
4	erUserSlmpCanReadObject	SLMP ReadObject 要求コマンド受信処理	※1	5.9.4
5	erUserSlmpCanWriteObject	SLMP WriteObject 要求コマンド受信処理	※1	5.9.5
6	erUserSlmpCanObjectSubIDReadBlock	SLMP ObjectSubIDReadBlock 要求コマンド受信処理	※1	5.9.6
7	erUserSlmpCanObjectSubIDWriteBlock	SLMP ObjectSubIDWriteBlock 要求コマンド受信処理	※1	5.9.7
8	erUserSlmpCanNmtStateUpload	SLMP NMTStateUpload 要求コマンド受信処理	※1	5.9.8
9	erUserSlmpCanNmtStateDownload	SLMP NMTStateDownload 要求コマンド受信処理	※1	5.9.9
10	usUserSlmpCanGetFinishCode	SLMP 終了コード取得処理	※1	5.9.10
11	iUserCanSetParameter	CANopen パラメータ設定処理	※1	5.9.11
12	gulWriteFunc1010	Index1010 書込み処理	※1	5.9.12
13	gulWriteFunc1011	Index1011 書込み処理	※1	5.9.13

【注】 ※1 CANopen 通信を行う場合、実装が必須です。



表 5.9 MCU 間 I/F 関連のユーザプログラム一覧

No.	関数名	概要	実装要否	参照
1	UserMculfInitial	MCU 間 I/F 初期化处理	※1	5.10.1
2	UserMculfAckGpioSet	GPIO 通信受付処理	※1	5.10.2
3	UserMculfAckGpioClear	GPIO 通信受付クリア処理	※1	5.10.3
4	erUserMculfRequestGpioSet	GPIO 通信イベント種別/コード設定処理	※1	5.10.4
5	UserMculfSafetyPduTransfer	安全 PDU 転送処理(MCU 内部→外部)	※1	5.10.5
6	UserMculfSafetyPduReceived	安全 PDU 転送処理(MCU 内部←外部)	※1	5.10.6
7	UserMculfSafetyPduTransferResponse	安全 PDU 転送準備完了(MCU 内部→外部)受付受信処理	※1	5.10.7
8	UserMculfSafetyPduReceivedResponse	安全 PDU 転送準備完了(MCU 内部←外部)受付受信処理	※1	5.10.8
9	UserMculfSafetyPduReceivedComplete	安全 PDU 転送完了処理(MCU 内部←外部)	※1	5.10.9

【注】※1 安全 PDU を送受信する場合、実装が必須です。

表 5.10 H/W テスト関連のユーザプログラム一覧

No.	関数名	概要	実装要否	参照
1	UserIIEETest	H/W テスト処理(IEEE802.3ab コンプライアンステスト)	必須	5.11.1
2	UserLoopBackTest	H/W テスト処理(折り返し通信テスト)	任意	5.11.2
3	UserLoopBackTestInitial	折り返し通信テスト準備処理	任意	5.11.3
4	UserLoopBackTestLinkCheck	折り返し通信テストリンク状態判定処理	任意	5.11.4
5	UserLoopBackTestSend	折り返し通信テストデータ送信処理	任意	5.11.5
6	UserLoopBackTestReceived	折り返し通信テスト受信結果判定処理	任意	5.11.6
7	UserLoopBackTestExit	折り返し通信テスト完了処理	任意	5.11.7

## 5.2 ユーザプログラムのタスク

ユーザプログラムにおける RTOS のコンフィグレーションファイルに記述する生成情報を示します。各種設定値の詳細は、「R-IN32M4-CL3 プログラミング・マニュアル OS 編」を参照してください。

なお、各種設定値は特別な理由がない限り、変更しないでください。

ユーザにてタスクを新規生成する場合は、「6.3 R-IN32M4-CL3 ドライバのタスク」に記載する生成情報も確認のうえ、生成してください。

表 5.11 ユーザプログラムのタスク一覧

No.	タスク名称	タスク ID	T_CTSK 構造体メンバ					
			タスク属性	拡張情報	起動番地 (関数名で記載)	起動時 優先度	スタック サイズ	スタック領域 の先頭番地
1	イニシャルタスク	TSKID_INITIAL(1)	TA_HLNG   TA_ACT	0000H	vTask_Initial	1	400H	NULL
2	定周期処理タスク	TSKID_PERIODIC(3)	TA_HLNG	0000H	vTask_Periodic	3	400H	NULL
3	アイドルタスク	TSKID_IDLE(2)	TA_HLNG	0000H	vTask_Idle	15	1000H	NULL
4	エラー管理タスク	TSKID_ERR(5)	TA_HLNG	0000H	vTask_Err	10 (13) <sup>※1</sup>	1800H	NULL
5	同期タイミング処理 タスク	TSKID_SYNCPROC(9)	TA_HLNG	0000H	vTask_SyncProc	2	400H	NULL
6	同期解列時処理 タスク	TSKID_PERIO_DLERR(10)	TA_HLNG	0000H	vTask_Periodic_ DLinkErr	3	400H	NULL
7	スケジューラ タスク	TSKID_SCHEDULER(11)	TA_HLNG	0000H	vTask_Scheduler	12	500H	NULL
8	GPIO 通信送信 タスク	TSKID_MSGP_SEND(12)	TA_HLNG	0000H	vTask_MsgGpio_Se nd	3	600H	NULL
9	GPIO 通信応答受信 タスク	TSKID_MSGP_ANS(13)	TA_HLNG	0000H	vTask_MsgGpio_Ans	3	400H	NULL
10	GPIO 通信受信 タスク	TSKID_MSGP_RECV(14)	TA_HLNG	0000H	vTask_MsgGpio_Re cv	3	500H	NULL
11	RT DMA 転送完了 タスク	TSKID_RTDMA_DONE(15)	TA_HLNG	0000H	vTask_RtDma_Done	3	500H	NULL
12	サイクリックフレーム 送信処理タスク (CC-Link IE TSN Class A)	TSKID_SENDCYCFCRM _CLASS_A(16)	TA_HLNG	0000H	vTask_SendCyclic FrameClassA	4	400H	NULL

【注】 ※1 MCU 間 I/F 機能を有効(コンパイルスイッチ「MCUIF\_ENABLE」を有効)にした場合、時間計測するスケジューラタスクの動作を妨げないように優先度を 10 から 13 へ下げています。

CC-Link IE TSN Class およびネットワーク同期通信の設定有無によって、起動するタスクが異なります。それぞれの場合で起動するタスクの一覧を示します。

なお、ネットワーク同期通信は、CC-Link IE TSN Class B 動作時、かつマスタ局が同期通信設定で「同期する」を指定した場合に行われます。CC-Link IE TSN Class A 動作時では、ネットワーク同期通信を使用できません。

表 5.12 通信設定ごとの起動タスク一覧

No.	名称	ネットワーク同期通信なし		ネットワーク同期通信あり
		CCIETSN Class A	CCIETSN Class B	CCIETSN Class B
1	イニシャルタスク	○	○	○
2	定周期処理タスク	○	○	○
3	アイドルタスク	○	○	○
4	エラー管理タスク	○	○	○
5	同期タイミング処理タスク	×	×	○
6	同期解列時処理タスク	×	×	○
7	スケジューラタスク	○※1	○※1	×
8	GPIO 通信送信タスク	○※1※2	○※1※2	×
9	GPIO 通信応答受信タスク	○※1※2	○※1※2	×
10	GPIO 通信受信タスク	○※1※2	○※1※2	×
11	RT DMA 転送完了タスク	○※1※2	○※1※2	×
12	サイクリックフレーム送信処理タスク (CC-Link IE TSN Class A)	○	×	×

○：起動する，×：起動しない

【注】 ※1 MCU 間 I/F 機能を有効(コンパイルスイッチ「MCUIF\_ENABLE」を有効)にした場合に起動します。

※2 外部 MCU と GPIO 通信開始時に起動します。

表 5.13 セマフォの生成情報

No.	名称	ID	セマフォ属性※1	資源数の初期値	最大資源数
1	Fatal エラー管理セマフォ	SEMID_FATALERROR (128)	TA_TFIFO	1	1
2	エラー履歴管理セマフォ	SEMID_ERRHIS (127)	TA_TFIFO	1	1
3	PHY 管理セマフォ	SEMID_PHY (126)	TA_TFIFO	1	1
4	SLMP 受信情報管理セマフォ	SEMID_SLMP_RES_INFO (125)	TA_TFIFO	1	1

【注】 ※1 μITRON ではセマフォ属性を以下のとおり定義しています。(μITRON4.0 仕様 ver4.03)

TA\_TFIFO(00H) …タスクの待ち行列を FIFO 順に管理する

TA\_TPRI (01H)…タスクの待ち行列をタスクの優先度順に管理する

表 5.14 メールボックスの生成情報

No.	メールボックス名称	メールボックス ID	T_CMBX 構造体メンバ		
			メールボックス属性	メッセージの優先度最大値	優先度別のメッセージキューヘッダ領域の先頭番地
1	GPIO 通信送信※1	MBXID_MSGP(50)	TA_TPRI   TA_MFIFO	1	NULL

【注】 ※1 MCU 間 I/F 機能を有効(コンパイルスイッチ「MCUIF\_ENABLE」を有効)にした場合

表 5.15 Hardware ISR の設定

No.	Hardware ISR 名称	対象割り込み番号		例外番号	割り込み発生時に自動実行するサービスコール	自動実行するサービスコールの対象オブジェクト ID
1	定周期処理	0 (TAUJ2 チャンネル 0 割り込み)		16	HWISR_WUP_TSK	TSKID_PERIODIC
2	エラー管理	一般通信	62 (INTPZ15 入力/TAUJ2 チャンネル 9 割り込み)	78	HWISR_WUP_TSK	TSKID_ERR
		安全通信※2	2 (TAUJ2 チャンネル 2 割り込み)			
3	同期タイミング処理※1	114 (同期信号出力(アプリケーション用))		130	HWISR_WUP_TSK	TSKID_SYNCPROC
4	同期解列時処理※1	1 (TAUJ2 チャンネル 1 割り込み)		17	HWISR_WUP_TSK	TSKID_PERIO_DLERR
5	スケジューラタスク※2	1 (TAUJ2 チャンネル 1 割り込み)		17	HWISR_WUP_TSK	TSKID_SCHEDULER
6	RT DMA 転送完了タスク※2	26 (リアルタイム・ポート用 DMAC 転送完了割り込み)		42	HWISR_WUP_TSK	TSKID_RTDMA_DONE
7	GPIO 通信応答受信タスク※2	61 (INTPZ14 入力/TAUD チャンネル 8 割り込み)		77	HWISR_WUP_TSK	TSKID_MSGP_ANS
8	GPIO 通信受信タスク※2	62 (INTPZ15 入力/TAUD チャンネル 9 割り込み)		78	HWISR_WUP_TSK	TSKID_MSGP_RECV

【注】 ※1 ネットワーク同期通信時

※2 MCU 間 I/F 機能を有効(コンパイルスイッチ「MCUIF\_ENABLE」を有効)にした場合

表 5.16 タイマの使用一覧

No.	分類	チャンネル	用途		基準時間の 番号	動作 クロック	時間
1	32bit タイマ (TAUJ2)	0	定周期処理の定周期を刻むために使用		基準時間 1	CK0	0.01 $\mu$ s (PCLK/20)
2		1	一般通信	同期解列処理の定周期を刻むために使用			
			安全通信※1	スケジューラタスクの定周期を刻むために使用			
3		2	一般通信	未使用			
			安全通信※1	エラー管理(SNMP)の定周期を刻むために使用			
4	16bit タイマ (TAUD)	9	一般通信	エラー管理(SNMP)の定周期を刻むために使用	基準時間 4	CK3_PRE	0.16 $\mu$ s (PCLK/24)
			安全通信※1	未使用			

【注】 ※1 MCU 間 I/F 機能を有効(コンパイルスイッチ「MCUIF\_ENABLE」を有効)にした場合

タイマの詳細は「R-IN32M4-CL3 ユーザーズ・マニュアル ハードウェア編」を参照してください。

## 5.2.1 イニシャルタスク

### (1) タスクの処理概要

本タスクは、使用するタイマ(TAUJ2, TAUD)の初期化、設定、起動と、アイドルタスクの起動を行います。

本タスクは、起動時に一度だけ処理を実行します。

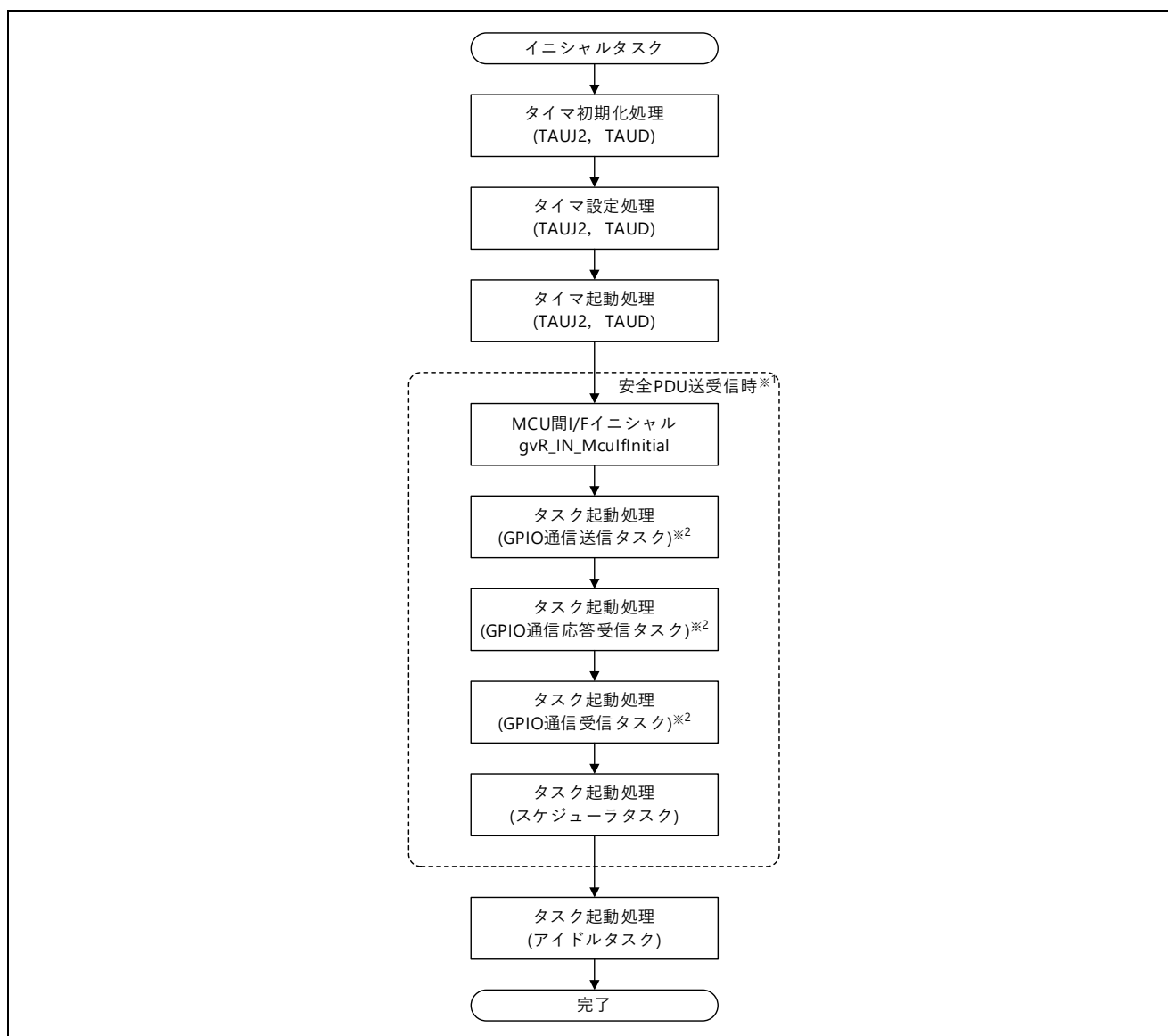


図 5.1 イニシャルタスク処理概要

【注】 ※1 MCU 間 I/F 機能を有効(コンパイルスイッチ「MCUIF\_ENABLE」を有効)にした場合に実行します。

※2 マスタ局が安全通信使用有無設定で「使用する」を指定した場合に起動します。

各タイマに設定する値は以下のとおりです。

表 5.17 タイマの使用一覧

使用するタイマ		クロック設定値	タイムアウト設定値
32bit タイマ (TAUJ2)	TM00	CK0 (0.01us = PCLK / (2 <sup>40</sup> ))	20000 - 1 (1 クロック 10ns で 200us 周期)
	TM01		
16bit タイマ (TAUD)	TM09	CK3 (0.16us = PCLK / (2 <sup>4</sup> ))	62500 - 1 (1 クロック 160ns で 10ms 周期)

## 5.2.2 アイドルタスク

### (1) タスクの処理概要

本タスクは、起動時に一度だけ、F/W と H/W を初期化します。初期化処理の実施後に、無限ループにより、状態管理・トランジェントメイン処理を繰り返し実行します。

本タスクは、タスクの優先度が最も低いタスクであり、時間制約の緩い処理を無限ループで処理します。他タスクが実施されていないときに処理を実施します。タスクの処理概要を示します。

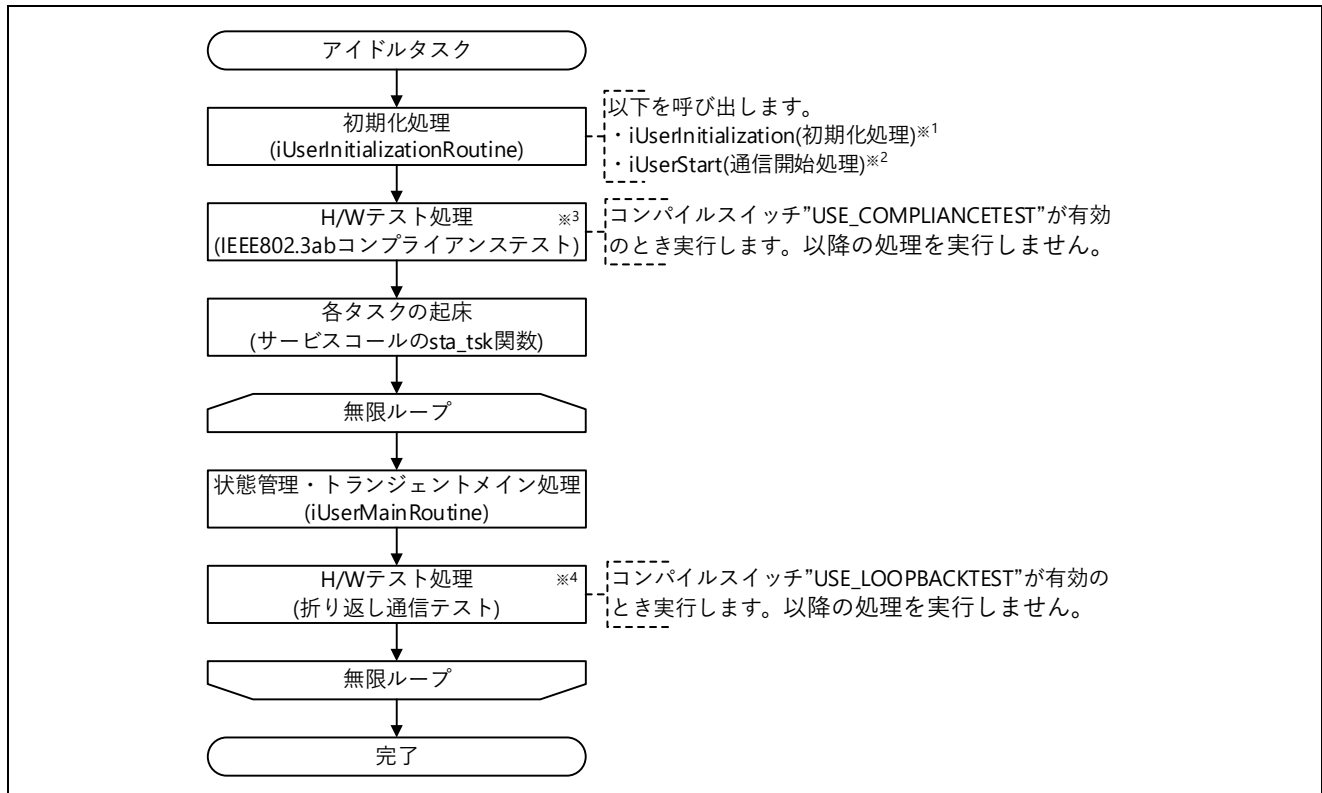


図 5.2 アイドルタスク処理概要

【注】※1：「iUserInitialization」(5.3.1 初期化処理)を参照してください。

※2：「iUserStart」(5.3.2 通信開始処理)を参照してください。

※3：「UserIEEETest」(5.11.1H/W テスト処理(IEEE802.3ab コンプライアンステスト))を参照してください。

※4：「UserLoopBackTest」(5.11.2H/W テスト処理(折り返し通信テスト))を参照してください。

## (2) タスクのメイン処理

アイドルタスクの状態管理・トランジェントメイン処理の概略フローを示します。

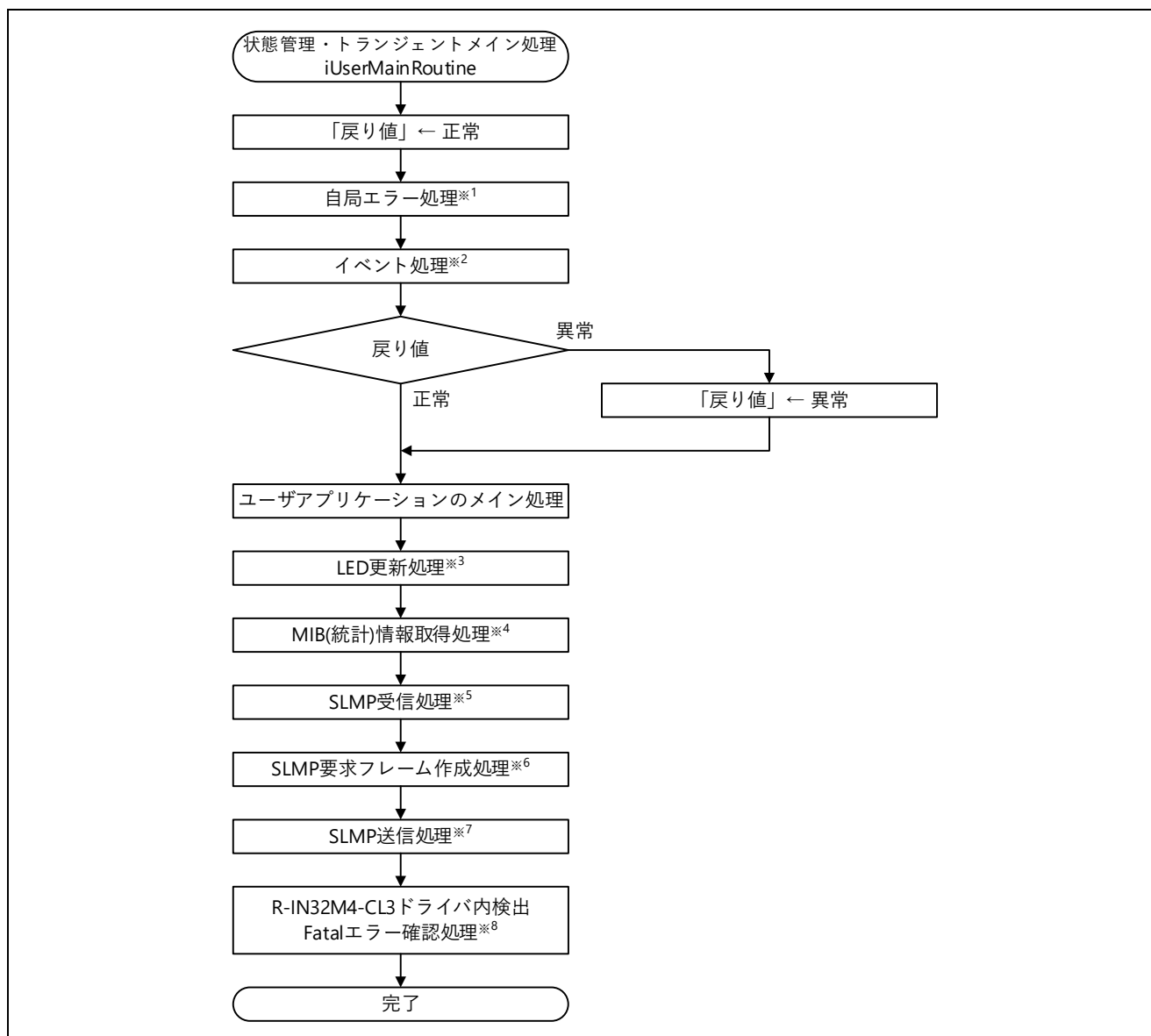


図 5.3 状態管理・トランジェントメイン処理フロー図

- 【注】※1：「UserForceStop」(5.5.1 自局エラー処理)を参照してください。  
 ※2：「iUserExecuteMain」(5.5.2 イベント処理)を参照してください。  
 ※3：「UserUpdateLed」(5.5.3 LED 更新処理)を参照してください。  
 ※4：「UserGetMIB」(5.5.4 MIB(統計)情報取得処理)を参照してください。  
 ※5：「UserSlmpReceive」(5.5.5 SLMP 受信処理)を参照してください。  
 ※6：「UserSlmpMakeRequest」(5.5.7 SLMP 要求フレーム作成処理)を参照してください。  
 ※7：「UserSendSlmp」(5.5.6 SLMP 送信処理)を参照してください。  
 ※8：「UserConfirmFatalError」(5.5.9 R-IN32M4-CL3 ドライバ内検出 Fatal エラー確認処理)を参照してください。

### 5.2.3 定周期処理タスク

本タスクは、サイクリックデータを定期的に更新します。無限ループにより、「タスクの起床待ち」と「サイクリックメイン処理」を繰り返し実行します。

#### (1) タスクの処理概要

本タスクは、定周期で処理を実施するタスクです。RTOS がタイマ割込みを検出すると、Hardware ISR がタスクを起床します。タスクの処理概要を示します。

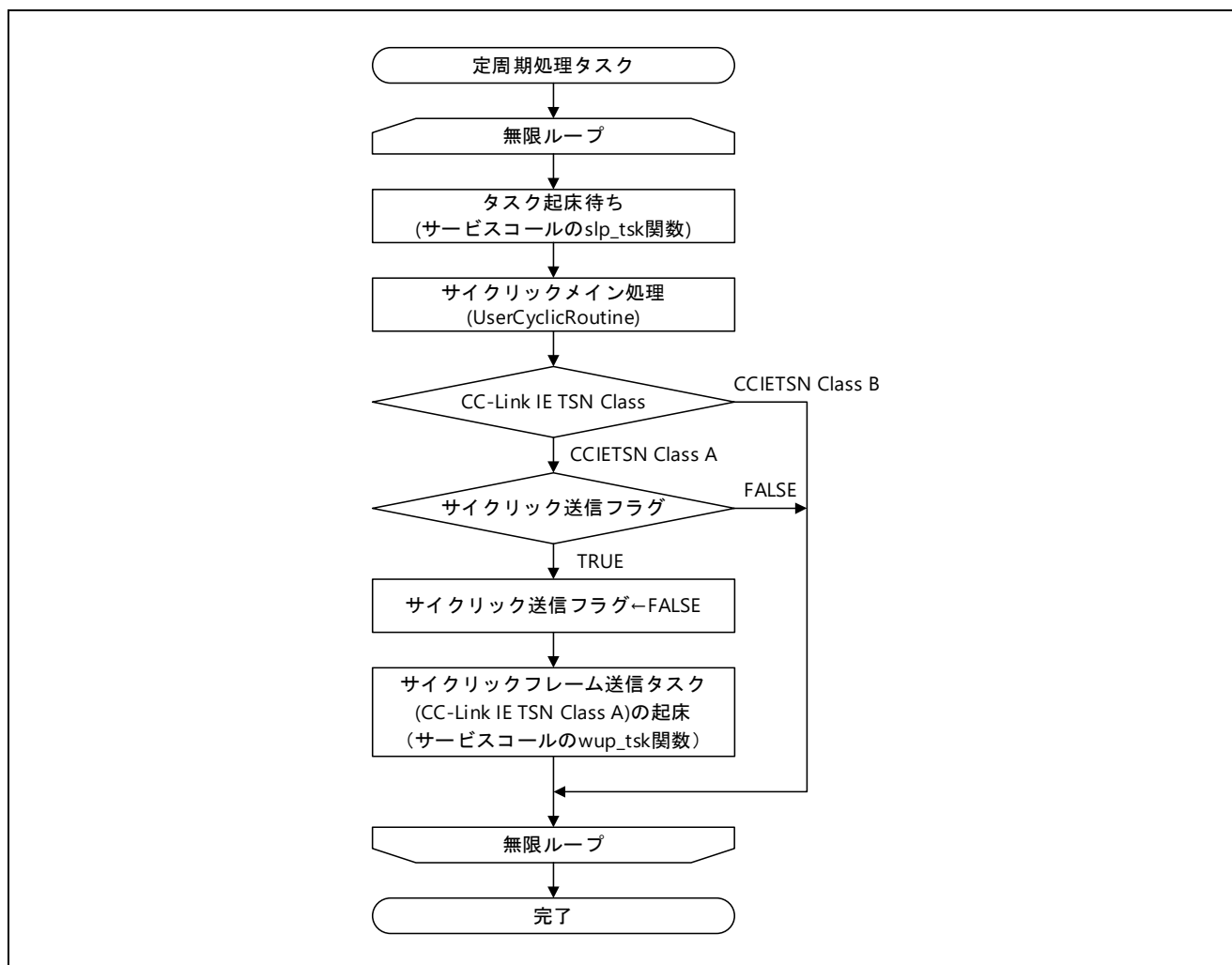


図 5.4 定周期処理タスク処理概要

#### <注意事項>

ネットワーク同期通信の有効/無効(同期する/同期しない)が確定した後に、リモート局のネットワーク同期通信機能の有効/無効を変更することはできません。ネットワーク同期通信設定を変更する場合は、起動済みタスクを停止させるため、対象機器の電源 OFF→ON またはシステムリセットが必要です。



## (2) タスクのメイン処理

サイクリックメイン処理の概略フローを示します。

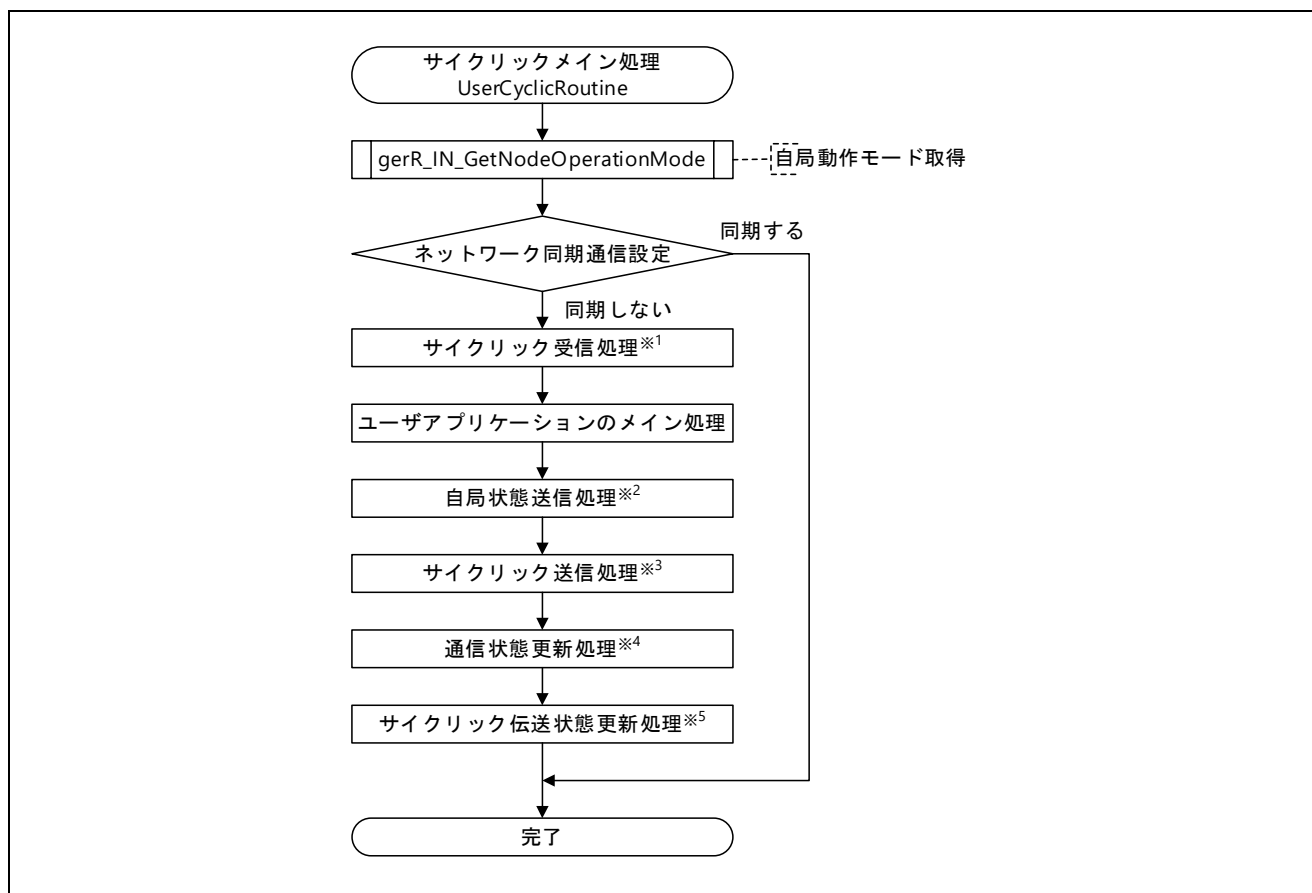


図 5.5 サイクリックメイン処理フロー図

- 【注】 ※1 「UserReceiveCyclic」 (5.4.1 サイクリック受信処理)を参照してください。  
 ※2 「UserSendNodeStatus」 (5.4.2 自局状態送信処理)を参照してください。  
 ※3 「UserSendCyclic」 (5.4.3 サイクリック送信処理)を参照してください。  
 ※4 「UserUpdateStatus」 (5.4.4 通信状態更新処理)を参照してください。  
 ※5 「UserGetCyclicStatus」 (5.4.5 サイクリック伝送状態更新処理)を参照してください。

## 5.2.4 エラー管理タスク

本タスクは CC-Link IE TSN 診断において、SNMP を使用して自局の診断情報をマスタ局へ送信します。無限ループにより、「タスクの起床待ち」と「SNMP 定周期処理」を繰り返し実行します。

本タスクは定周期で処理を実施するタスクです。RTOS がタイマ割込みを検出すると、Hardware ISR がタスクを起床します。タスクの処理概要を示します。

なお、「SNMP 定周期処理」は RIN32M4-CL3 ドライバが実行します。

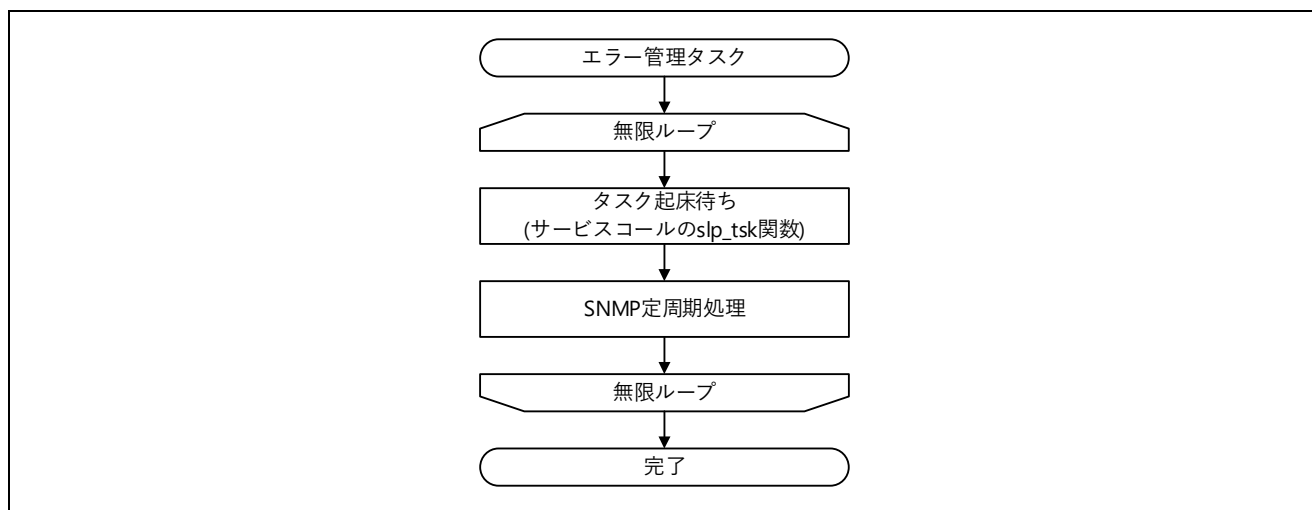


図 5.6 エラー管理タスク処理概要

## 5.2.5 同期タイミング処理タスク

### (1) タスクの処理概要

本タスクは、同期通信において他のデバイス局とタイミングを合わせたいユーザ処理を実行します。

無限ループにより「タスクの起床待ち」と「同期タイミング処理」を繰り返し実行します。

本タスクは、割込みが発生したときに処理を実施します。RTOS が INTSYNCO 割込み※1を検出すると、Hardware ISR がタスクを起床します。タスクの処理概要を示します。

※1：INTSYNCO 割込みは同期信号出力(アプリ用)の立下りで発生します。同期信号出力(アプリ用)の立上りなどで割込みを発生させたい場合は、割込みモード・レジスタ(INTM11)にてトリガ・モード(76H)を変更してください。INTM11 レジスタの詳細は、「R-IN32M4-CL3 ユーザズ・マニュアル ハードウェア編」を参照してください。

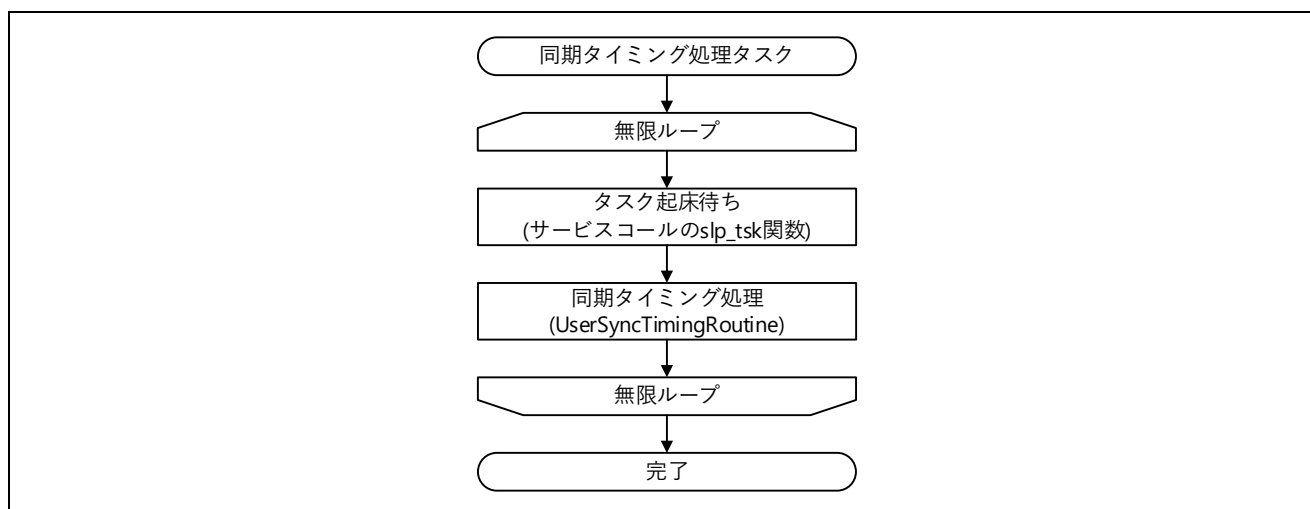


図 5.7 同期タイミング処理タスク処理概要

#### <注意事項>

本タスクは、マスタ局がネットワーク同期通信設定で「同期する」を指定した場合に起動します。

ネットワーク同期通信の有効/無効(同期する/同期しない)が確定した後に、リモート局のネットワーク同期通信機能の有効/無効を変更することはできません。ネットワーク同期通信設定を変更する場合は、起動済みタスクを停止させるため対象機器の電源 OFF→ON またはシステムリセットが必要です。

## (2) タスクのメイン処理

同期タイミング処理の概略フローを示します。

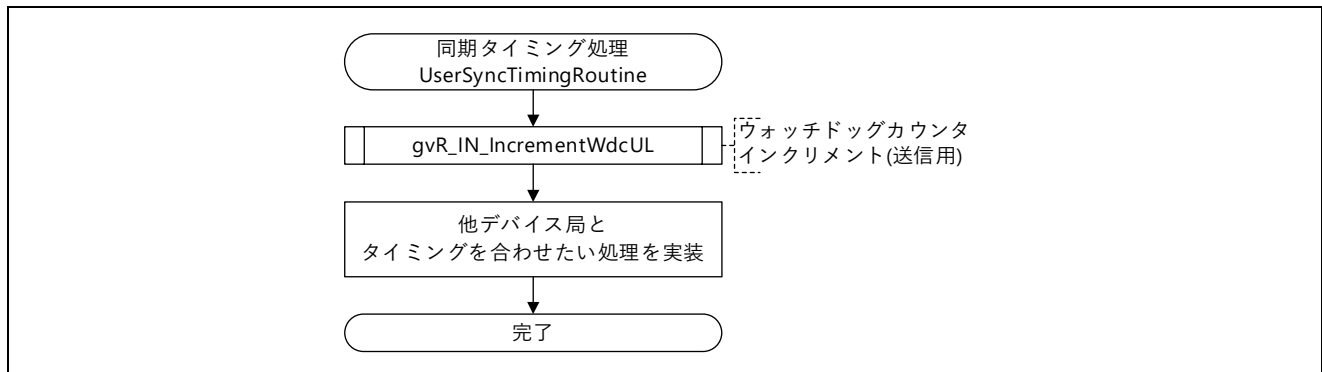


図 5.8 同期タイミング処理フロー図

## 5.2.6 同期解列時処理タスク

### (1) タスクの処理概要

本タスクは、同期通信において自局が解列したときにユーザが実行したい処理を実行します。

無限ループにより「タスクの起床待ち」と「同期解列時処理」を繰り返し実行します。

本タスクは、定周期で処理を実施するタスクです。RTOS がタイマ割込みを検出すると、Hardware ISR がタスクを起床します。タスクの処理概要を示します。

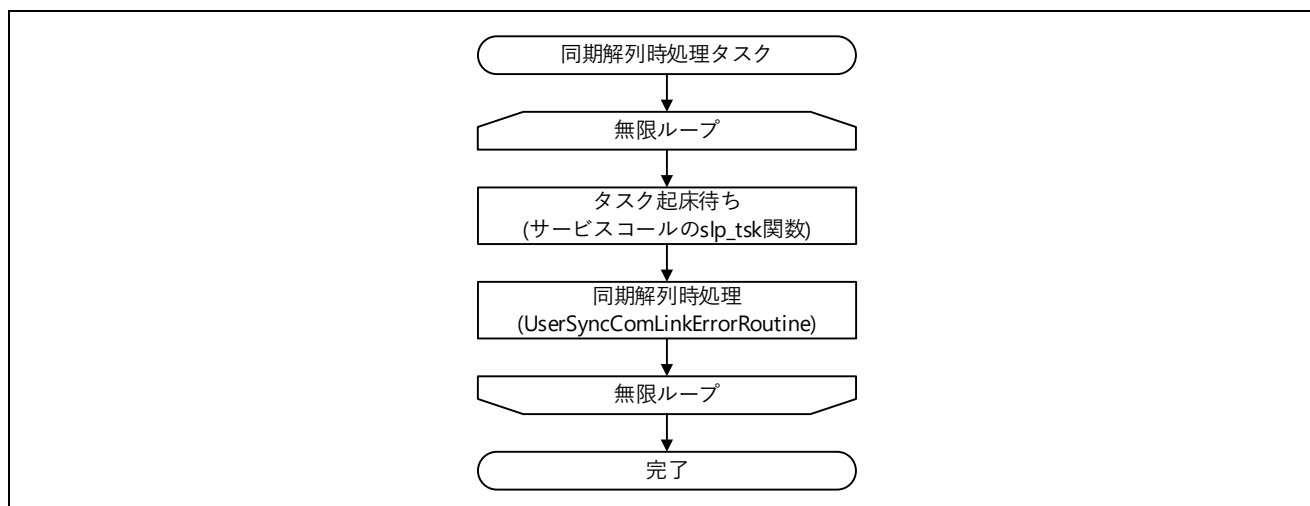


図 5.9 同期解列時処理タスク処理概要

#### <注意事項>

本タスクは、マスタ局がネットワーク同期通信設定で「同期する」を指定した場合に起動します。

ネットワーク同期通信の有効/無効(同期する/同期しない)が確定した後に、リモート局のネットワーク同期通信機能の有効/無効を変更することはできません。ネットワーク同期通信設定を変更する場合は、起動済みタスクを停止させるため、対象機器の電源 OFF→ON またはシステムリセットが必要です。

## (2) タスクのメイン処理

同期解列時処理の概略フローを示します。

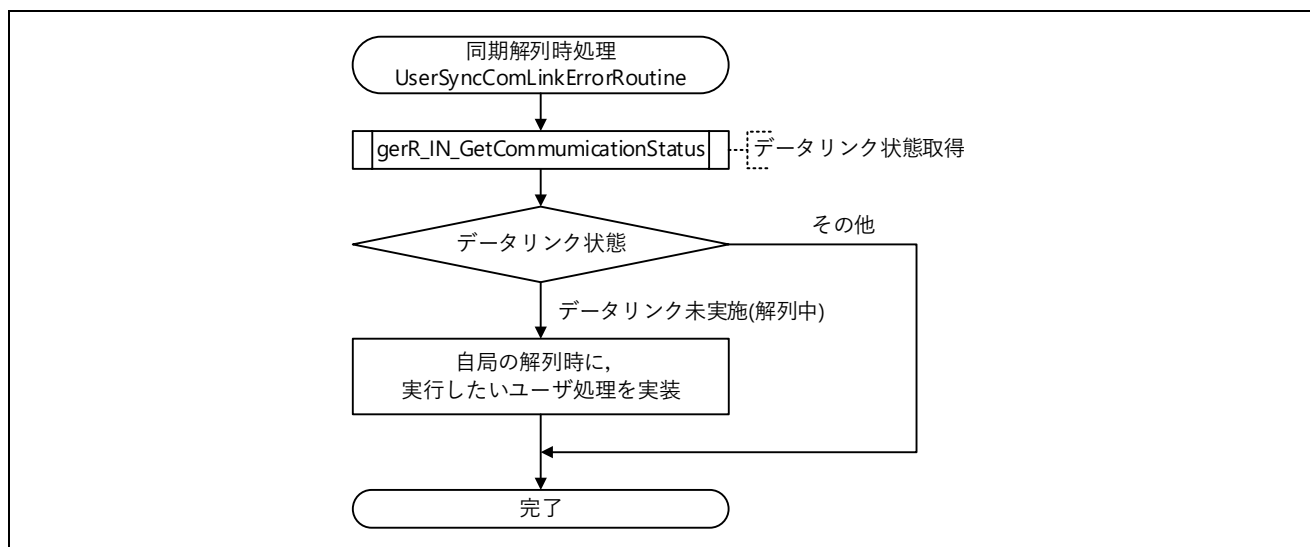


図 5.10 同期解列時処理フロー図

## 5.2.7 スケジューラタスク

### (1) タスクの処理概要

本タスクは、無限ループにより「タスクの起床待ち」と「スケジューラタスクメイン処理」を繰り返し実行します。

本タスクは、定周期で処理を実施するタスクです。RTOS がタイマ割込みを検出すると、Hardware ISR がタスクを起床します。タスクの処理概要を示します。

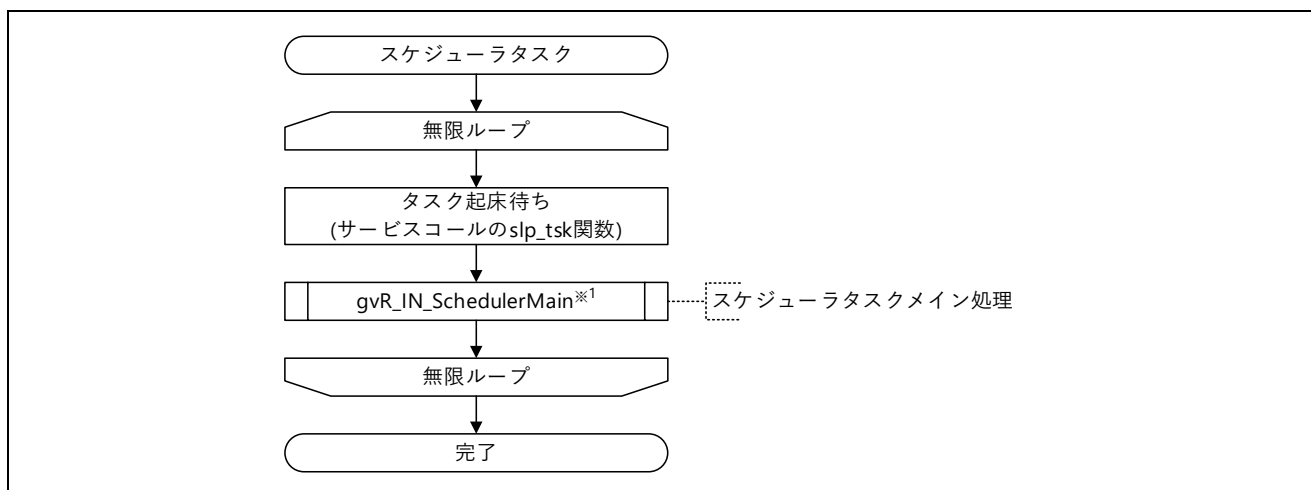


図 5.11 スケジューラタスク処理概要

【注】 ※1 「6.4.17(1) gvR\_IN\_SchedulerMain」を参照してください。

#### <注意事項>

本タスクは、MCU 間 I/F 機能を有効(コンパイルスイッチ「MCUIF\_ENABLE」を有効)にした場合に起動します。

## 5.2.8 GPIO 通信送信タスク

### (1) タスクの処理概要

本タスクは、無限ループにより「GPIO 通信送信タスクメイン処理」を繰り返し実行します。

本タスクは、メッセージを受信したときに処理を実施し、メッセージがないときはメッセージ待ち状態に移行するタスクです。タスクの処理概要を示します。

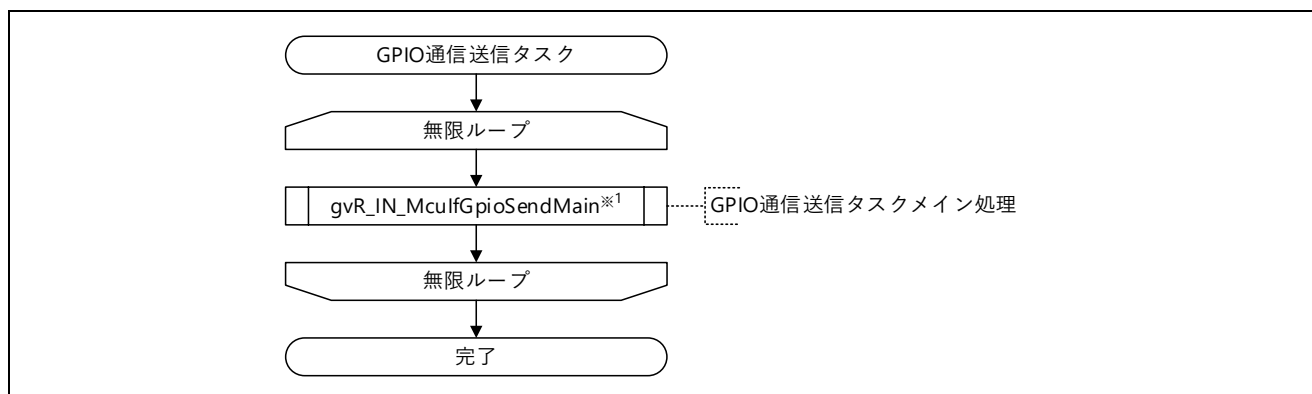


図 5.12 GPIO 通信送信タスク処理概要

【注】 ※1 「6.4.17(2) gvR\_IN\_MculfGpioSendMain」を参照してください。

GPIO 通信送信タスクが受け付けるメッセージを以下に示します。

表 5.18 メッセージ一覧

No.	メッセージ名称	メッセージ ID	メッセージ送信元
1	イベント通知送信要求 (MCU 内部→外部)	MSGP_MID_SEND_EVENTNOTICE_MPUA (1)	GPIO 通信受信タスク
2	イベント応答受信タイムアウト (MCU 内部→外部)	MSGP_MID_TIMEOUT_RESPNOTICE_MPUA (2)	スケジューラタスク
3	次回イベント通知確認 (MCU 内部→外部)	MSGP_MID_CHECK_NEXTEVENT_MPUA (3)	GPIO 通信受信タスク

#### <注意事項>

本タスクは、MCU 間 I/F 機能を有効(コンパイルスイッチ「MCUIF\_ENABLE」を有効)にした上で、マスタ局が安全通信使用有無設定で「使用する」を指定した場合に起動します。



## 5.2.9 GPIO 通信応答受信タスク

### (1) タスクの処理概要

本タスクは、無限ループにより「割り込み発生待ち」と「GPIO 通信応答受信タスクメイン処理」を繰り返し実行します。

本タスクは、割り込みが発生したときに処理を実施します。RTOS が INTPZ14 割り込みを検出すると、Hardware ISR がタスクを起床します。タスクの処理概要を示します。

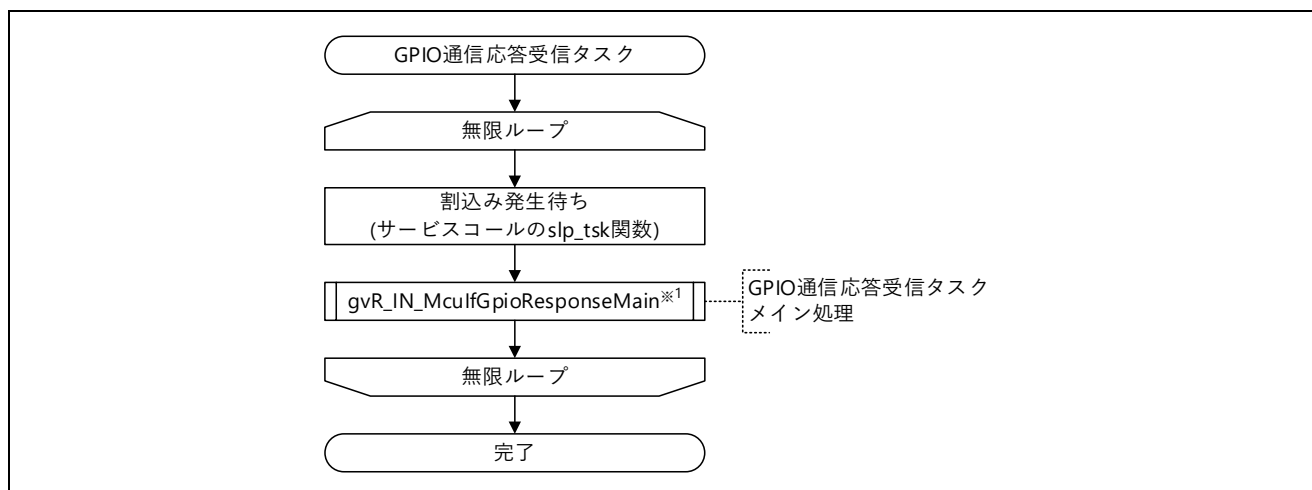


図 5.13 GPIO 通信応答受信タスク処理概要

【注】 ※1 「6.4.17(3) gvR\_IN\_McufGpioResponseMain」を参照してください。

#### <注意事項>

本タスクは、MCU 間 I/F 機能を有効(コンパイルスイッチ「MCUIF\_ENABLE」を有効)にした上で、マスタ局が安全通信使用有無設定で「使用する」を指定した場合に起動します。

## 5.2.10 GPIO 通信受信タスク

### (1) タスクの処理概要

本タスクは、無限ループにより「割り込み発生待ち」と「GPIO 通信受信タスクメイン処理」を繰り返し実行します。

本タスクは、割り込みが発生したときに処理を実施します。RTOS が INTPZ15 割り込みを検出すると、Hardware ISR がタスクを起床します。タスクの処理概要を示します。

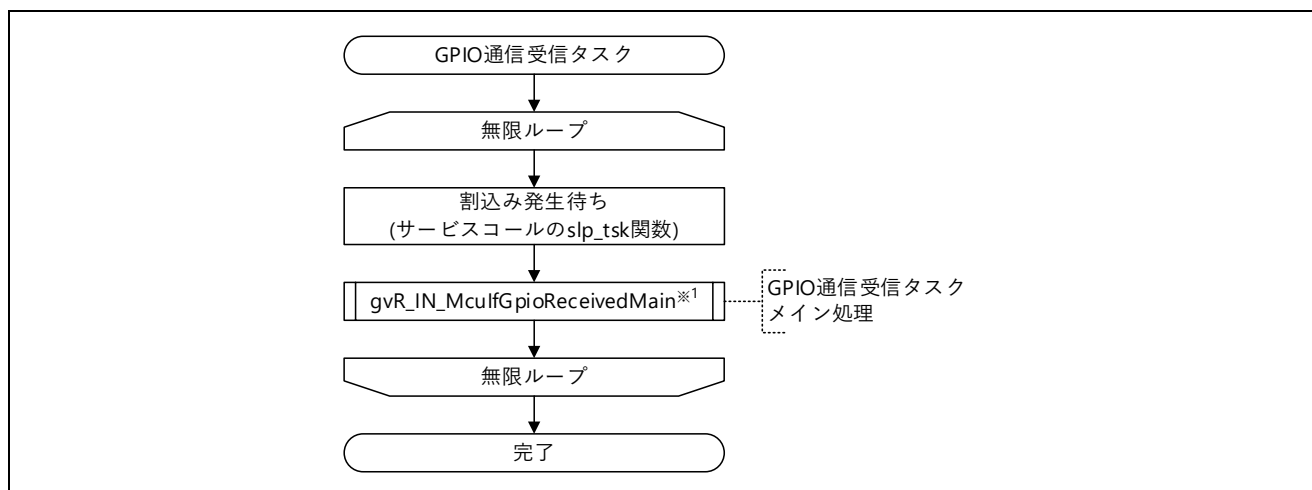


図 5.14 GPIO 通信応答受信タスク処理概要

【注】 ※1 「6.4.17(4) gvR\_IN\_MculfGpioReceivedMain」を参照してください。

#### <注意事項>

本タスクは、MCU 間 I/F 機能を有効(コンパイルスイッチ「MCUIF\_ENABLE」を有効)にした上で、マスタ局が安全通信使用有無設定で「使用する」を指定した場合に起動します。

### 5.2.11 RT DMA 転送完了タスク

#### (1) タスクの処理概要

本タスクは、無限ループにより「割り込み発生待ち」と「RT DMA 転送完了タスクメイン処理」を繰り返し実行します。

本タスクは、割り込みが発生したときに処理を実施します。RTOS がリアルタイム・ポート用 DMAC 転送完了割り込みを検出すると、Hardware ISR がタスクを起床します。タスクの処理概要を示します。

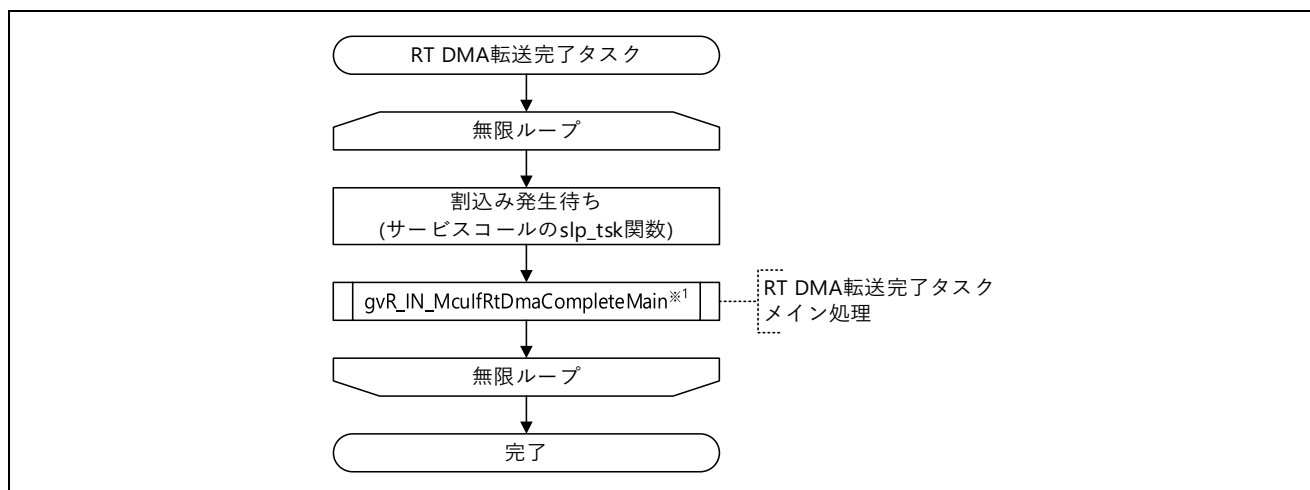


図 5.15 GPIO 通信応答受信タスク処理概要

【注】 ※1 「6.4.17(5) gvR\_IN\_McufRtDmaCompleteMain」を参照してください。

#### <注意事項>

本タスクは、MCU 間 I/F 機能を有効(コンパイルスイッチ「MCUIF\_ENABLE」を有効)にした上で、マスタ局が安全通信使用有無設定で「使用する」を指定した場合に起動します。

## 5.2.12 サイクリックフレーム送信タスク(CC-Link IE TSN Class A)

### (1) タスクの処理概要

本タスクは、自局が CC-Link IE TSN Class A として動作する場合に、サイクリックフレームを送信する処理を実行します。無限ループにより、「タスクの起床待ち」と「サイクリックフレーム送信処理(CC-Link IE TSN Class A)」を繰り返し実行します。

本タスクは、アイドルタスクにより起動し、定周期処理タスクにより待機から起床へ移行し処理を実施します。タスクの処理概要を示します。

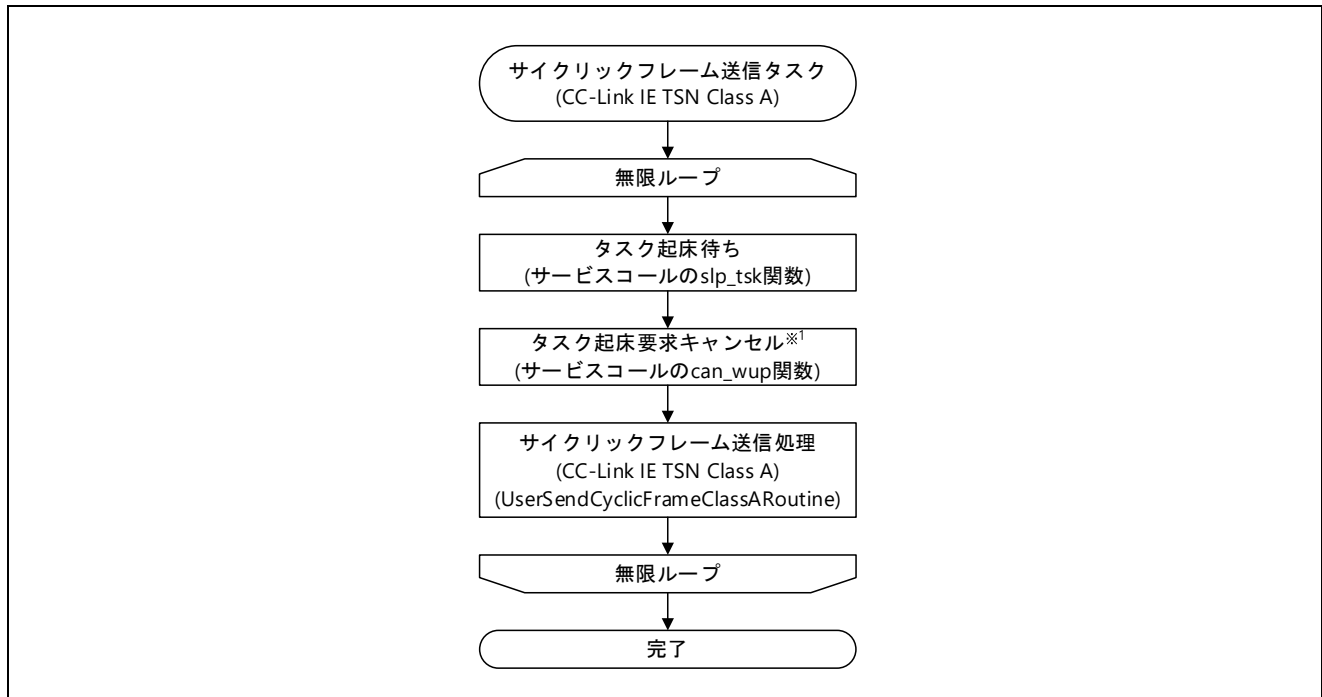


図 5.16 サイクリックフレーム送信タスク(CC-Link IE TSN Class A)処理概要

【注】※1 定周期処理タスクと本タスクは 1 対 1 で動作します。ただし、定周期処理タスクの負荷が高い場合など、定周期処理タスクが複数回動作する間に、本タスクが 1 回しか動作しないようなケースが想定されます。そのような場合に、本タスクが不必要に起床されることを防ぐために、can\_wup 関数により不要な起床要求をキャンセルします。

### (2) タスクの処理概要

サイクリックフレーム送信処理(CC-Link IE TSN Class A)の概略フローを示します。

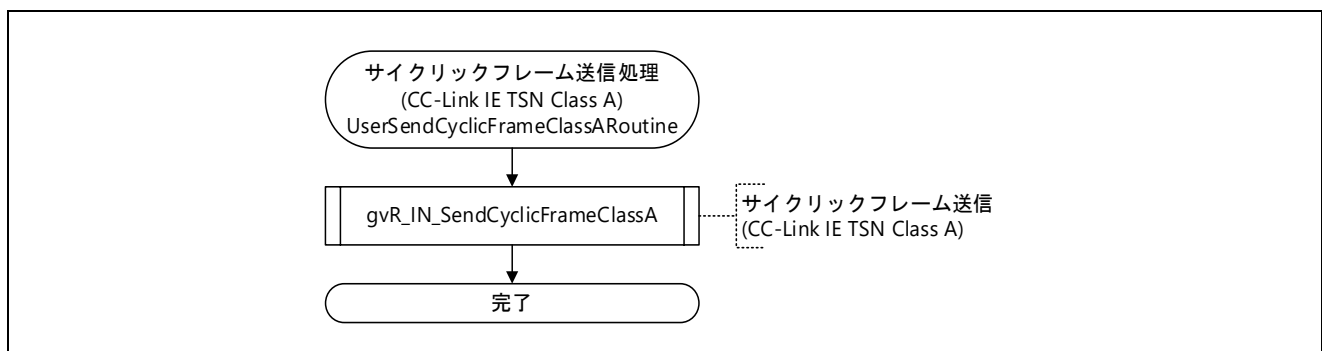


図 5.17 サイクリックフレーム送信処理(CC-Link IE TSN Class A)フロー図

## 5.3 ユーザプログラム詳細(イニシャル関連)

### 5.3.1 初期化処理

R-IN32M4-CL3 の初期化、および IP アドレスを設定します。

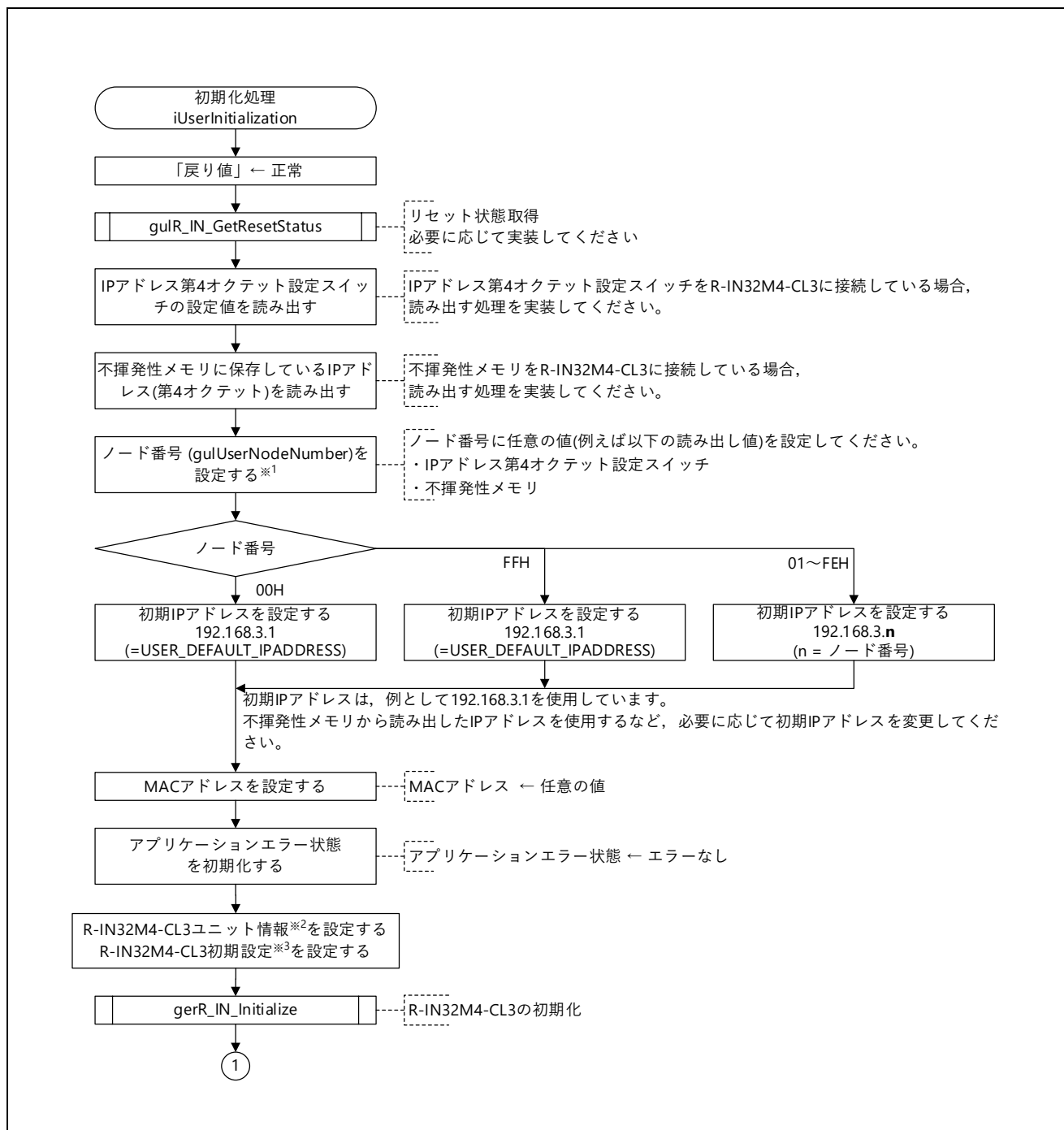


図 5.18 初期化処理フロー図

【注】 ※1 ノード番号(gulUserNodeNumber)に 00H を設定することで、デバイス局の IP アドレス設定が実施できます。詳細は「5.6.7 SLMP IP アドレス変更要求コマンド受信処理」を参照してください。

※2 「表 6.9 R\_IN\_UNITINFO\_T 一覧」を参照してください。

※3 「表 6.17 R\_IN\_UNITINIT\_T 一覧」を参照してください。

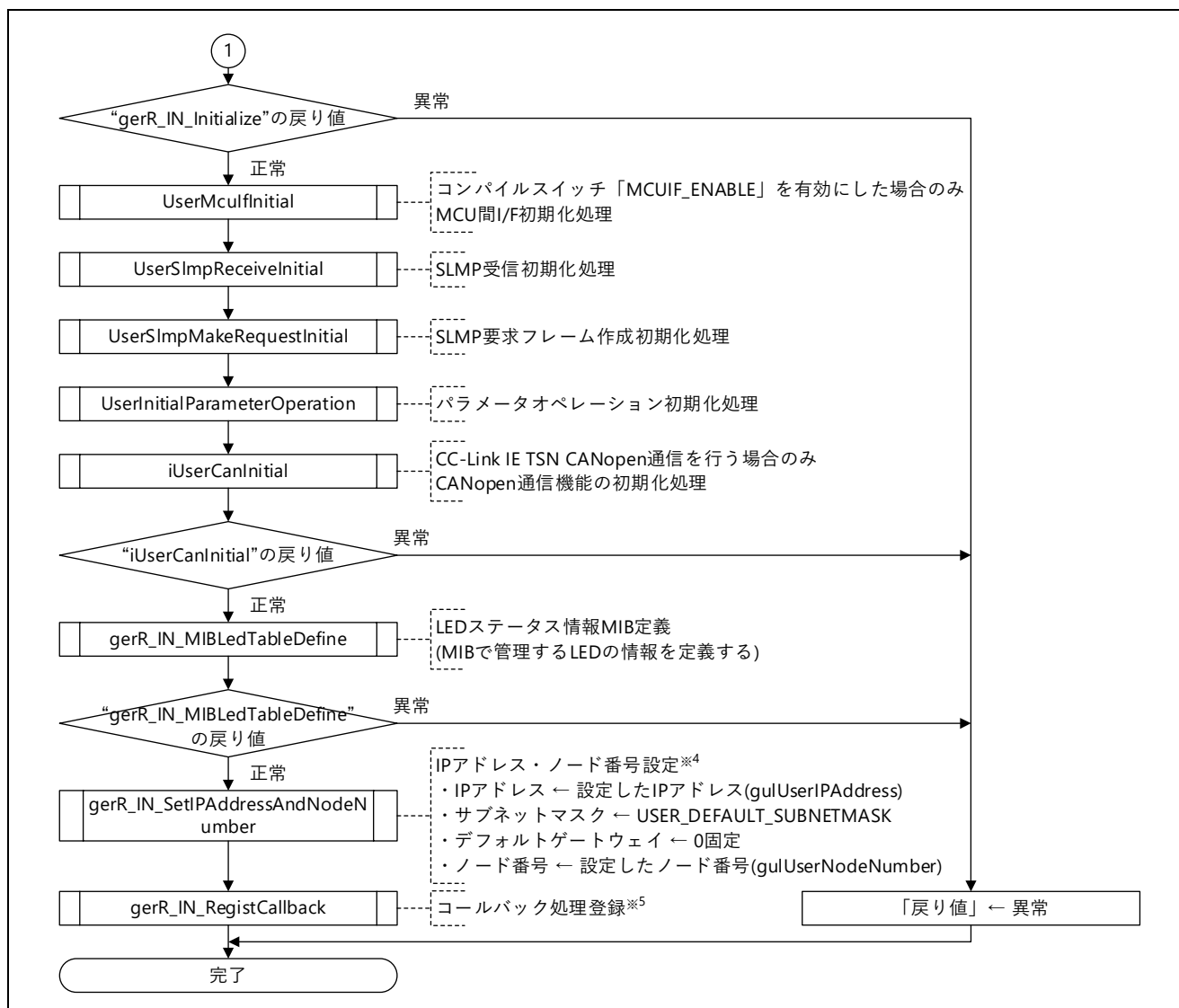


図 5.19 初期化处理フロー図

【注】 ※4 R-IN32M4-CL3 ドライバへ IP アドレス関連の初期値を設定します。IP アドレス第 1～第 3 オクテットとサブネットマスクは CC-Link IE TSN 通信を開始するときに R-IN32M4-CL3 ドライバによってマスタ局が指定する値に変更されます。変更タイミングは「図 5.20 IP アドレスの書き込みタイミング」を参照してください。

※5 登録する処理は「表 6.19 コールバック処理仕様」を参照してください。

## &lt;IP アドレス第 1～第 3 オクテットの補足&gt;

iUserInitialization 実行時は、IP アドレス「gulUserIPAddress」が「gerR\_IN\_SetIPAddressAndNodeNumber」(6.4.1(4))によって、IP アドレス第 1～第 3 オクテットに書き込まれます。

iUserInitialization 完了後、iUserStart (5.3.2 通信開始処理)にて CC-Link IE TSN 通信を開始する時は、(Detection フレームをもとに算出した)マスタ局の指定する値が R-IN32M4-CL3 ドライバによって、IP アドレス第 1～第 3 オクテットに書き込まれます。

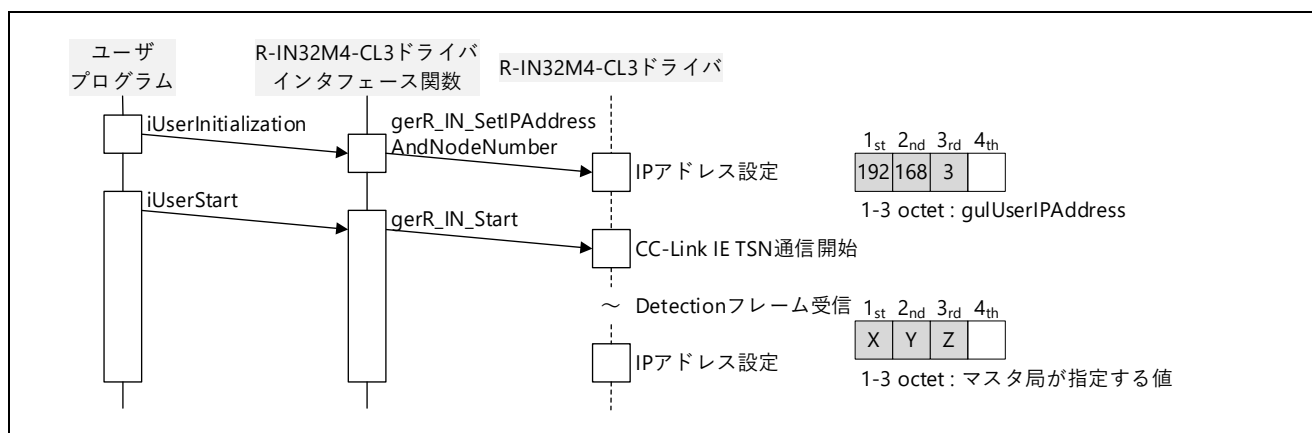


図 5.20 IP アドレスの書き込みタイミング

### 5.3.2 通信開始処理

R-IN32M4-CL3 に通信の開始を指示します。本処理は初期化処理後、一度だけ実行してください。

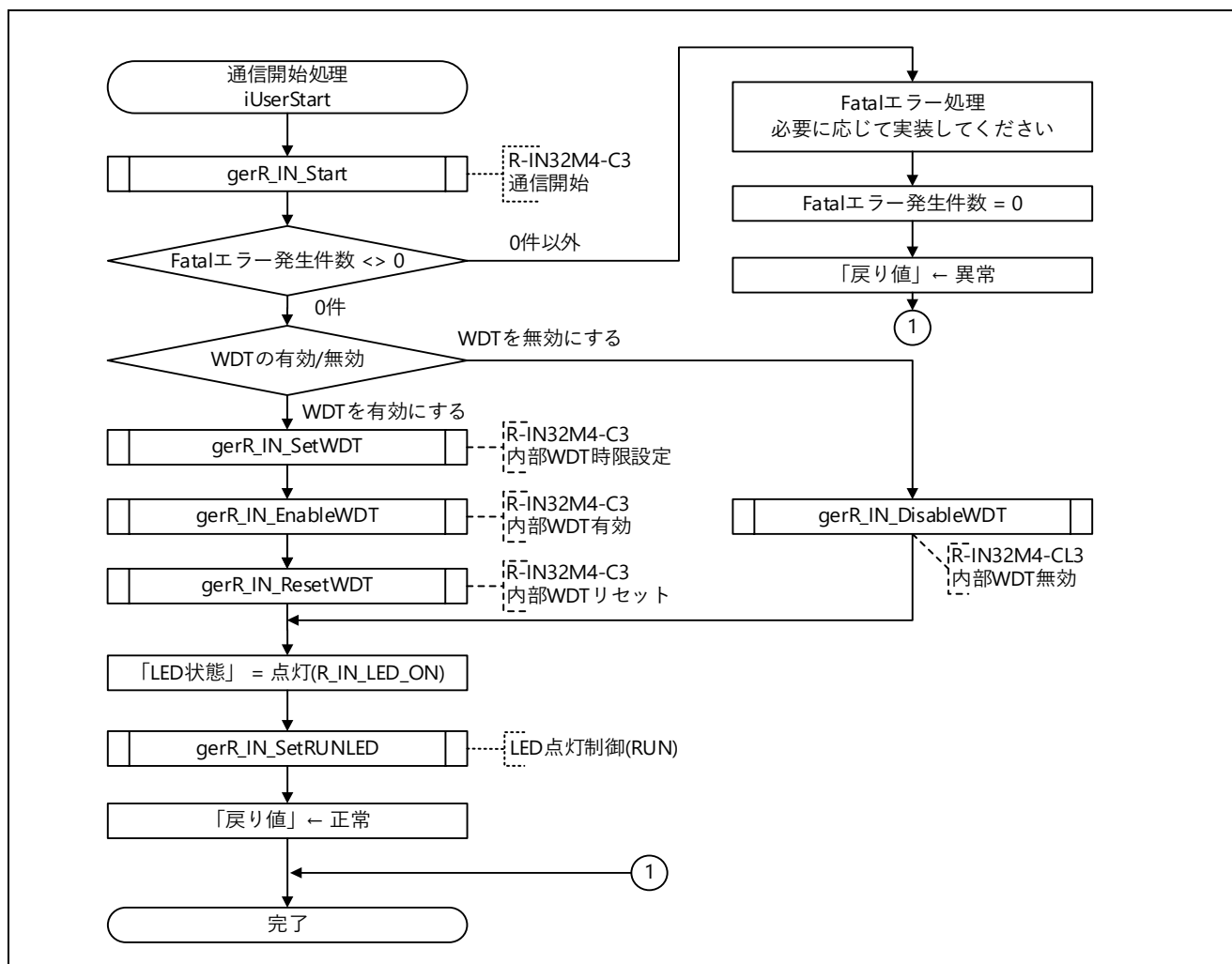


図 5.21 通信開始処理フロー図

< USNetPlus の API 関数を使用して、Socket プログラムを作成する際の注意事項>

■ マスタ局を接続しないで汎用 Ethernet 通信を行う場合

USNetPlus の初期化は「gerR\_IN\_Start」(6.4.1(5))で実行しているため、gerR\_IN\_Start を実行したあとに、Socket プログラムを実行するように実装してください。

■ マスタ局を接続して汎用 Ethernet 通信を行う場合

USNetPlus の初期化を gerR\_IN\_Start 内で実行していますので、gerR\_IN\_Start を実行したあとに、データリンク中であることを確認して、Socket プログラムを実行するように実装してください。

データリンクの確認は、「gerR\_IN\_GetCommunicationStatus」(6.4.6(4))を実行し、データリンク状態を取得してください。引数のデータリンク状態がデータリンク未実施(解列中)以外となった場合に、Socket プログラムを実行するように実装してください。



### 5.3.3 SLMP 受信初期化処理

SLMP 要求コマンド実行関数テーブル「R\_IN\_SLMP\_FUNCTION\_REQUEST\_TBL\_T」および SLMP 応答コマンド実行関数テーブル「R\_IN\_SLMP\_FUNCTION\_RESPONSE\_TBL\_T」のポインタを R-IN32M4-CL3 ドライバに通知します。

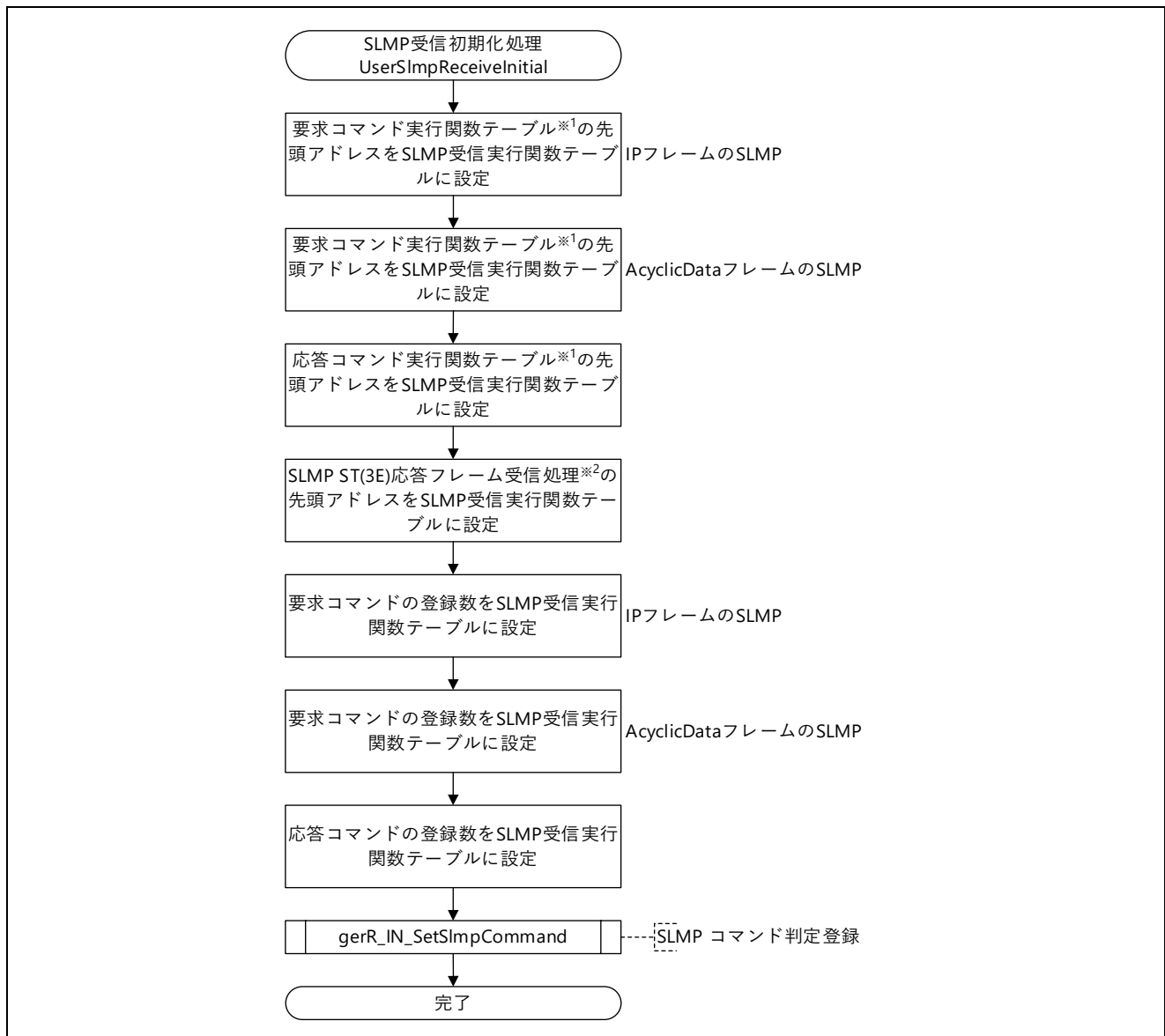


図 5.22 SLMP 受信初期化処理フロー図

【注】 ※1 実行関数テーブルの詳細は「表 6.30」および「表 6.33」を参照してください。

※2 詳細は「UserSmpReceive3EFrame」(5.5.8 SLMP ST(3E)応答フレーム受信処理)を参照してください。

### 5.3.4 SLMP 要求フレーム作成初期化処理

「UserSimpMakeRequest」(5.5.7 SLMP 要求フレーム作成処理)で使用するグローバル変数の初期化を行います。

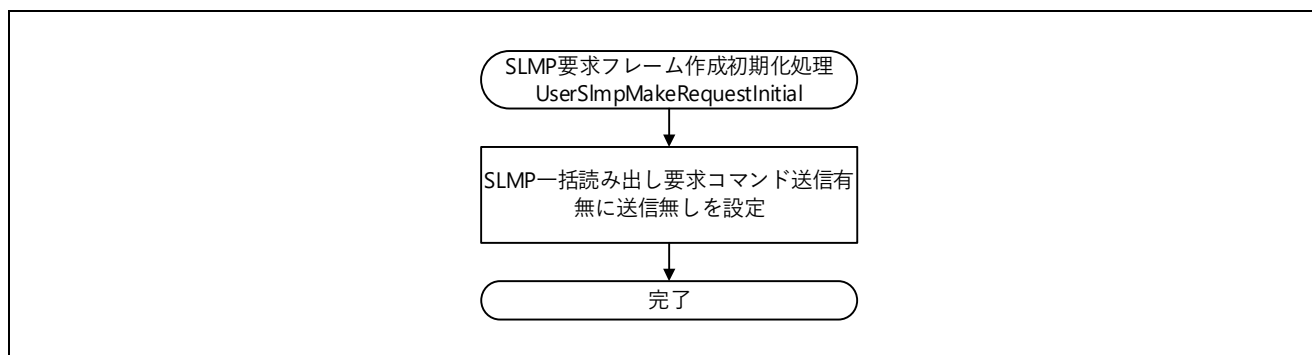


図 5.23 SLMP 要求フレーム作成初期化処理フロー図

### 5.3.5 パラメータオペレーション初期化処理

デバイス局パラメータ自動設定(パラメータオペレーション)で使用するグローバル変数の初期化を行います。

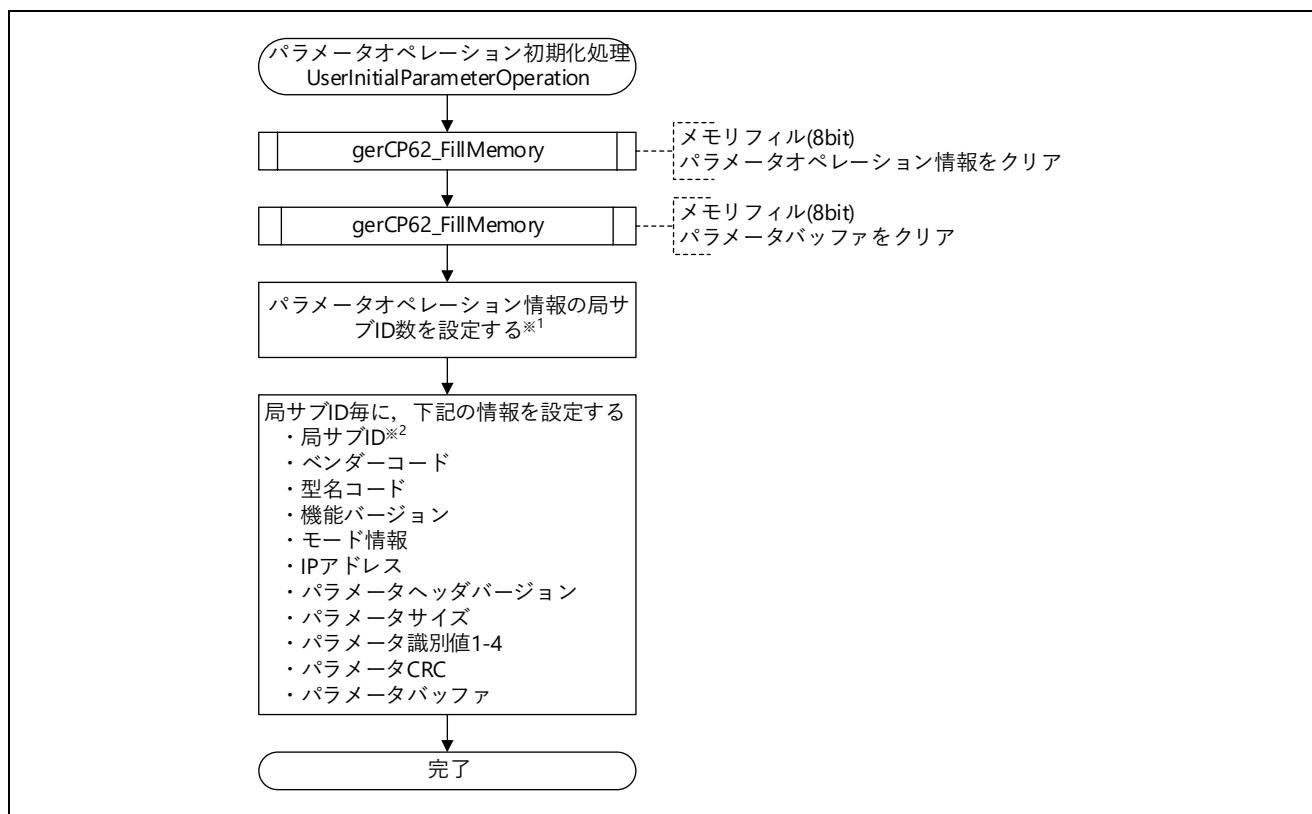


図 5.24 パラメータオペレーション初期化処理フロー図

※1：局サブ ID 数 (usStationSubIdNum) の値は、増設ユニットなしの場合、0001H を設定してください。増設ユニットありの場合、基本ユニットと増設ユニットの合計台数を設定してください。

※2：局サブ ID (usStationSubId) の値は、増設ユニットなしの場合、固定値 8000H を設定してください。増設ユニットありの場合、基本ユニットに固定値 8000H を設定し、増設ユニットには 0000H～0FFFH の範囲で設定してください。  
GX Works3 でパラメータ自動設定を行う場合、増設ユニットの局サブ ID 値は 0000H から設定してください。

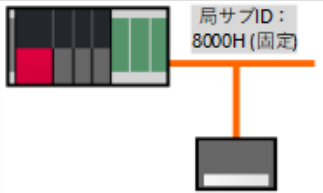
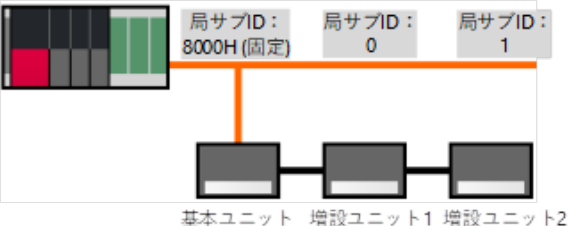
増設ユニット 1 台目：局サブ ID=0

増設ユニット 2 台目：局サブ ID=1

...

増設ユニット n 台目：局サブ ID=n-1

表 5.19 局サブ ID に関する設定値

増設ユニットの有無	局サブ ID 数	局サブ ID の設定値	オプション 情報数(参考)
増設ユニットなし (増設ユニットに非対応、 または増設ユニットを 未装着の基本ユニット)	0001H		00H
増設ユニットあり (増設ユニット 2 台の例)	0003H		02H

## 5.4 ユーザプログラム詳細(サイクリック伝送関連)

### 5.4.1 サイクリック受信処理

サイクリック受信フレームからサイクリックデータ(RY および RWw)を取得します。

安全 PDU 送受信時(コンパイルスイッチ「SAFETY\_PDU\_ENABLE」を有効にした場合)は、RY および RWw に加え RSPDU を取得します。

本処理は、サイクリックデータを取得する処理例として、3 つの処理を記述しています。使用時は、どれか 1 つのみ使用して、他のサイクリック受信処理はコメントアウトなどで無効化してください。

なお、CANopen 通信時は「5.9.2 サイクリック受信処理(RPDO 更新)」を使用してください。以下 3 つの処理は使用できません。

#### (1) サイクリック受信処理(高速)

【概要】 R-IN32M4-CL3 ドライバインタフェース関数を呼ばず、RY、RWw および RSPDU 領域のアドレスを直接指定して読み出します。

【特長】 受信サイズが 1420～2400 バイトに渡る場合でも、高速に処理できます。パフォーマンスを優先したい場合にご利用ください。

#### (2) サイクリック受信処理(一括)

【概要】 R-IN32M4-CL3 ドライバインタフェース関数を呼び出して、RY、RWw および RSPDU 全領域を毎周期で読み出します。

【特長】 R-IN32M4-CL2 サンプルコード「UserReceiveCyclic」関数と互換できるよう記述しています。過去に CC-Link IE Field 接続対応製品を開発したことがある場合にご利用ください。

#### (3) サイクリック受信処理(個別)

【概要】 R-IN32M4-CL3 ドライバインタフェース関数を呼び出して、RY、RWw および RSPDU 領域の一部だけを毎周期で読み出します。

【特長】 RY、RWw および RSPDU 領域を高頻度に読み出す領域と低頻度に読み出す領域に分けている場合にご利用ください。

#### ポイント

サイクリック伝送機能は、アプリケーションの規模による負荷変動によって、送受信できる最大サイズが異なります。定周期処理タスクが 200  $\mu$ s 周期で動作した場合で、送受信可能なサイズ (バイト) の目安を示します。

ユーザプログラム	CCIETSN Class B	CCIETSN Class A	
	1Gbps/100Mbps	1Gbps	100Mbps
サイクリック送信処理(高速) / サイクリック受信処理(高速)	1426	1116	606
サイクリック送信処理(一括) / サイクリック受信処理(一括)	678	548	368
サイクリック送信処理(個別) / サイクリック受信処理(個別)	490	368	232

送受信最大サイズ(2400 バイト)まで使用する場合は、サイクリック送信処理(高速)/サイクリック受信処理(高速)を使用してください。その際には、定周期処理タスクの処理間隔を 400  $\mu$ s 周期程度に設定してください。

## (1) サイクリック受信処理(高速)

サイクリック受信処理(高速)は、RY、RWw および RSPDU の格納先アドレスを直接指定してデータを読み出します。  
(送受信最大サイズ 2400 バイトまで使用する場合にご利用ください)

サイクリック受信処理(高速)と、他のサイクリック受信処理と同時に使用することはできません。

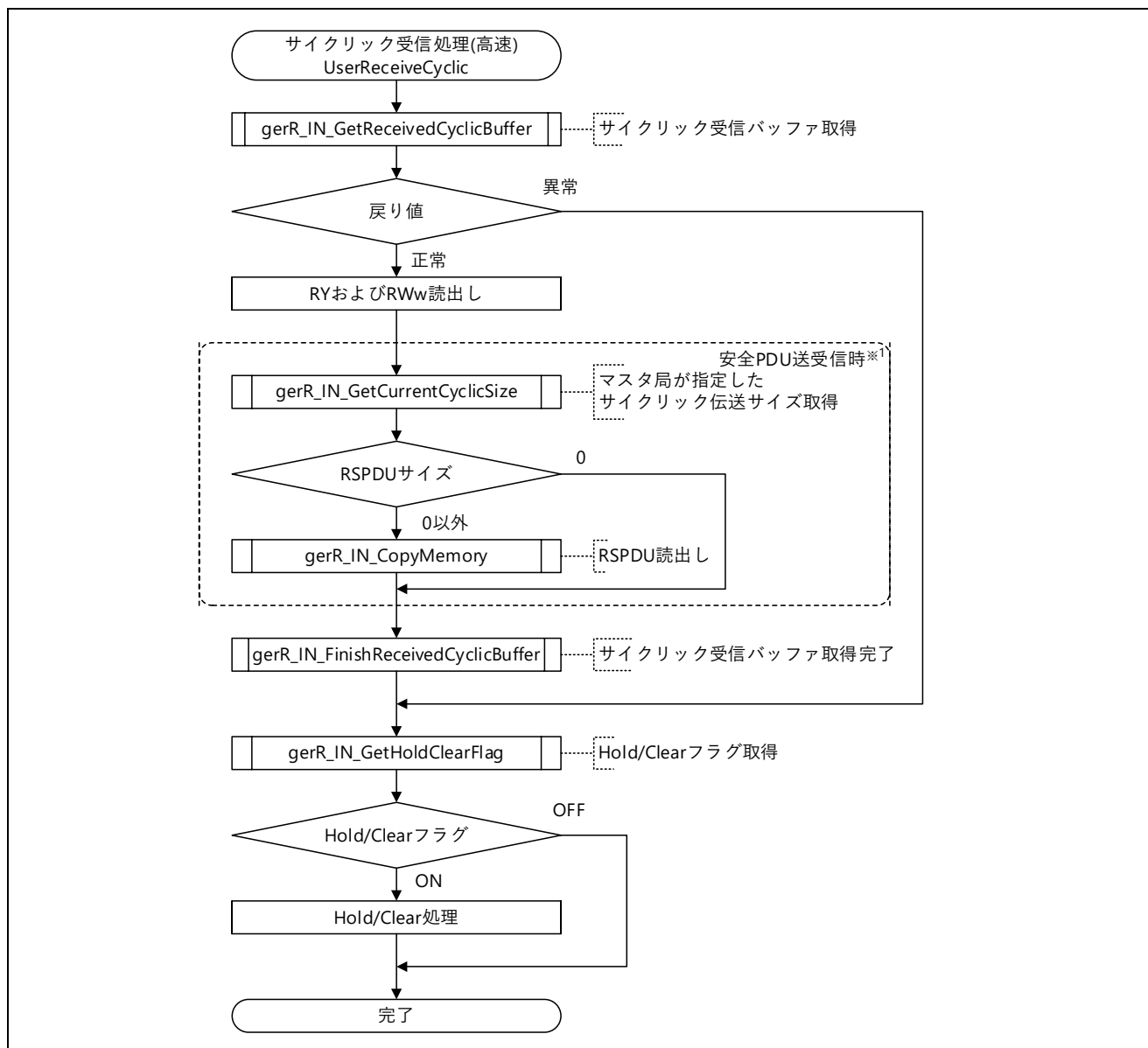


図 5.25 サイクリック受信処理(高速)フロー図

【注】 ※1 コンパイルスイッチ「SAFETY\_PDU\_ENABLE」が有効の場合に実行します。

## (2) サイクリック受信処理(一括)

サイクリック受信処理(一括)は、RY、RWw および RSPDU 領域のそれぞれ全域(先頭から終端)を、一括で読み出します。(RY および RWw 全領域を毎周期で読み出す場合などにご利用ください)

サイクリック受信処理(一括)と、他のサイクリック受信処理と同時に使用することはできません。

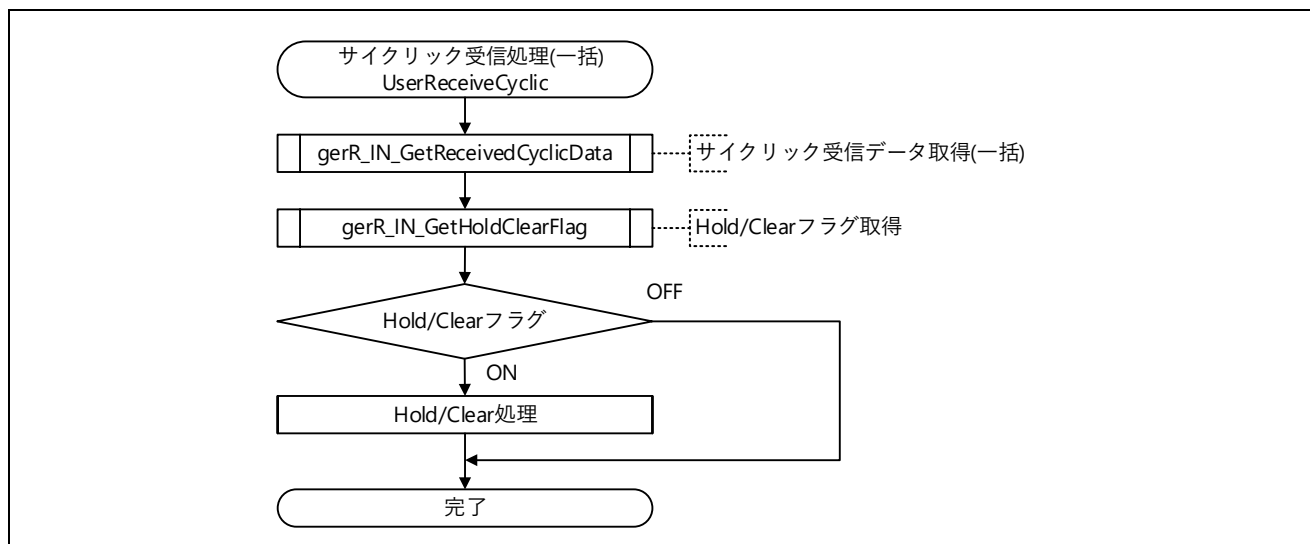


図 5.26 サイクリック受信処理(一括)フロー図

### (3) サイクリック受信処理(個別)

サイクリック受信処理(個別)は、RY、RWw および RSPDU 領域のそれぞれ指定位置(領域先頭から任意のオフセット位置)を、個別に読み出します。

(RY、RWw および RSPDU 領域の一部だけ毎周期で読み出す場合などにご利用ください)

サイクリック受信処理(個別)と、他のサイクリック受信処理と同時に使用することはできません。

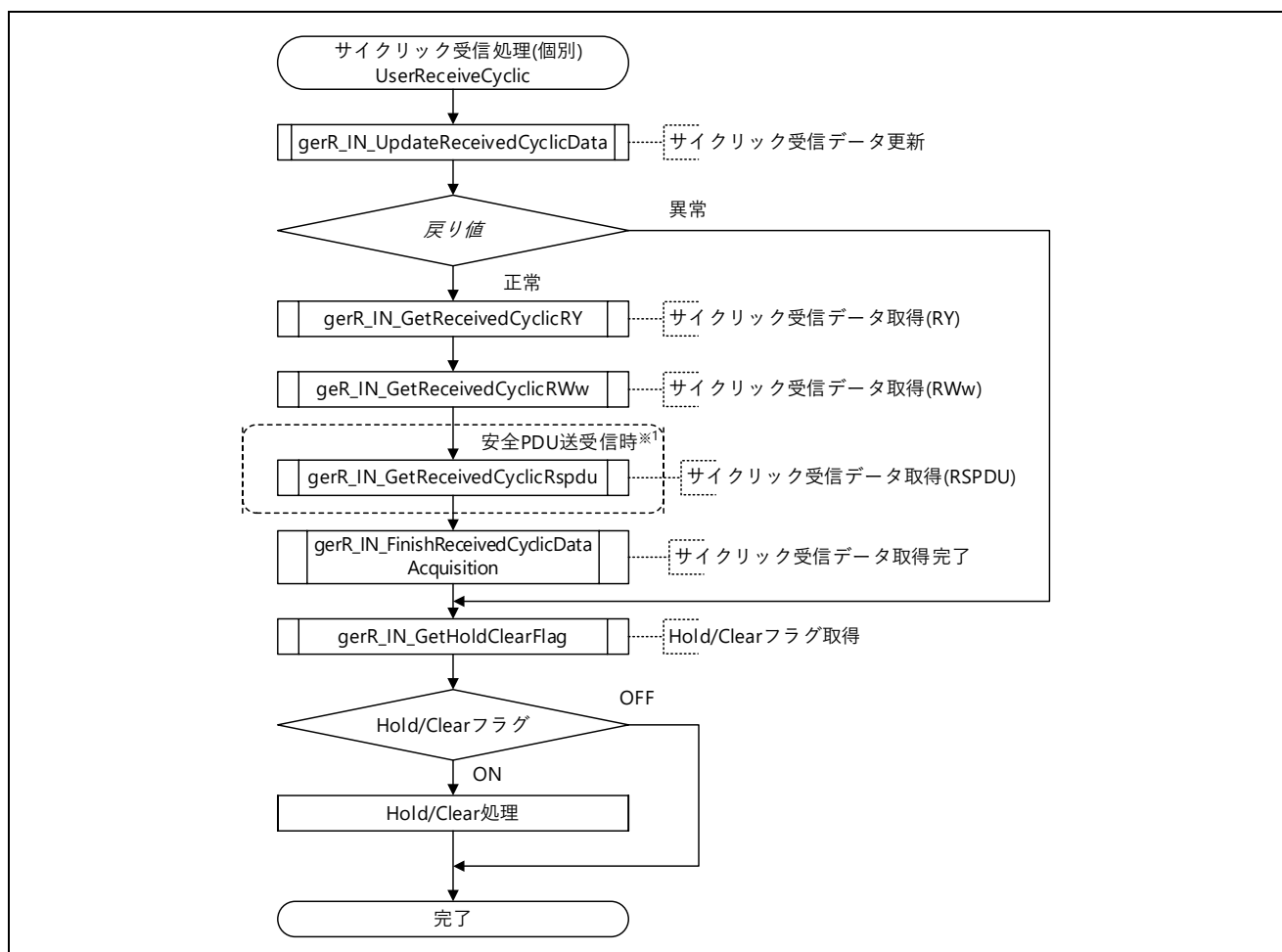


図 5.27 サイクリック受信処理(個別)フロー図

【注】 ※1 コンパイルスイッチ「SAFETY\_PDU\_ENABLE」が有効の場合に実行します。

#### (4) Hold/Clear 処理について

Hold/Clear 処理は、自局がデータリンクから解列したとき、またはマスタ局アプリケーションが停止/異常のとき、受信したサイクリックデータの自局外部への出力を続行する(Hold)、または停止する(Clear)処理です。

フェールセーフとして、以下を考慮して Hold/Clear 処理を実装してください。(ユーザの任意処理になります。)

##### 【マスタ局が送信するサイクリックデータ(RY および RWw)について】

マスタ局アプリケーション停止/異常のとき、マスタ局が送信するサイクリックデータは、マスタ局の設定によって Hold または Clear されています。デバイス局(自局)は、マスタ局の送信するサイクリックデータが Hold あるいは Clear どちらのデータか、あらかじめ検出することができません。

(マスタ局に RJ71GN11-T2 を使用する場合、“CPU ユニット STOP 時の出力状態設定” および “CPU 停止エラー時の出力状態設定” で Hold/Clear 処理を設定します。)

### 5.4.2 自局状態送信処理

StsW のうちアプリケーションエラー状態を設定します。※1

アプリケーションエラー状態は、StsW に格納し、サイクリック伝送でマスタ局へ通知されます。

※1 : StsW のうち、アプリケーションエラー状態のみ、ユーザプログラム(本処理)で設定します。その他の項目は、R-IN32M4-CL3 ドライバで設定しています。

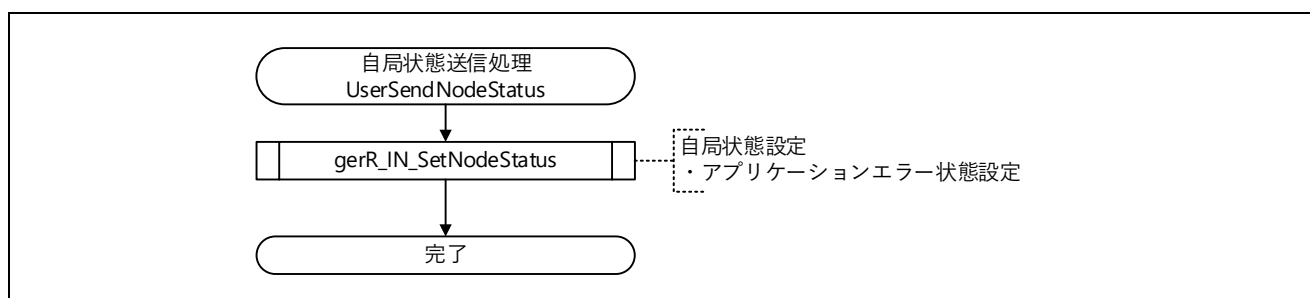


図 5.28 自局状態送信処理フロー図

引数に設定するアプリケーションエラー状態は、以下の 4 種類があります。エラー状態の度合いに規定はありませんので、任意に定義してください。参考として、三菱電機製リモート I/O ユニットのエラー状態を例示します。

表 5.20 アプリケーションエラー状態の例 (三菱電機製リモート I/O ユニットの例)

アプリケーションエラー状態	内容
エラーなし (0)	-
軽度異常 (1)	ユニットが動作を続行するエラー
中度異常 (2)	ユニットが動作を停止し、エンドユーザの操作で処置が可能なエラー
重度異常 (3)	ユニットが動作を停止し、エンドユーザでは復旧不可能なエラー

RJ71GN11-T2 をマスタ局とする場合、以下のリンク特殊レジスタ(SW)で各局のアプリケーションエラー状態をモニタできます。各局のアプリケーションに異常がある場合、直前のサイクリックデータを保持します。

- ・ SW0100 - SW0107(各局 CPU 中重度異常発生状態)
- ・ SW0110 - SW0117(各局 CPU 軽度異常発生状態)



### 5.4.3 サイクリック送信処理

サイクリックデータ(RX および RWr)をサイクリック送信フレームに設定します。

安全 PDU 送受信時(コンパイルスイッチ「SAFETY\_PDU\_ENABLE」を有効にした場合)は、RX および RWr に加え SSPDU を設定します。

本処理は、サイクリックデータを設定する処理例として、3 つの処理を記述しています。使用時は、どれか 1 つのみを使用して、他のサイクリック送信処理はコメントアウトなどで無効化してください。

なお、CANopen 通信時は「5.9.3 サイクリック送信処理(TPDO 更新)」を使用してください。以下の 3 つの処理は使用できません。

#### (1) サイクリック送信処理(高速)

【概要】 R-IN32M4-CL3 ドライバインタフェース関数を呼ばず、RX、RWr および SSPDU 領域のアドレスを直接指定して書き込みます。

【特長】 送信サイズが 1420~2400 バイトに渡る場合でも、高速に処理できます。パフォーマンスを優先したい場合にご利用ください。

#### (2) サイクリック送信処理(一括)

【概要】 R-IN32M4-CL3 ドライバインタフェース関数を呼び出して、RX、RWr および SSPDU 全領域を毎周期で書き込みます。

【特長】 R-IN32M4-CL2 サンプルコード「UserSendCyclic」関数と互換できるよう記述しています。  
過去に CC-Link IE Field 接続対応製品を開発したことがある場合にご利用ください。

#### (3) サイクリック送信処理(個別)

【概要】 R-IN32M4-CL3 ドライバインタフェース関数を呼び出して、RX、RWr および SSPDU 領域の一部だけを毎周期で書き込みます。

【特長】 RX、RWr および SSPDU 領域を高頻度で書き込む領域と低頻度で書き込む領域に分けている場合にご利用ください。

#### ポイント

サイクリック伝送機能は、アプリケーションの規模による負荷変動によって、送受信できる最大サイズが異なります。定周期処理タスクが 200  $\mu$ s 周期で動作した場合で、送受信可能なサイズ (バイト) の目安を示します。

ユーザプログラム	CCIETSN Class B		CCIETSN Class A	
	1Gbps/100Mbps		1Gbps	100Mbps
サイクリック送信処理(高速) / サイクリック受信処理(高速)	1426		1116	606
サイクリック送信処理(一括) / サイクリック受信処理(一括)	678		548	368
サイクリック送信処理(個別) / サイクリック受信処理(個別)	490		368	232

送受信最大サイズ(2400 バイト)まで使用する場合は、サイクリック送信処理(高速)/サイクリック受信処理(高速)を使用してください。その際には、定周期処理タスクの処理間隔を 400  $\mu$ s 周期程度に設定してください。

## (1) サイクリック送信処理(高速)

サイクリック送信処理(高速)は、RX、RWr および SSPDU の格納先アドレスを直接指定してデータを書き込みます。  
(送受信最大サイズ 2400 バイトまで使用する場合にご利用ください)

サイクリック送信処理(高速)と、他のサイクリック送信処理と同時に使用することはできません。

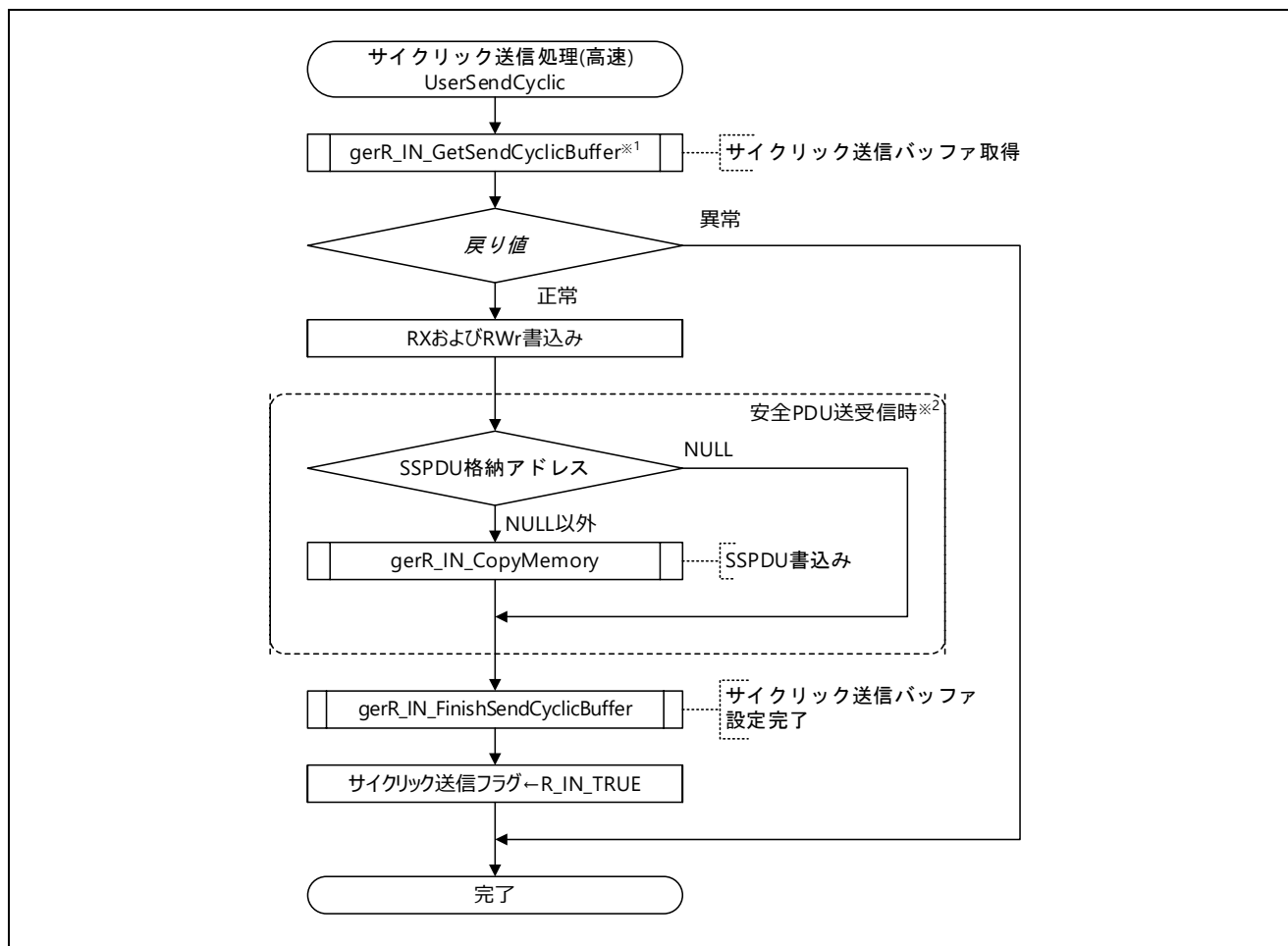


図 5.29 サイクリック送信処理(高速)フロー図

【注】 ※1 送信フレームが 1518 バイトを越え 2 フレームに分割される場合、「サイクリック分割送信バッファ取得」(6.4.4(17) gerR\_IN\_GetSplitSendCyclicBuffer)を使用してください。

※2 コンパイルスイッチ「SAFETY\_PDU\_ENABLE」が有効の場合に実行します。

## (2) サイクリック送信処理(一括)

サイクリック送信処理(一括)は、RX、RWr および SSPDU 領域のそれぞれ全域(先頭から終端)へ、一括で書き込みます。  
(RX、RWr および SSPDU および RWr 全領域を毎周期で書き込む場合などにご利用ください)

サイクリック送信処理(一括)と、他のサイクリック送信処理と同時に使用することはできません。

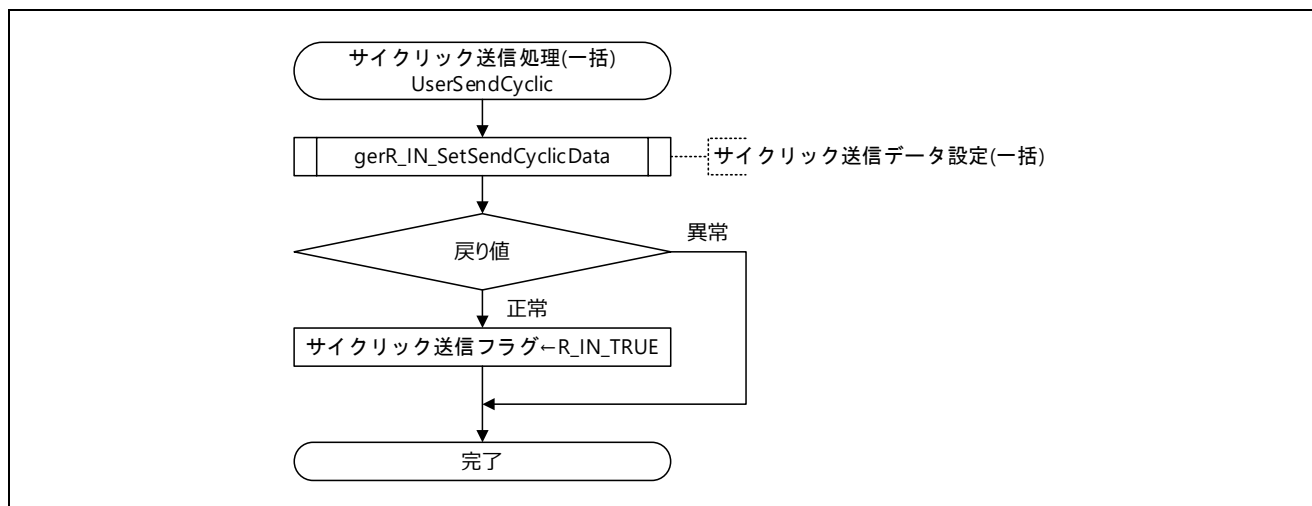


図 5.30 サイクリック送信処理(一括)フロー図

### (3) サイクリック送信処理(個別)

サイクリック送信処理(個別)は、RX、RWr および SSPDU 領域のそれぞれ指定位置(領域先頭から任意のオフセット位置)へ、個別に書き込みます。(RX および RWr 領域の一部だけ毎周期で書き込む場合などにご利用ください)

サイクリック送信処理(個別)と、他のサイクリック送信処理と同時に使用することはできません。

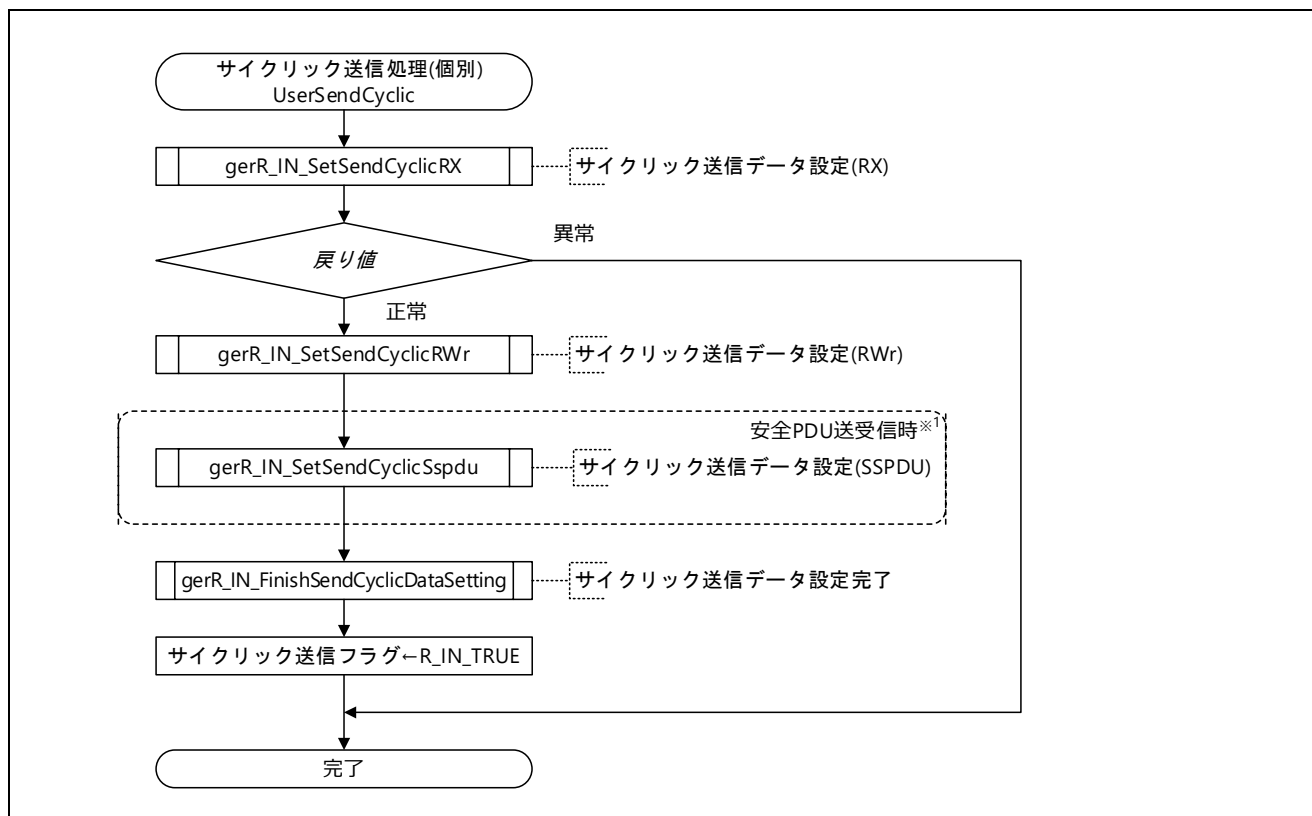


図 5.31 サイクリック送信処理(個別)フロー図

【注】 ※1 コンパイルスイッチ「SAFETY\_PDU\_ENABLE」が有効の場合に実行します。

#### 5.4.4 通信状態更新処理

自局のデータリンク状態を取得し、データリンク状態に応じて ERR. LED 制御フラグ(gulErrCtrl)を設定します。

ERR. LED 制御フラグ(gulErrCtrl)は、「UserUpdateLed」(5.5.3 LED 更新処理)において、ERR. LED の点灯/消灯/点滅制御に使います。

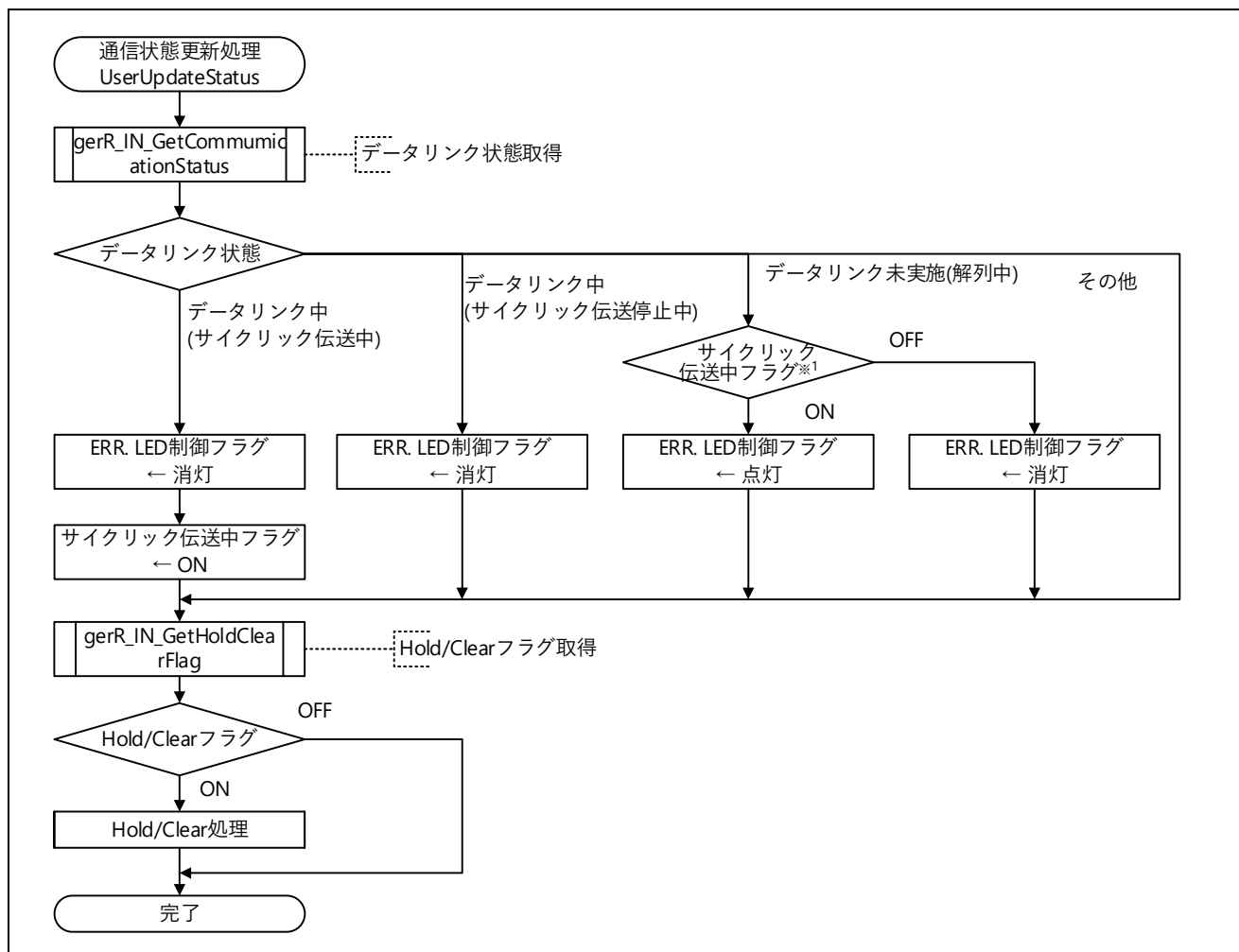


図 5.32 通信状態更新処理フロー図

【注】※1 サイクリック伝送実行中の解列か、リンクアップしてもサイクリック伝送を実行できず解列した状態かを判定します。

#### Hold/Clear 処理について

Hold/Clear 処理は、自局がデータリンクから解列したとき、またはマスタ局アプリケーションが停止/異常のとき、受信したサイクリックデータの自局外部への出力を続行する(Hold)、または停止する(Clear)処理です  
詳細は「5.4.1 サイクリック受信処理」を参照してください。

### 5.4.5 サイクリック伝送状態更新処理

マスタ局から指定されたサイクリック伝送サイズと、サイクリック伝送状態を取得します。

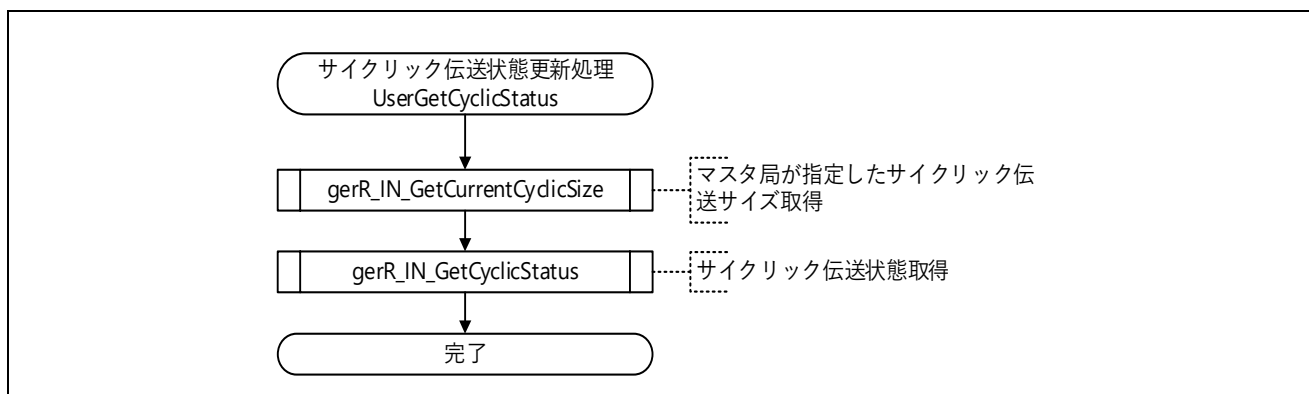


図 5.33 サイクリック伝送状態更新処理フロー図

## 5.5 ユーザプログラム詳細(状態管理およびトランジェント伝送関連)

### 5.5.1 自局エラー処理

ユーザアプリケーションのエラー処理を、ユーザ作成・追加のタスクではなく、サンプルコードのアイドルタスクに記述する場合、本関数にエラー処理を追加してください。

本処理はユーザの任意で実装する処理です。

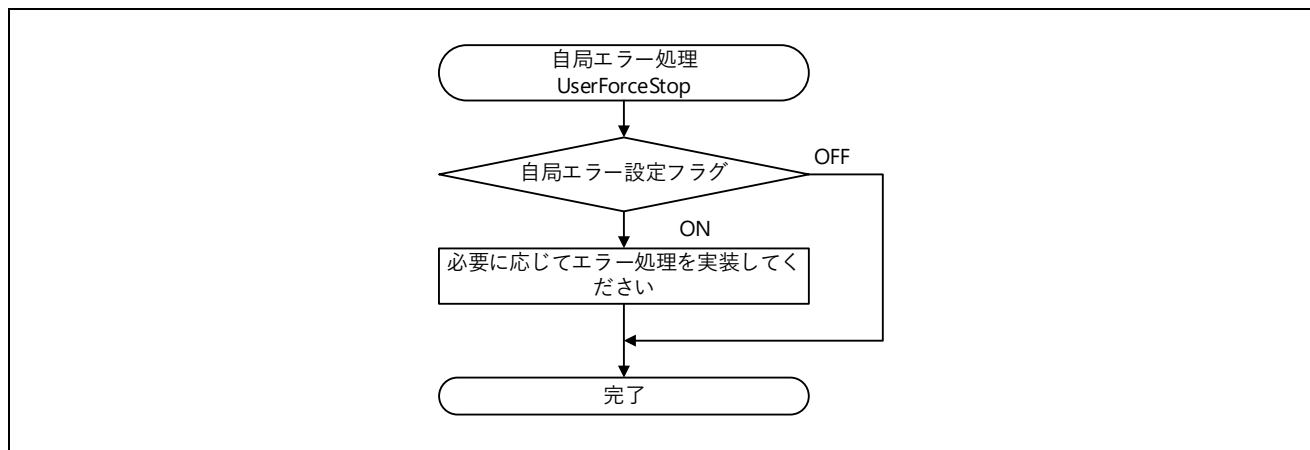


図 5.34 自局エラー処理フロー図

### 5.5.2 イベント処理

R-IN32M4-CL3 で発生したイベントをもとに対応する処理を実行します。

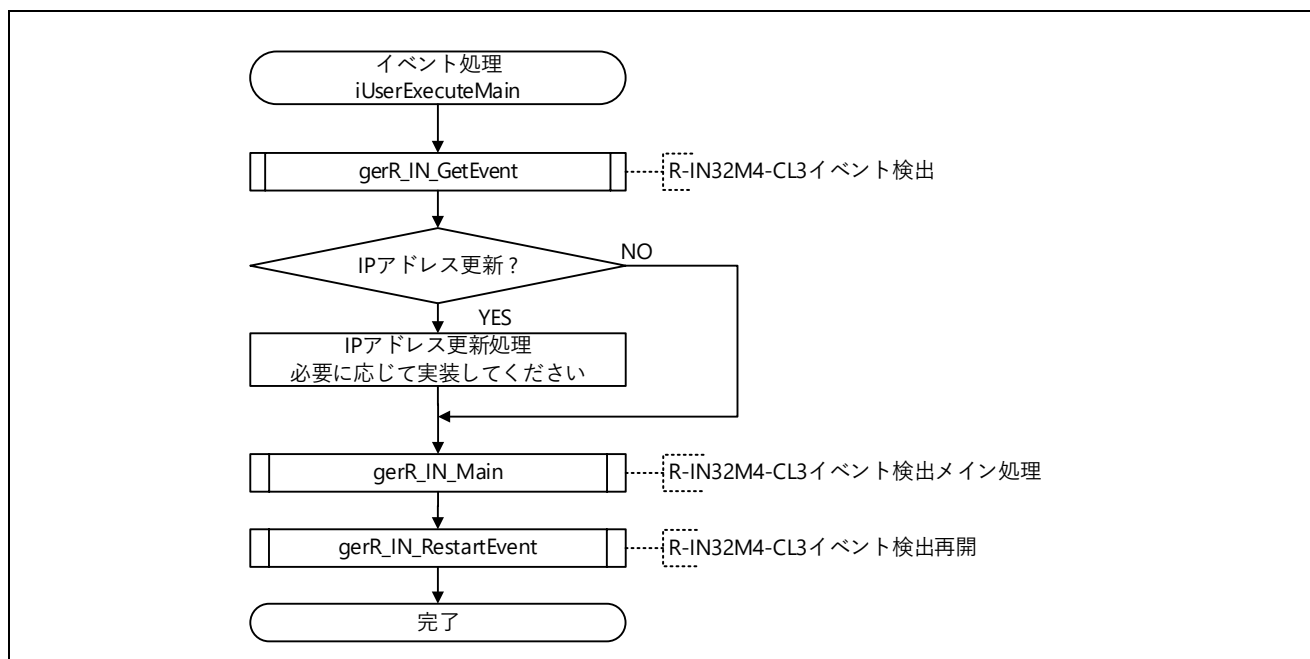


図 5.35 イベント処理フロー図

### 5.5.3 LED 更新処理

自局のデータリンク状態に応じて、RUN LED、D LINK LED、L ER1 LED、L ER2 LED および ERR. LED の点灯/消灯/点滅の制御を行います。

なお、ERR. LED の点灯/消灯/点滅の制御は、「UserUpdateStatus」(5.4.4 通信状態更新処理)で設定した ERR. LED 制御フラグ(gulErrCtrl)の値に応じて行います。

ERR. LED の点灯/消灯/点滅の制御をユーザアプリケーションの都合で行う場合、ERR. LED 制御フラグ(gulErrCtrl)の値を本関数内で変更してください。

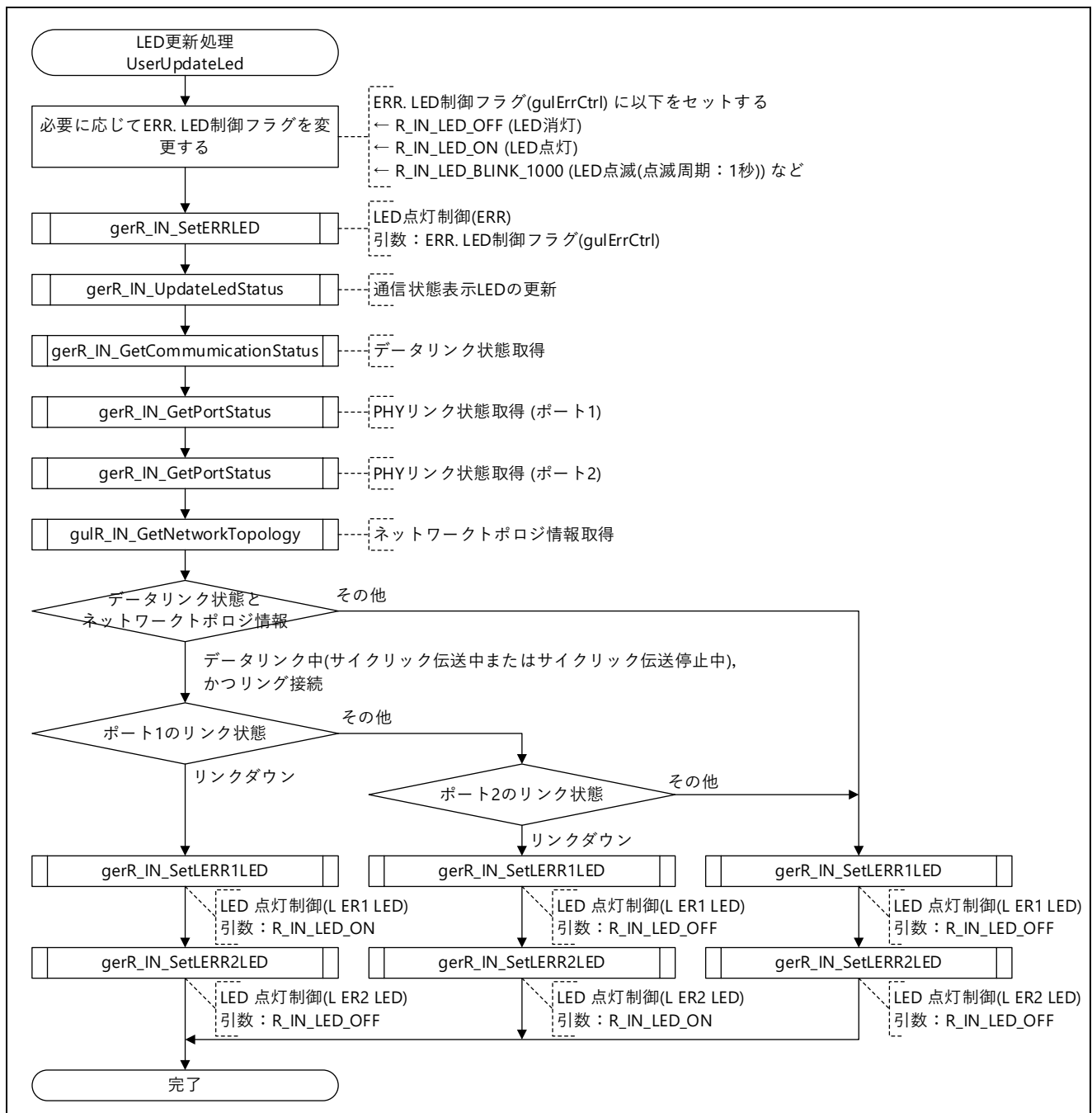


図 5.36 LED 更新処理フロー図



### 5.5.4 MIB(統計)情報取得処理

MIB(統計)情報を取得する処理を行います。

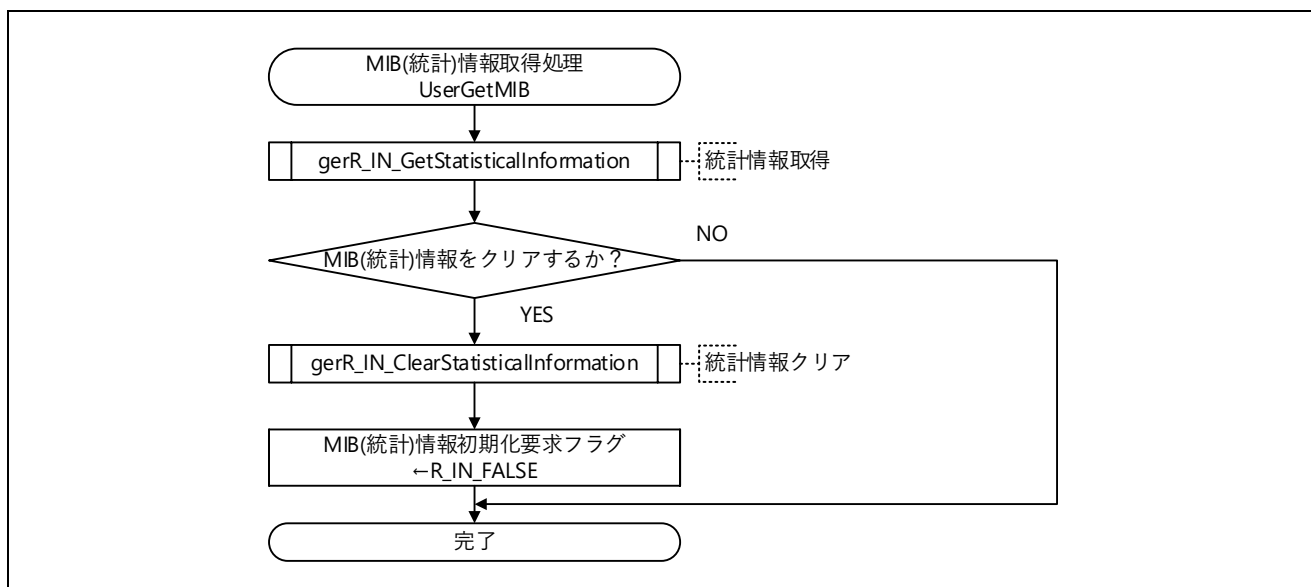


図 5.37 MIB(統計)情報取得処理フロー図

### 5.5.5 SLMP 受信処理

自局が受信した SLMP フレームを取得し、データを処理します。

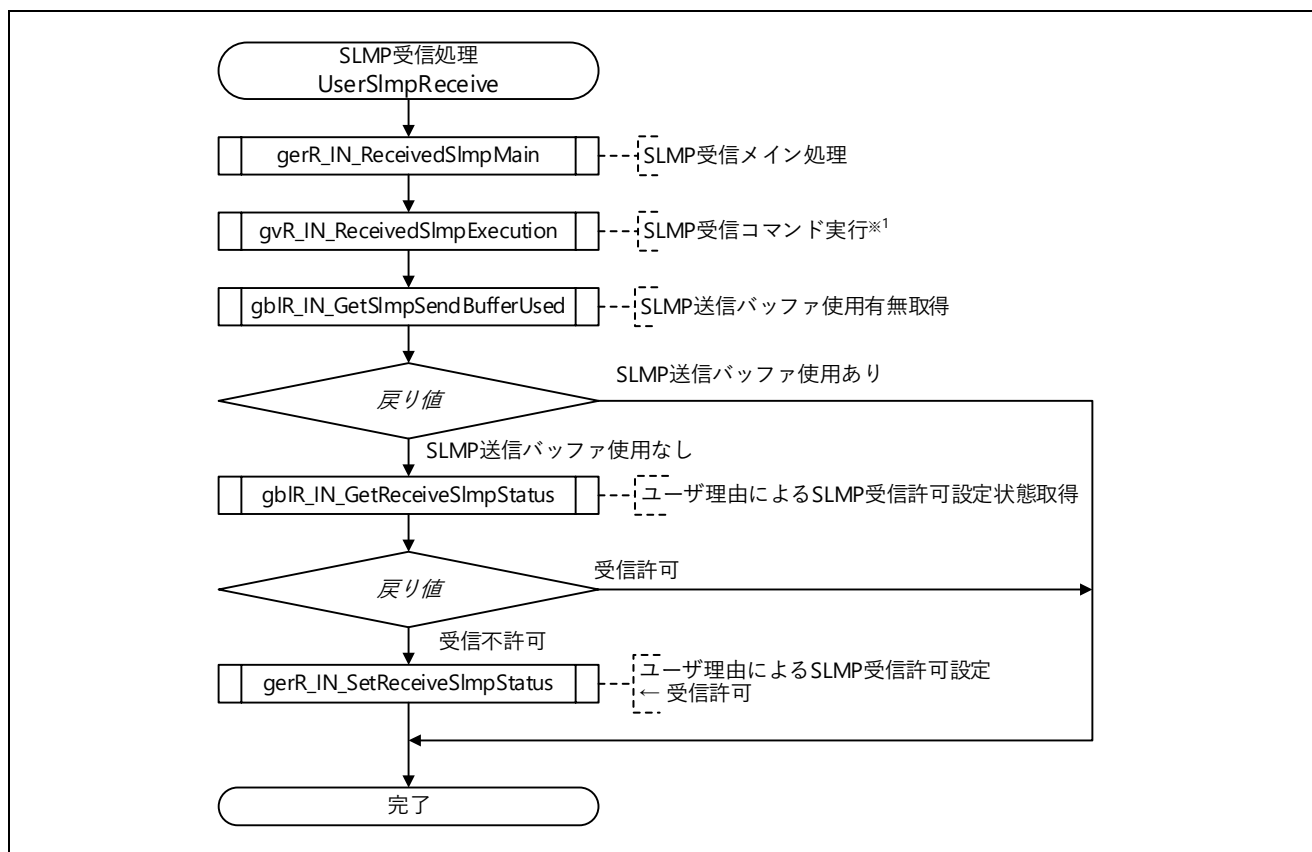


図 5.38 SLMP 受信処理フロー図

【注】 ※1 受信した SLMP フレームのコマンドが対応可能であれば、該当するコマンド受信処理関数を呼び出します。初期状態のサンプルコードで対応できるコマンドは「4.2.4 SLMP のコマンドについて」を参照してください。

### (1) 自局がサーバのとき(要求コマンドを受信するとき)

「4.2.4 SLMP のコマンドについて」の要求コマンド(他局が自局へ要求したコマンド)を受信したとき、該当するコマンド関数が要求コマンドに応じた処理(要求受信および応答送信処理)を行います。

「4.2.4 SLMP のコマンドについて」以外の SLMP 要求コマンドを受信する場合、任意の SLMP コマンド、SLMP サブコマンド、およびユーザ作成のコマンド受信処理関数を実行関数テーブル

「R\_IN\_SLMP\_FUNCTION\_REQUEST\_TBL\_T」(R\_IN32M4\_CL3\_Slmp\_Receive.c ファイル内)に追加してください。

### (2) 自局がクライアントのとき(応答コマンドを受信するとき)

「4.2.4 SLMP のコマンドについて」の応答コマンド(自局が他局へ要求したコマンドに対する応答)を受信したとき、該当するコマンド関数が応答コマンドに応じた処理(応答受信処理)を行います。

「4.2.4 SLMP のコマンドについて」以外の SLMP 応答コマンドを受信する場合、任意の SLMP コマンド、SLMP サブコマンド、およびユーザ作成のコマンド受信処理関数を実行関数テーブル

「R\_IN\_SLMP\_FUNCTION\_RESPONSE\_TBL\_T」(R\_IN32M4\_CL3\_Slmp\_Receive.c ファイル内)に追加してください。  
その上で、送信したいコマンドの要求フレーム作成処理を、「UserSlmpMakeRequest」(5.5.7 SLMP 要求フレーム作成処理)に追加してください。

なお、ST 型の応答フレームを受信した場合、その応答フレーム内にコマンドを特定する情報(シリアル番号、コマンド、サブコマンド)が無い場合、いずれのコマンドであっても受信処理を行う「UserSlmpReceive3EFrame」(5.5.8 SLMP ST(3E)応答フレーム受信処理)を呼び出します。

### 5.5.6 SLMP 送信処理

SLMP フレームの送信を行います。

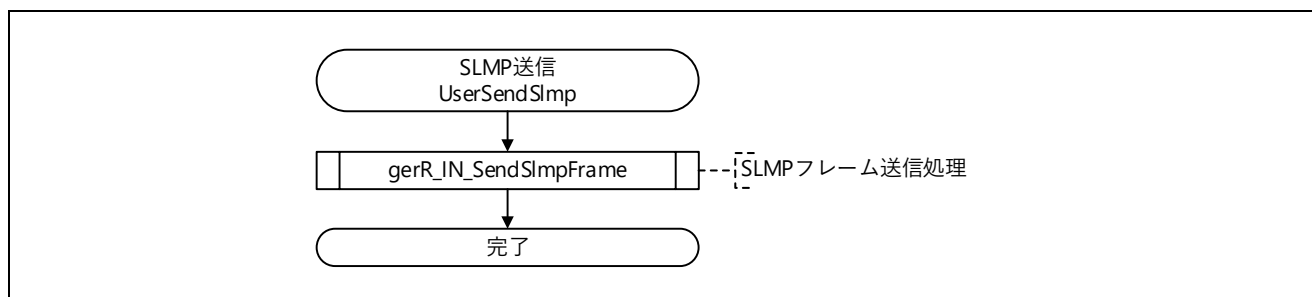


図 5.39 SLMP 送信処理フロー図

### 5.5.7 SLMP 要求フレーム作成処理

本処理は、自局がクライアントの場合に行います。

送信ありの SLMP コマンドがある場合、当該 SLMP コマンドの SLMP 要求フレーム作成処理関数を呼び出します。本サンプルコードは、例として SLMP メモリ読出し要求のフレーム作成処理を呼び出します。送信したい要求コマンドのフレーム作成処理を追加してください。

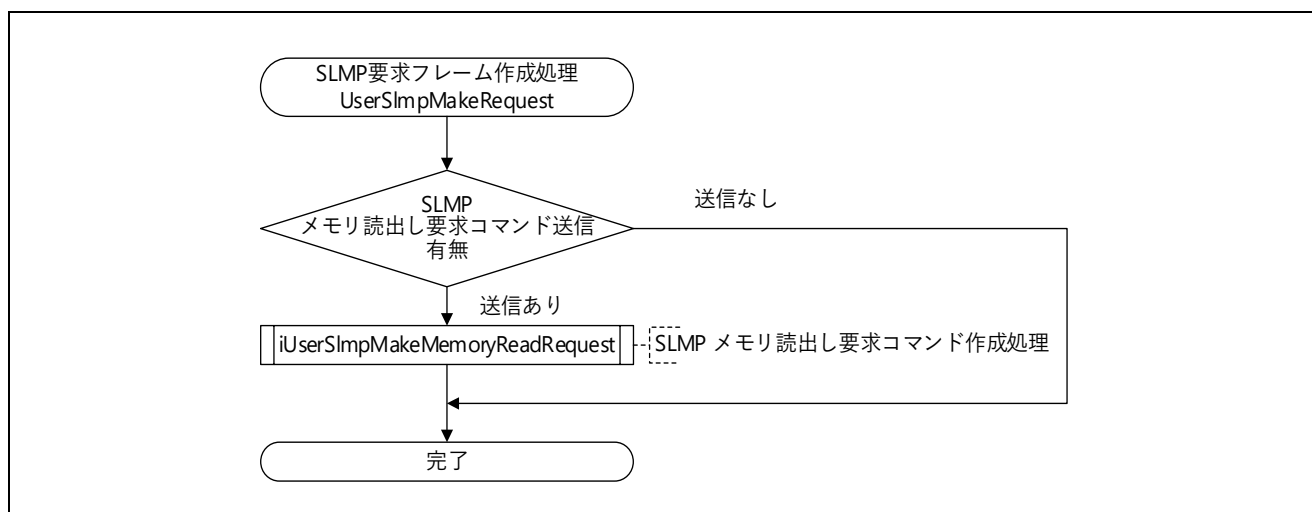


図 5.40 SLMP 要求フレーム作成処理フロー図

### 5.5.8 SLMP ST(3E)応答フレーム受信処理

本処理は、自局がクライアントの場合に行います。

SLMP ST(3E)フレームで他局へ要求し、その応答フレームを受信した際の共通受信処理を実行します。

ST(3E)応答フレームにはコマンドを特定する情報(シリアル番号、コマンド、サブコマンド)が無い場合、いずれのコマンドであっても本処理が呼び出されます。

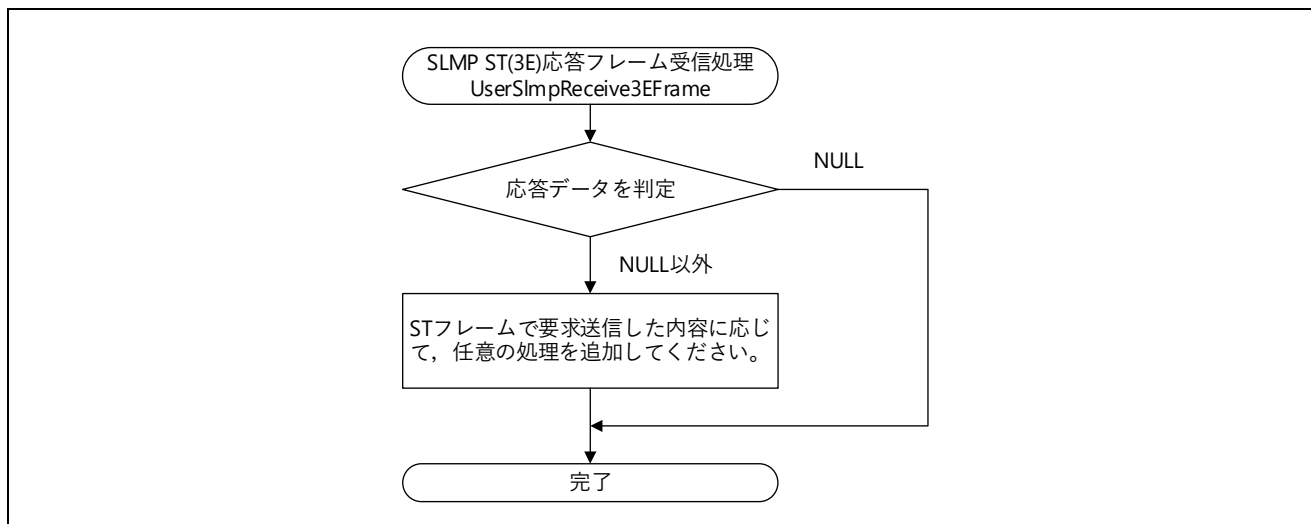


図 5.41 SLMP ST(3E)応答フレーム受信処理フロー図

### 5.5.9 R-IN32M4-CL3 ドライバ内検出 Fatal エラー確認処理

R-IN32M4-CL3 ドライバがユーザプログラムからの要求に関係なく検知した Fatal エラーについて検出状態を確認します。Fatal エラーが発生していれば、発生した Fatal エラーに対する処理を実行します。

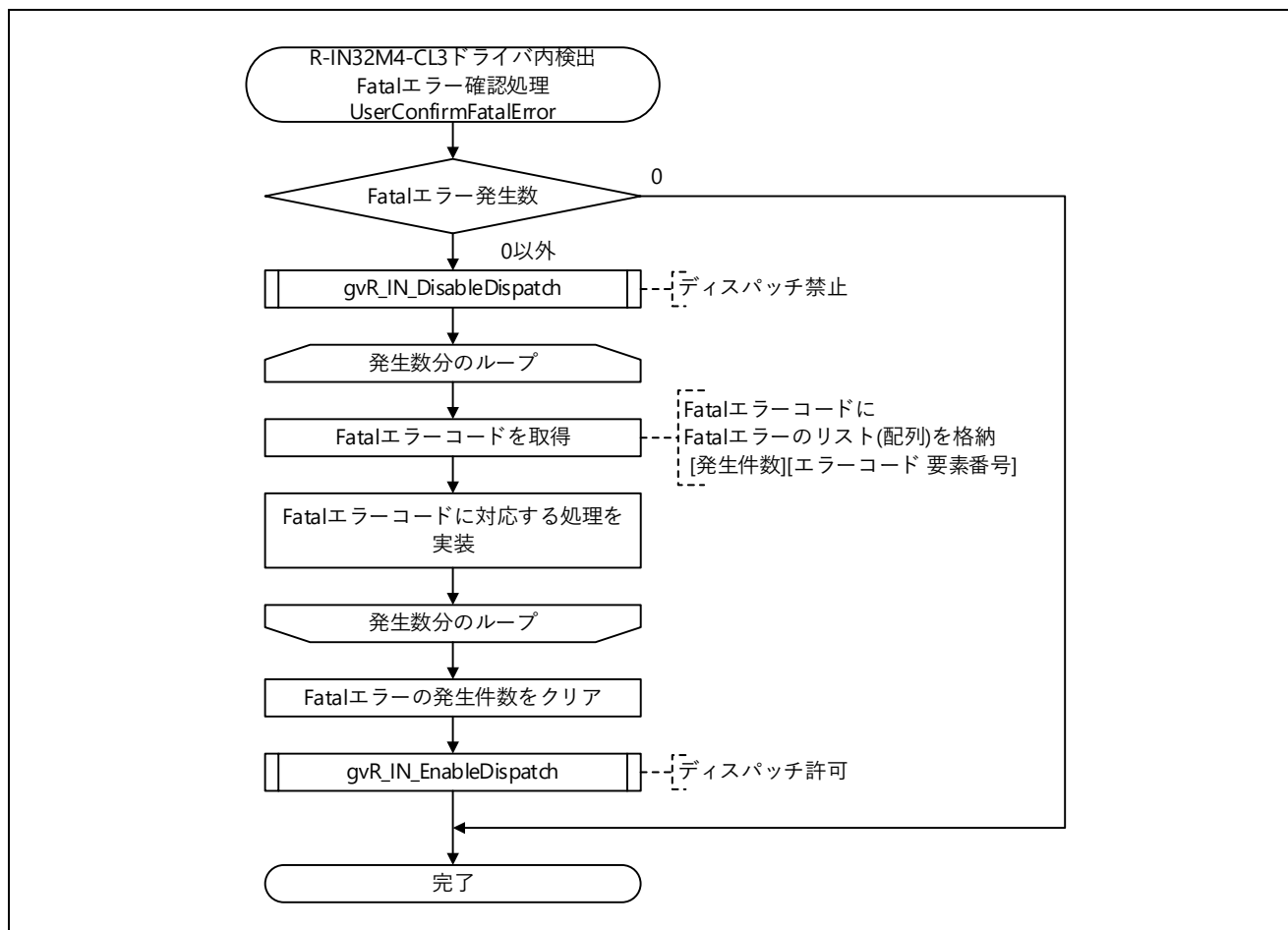


図 5.42 R-IN32M4-CL3 ドライバ内検出 Fatal エラー確認処理フロー図

表 5.21 Fatal エラーコード

Fatal エラーコード	値	内容
R_IN_FATALERROR_MDIOCOMMAND_TIMEOUT_ERROR	0000 D52AH	MDIO コマンド待ちエラー
R_IN_FATALERROR_LOOPBACKTEST_SEND_ERROR	0000 D530H	折り返しテスト時、送信エラー発生
R_IN_FATALERROR_LOOPBACKTEST_RECEIVE_FRAME_ERROR	0000 D531H	折り返しテスト時、FCS エラーフレーム受信
R_IN_FATALERROR_LOOPBACKTEST_RECEIVE_COUNT_ERROR	0000 D532H	折り返しテスト時、受信フレーム数異常
R_IN_FATALERROR_LOOPBACKTEST_RECEIVE_DISCARD_COUNT_ERROR	0000 D533H	折り返しテスト時、受信フレーム破棄あり
R_IN_FATALERROR_SYNCMODE_CHANGE	1000 0001H	同期モード変更

Fatal エラーの発生要因として、ユーザプログラムからの処理要求に対して Fatal エラーが発生するケースと、ユーザプログラムからの要求とは無関係に R-IN32M4-CL3 ドライバ内部で発生するケースがあります。

(1) ユーザプログラムからの要求により Fatal エラーが発生するケース

アイドルタスク内の「初期化处理」中の、通信開始処理、H/W テストなど。

この場合、それぞれ関数呼出し後に Fatal エラーの発生有無を確認します。

(2) ユーザプログラムからの要求とは無関係に R-IN32M4-CL3 ドライバ内部で発生するケース

アイドルタスク内の「状態管理・トランジェントメイン処理」中の、例えば、マスタ局からの SLMP 要求伝文が不正であった場合など。

この場合、メイン処理の最後で本処理(UserConfirmFatalError)を実行することで、R-IN32M4-CL3 ドライバからの Fatal エラーの発生有無を確認します。

## 5.6 ユーザプログラム詳細(SLMP コマンド実行関連)

### 5.6.1 SLMP メモリ読出し要求コマンド受信処理

SLMP デュアルポートメモリ一括読出し要求フレーム(SLMP コマンド : 0613H)を受信したとき、指定された自局バッファメモリのデータを読み出し、応答フレームに設定します。

本コマンド受信処理は、LMT / MT / ST 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

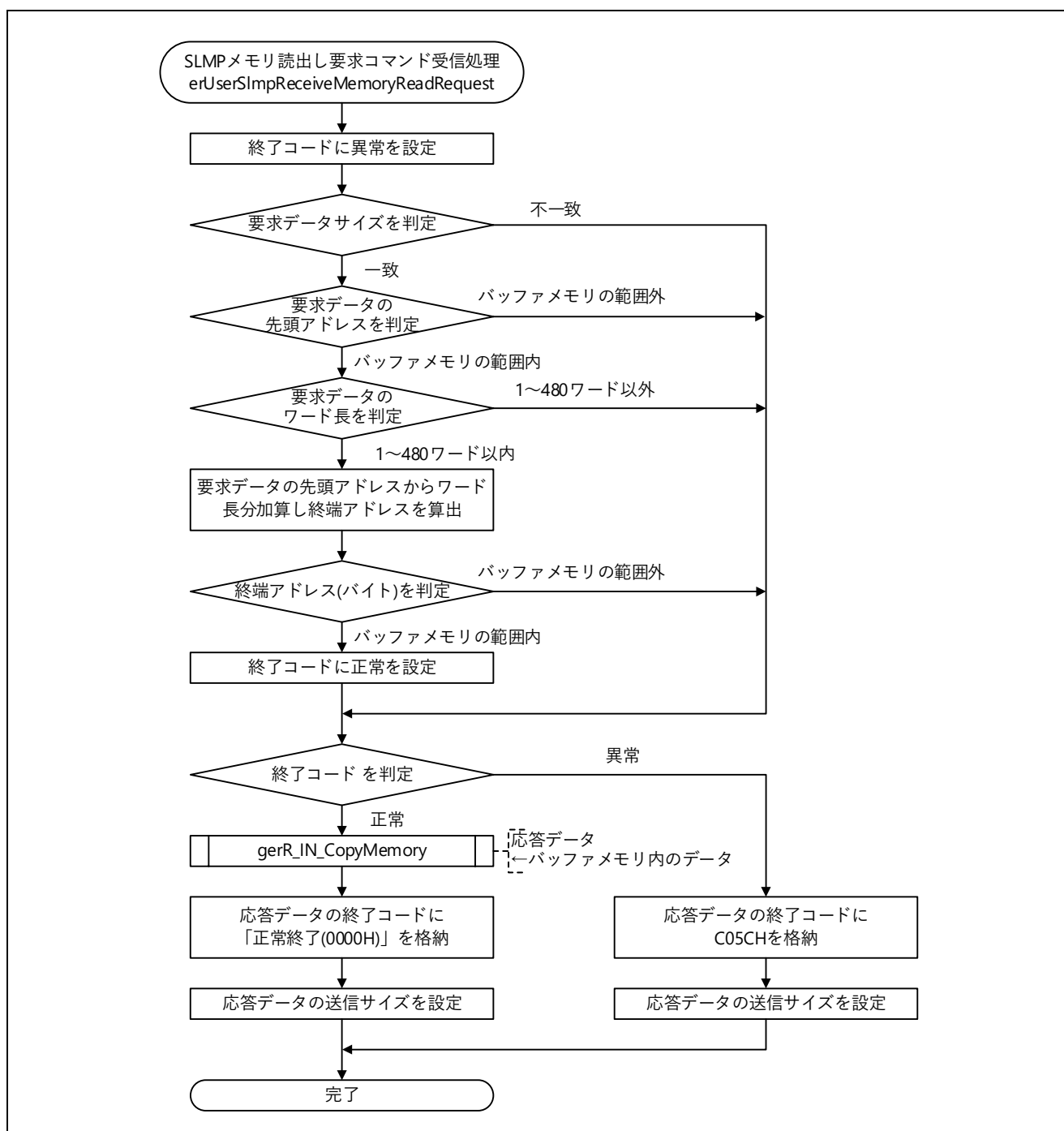


図 5.43 SLMP メモリ読出し要求コマンド受信処理フロー図



## 5.6.2 SLMP メモリ書き込み要求コマンド受信処理

SLMP デュアルポートメモリー括書き込み要求フレーム(SLMP コマンド : 1613H)を受信したとき、指定された自局バッファメモリにデータを書き込み、応答フレームを送信します。

本コマンド受信処理は、LMT / MT / ST 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

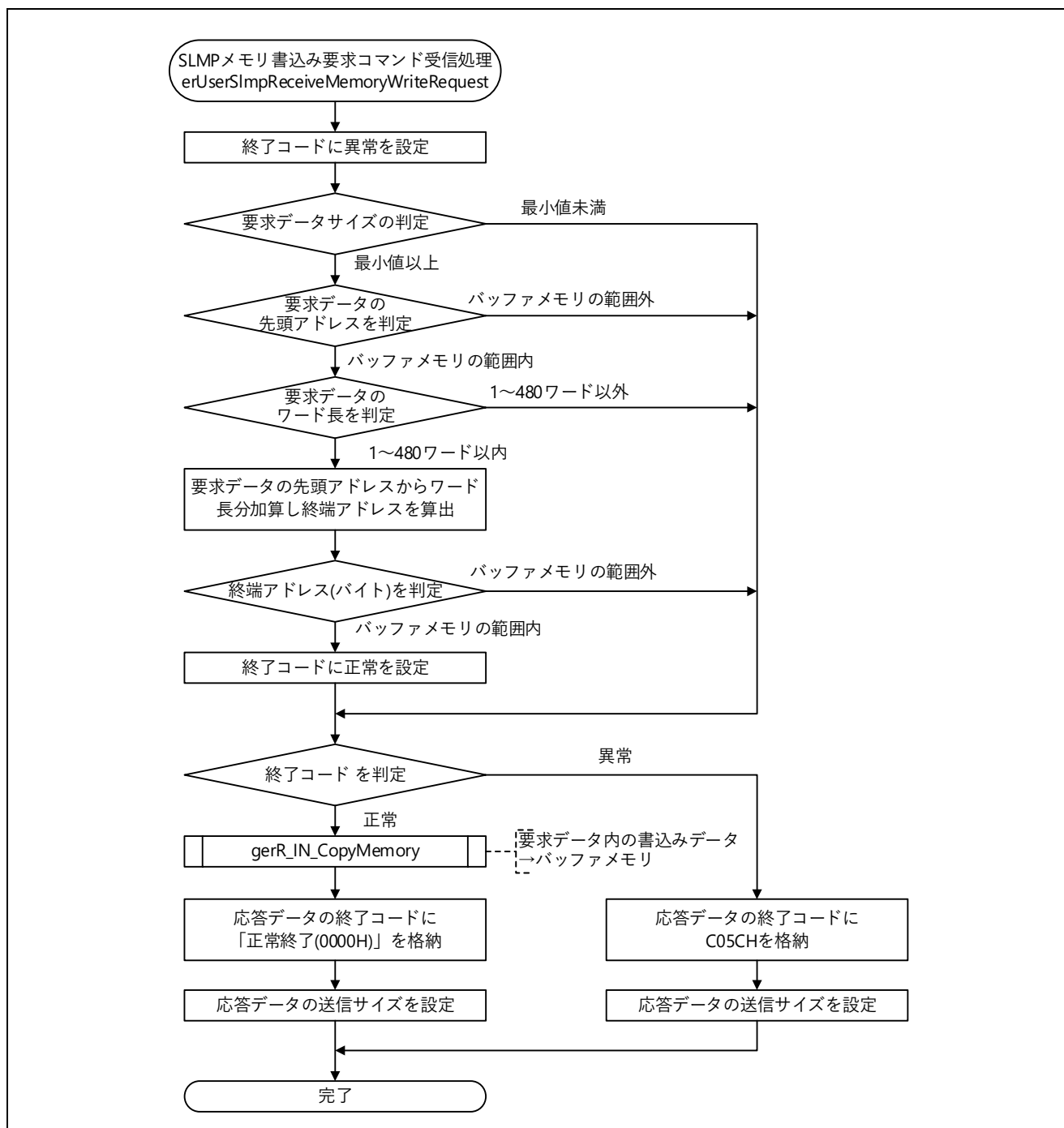


図 5.44 SLMP メモリ書き込み要求コマンド受信処理フロー図

### 5.6.3 SLMP メモリ読出し要求コマンド作成処理

SLMP デュアルポートメモリー括読出しコマンド(SLMP コマンド : 0613H)の要求フレームを作成します。

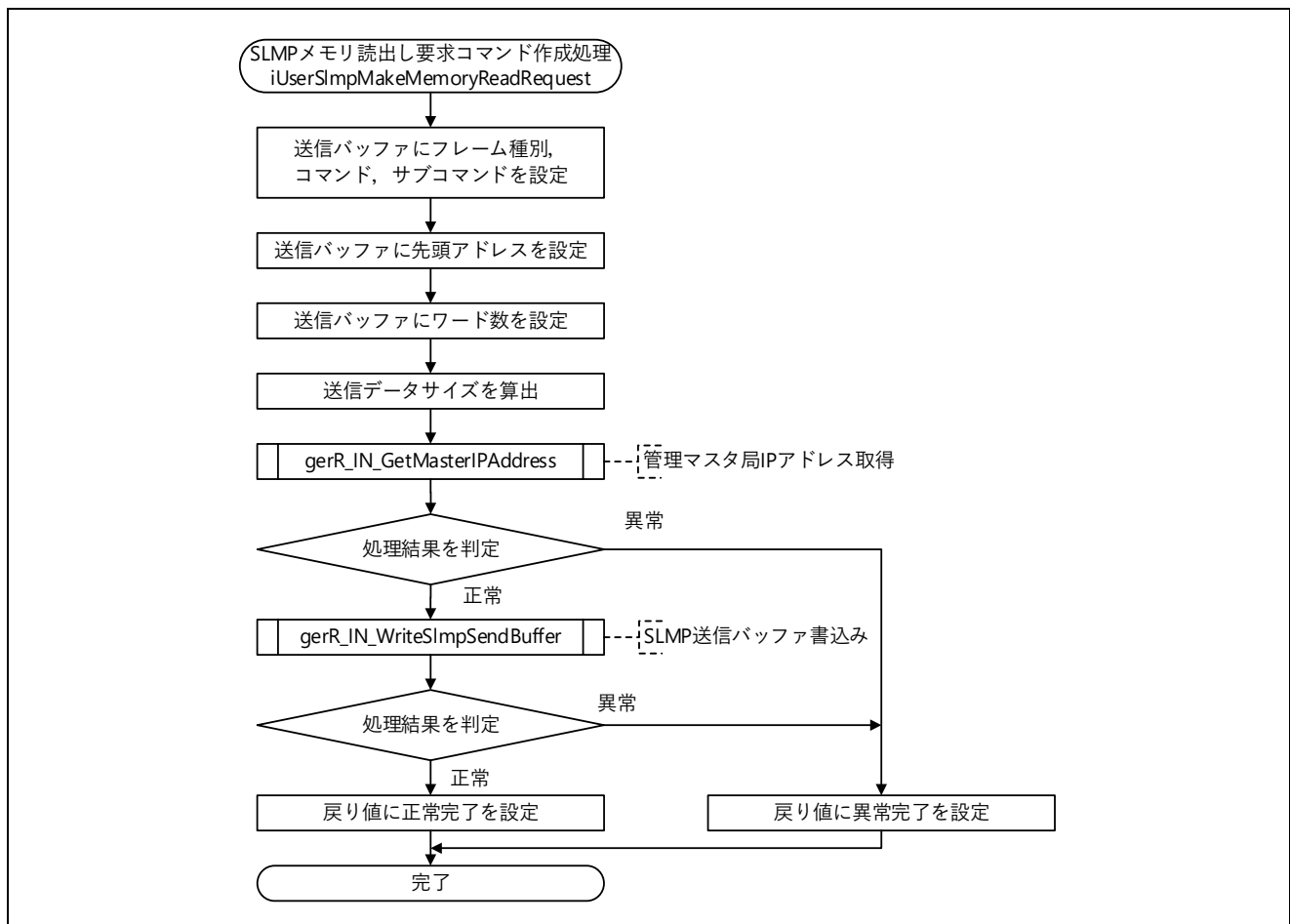


図 5.45 SLMP メモリ読出し要求コマンド作成処理フロー図

要求フレームの構成を示します。各メンバの詳細は、CC-Link 協会「SLMP 仕様書」または関連マニュアル「SLMP リファレンスマニュアル」を参照してください。

表 5.22 R\_IN\_SLMP\_6E\_FRAME\_REQUEST\_T

No	メンバ		内容
1	R_IN_SLMP_6E_FRAME_HEAD_T	stSlmpHead	SLMP ヘッダ情報(表 5.23)
2	USHORT	usExtendedStationNumber	要求先局拡大局番号
3	USHORT	usByteSize	要求データ長
4	USHORT	usTimer	タイマ
5	USHORT	usCommand	コマンド
6	USHORT	usSubCommand	サブコマンド
7	UCHAR	uchReserve	予約
8	UCHAR	uchDataId	伝文識別値
9	USHORT	usDataDevideNum	総分割数
10	USHORT	usDataNumber	分割番号
11	ULONG	aulDataArea[361]	要求データ[Dword サイズ]

表 5.23 R\_IN\_SLMP\_6E\_FRAME\_HEAD\_T

No	メンバ		内容
1	USHORT	usFrameType	フレーム種別
2	USHORT	usSerialNumber	シリアル番号
3	USHORT	usReserve1	予約
4	UCHAR	uchNetworkNumber	要求先局ネットワーク番号
5	UCHAR	uchStationNumber	要求先局局番号
6	USHORT	usProcessorNumber	要求先局プロセッサ番号
7	UCHAR	uchMultiDropNo	要求先局プロセッササブ番号
8	UCHAR	uchReserve3	予約

#### 5.6.4 SLMP メモリ読出し応答コマンド受信処理

SLMP デュアルポートメモリー括読出し応答フレーム (SLMP コマンド : 0613H)の受信処理を行います。

自局が他局へ要求を送信した後の、応答を受信します。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームのサブコマンド以降のデータ長異常を検出した場合、受信データを破棄し、本コマンド受信処理は実行されません。

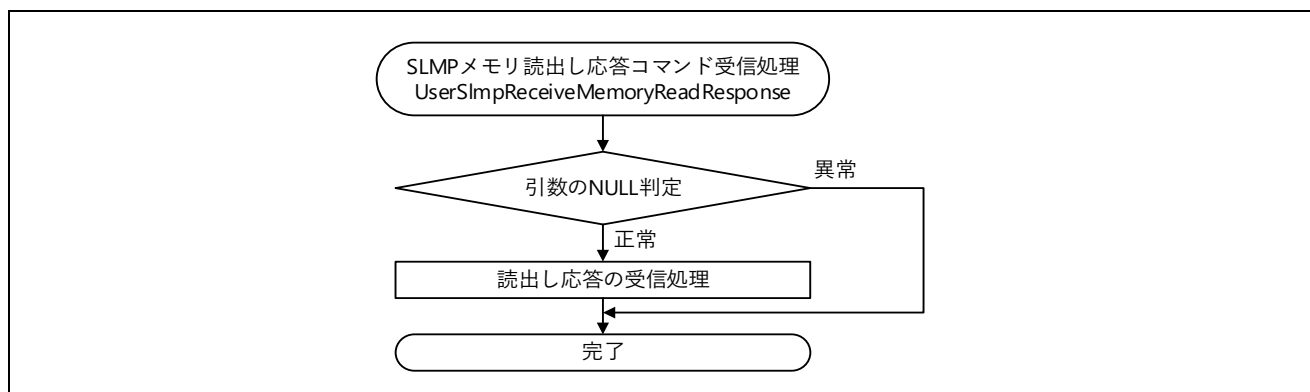


図 5.46 SLMP メモリ読出し応答コマンド受信処理フロー図

### 5.6.5 SLMP リモートリセット要求コマンド受信処理

SLMP リモートリセット要求フレーム(SLMP コマンド：1006H)を受信したとき、自局(R-IN32M4-CL3)をリセットします。

本コマンド受信処理は、LMT / MT / ST 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答し、本コマンド受信処理は実行されません。

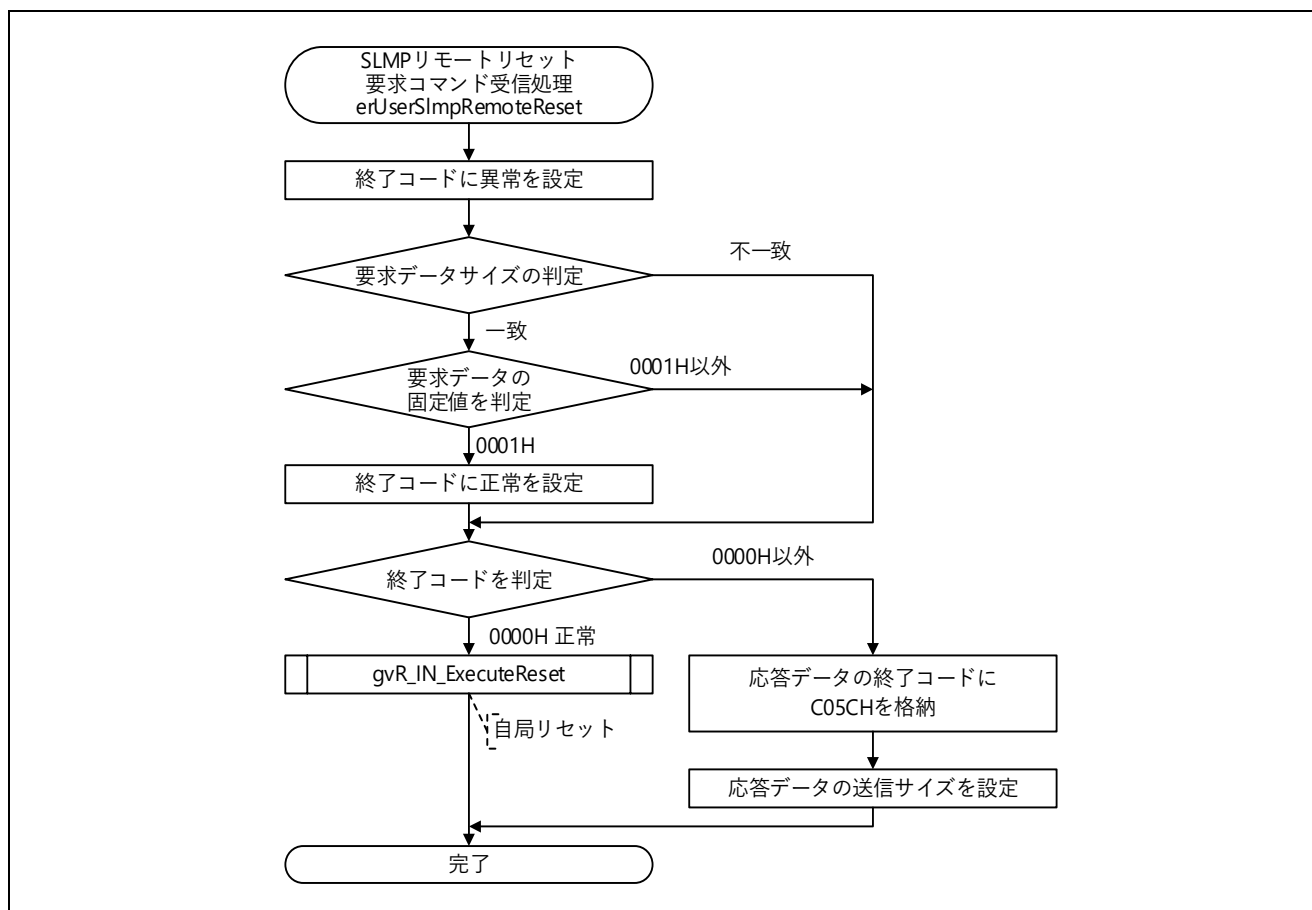


図 5.47 SLMP リモートリセット要求受信処理フロー図

## 5.6.6 SLMP インディケータ表示要求コマンド受信処理

SLMP より指定された表示指示に従って USER LED の点滅/消灯を行います。

本コマンド受信処理は、LMT / MT / ST 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが要求データ(受信)付属情報の SLMP 受信結果にエラーを設定します。

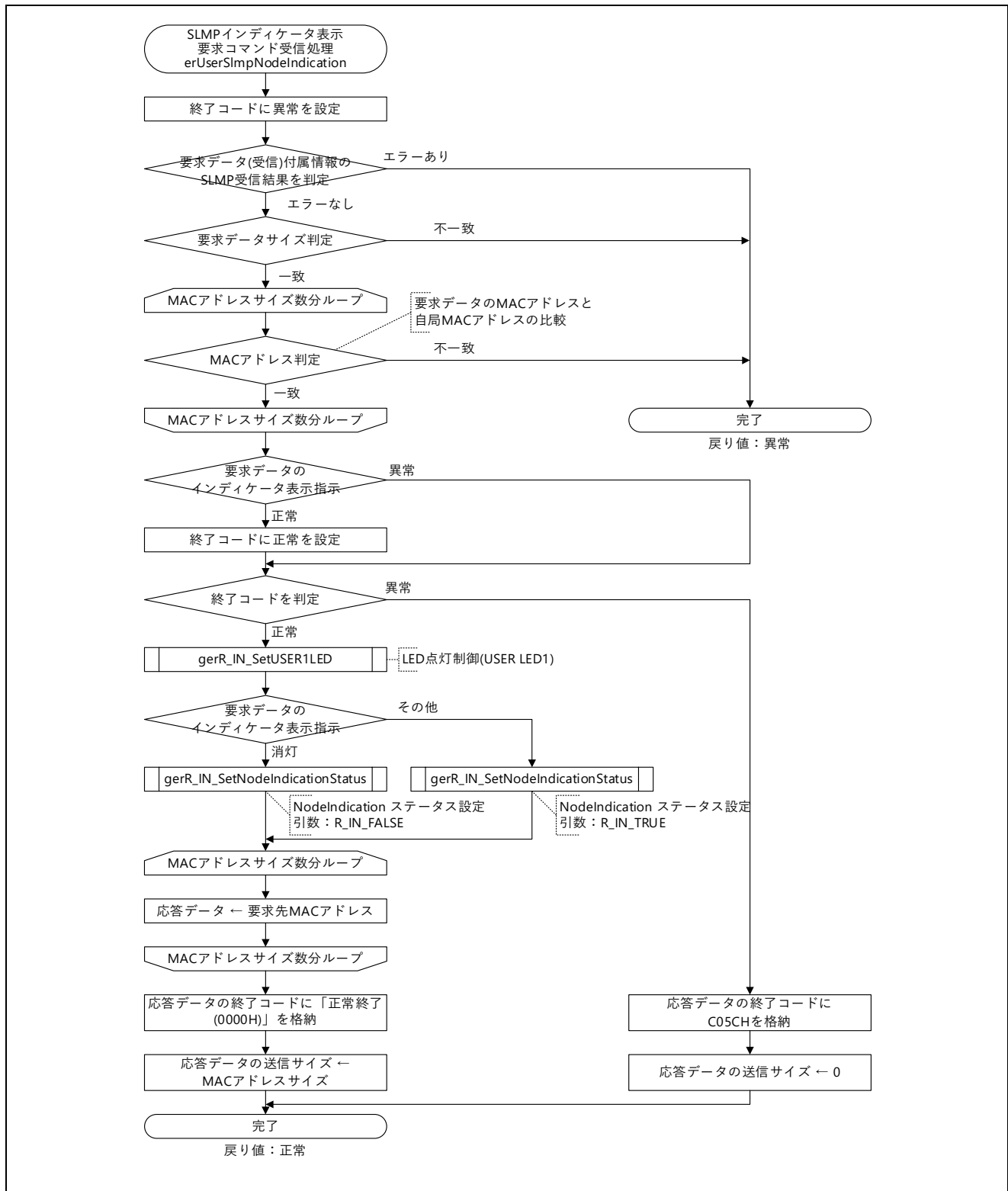


図 5.48 SLMP インディケータ表示要求コマンド受信処理フロー図

### 5.6.7 SLMP IP アドレス変更要求コマンド受信処理

SLMP より指定されたサーバの IP アドレスを取得します。

本コマンド受信処理は、LMT / MT / ST 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが要求データ(受信)付属情報の SLMP 受信結果にエラーを設定します。

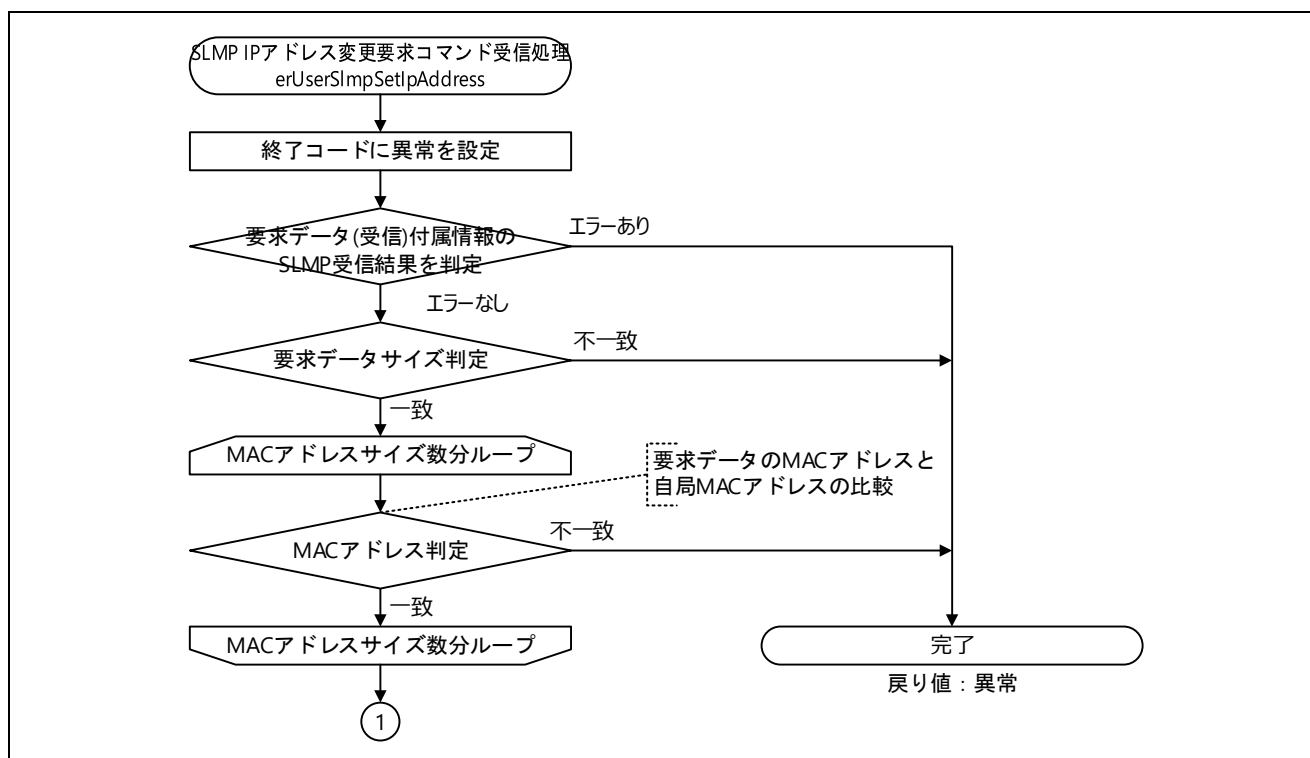


図 5.49 SLMP IP アドレス変更要求コマンド受信処理フロー図 (1/2)

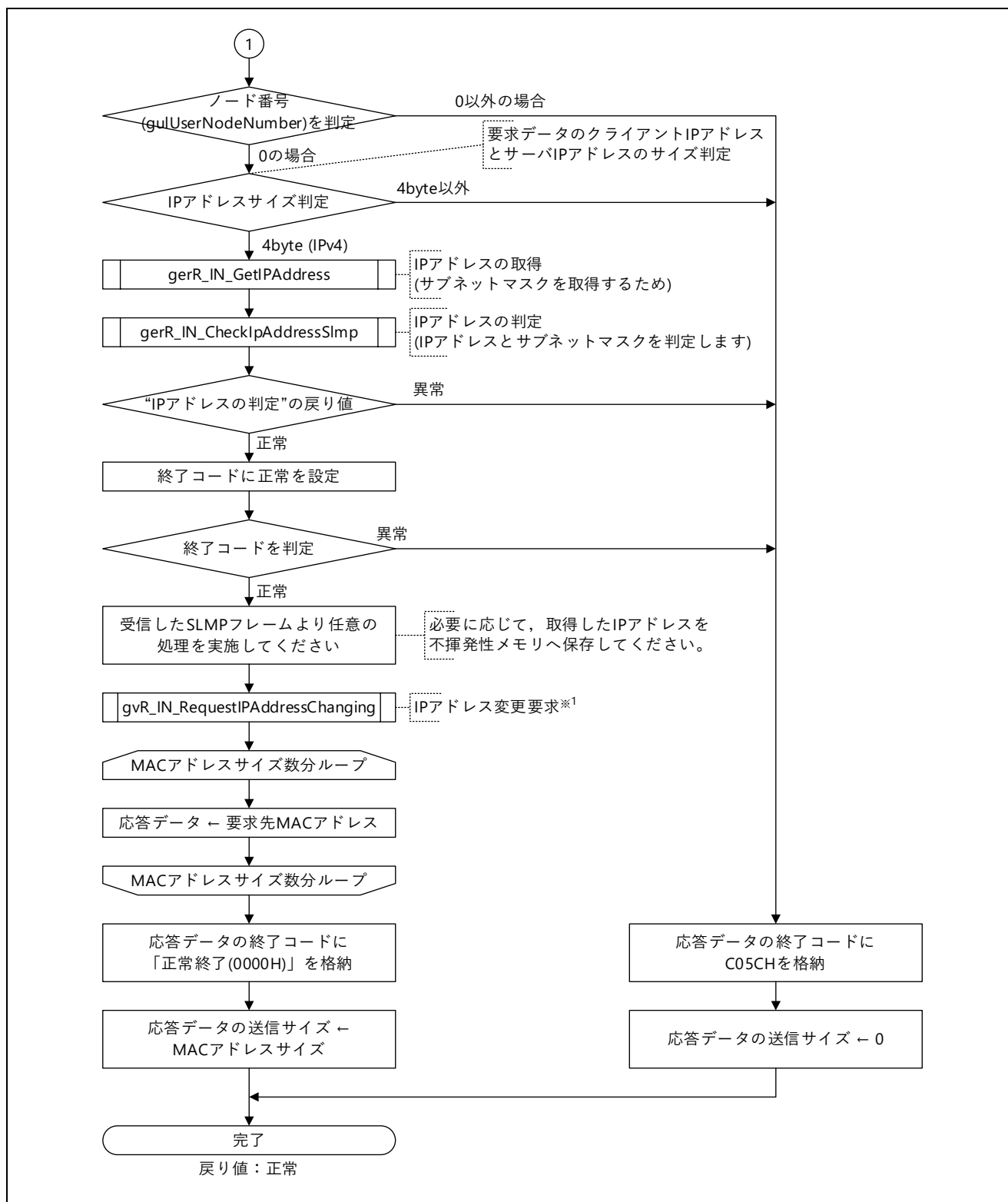


図 5.50 SLMP IP アドレス変更要求コマンド受信処理フロー図 (2/2)

【注】 ※1 IP アドレスの変更結果がコールバック関数より通知されます。

変更結果の詳細は「gvR\_IN\_CallbackIPAddressChangingResult」(6.6(10))を参照してください。

デバイス局の IP アドレス設定における一連の処理の流れを示します。

(IP アドレス第 4 オクテットを 01H から 03H に変更する場合)

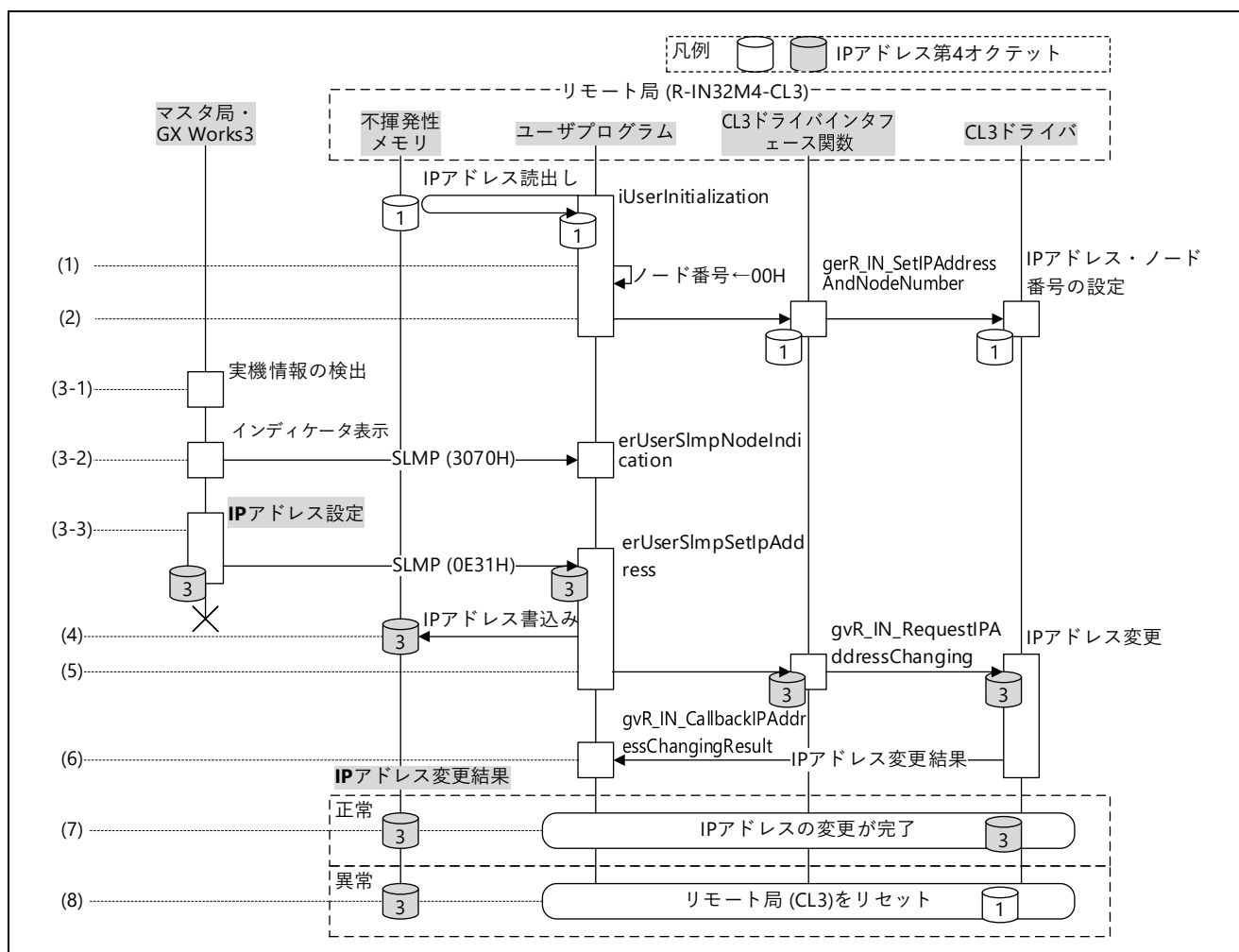


図 5.51 デバイス局の IP アドレス設定における処理の流れ



No	内容	関連
(1)	ユーザは任意の方法でリモート局(R-IN32M4-CL3)のノード番号(gulUserNodeNumber)を 00H に設定してください。リモート局(R-IN32M4-CL3)は、電源 ON 時にノード番号が 00H であれば、デバイス局の IP アドレス設定を実施できます。	5.3.1
(2)	不揮発性メモリから読み出した IP アドレス、あるいは任意の IP アドレスを gerR_IN_SetIPAddressAndNodeNumber の引数にセットし、R-IN32M4-CL3 起動時の IP アドレスを設定します。	6.4.1(4)
(3)	<p>ユーザは GX Works3 の CC-Link IE TSN 構成ウィンドウにて、“IP アドレスの設定”ウィンドウを表示させ 1～3 を実施してください。</p> <p>1.実機情報の検出 マスタ局に接続している局が IP アドレスの設定ウィンドウに表示されます。※1</p> <p>2.インディケータ表示 (IP アドレス設定の対象機器を目視確認するために実施します。目視確認が不要な場合は不要です。) マスタ局が、IP アドレス設定の対象局に SLMP 要求(3070H：インディケータ表示)を送信します。 ※2 IP アドレス設定の対象局は USER LED が点滅します。</p> <p>3.IP アドレス設定 マスタ局が、IP アドレス設定の対象局に SLMP 要求(0E31H：接続機器の IP アドレス設定)を送信します。</p>	-
(4)	erUserSimpSetIpAddress 内のユーザ作成処理が、不揮発性メモリに IP アドレスを保存します。	5.6.7
(5)	gvR_IN_RequestIPAddressChanging が、R-IN32M4-CL3 ドライバへ IP アドレス変更要求を行います。	6.4.11(8)
(6)	gvR_IN_CallbackIPAddressChangingResult が、IP アドレス変更の結果を通知します。	6.6(10)
(7)	リモート局(R-IN32M4-CL3)がマスタ局と通信をまだ開始していないとき、変更結果は正常完了になります。マスタ局から送られてきた IP アドレスが R-IN32M4-CL3 ドライバに反映されます。(図 5.51 の場合、IP アドレス第 4 オクテットは” 3” が設定されます。)	-
(8)	<p>リモート局(R-IN32M4-CL3)がマスタ局とデータリンクを既に開始しているとき、変更結果は異常完了になります。</p> <p>初期化処理(図中の(2))で設定した IP アドレスを通信に使用しているため、マスタ局から送られてきた IP アドレスは R-IN32M4-CL3 ドライバに反映されません。(図 5.51 の場合、IP アドレス第 4 オクテットは” 1” が維持されます。)</p> <p>IP アドレスを変更するためには、リモート局(R-IN32M4-CL3)を電源 OFF→ON などでリセットしてください。</p> <p>リセット後のリモート局(R-IN32M4-CL3)は、初期化処理(図中の(2))にて不揮発性メモリから読み出した IP アドレスを gerR_IN_SetIPAddressAndNodeNumber の引数にセットし、IP アドレスを R-IN32M4-CL3 ドライバに設定します。</p>	-

【注】 ※1 実機情報の検出では増設ユニットは表示されません。

※2 受信処理の詳細は「5.6.6 SLMP インディケータ表示要求コマンド受信処理」を参照してください。

## 5.6.8 SLMP エラー履歴クリア要求コマンド受信処理

エラー履歴をすべてクリアします。

増設ユニット/スライス I/O 装着し、MIB 現在エラー情報のオプション情報を使用する場合(コンパイルスイッチ「CURERR\_OPTIONINFO\_ENABLE」を有効にした場合)は、要求データ(受信)付属情報構造体の要求先局プロセッサ番号をチェックし、コントローラ情報のエラー履歴、またはオプション情報のエラー履歴をクリアします。

本コマンド受信処理は、LMT / MT / ST 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

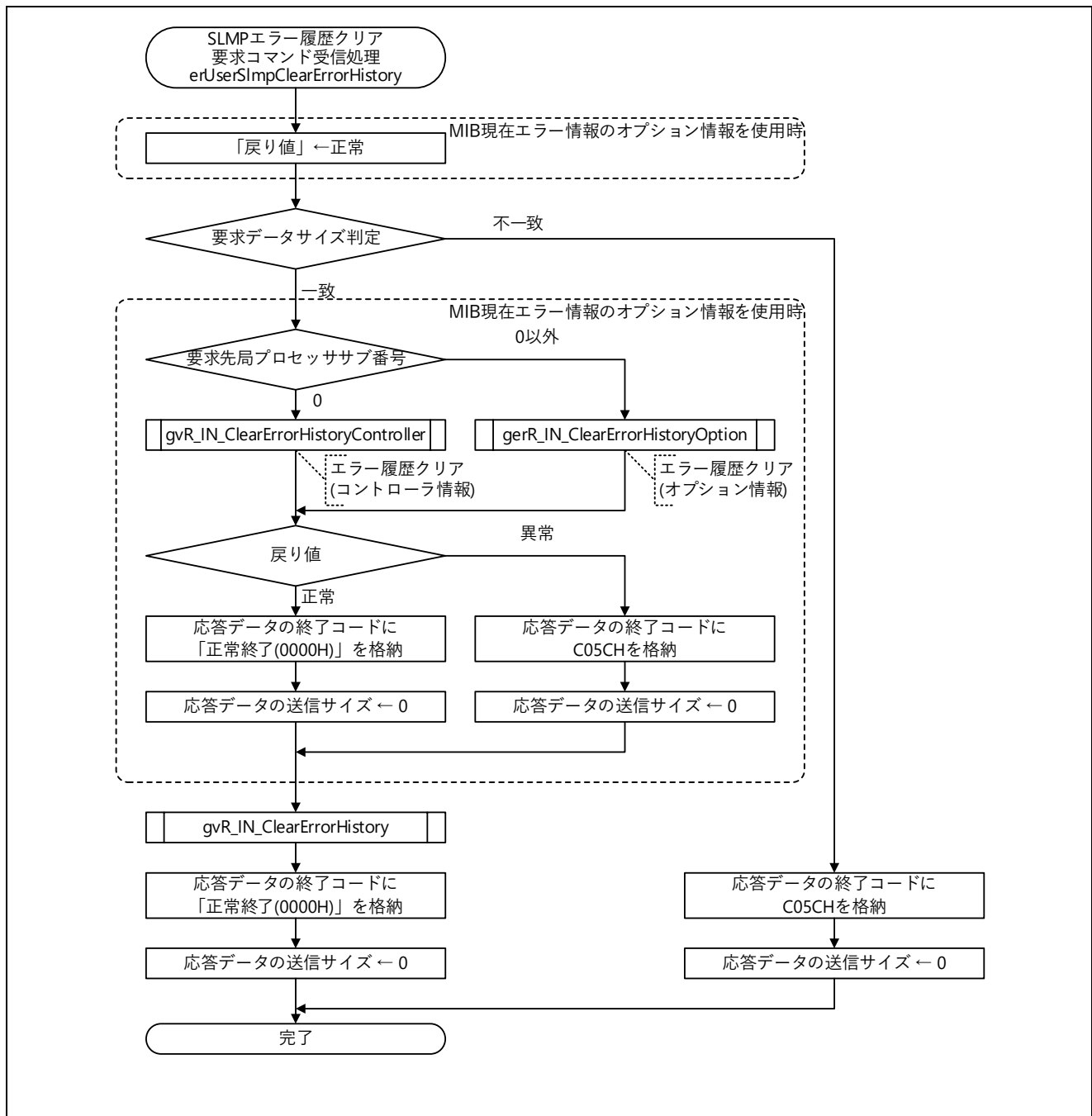


図 5.52 SLMP エラー履歴クリア要求コマンド受信処理フロー図

### 5.6.9 SLMP ネットワーク時刻オフセット配信コマンド受信処理

ネットワーク時刻のオフセットを時計機能へ設定します。

本コマンド受信処理は、LMT / MT / ST 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが要求データ(受信)付属情報の SLMP 受信結果にエラーを設定します。

なお、本コマンドの応答送信は不要です。応答処理を行わないよう、正常完了の場合でも戻り値に「異常(データがない)」を設定します。

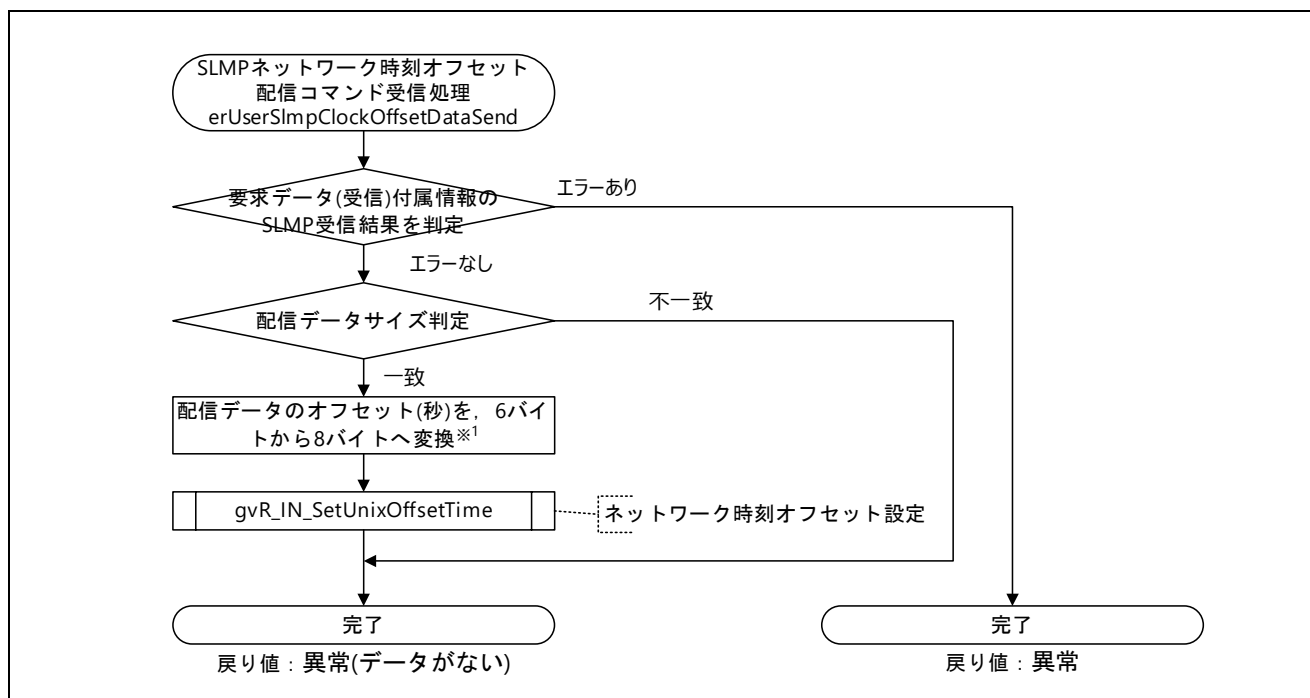


図 5.53 SLMP ネットワーク時刻オフセット配信コマンド受信処理フロー図

【注】※1 配信データのオフセット(秒)は、符号付き 6byte データです。6byte→8byte 変換の際、6byte 側の最上位 bit を判定し、負の値であれば、変換先の 8byte の内、上位 2byte のエリアにオール F へ設定する。

### 5.6.10 SLMP ネットワーク時刻配信コマンド受信処理

本処理は、CC Link IE TSN Class A の場合のみ使用できます。

ネットワーク時刻を時計機能へ設定します。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが要求データ(受信)付属情報の SLMP 受信結果にエラーを設定します。

なお、本コマンドの応答送信は不要です。応答処理を行わないよう、正常完了の場合でも戻り値に「異常(データがない)」を設定します。

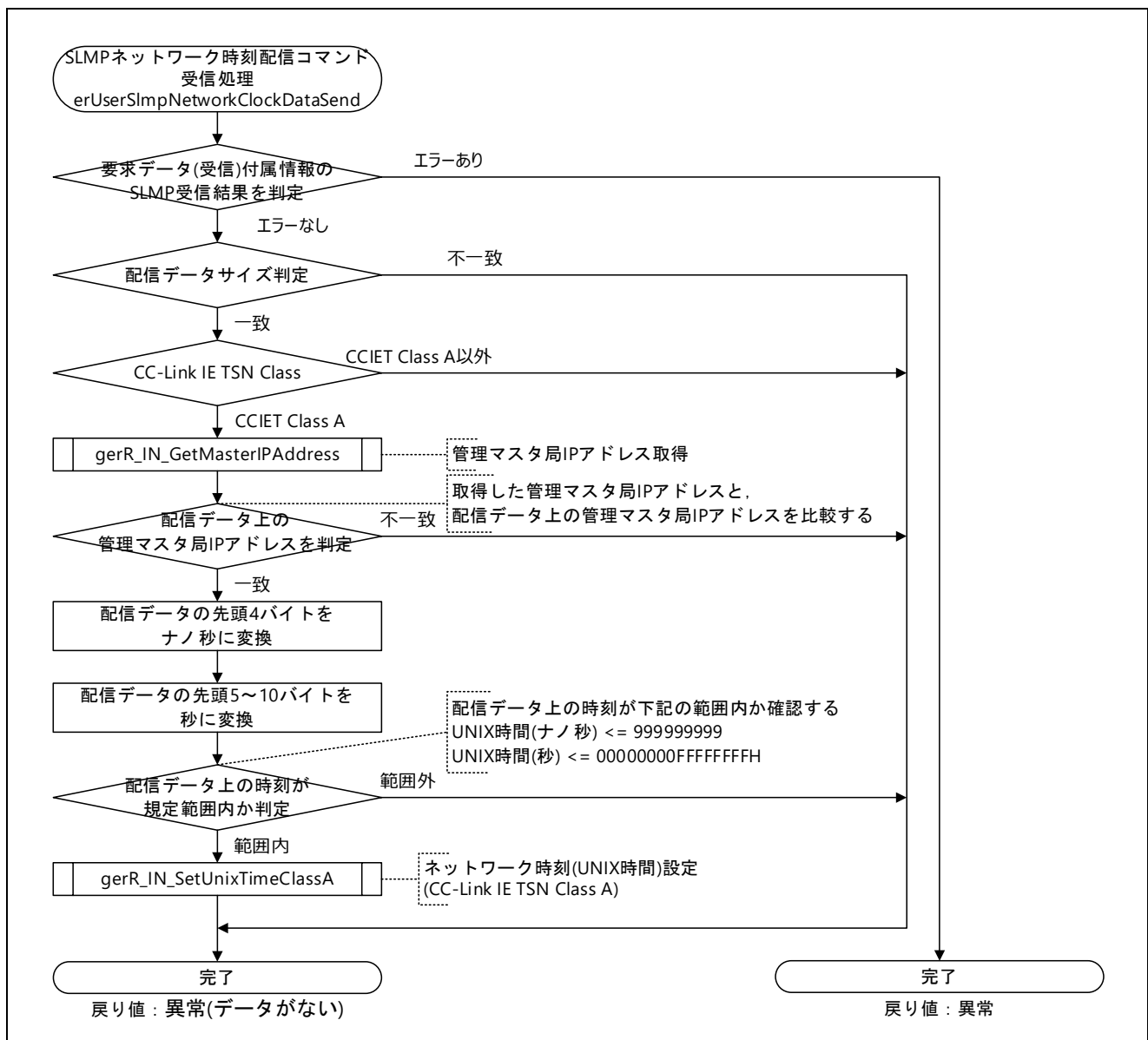


図 5.54 SLMP ネットワーク時刻配信コマンド受信処理フロー図

### 5.6.11 SLMP ウォッチドッグカウンタ情報設定要求コマンド受信処理

SLMP ウォッチドッグカウンタ設定コマンド(3210H)の要求伝文で指定されたウォッチドッグカウンタチェック異常連続カウンタ閾値を R-IN32M4-CL3 に設定し、応答伝文でウォッチドッグカウンタのオフセット情報などを返します。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません

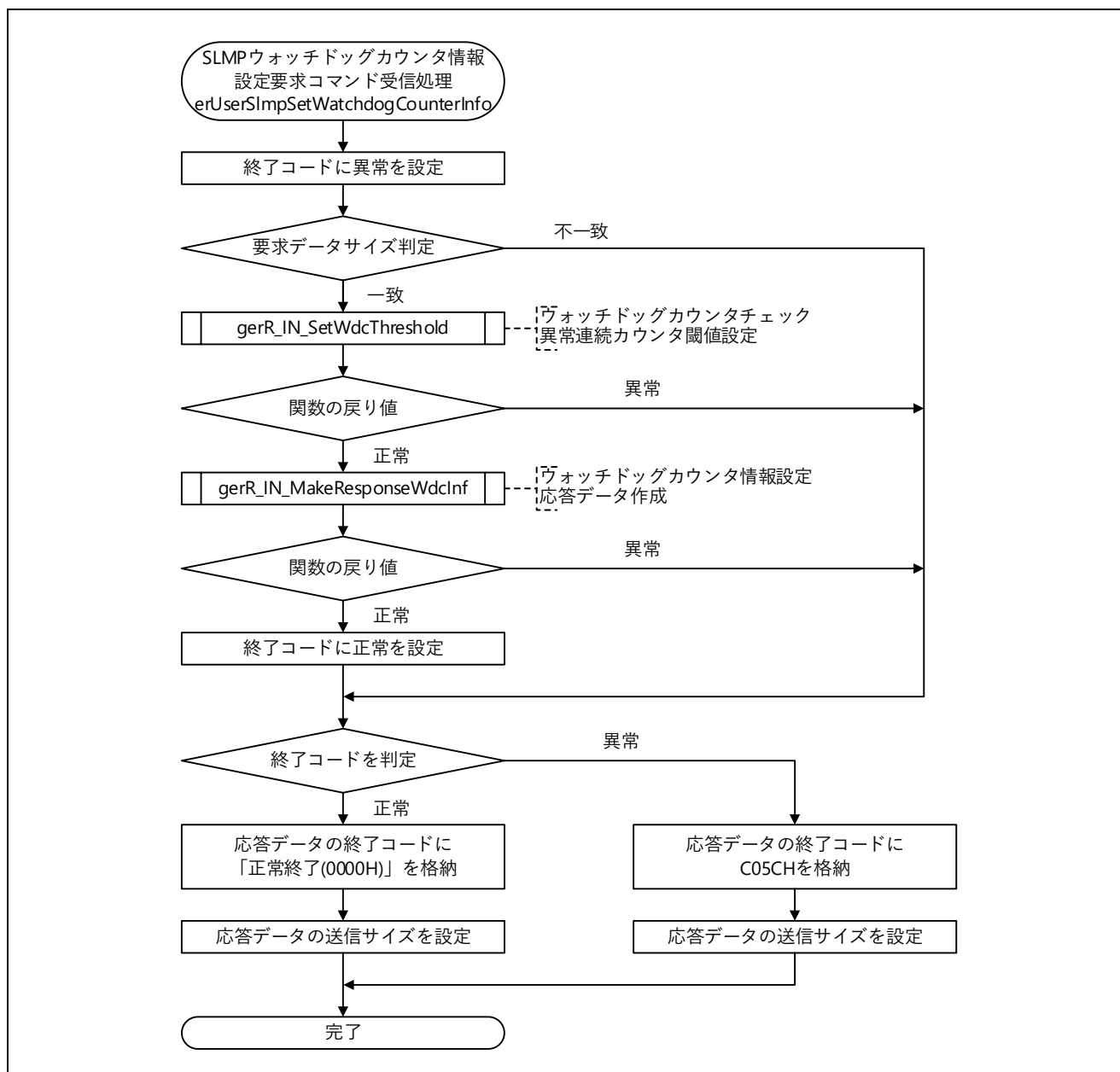


図 5.55 SLMP ウォッチドッグカウンタ情報設定要求コマンド受信処理フロー図

### 5.6.12 SLMP SetCommunityName 要求コマンド受信処理

SLMP SetCommunityName コマンド(3084H)の要求伝文で指定された SNMP のコミュニティ名を R-IN32M4-CL3 に設定します。本機能を有効にするには、コンパイルスイッチで「SNMP\_COMMUNITY\_NAME\_SETTING\_ENABLE」を定義してください。

本コマンド受信処理は、管理マスタ局から受信したデータに対して以下のチェックを行います。

No	内容
1	データリンク中か?
2	SLMP 受信結果が正常か?
3	要求データが SetCommunityName 要求コマンドに必要なサイズ以上か?
4	送信元が管理マスタ局か?
5	コミュニティ名長が 0 より大きいか?
6	コミュニティ名に使用されている文字列が使用可能文字(半角英数字(大文字、小文字を区別)とアンダースコア“_”)か?

チェック結果が異常の場合、戻り値に「R\_IN\_ERR」を設定し、応答を返しません。

チェック結果が正常の場合、管理マスタ局から送信された SNMP コミュニティ名を、R-IN32M4-CL3 ドライバインタフェース関数「gerR\_IN\_SetSNMPCommunityName」SNMP コミュニティ名設定を使用して、R-IN32M4-CL3 に設定します。

なお、本コマンドの応答送信は不要です。応答処理を行わないよう、正常完了の場合でも戻り値に「異常(データがない)」を設定します。

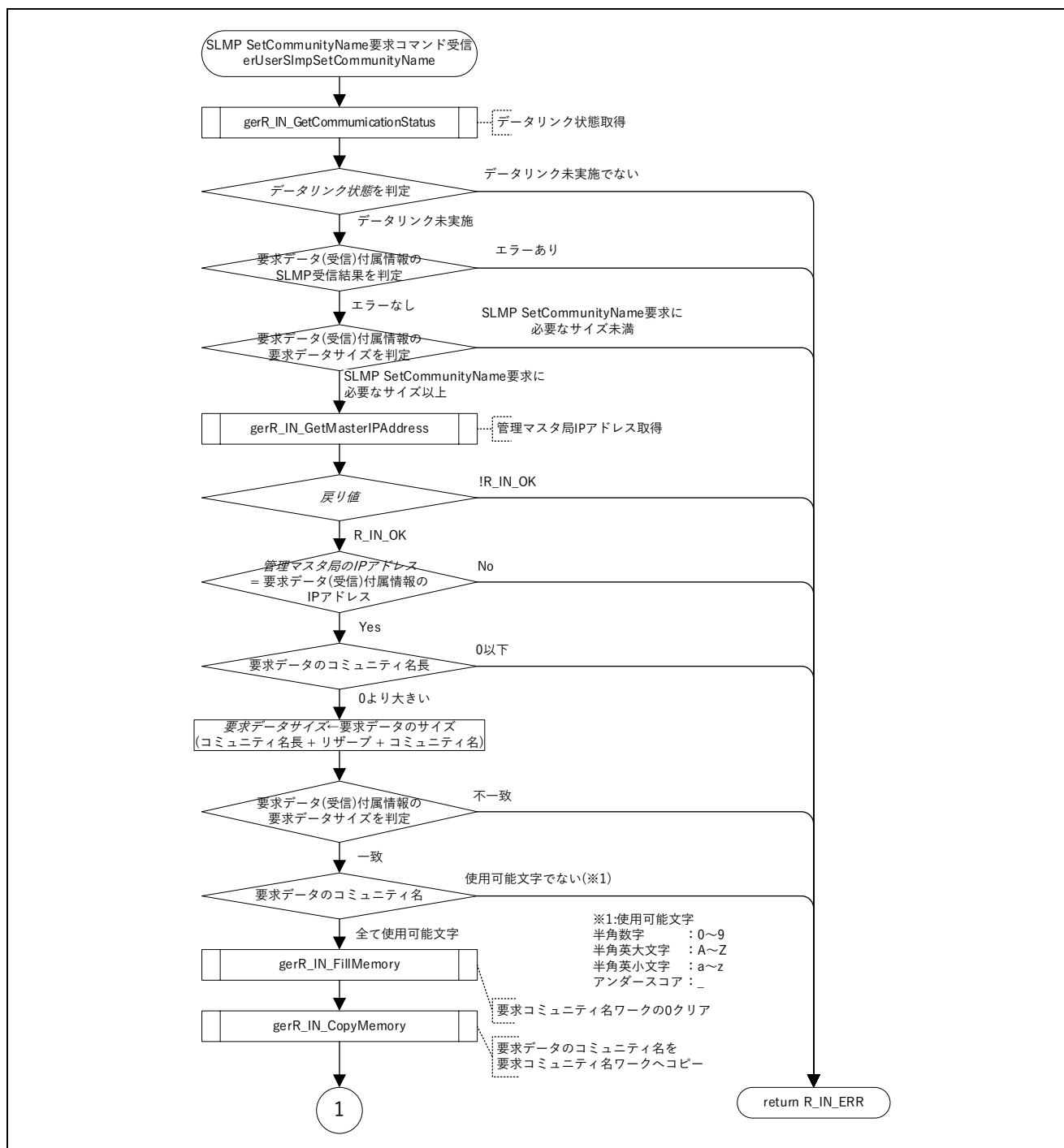


図 5.56 SLMP SetCommunityName 要求コマンド受信処理フロー図 1

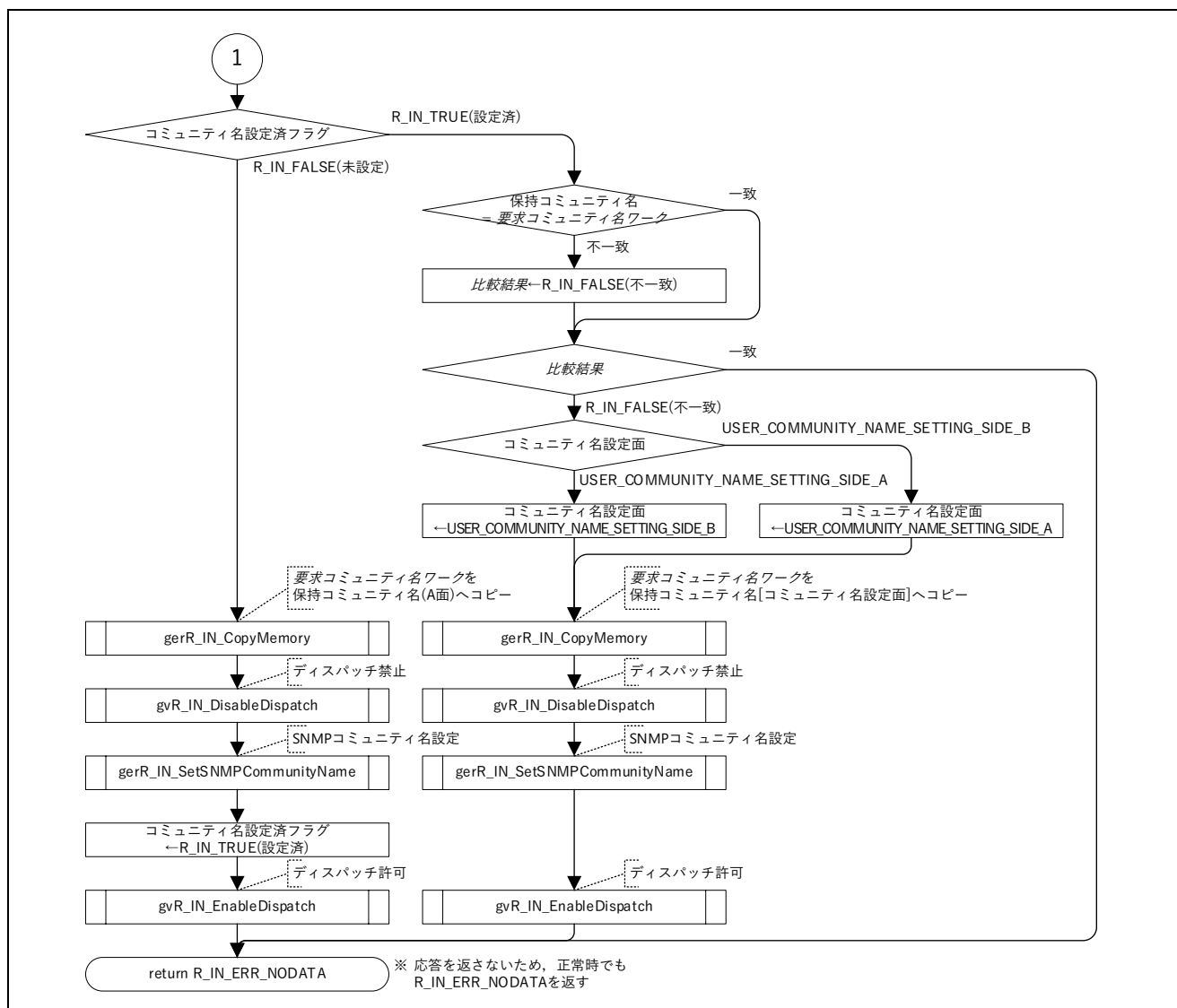


図 5.57 SLMP SetCommunityName 要求コマンド受信処理フロー図 2



## 5.7 ユーザプログラム詳細(デバイス局パラメータ自動設定関連)

### 5.7.1 SLMP 通信設定取得要求コマンド受信処理

通信ポートとタイムアウト値を送信バッファに設定します。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

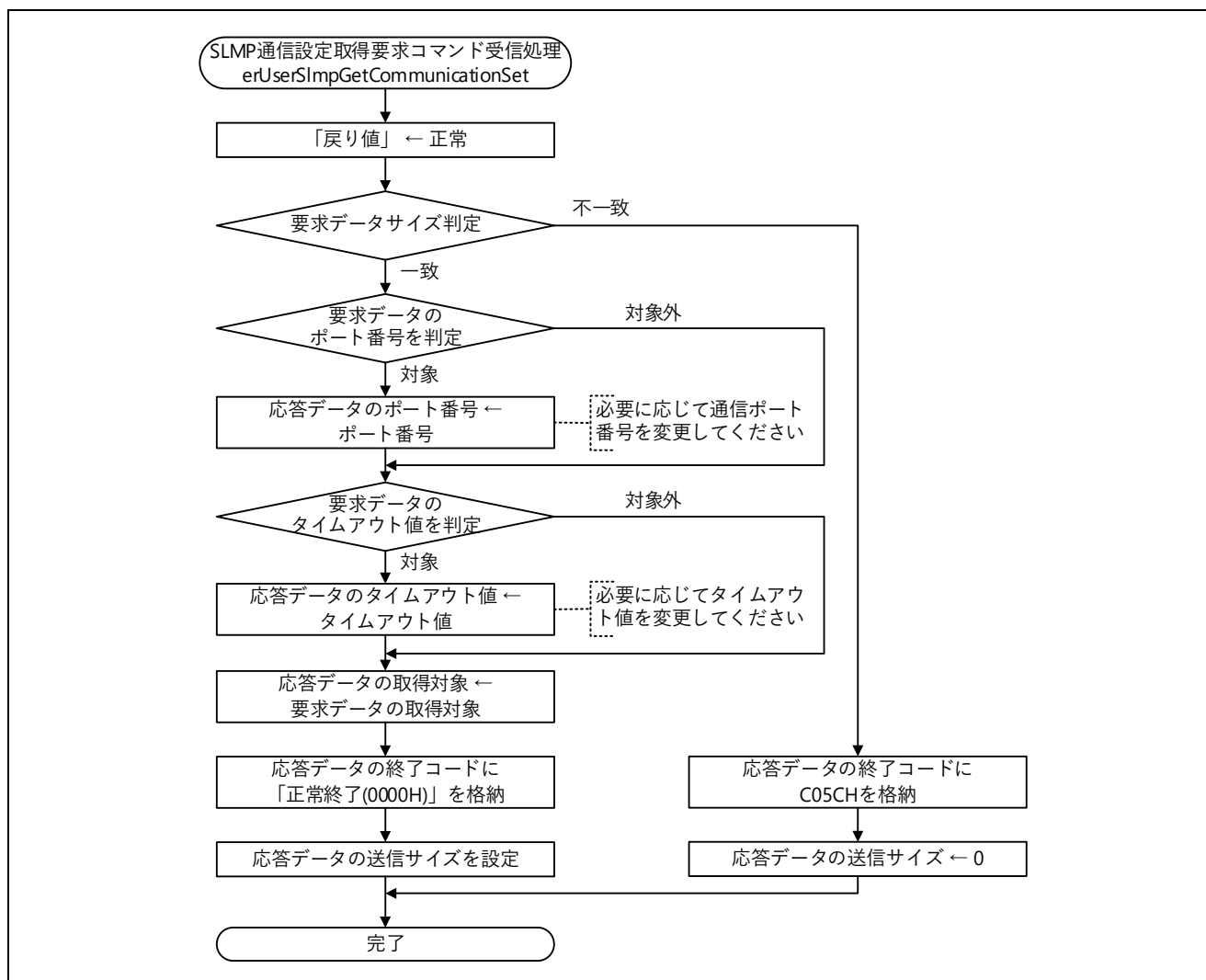


図 5.58 SLMP 通信設定取得要求コマンド受信処理フロー図

## 5.7.2 SLMP パラメータ配信可否チェック要求コマンド受信処理

パラメータ配信可否チェック処理を実行します。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

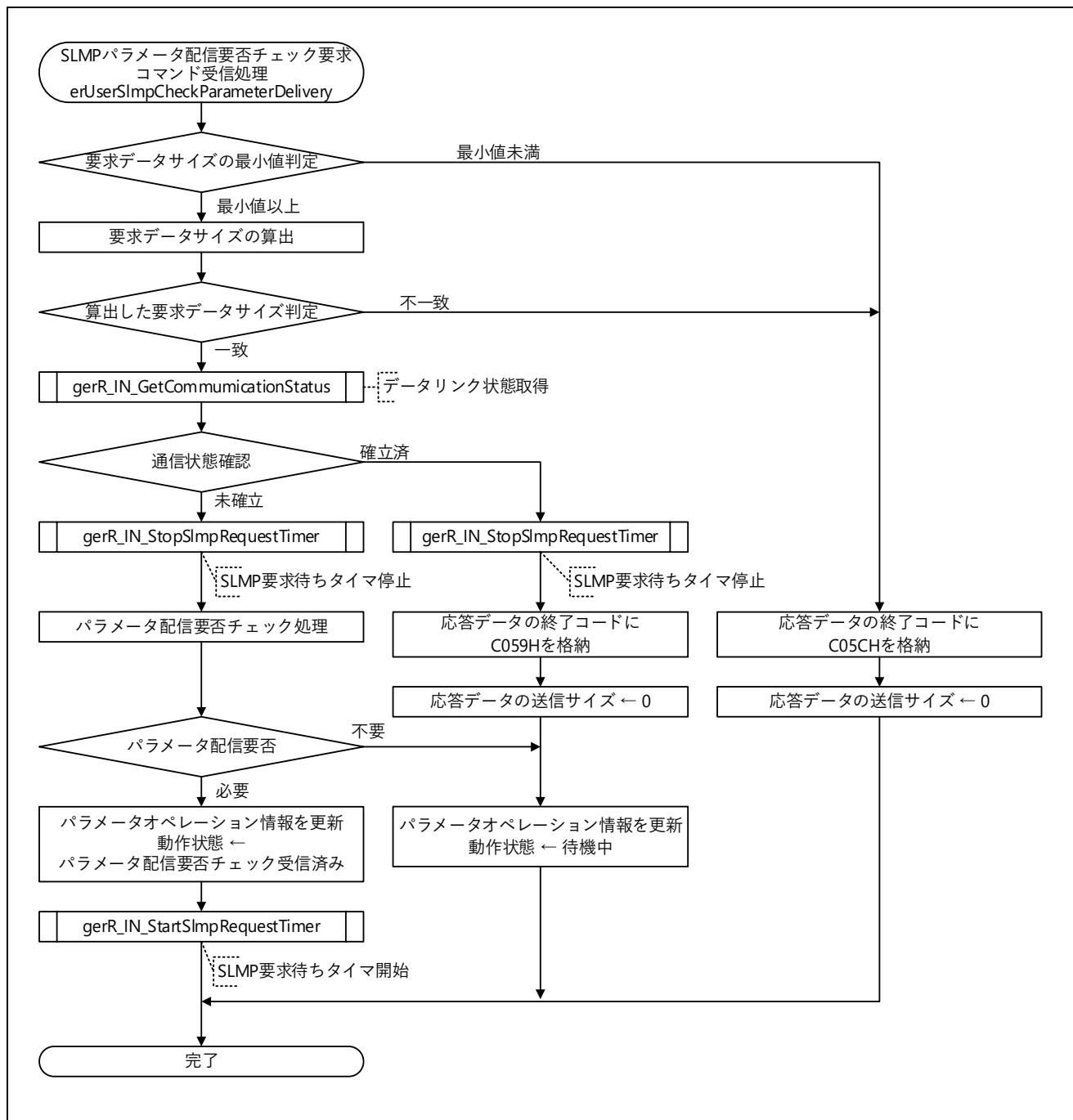


図 5.59 SLMP パラメータ配信可否チェック要求コマンド受信処理フロー図

## 5.7.3 SLMP パラメータ配信要否チェック処理

局サブ ID ごとにパラメータ配信が必要か否か判定を行い、応答データを作成します。

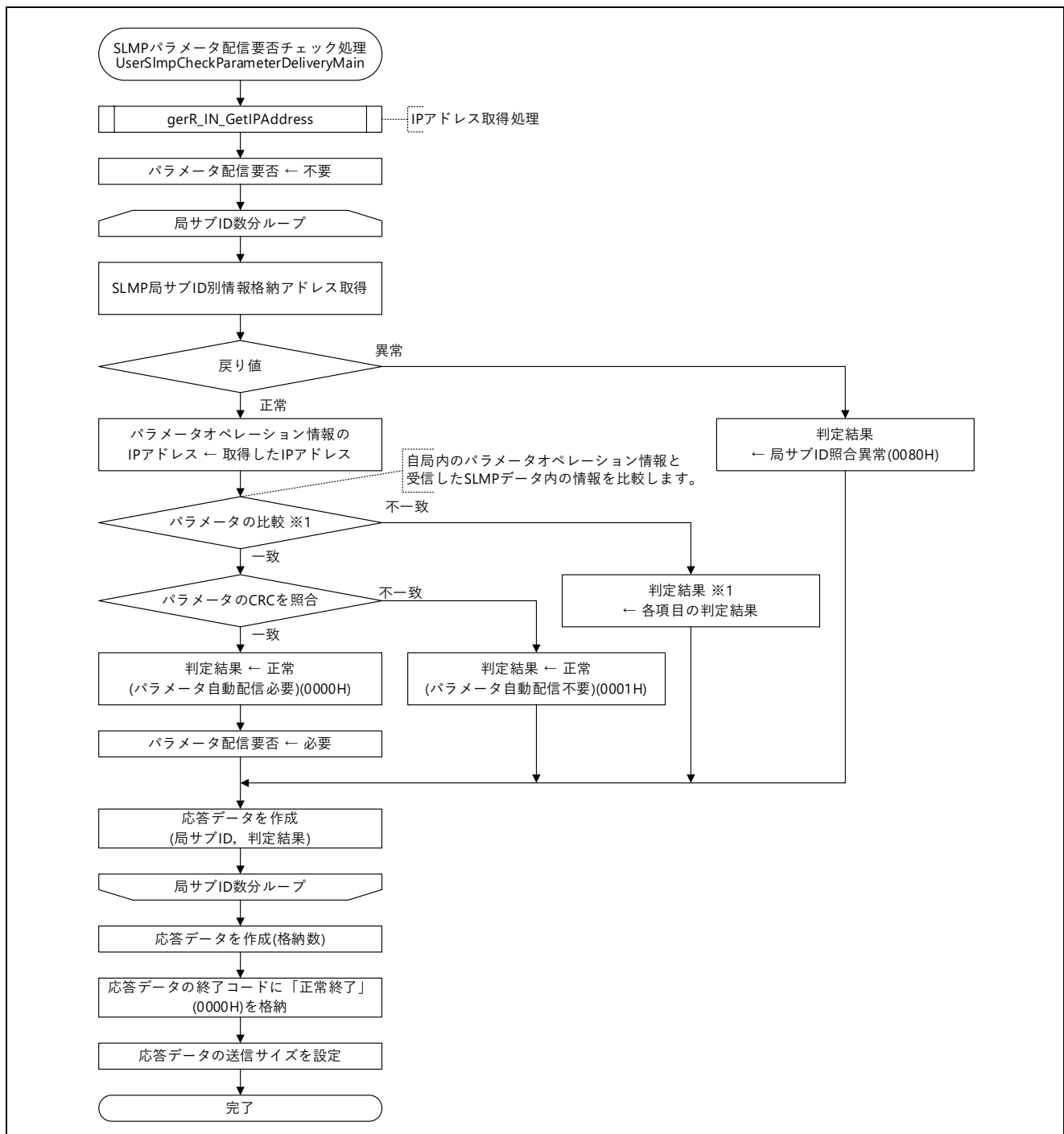


図 5.60 SLMP パラメータ配信要否チェック処理フロー図

【注】 ※1 比較対象データと判定結果は以下です。

- ・ ヘッダバージョン : ヘッダバージョン照合異常(0010H)
- ・ ベンダーコード : ベンダーコード照合異常(0020H)
- ・ 型名コード : 型名コード照合異常(0030H)
- ・ 機器バージョン : 機器バージョン照合異常(0040H)
- ・ IP アドレス : IP アドレス照合異常(0060H)

## 5.7.4 SLMP リストア開始通知要求コマンド受信処理

リストア事前処理を行います。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

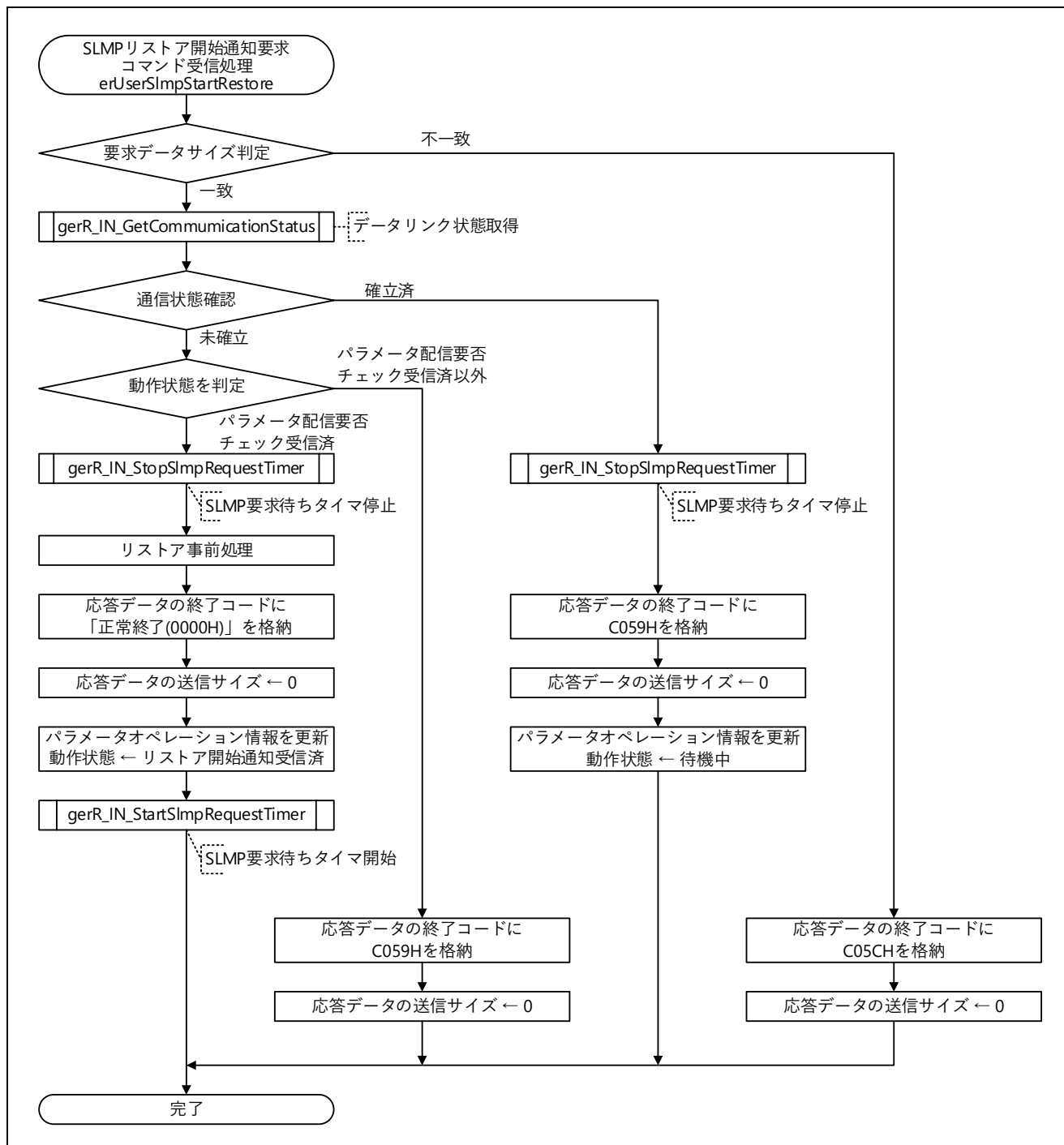


図 5.61 SLMP リストア開始通知要求コマンド受信処理フロー図

### 5.7.5 SLMP リストア終了通知要求コマンド受信処理

リストア事後処理を行います。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

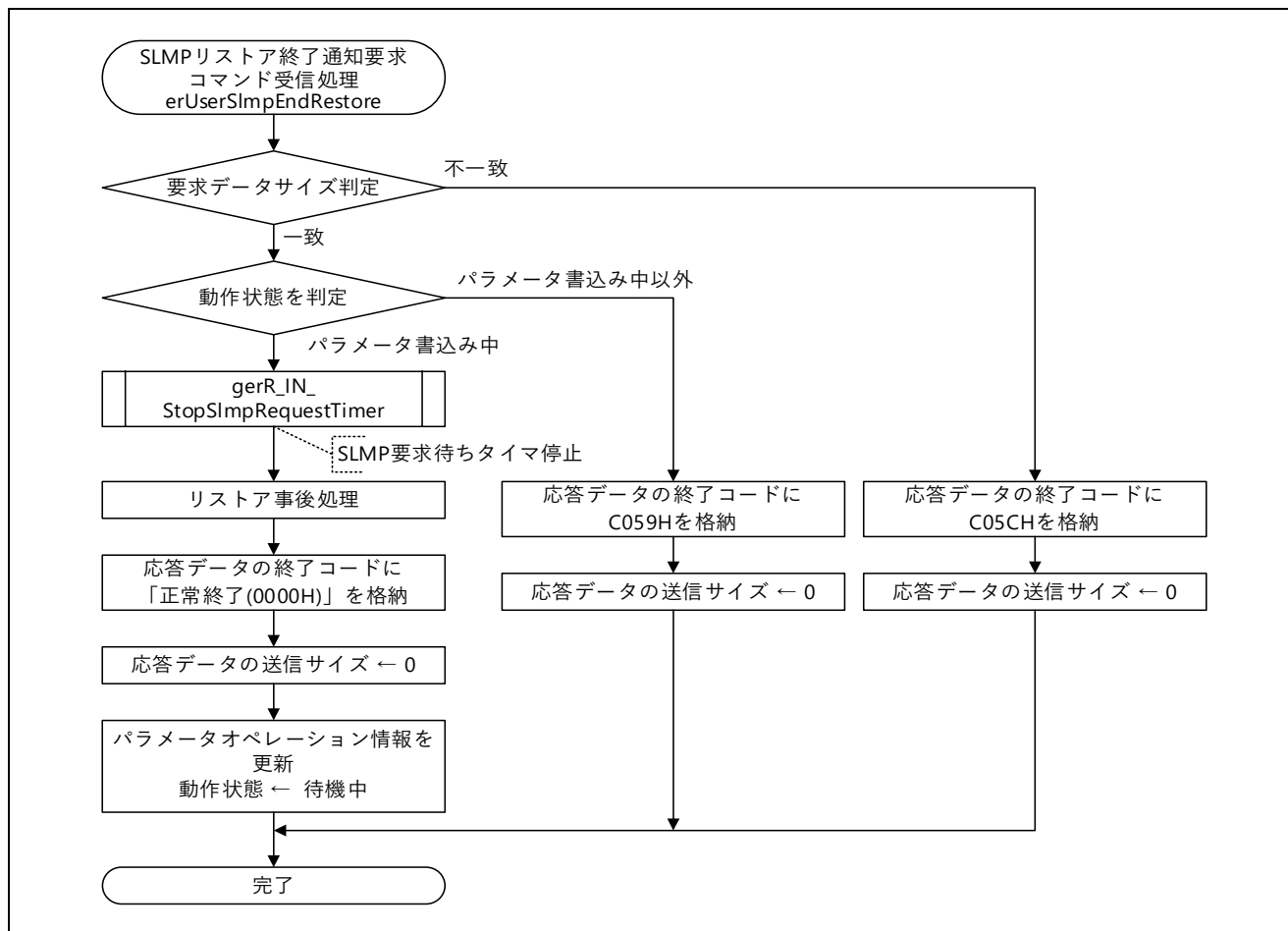


図 5.62 SLMP リストア終了通知要求コマンド受信処理フロー図

### 5.7.6 SLMP パラメータデータ書込み要求コマンド受信処理

パラメータ書込み処理を行います。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

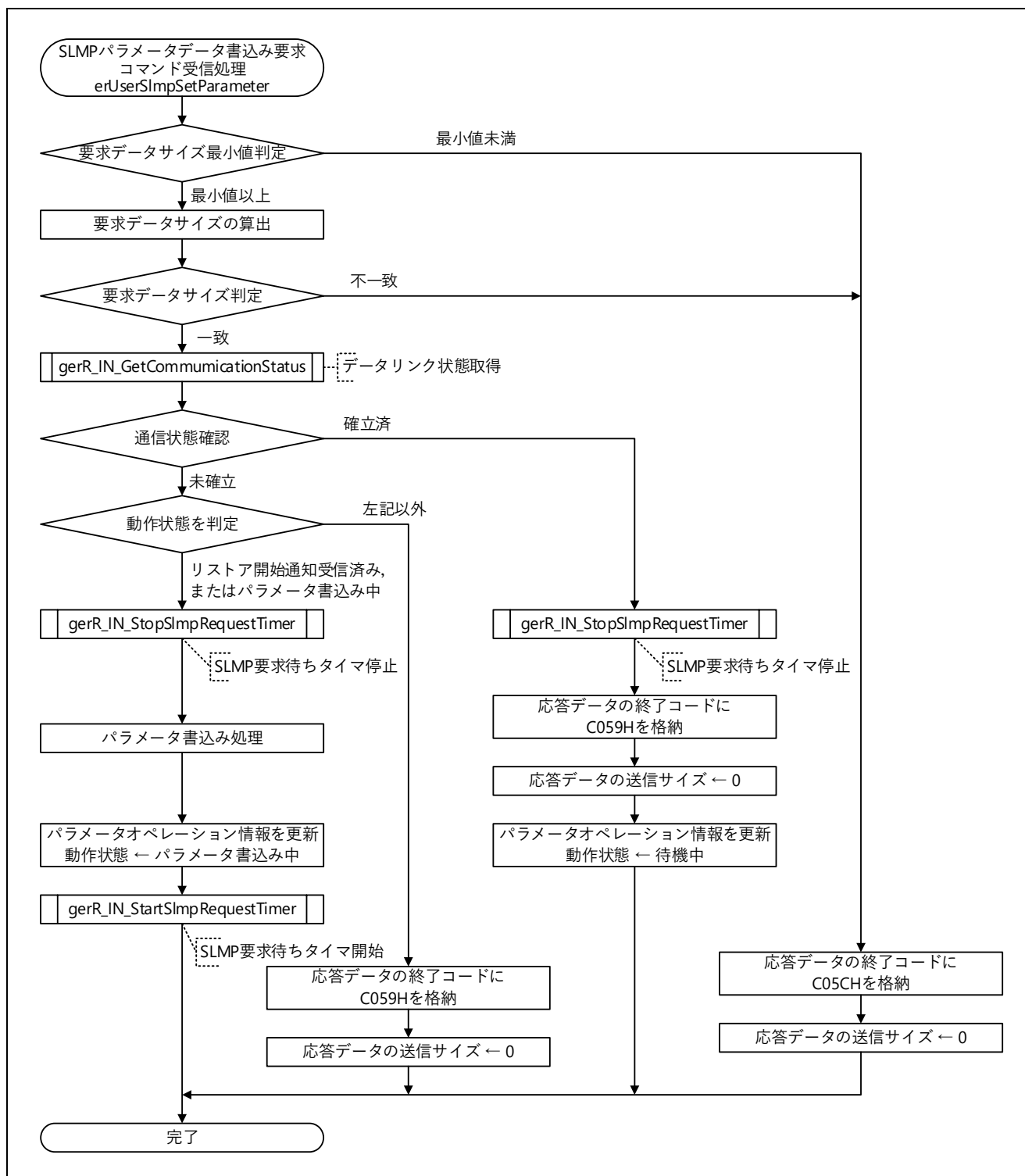


図 5.63 SLMP パラメータデータ書き込み要求コマンド受信処理フロー図

### 5.7.7 SLMP パラメータデータ書き込み処理

受信したパラメータを、対応する局サブ ID のパラメータバッファへ格納します。

本処理はパラメータデータのサイズが 1438byte 以下に対応しています。

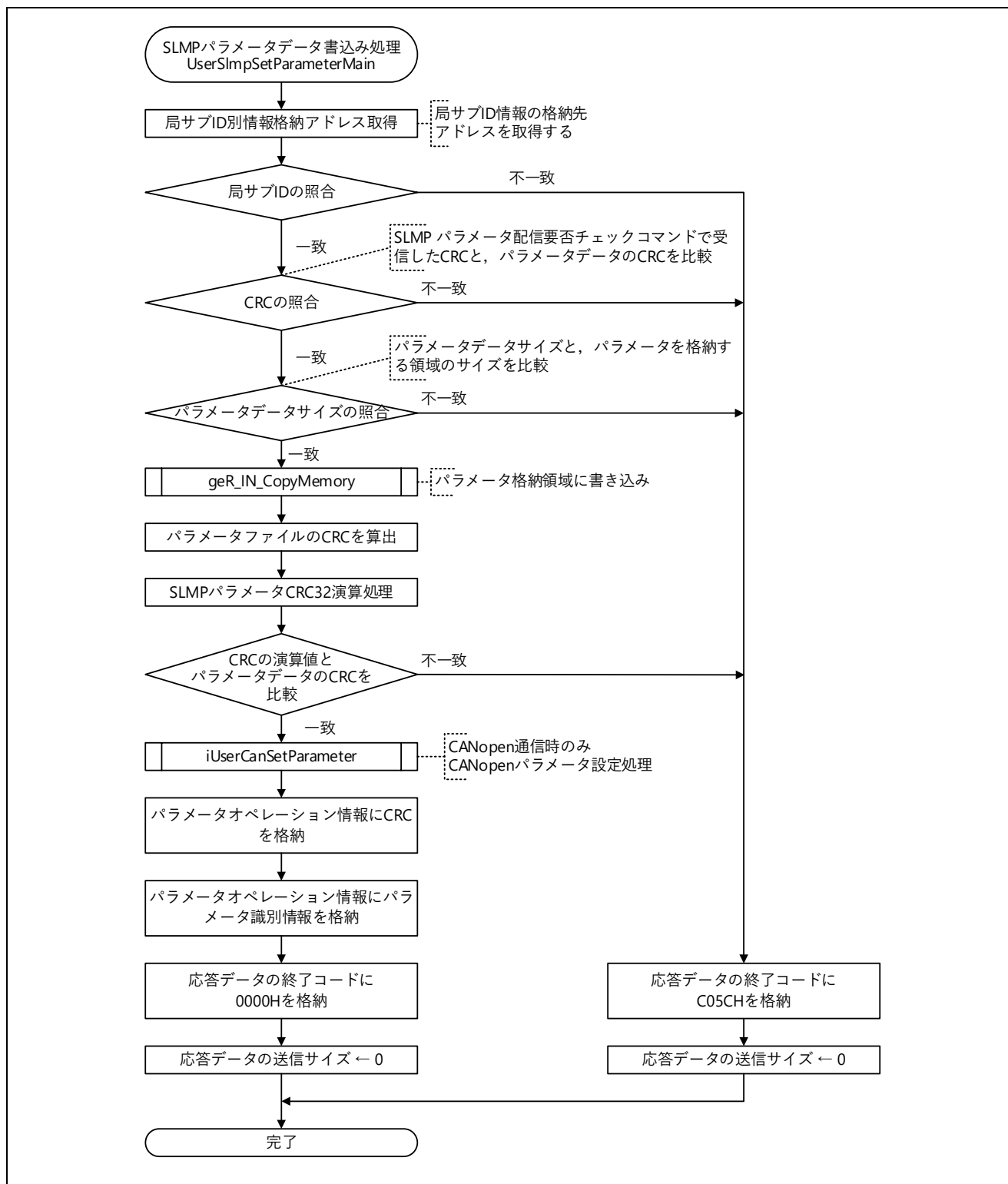


図 5.64 SLMP パラメータデータ書き込み処理フロー図

## 5.8 ユーザプログラム詳細(SLMP 経由の通信速度・CC-Link IE TSN Class 設定関連)

R-IN32M4-CL3 適用製品の通信速度/CC-Link IE TSN Class を設定ツールで設定します。

設定ツールが SLMP を使用して通信速度/CC-Link IE TSN Class 設定データを R-IN32M4-CL3 適用製品に送信します。R-IN32M4-CL3 適用製品は受信した設定データを不揮発性メモリに書き込み、次回電源 OFF→ON 時に設定を反映します。

設定ツール(SLMP のクライアント)には、CC-Link 協会の CC-Link IE TSN 設定ツール(CC-Link IE TSN Configurator)を使用します。CC-Link IE TSN 設定ツールの操作の詳細は、CC-Link IE TSN 設定ツールユーザズマニュアル(BAP-C3009-001)を参照してください。通信速度を 100Mbps から 1Gbps に変更する場合で、一連の処理の流れを示します。

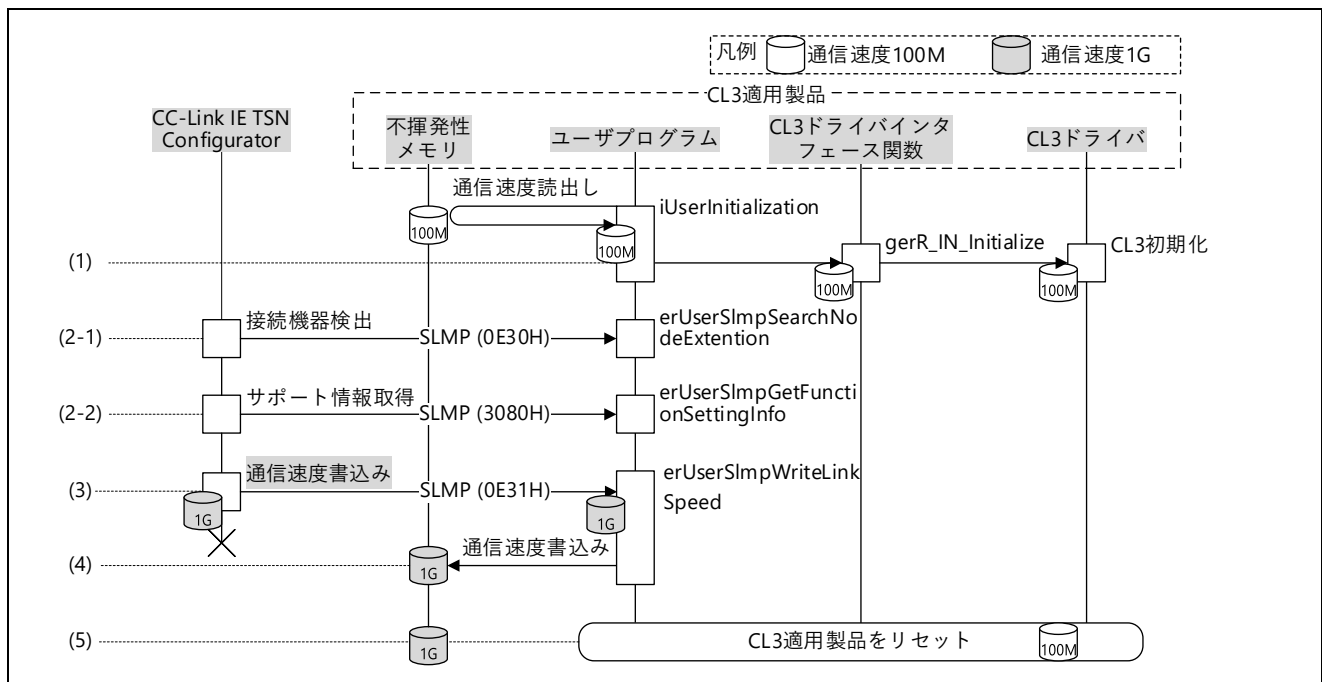


図 5.65 SLMP 経由の通信速度設定の概略手順

手順	内容	関連
(1)	CC-Link IE TSN 設定ツールをインストールしたパソコンと、通信速度設定対象の R-IN32M4-CL3 適用製品を Ethernet ケーブルで接続してください。R-IN32M4-CL3 適用製品の電源を ON してください。不揮発性メモリから読み出した通信速度、あるいはほかの方法で設定した通信速度を gerR_IN_Initialize の引数にセットし、R-IN32M4-CL3 起動時の通信速度を設定します。	5.3.1 6.4.1(2)
(2-1)	CC-Link IE TSN 設定ツールの「ツール」→「リモート局の機能設定一括/個別実行機能」を選択してください。	5.8.1
(2-2)	画面起動時に CC-Link IE TSN 設定ツールが 2 つの SLMP 要求を送信します。応答することで、画面に R-IN32M4-CL3 適用製品を表示できます。	5.8.2
(3)	起動した画面内の「実行する処理」から「通信速度書き込み」を選択し実行してください。CC-Link IE TSN 設定ツールが、SLMP(3082H: 機能設定の書き込み(通信速度))を送信します。R-IN32M4-CL3 適用製品は、erUserSmpWriteLinkSpeed による受信処理を行います。	5.8.4
(4)	erUserSmpWriteLinkSpeed 内のユーザ作成処理が、不揮発性メモリに通信速度の設定値を書き込みます。書き込み完了後に電源を OFF してください。-	-
(5)	次回電源 ON 時に、不揮発性メモリに書き込まれたデータを読み出し、gerR_IN_Initialize の引数(stUnitInfo.ulLinkSpeed)へセットします。gerR_IN_Initialize が R-IN32M4-CL3 の内蔵 GbE-PHY に通信速度を設定します。gerR_IN_Initialize の完了後、R-IN32M4-CL3 ドライバが内蔵 GbE-PHY をリセットし、通信速度の変更が適用されます。	5.3.1 6.4.1(2)



### 5.8.1 SLMP 接続機器検出(拡張)要求コマンド受信処理

接続機器検出要求に応答します。

本要求コマンドは、CC-Link 協会の CC-Link IE TSN 設定ツールにて、「リモート局の機能設定一括/個別実行機能」を選択、または「接続/切断した機器の検出」を選択した際に送信されます。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが要求データ(受信)付属情報の SLMP 受信結果にエラーを設定します。

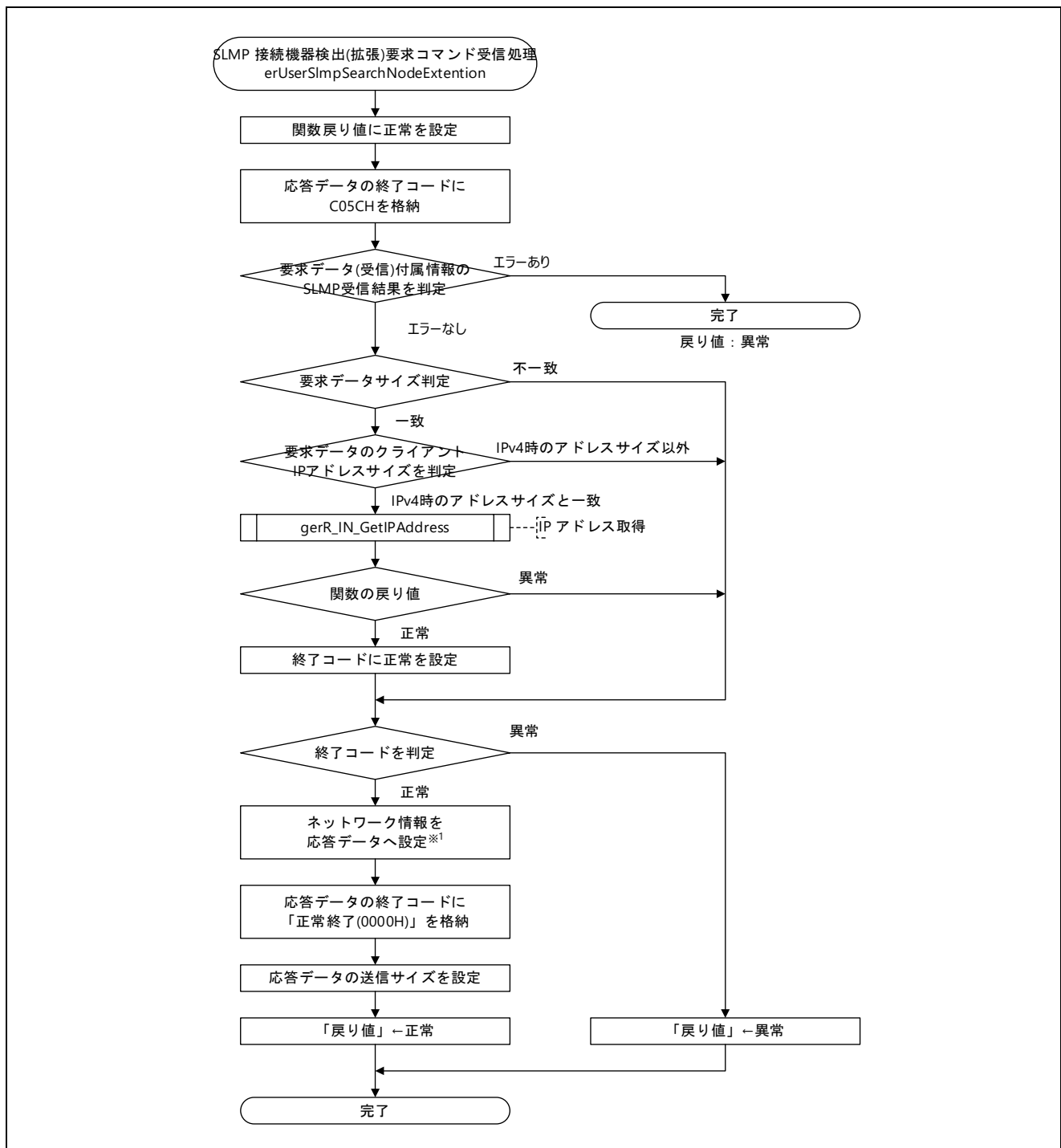


図 5.66 SLMP 接続機器検出(拡張)要求コマンド受信処理フロー図

【注】※1 応答データの詳細は、「表 5.24」を参照してください。

表 5.24 USER\_SLMP\_COMMAND\_SEARCH\_NODE\_EXTENTION\_RESPONSE\_T

No	メンバ		内容
1	UCHAR	auchClientMACAddress[R_IN_MACADR_SZ]	クライアント MAC アドレス (6 バイト)
2	UCHAR	uchClientIPAddressSize	クライアント IP アドレスサイズ
3	ULONG	ulClientIPAddress	クライアント IP アドレス
4	UCHAR	auchServerMACAddress[R_IN_MACADR_SZ]	サーバ MAC アドレス (6 バイト)
5	UCHAR	uchServerIPAddressSize	サーバ IP アドレスサイズ
6	ULONG	ulServerIPAddress	サーバ IP アドレス
7	ULONG	ulServerSubnetmask	サーバサブネットマスク
8	ULONG	ulServerDefaultGateway	サーバデフォルトゲートウェイ IP アドレス
9	UCHAR	uchServerHostNameSize	サーバホスト名サイズ (0 固定)
10	USHORT	usServerVendorCode	サーバベンダーコード
11	ULONG	ulServerModelCode	サーバ形名コード
12	UCHAR	uchServerModelNameSize	サーバ形名サイズ
13	UCHAR	auchServerModelName[R_IN_MODEL_NAME_LENGTH]	サーバ形名 (最大 20 バイト)
14	USHORT	usServerMachineVersion	サーバ機器バージョン
15	UCHAR	uchTargetUnitIPAddressSize	交信相手ユニット IP アドレスサイズ
16	ULONG	ulTargetUnitIPAddress	交信相手ユニット IP アドレス
17	USHORT	usTargetUnitPortNumber	交信相手ユニット通信ポート番号
18	USHORT	usServerStatus	サーバステータス
19	USHORT	usServerPortNumber	サーバ通信ポート番号
20	UCHAR	uchServerProtocol	サーバ通信プロトコル設定

## 5.8.2 SLMP 機能設定(サポート情報取得)要求コマンド受信処理

自局における機能設定の対応情報を応答します。

本要求コマンドは、CC-Link 協会の CC-Link IE TSN 設定ツールにて、「リモート局の機能設定一括/個別実行機能」を選択、または「接続/切断した機器の検出」を選択した際に送信されます。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

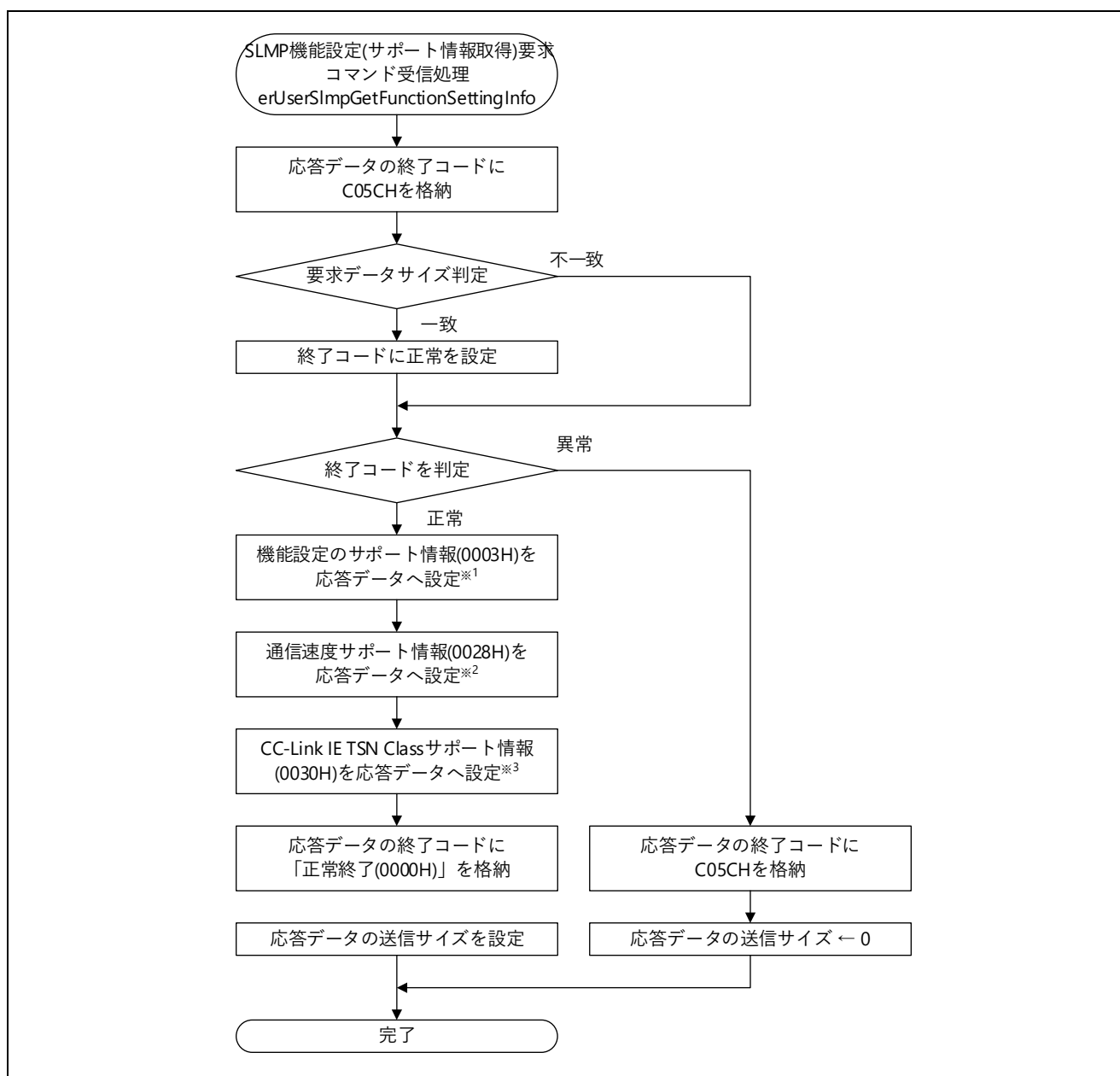


図 5.67 SLMP 機能設定(サポート情報取得)要求コマンド受信処理フロー図

【注】※1 詳細は「表 5.25」を参照してください。

※2 詳細は「表 5.26」を参照してください。

※3 詳細は「表 5.27」を参照してください。

表 5.25 USER\_SLMP\_FUNCTION\_SETTING\_INFO

ビット	項目	内容	初期値
15-3	予約	0 固定	0
2	SNMP コミュニティ名設定	0 : 未対応 / 1 : 対応	1b
1	CC-Link IE TSN Class	0 : 未対応 / 1 : 対応	1b
0	通信速度	0 : 未対応 / 1 : 対応	1b

表 5.26 USER\_SLMP\_FUNCTION\_SETTING\_INFO\_LINKSPEED\_100M\_1G\_FULL

ビット	項目	内容	初期値
31-6	予約	0 固定	0
5	1Gbps (全二重)	0 : 未対応 / 1 : 対応	1b
4	予約	0 固定	0b
3	100Mbps(全二重)	0 : 未対応 / 1 : 対応	1b
2-0	予約	0 固定	000b

表 5.27 USER\_SLMP\_FUNCTION\_SETTING\_INFO\_CCIETSN\_CLASS

ビット	項目	内容	初期値
15-6	予約	0 固定	0
5	CC-Link IE TSN Class B ver.2.0	0 : 未対応 / 1 : 対応	1b
4	CC-Link IE TSN Class A ver.2.0 (ネットワーク時刻配信対応)	0 : 未対応 / 1 : 対応	1b
3	CC-Link IE TSN Class A ver.2.0 (ネットワーク時刻配信非対応)	0 : 未対応 / 1 : 対応	0b
2	CC-Link IE TSN Class B ver.1.0	0 : 未対応 / 1 : 対応	0b
1	予約	0 固定	0b
0	CC-Link IE TSN Class A ver.1.0	0 : 未対応 / 1 : 対応	0b

### 5.8.3 SLMP 機能設定読出し(通信速度)要求コマンド受信処理

自局が動作している通信速度を応答します。

本要求コマンドは、CC-Link 協会の CC-Link IE TSN 設定ツールにて、「リモート局の機能設定一括/個別実行機能」の「通信速度読出し」を選択した際に送信されます。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

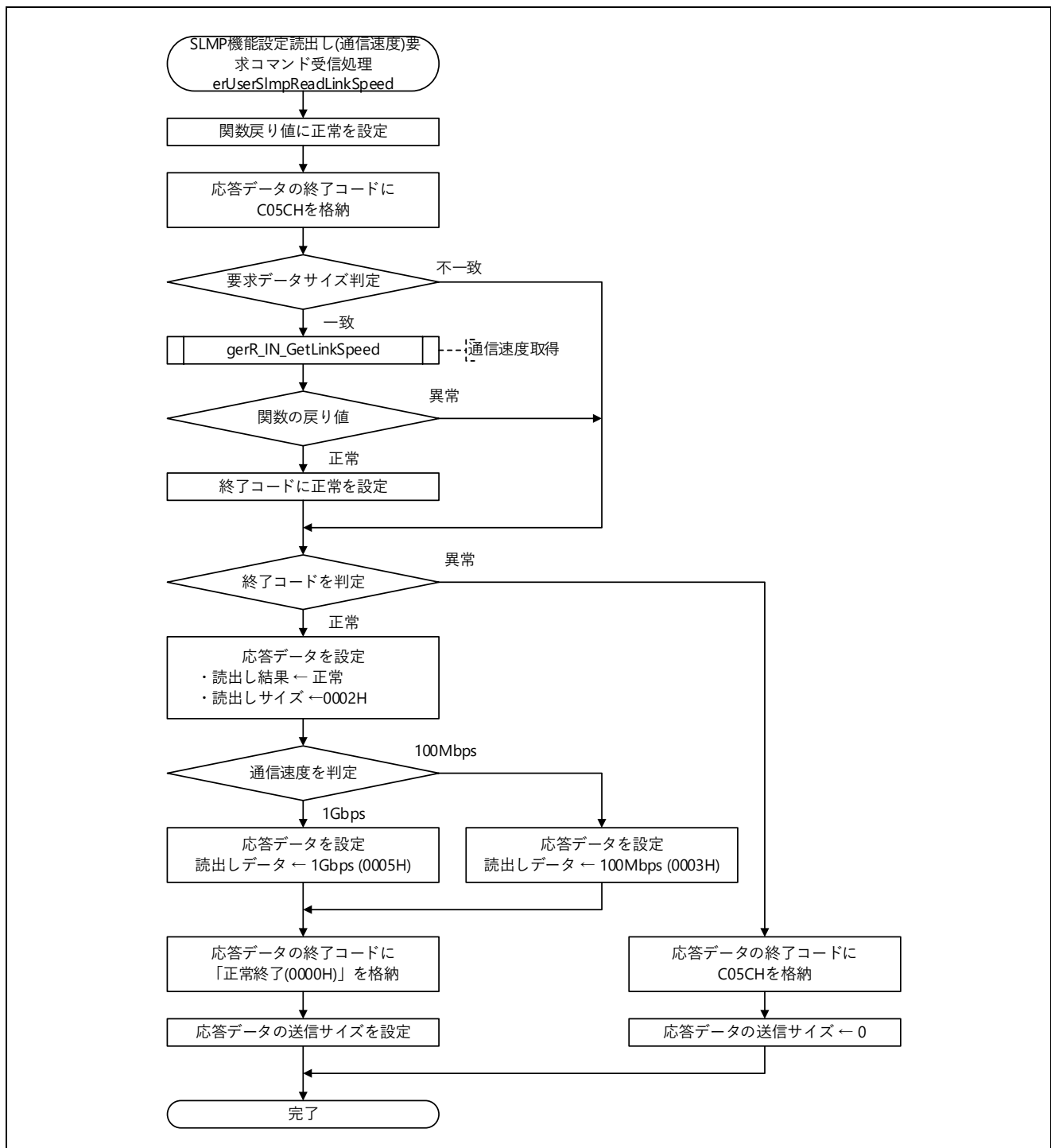


図 5.68 SLMP 機能設定読出し(通信速度)要求コマンド受信処理フロー図

#### 5.8.4 SLMP 機能設定書込み(通信速度)要求コマンド受信処理

指定された通信速度を SLMP フレームから取得し、不揮発性メモリに保存します。

本要求コマンドは、CC-Link 協会の CC-Link IE TSN 設定ツールにて、「リモート局の機能設定一括/個別実行機能」の「通信速度書込み」を選択した際に送信されます。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

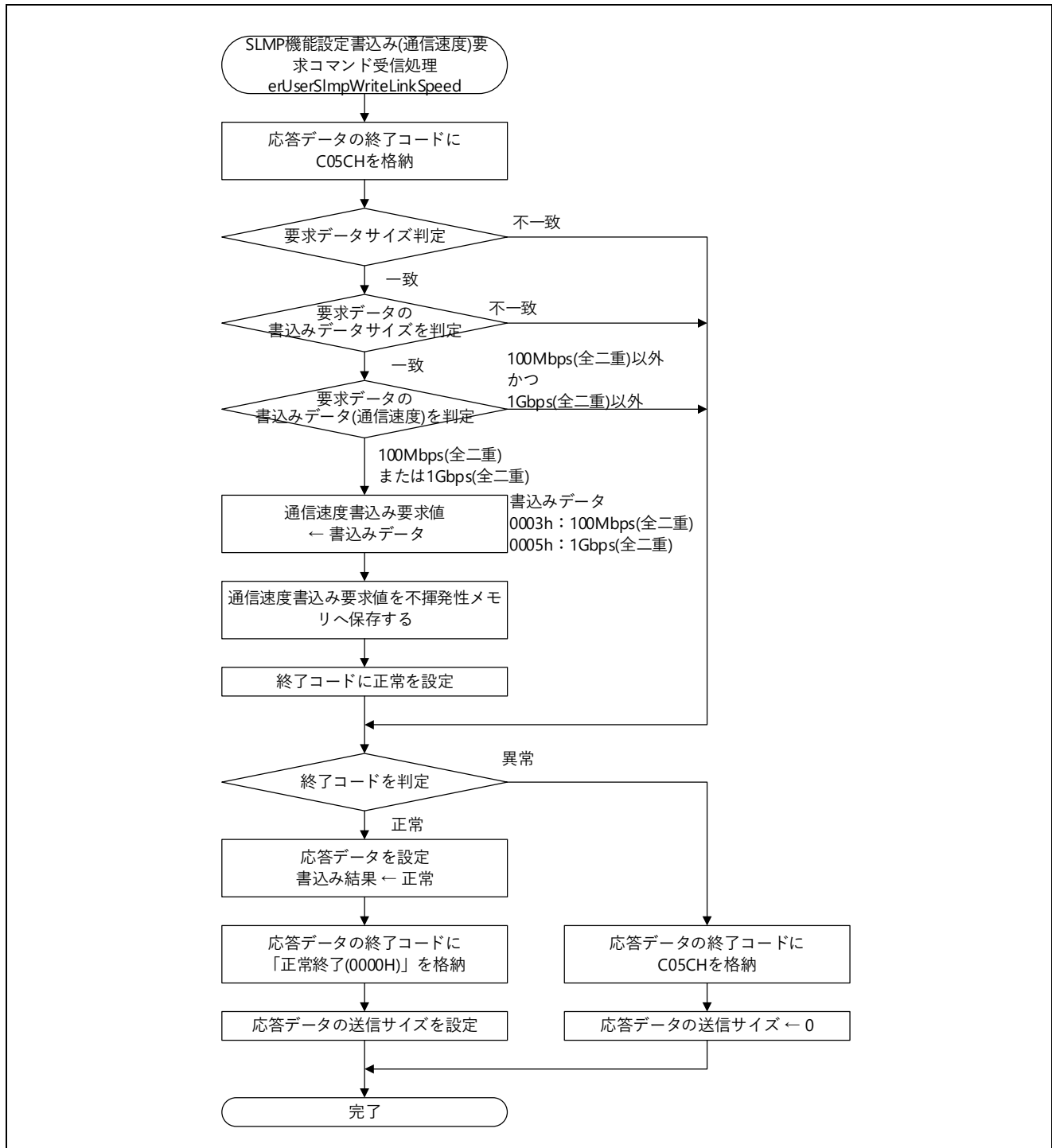


図 5.69 SLMP 機能設定書込み(通信速度)要求コマンド受信処理フロー図

### 5.8.5 SLMP 機能設定読出し(CC-Link IE TSN Class)要求コマンド受信処理

自局が動作している CC Link IE TSN Class を応答します。

本要求コマンドは、CC-Link 協会の CC-Link IE TSN 設定ツールにて、「リモート局の機能設定一括/個別実行機能」の「CC Link IE TSN Class 読出し」を選択した際に送信されます。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

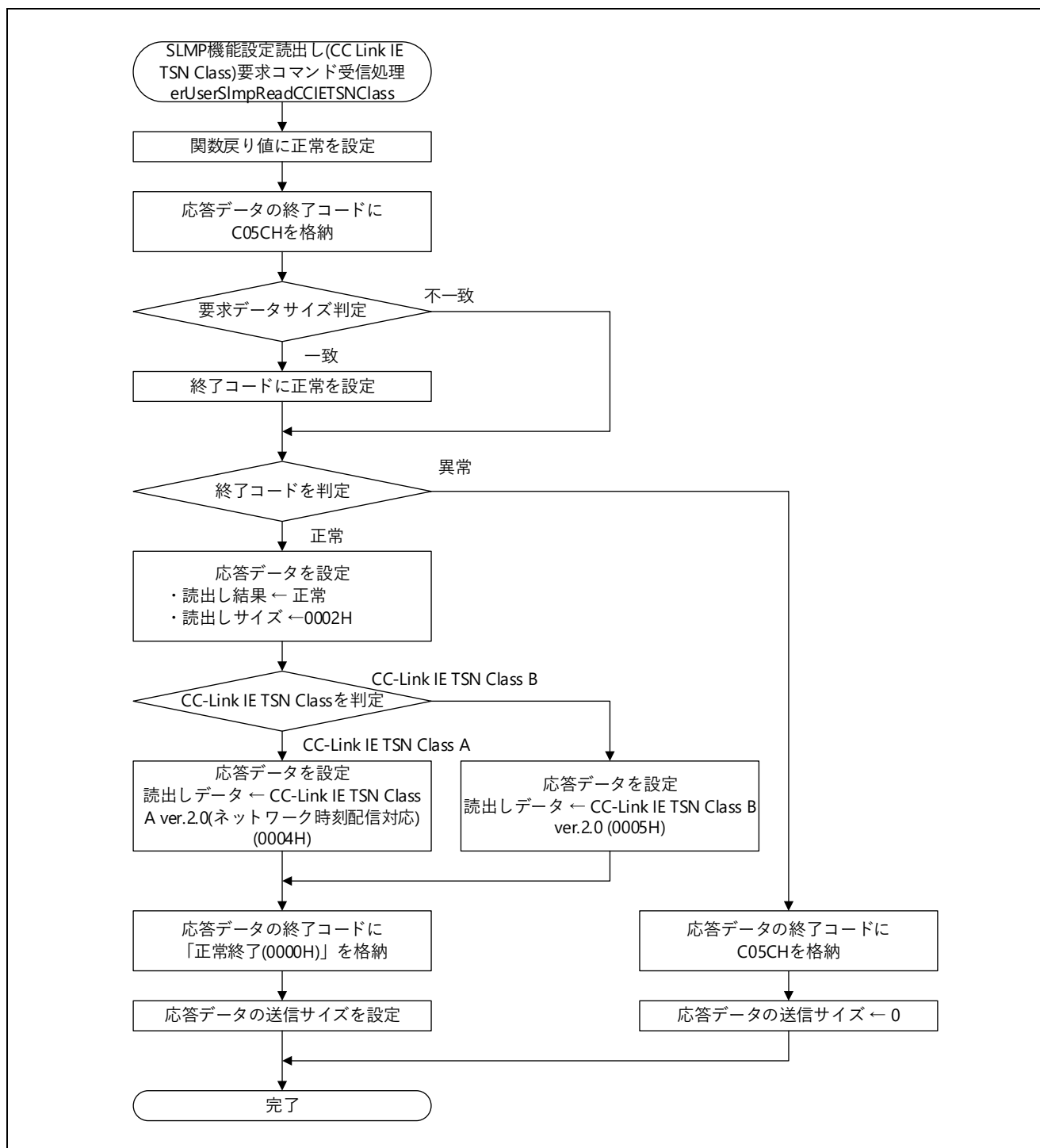


図 5.70 SLMP 機能設定読出し(CC-Link IE TSN Class)要求コマンド受信処理フロー図

### 5.8.6 SLMP 機能設定書込み(CC-Link IE TSN Class)要求コマンド受信処理

指定された CC-Link IE TSN Class を SLMP フレームから取得し、不揮発性メモリに保存します。

本要求コマンドは、CC-Link 協会の CC-Link IE TSN 設定ツールにて、「リモート局の機能設定一括/個別実行機能」の「CC-Link IE TSN Class 書込み」を選択した際に送信されます。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

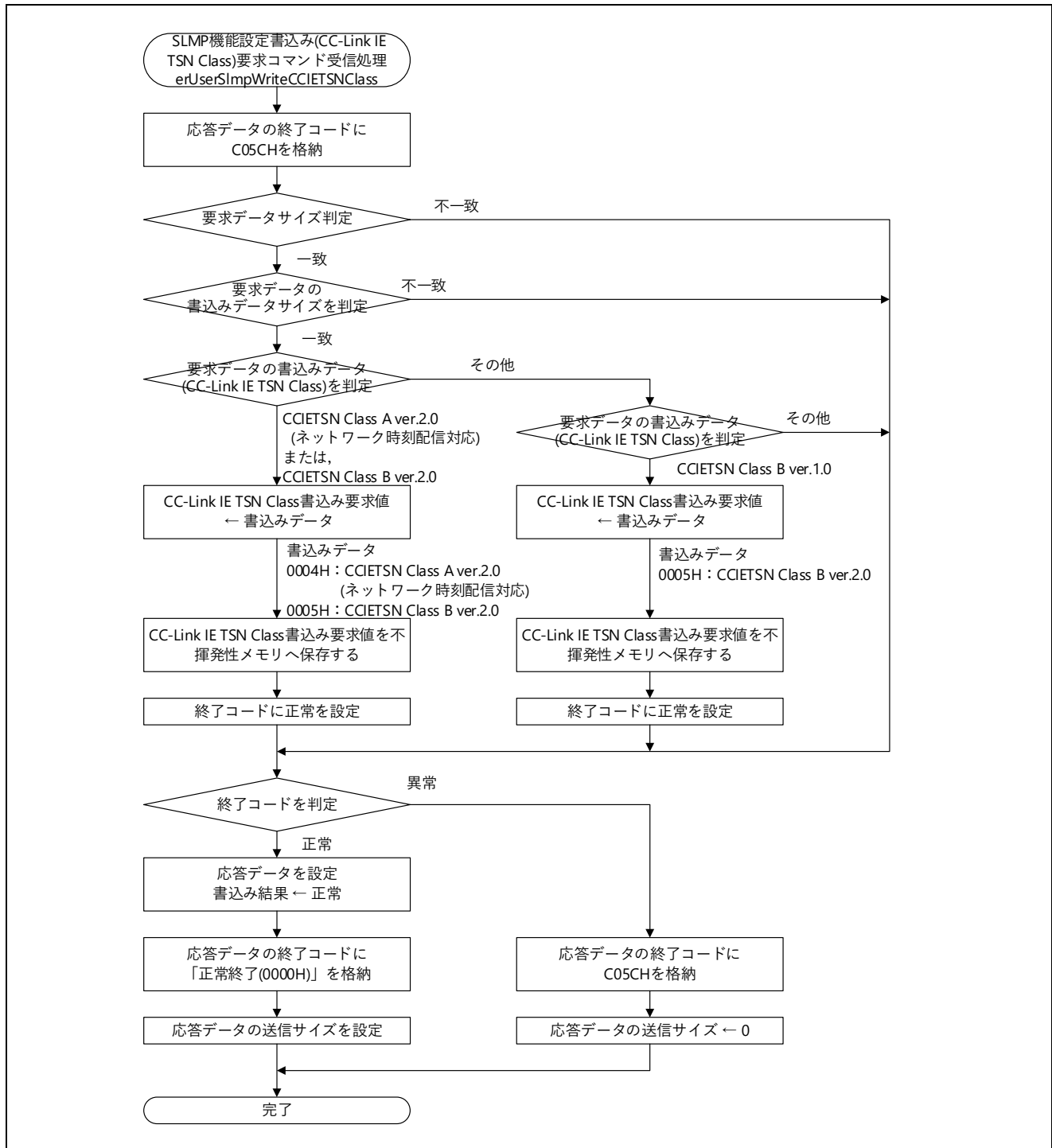


図 5.71 SLMP 機能設定書込み(CC-Link IE TSN Class)要求コマンド受信処理フロー図



## 5.9 ユーザプログラム詳細(CANopen 通信関連)

### (1) Object Dictionary について

Object Dictionary はパラメータや指令値などの機器に依存するデータを持つため、ユーザ側で定義し R-IN32M4-CL3 に設定します。Object Dictionary 内のアクセス対象オブジェクトは、Index および SubIndex により指定されます。

構造の詳細は、CC-Link 協会の CC-Link IE TSN 仕様書(概要編)を参照してください。

表 5.28 Object Dictionary の構造

Index	Object dictionary area	内容
0000H~0FFFH	Data type area	データ型の定義を行います。
1000H~1FFFH	Communication profile area	通信固有オブジェクトの定義です。
2000H~5FFFH	Manufacturer specific profile area	メーカ固有オブジェクトの定義です。ユーザによる Object Dictionary への実装は任意です。
6000H~FFFFH	Device profile area	デバイスプロファイルで定義されたオブジェクトの定義です。ユーザによる Object Dictionary への実装は任意です。

### (2) Object Dictionary のデータ構造

Object Dictionary は R\_IN\_CAN\_OD\_T 型の配列構造となります。本配列は Index の昇順でソートしてください。また、Object Dictionary の設定はプロファイルで定義した Object Dictionary と同じ設定にしてください。

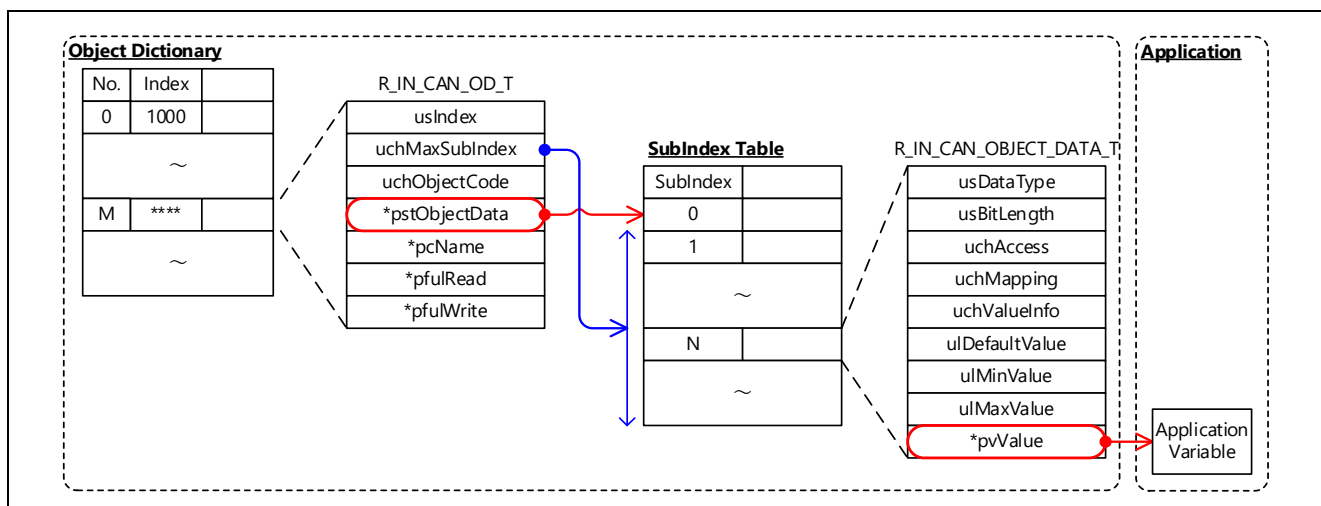


図 5.72 Object Dictionary のデータ構造イメージ

表 5.29 R\_IN\_CAN\_OD\_T 構造の概要

No	メンバ		内容
1	USHORT	usIndex	インデックス
2	UCHAR	uchMaxSubIndex	最大サブインデックス
3	UCHAR	uchObjectCode	オブジェクトコード
4	const R_IN_CAN_OBJECT_DATA_T	*pstObjectData	オブジェクトデータ
5	const CHAR	*pcName	名前配列の先頭アドレス
6	ULONG	(*pfulRead)(USHORT usIndex, UCHAR uchSubindex, ULONG ulSize, UCHAR *puchData)	読み出し関数
7	ULONG	(*pfulWrite)(USHORT usIndex, UCHAR uchSubindex, ULONG ulSize, UCHAR *puchData)	書き込み関数

R\_IN\_CAN\_OD\_T 構造体の詳細は「表 6.40 R\_IN\_CAN\_OD\_T 一覧」を参照してください。

## (3) PDO に関する Object Dictionary の定義について

Object Dictionary の Process data object (PDO)に関連する Object を定義する際には、下記の仕様を満たしてください。

表 5.30 PDO に関する Object Dictionary における PDO の仕様

No	対象 Index	項目	概要																
1	全体	文字列型の PDO マッピング	データタイプが「文字列型（R_IN_CAN_VISIBLESTRING）」のオブジェクトは、RPDO および TPDO マッピングできません。																
2	1C00H～1CFFH	使用可能な Index	1C00H から順に使用してください。 使用する Index の数以上の値を下記定義 No.1 に設定してください。 (Index：1C00H～1C01H の場合) <table><tr><th>No</th><th>定義</th><th>設定値</th></tr><tr><td>1</td><td>R_IN_CAN_PDO_CONFIG_OBJECT_NUM</td><td>2</td></tr></table>	No	定義	設定値	1	R_IN_CAN_PDO_CONFIG_OBJECT_NUM	2										
No	定義	設定値																	
1	R_IN_CAN_PDO_CONFIG_OBJECT_NUM	2																	
3	1600H～17FFH	使用可能な Index	1600H から順に使用してください。 使用する Index の数以上の値を下記定義 No.1 で設定してください。 使用する SubIndex の最大値を下記定義 No.2 で設定してください。 (Index：1600H～1601H、SubIndex：0～16 の場合) <table><tr><th>No</th><th>定義</th><th>設定値</th></tr><tr><td>1</td><td>R_IN_CAN_RPDO_MAPPING_OBJECT_NUM</td><td>2</td></tr><tr><td>2</td><td>R_IN_CAN_RPDO_APPLICATION_OBJECT_NUM</td><td>16</td></tr></table>	No	定義	設定値	1	R_IN_CAN_RPDO_MAPPING_OBJECT_NUM	2	2	R_IN_CAN_RPDO_APPLICATION_OBJECT_NUM	16							
No	定義	設定値																	
1	R_IN_CAN_RPDO_MAPPING_OBJECT_NUM	2																	
2	R_IN_CAN_RPDO_APPLICATION_OBJECT_NUM	16																	
4	1A00H～1BFFH	使用可能な Index	1A00H から順に使用してください。 使用する Index の数以上の値を下記定義 No.1 で設定してください。 使用する SubIndex の最大値を下記定義 No.2 で設定してください。 (Index：1A00H～1A01H、SubIndex：0～16 の場合) <table><tr><th>No</th><th>定義</th><th>設定値</th></tr><tr><td>1</td><td>R_IN_CAN_TPDO_MAPPING_OBJECT_NUM</td><td>2</td></tr><tr><td>2</td><td>R_IN_CAN_TPDO_APPLICATION_OBJECT_NUM</td><td>16</td></tr></table>	No	定義	設定値	1	R_IN_CAN_TPDO_MAPPING_OBJECT_NUM	2	2	R_IN_CAN_TPDO_APPLICATION_OBJECT_NUM	16							
No	定義	設定値																	
1	R_IN_CAN_TPDO_MAPPING_OBJECT_NUM	2																	
2	R_IN_CAN_TPDO_APPLICATION_OBJECT_NUM	16																	
5	1600H～17FFH 1A00H～1BFFH	ビット長の設定値	PDO Mapping Object の設定値の「ビット長」には、データタイプに対応するビット長を設定してください。 <table><tr><th>データタイプ</th><th>ビット長</th></tr><tr><td>R_IN_CAN_INTEGER8</td><td>8</td></tr><tr><td>R_IN_CAN_INTEGER16</td><td>16</td></tr><tr><td>R_IN_CAN_INTEGER32</td><td>32</td></tr><tr><td>R_IN_CAN_UNSIGNED8</td><td>8</td></tr><tr><td>R_IN_CAN_UNSIGNED16</td><td>16</td></tr><tr><td>R_IN_CAN_UNSIGNED32</td><td>32</td></tr><tr><td>R_IN_CAN_VISIBLESTRING</td><td>PDO マッピング不可</td></tr></table>	データタイプ	ビット長	R_IN_CAN_INTEGER8	8	R_IN_CAN_INTEGER16	16	R_IN_CAN_INTEGER32	32	R_IN_CAN_UNSIGNED8	8	R_IN_CAN_UNSIGNED16	16	R_IN_CAN_UNSIGNED32	32	R_IN_CAN_VISIBLESTRING	PDO マッピング不可
データタイプ	ビット長																		
R_IN_CAN_INTEGER8	8																		
R_IN_CAN_INTEGER16	16																		
R_IN_CAN_INTEGER32	32																		
R_IN_CAN_UNSIGNED8	8																		
R_IN_CAN_UNSIGNED16	16																		
R_IN_CAN_UNSIGNED32	32																		
R_IN_CAN_VISIBLESTRING	PDO マッピング不可																		
6	1600H～17FFH 1A00H～1BFFH	PDO マッピング オブジェクト最大 数	GX Works3 では、RPDO および TPDO に割付可能なオブジェクト数は、最大 64 オブジェクトです。																
7	1600H～17FFH 1A00H～1BFFH	Padding オブ ジェクト	GX Works3 では、1 バイトの PDO オブジェクト※1 をマッピングすると、1 バイトの Padding オブジェクトが挿入されます。  上記の点を考慮し、 「R_IN_CAN_RPDO_APPLICATION_OBJECT_NUM」、および 「R_IN_CAN_TPDO_APPLICATION_OBJECT_NUM」により PDO マッピング可能な最大オブジェクト数を定義してください。																

【注】※1 usDataType (データタイプ)が「R\_IN\_CAN\_INTEGER8」(8ビット)、「R\_IN\_CAN\_UNSIGNED8」(8ビット)のオブジェクト。リンクデバイスのデータサイズは2バイト単位のため、1バイトのPDOオブジェクトをリンクデバイスにマッピングした場合は、リンクデバイスの残り1バイトを空白でマッピングするべく、パディングします。

#### (4) CANopen 通信時の増設ユニットについて

R-IN32M4-CL3 適用製品が、例えば多軸一体型サーボアンプのように、通信を行う「基本ユニット(軸1)」と通信を行わない「増設ユニット(軸2～)」で構成する機器の場合、増設ユニットを装着する際は以下の項目を設定してください。

表 5.31 増設ユニットの設定項目概要

No	項目	内容
1	軸数	R_IN_CAN_MAX_ODTABLE_NUM (R_IN32M4_CL3CanConst.h)に使用する軸数(1~8)を設定してください。 軸数は、通信を行う基本ユニットと、通信を行わない増設ユニットの合計以上の値としてください。
2	PDO を格納するアドレス	使用する軸の PDO の格納先アドレスを、R_IN32M4_CL3MemoryAddress.h で定義してください。※1 R_IN_MEMORY_ADDRESS_RWW_EXT1 ~ R_IN_MEMORY_ADDRESS_RWW_EXT7 R_IN_MEMORY_ADDRESS_RWR_EXT1 ~ R_IN_MEMORY_ADDRESS_RWR_EXT7 アドレスは、軸1→軸2→軸3…のアドレス順となるように配置してください。
3	ユーザプログラム	使用する軸数分の Object Dictionary を定義し、gerR_IN_CanInit 関数(CANopen 通信機能の初期化)の引数に設定してください。 デバイス局パラメータ自動設定に関する SLMP コマンド受信処理(5.7.1 項~5.7.7 項)に、必要に応じて各軸の処理を追加してください。
4	軸の指定	NMT および SDO において軸を指定する場合は、SLMP フレームの「要求先局プロセッササブ番号」を使用します。 軸1~8 は、要求先局プロセッササブ番号の0~7に割り当たります。 軸1：要求先局プロセッササブ番号0 軸2：要求先局プロセッササブ番号1 軸3：要求先局プロセッササブ番号2
5	オプション情報	増設ユニットの装着数に応じて、オプション情報に関連する以下の定義を変更してください。 詳細は「表 6.9R_IN_UNITINFO_T 一覧」を参照してください。 ・ USER_OPTN_INFOFLG ・ USER_NUMBER_OF_OPTION ・ R_IN_OPTIONTABLE_ENTRY_SIZE

【注】※1 Object Dictionary、サンプルコード、および CSP+ファイルのアドレス設定値が一致するようにしてください。

表 5.32 アドレス設定

軸	PDO	Object Dictionary の 設定値	サンプルコードの設定値 (R_IN32M4_CL3MemoryAddress.h)	CSP+ファイルの設定値 (0x1234_RemoteSample_CAN _Base_1_en.CSPP)
1	RPDO	OD(軸 1) : Index 1C00h、 SubIndex 4	R_IN_MEMORY_ADDRESS_RWW	PDOConfigMemoryAddress1
	TPDO	OD(軸 1) : Index 1C01h、 SubIndex 4	R_IN_MEMORY_ADDRESS_RWR	PDOConfigMemoryAddress2
2	RPDO	OD(軸 2) : Index 1C00h、 SubIndex 4	R_IN_MEMORY_ADDRESS_RWW_EXT1	EXT1_RPDOConfigMemoryAd dress
	TPDO	OD(軸 2) : Index 1C01h、 SubIndex 4	R_IN_MEMORY_ADDRESS_RWR_EXT1	EXT1_TPDOConfigMemoryAd dress
...	...	...	...	...
8	RPDO	OD(軸 8) : Index 1C00h、 SubIndex 4	R_IN_MEMORY_ADDRESS_RWW_EXT7	EXT7_RPDOConfigMemoryAd dress
	TPDO	OD(軸 8) : Index 1C01h、 SubIndex 4	R_IN_MEMORY_ADDRESS_RWR_EXT7	EXT7_TPDOConfigMemoryAd dress

## (5) CANopen 通信時のネットワーク同期通信について

ネットワーク同期通信は、受信データが毎通信周期更新されていること監視するために、ウォッチドッグカウンタ(WDC)を使用します。

CANopen 通信時における WDC は、Object Dictionary ごとに配置することができます。

PDO マッピング設定にて WDC のオブジェクトをマッピングすることで有効となります。

WDC のオブジェクトは、下記の Index/SubIndex を使用します。

表 5.33 WDC の Index/SubIndex 設定

オブジェクト	Index	SubIndex
Watchdog Counter DL 1	0x1D01	0x01
Watchdog Counter UL 1	0x1D02	0x01

## 5.9.1 CANopen 通信機能の初期化処理

CANopen 通信機能を初期化します。

Object Dictionary の先頭アドレス、要素数を R-IN32M4-CL3 ドライバへ通知します。

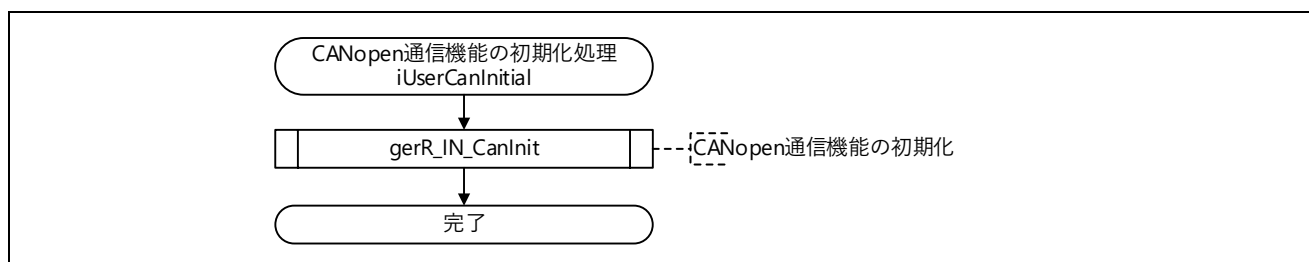


図 5.73 CANopen 通信機能の初期化処理フロー図

## (1) 初期化シーケンス

初期化処理からサイクリック伝送を開始するまでの流れを示します。

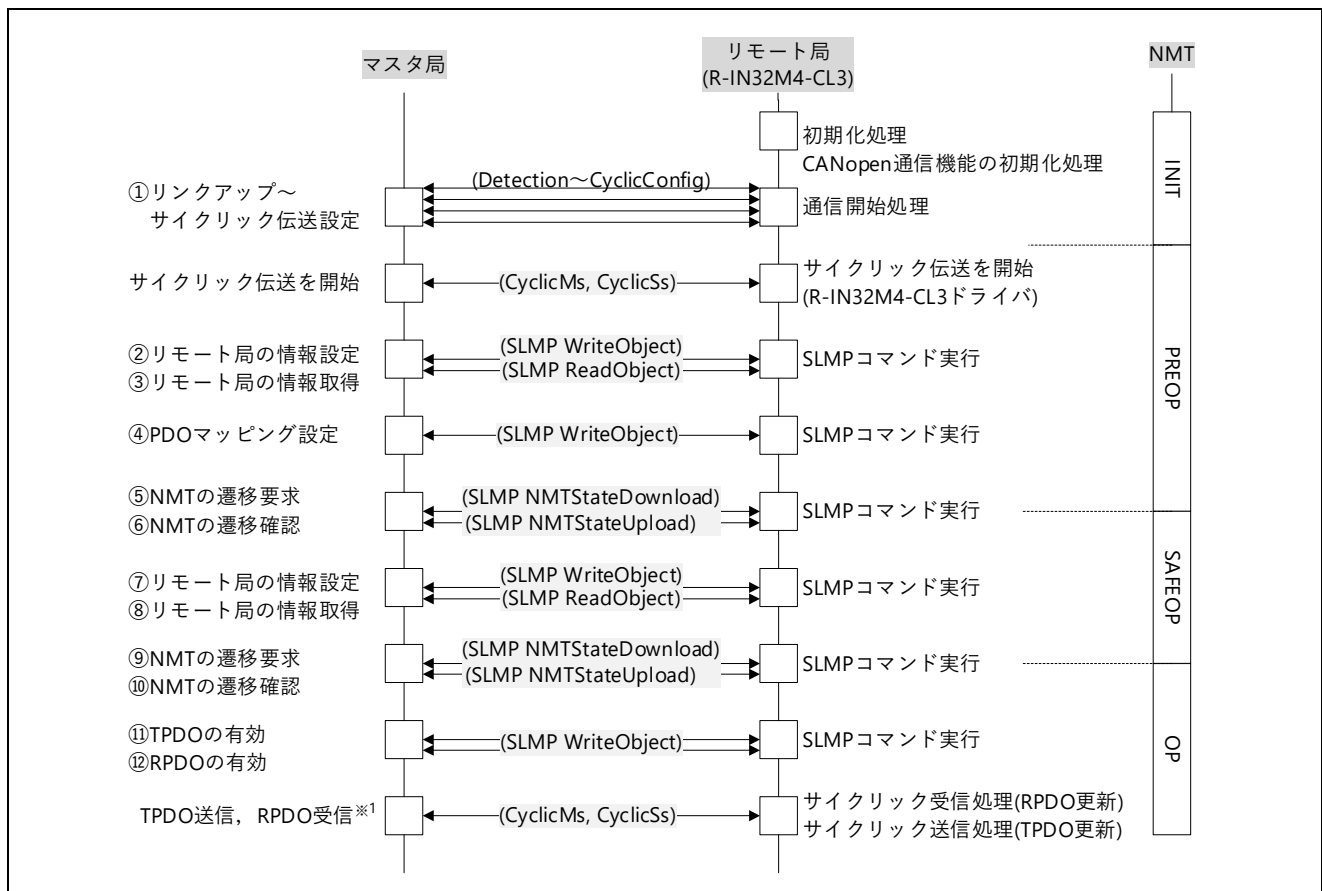


図 5.74 CANopen 通信時の初期化シーケンス(概略)

No	メンバ	参照
1	マスタ局はネットワーク立上げのための設定を行います。 リンクアップからサイクリック伝送設定までは、リンクデバイスを使用する場合と同じ手順です。	5.3.1 5.3.2
2	マスタ局は必要に応じて、リモート局(R-IN32M4-CL3)アプリケーションに初期設定を書き込みます。	5.9.5
3	マスタ局は必要に応じて、リモート局(R-IN32M4-CL3)アプリケーションの設定情報を読み出します。	5.9.4
4	マスタ局はユーザが設定した PDO の配置をリモート局(R-IN32M4-CL3)に書き込みます。	5.9.5
5	マスタ局はリモート局(R-IN32M4-CL3)の通信状態(NMT)を SAFEOP に変更します。	5.9.9
6	マスタ局はリモート局(R-IN32M4-CL3)の通信状態(NMT)が変更したことを確認します。	5.9.8
7	②と同様	-
8	③と同様	-
9	マスタ局はリモート局(R-IN32M4-CL3)の通信状態(NMT)を OP に変更します。	5.9.9
10	マスタ局はリモート局(R-IN32M4-CL3)の通信状態(NMT)が変更したことを確認します。	5.9.8
11	マスタ局はリモート局(R-IN32M4-CL3)の TPDO を有効にします。 完了後、リモート局(R-IN32M4-CL3)はマスタ局へ PDO を送信します。	5.9.5
12	マスタ局はリモート局(R-IN32M4-CL3)の RPDO を有効にします。 完了後、リモート局(R-IN32M4-CL3)はマスタ局からの PDO を受信します。	5.9.5

【注】※1 TPDO/RPDO のデータが有効となるのは SAFEOP と OP となります。

## (2) 増設ユニット装着時の NMT の遷移イメージ

R-IN32M4-CL3 適用製品が、例えば多軸一体型サーボアンプのように、通信を行う「基本ユニット(軸 1)」と通信を行わない「増設ユニット(軸 2～)」で構成する機器の場合、NMT は軸数分管理する必要があります。その際のリモート局における各軸の NMT 遷移は、R-IN32M4-CL3 ドライバが順次処理します。

基本ユニット(軸 1)に増設ユニット(軸 2～3)を装着する場合で、NMT の遷移イメージを示します。

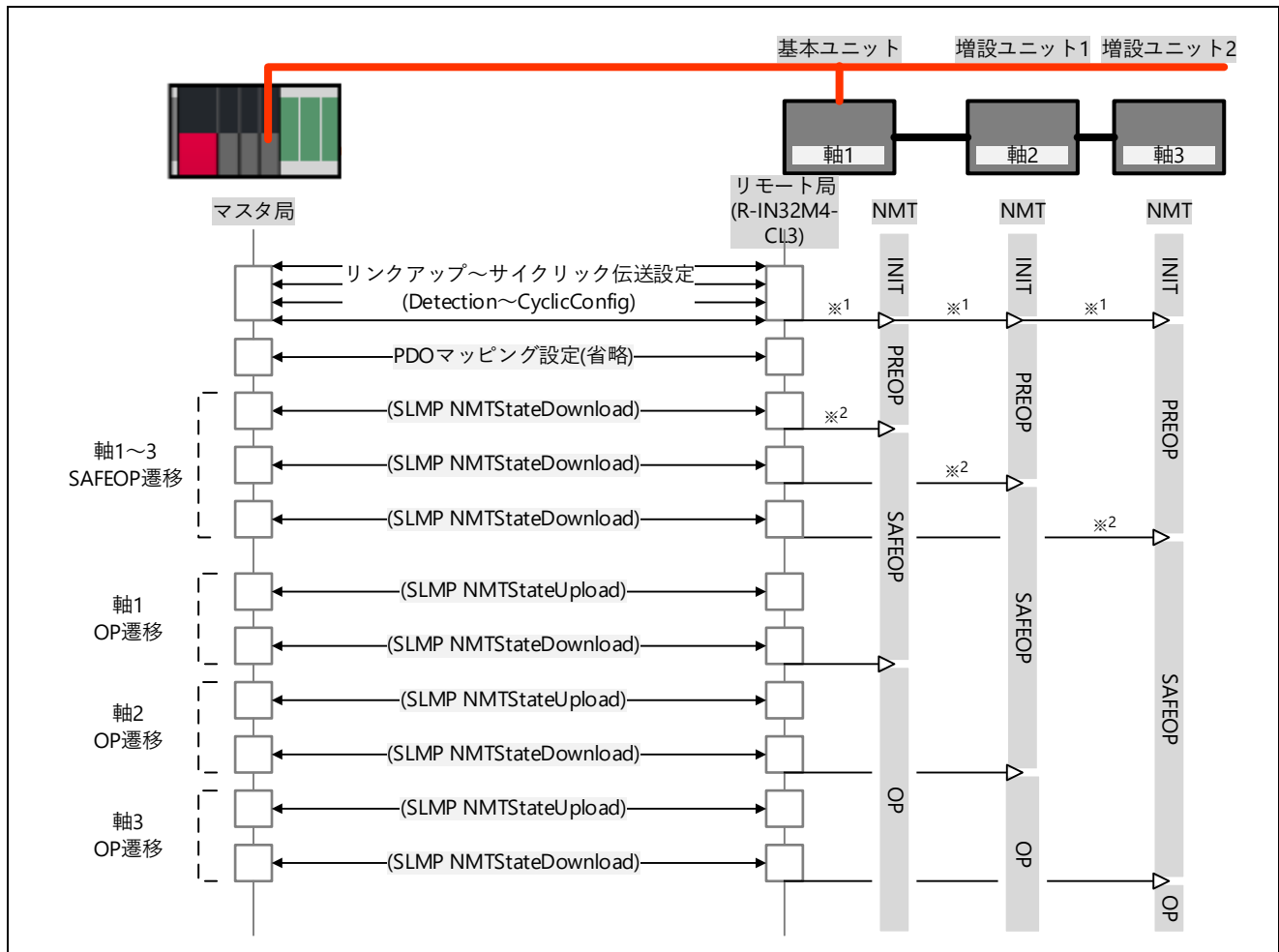


図 5.75 NMT の遷移イメージ(増設ユニット装着時)

NMT および SDO において軸を指定する場合は、SLMP フレームの「要求先局プロセッササブ番号」を使用します。本図における要求先局プロセッササブ番号の設定例を下記に示します。

軸 1 : 要求先局プロセッササブ番号 0

軸 2 : 要求先局プロセッササブ番号 1

軸 3 : 要求先局プロセッササブ番号 2

【注】 ※1 データリンクが可能になると、軸 1～3 のすべての NMT が INIT→PREOP へ遷移します。

※2 軸 1～3 のすべての NMT が SAFEOP もしくは OP の場合、RPDO の取得および TPDO の送信を行います。

PREOP 中に受信した PDO マッピングは、装着している軸の内、最後に PREOP から遷移(PREOP→SAFEOP もしくは PREOP→OP)する際に反映します。反映時に PDO マッピングの異常を検出した場合は、反映を中断し最初に異常となった軸の NMT を PREOP へ遷移させます。

### 5.9.2 サイクリック受信処理(RPDO 更新)

PDO マッピング設定に従い、受信した RWw のデータを Object Dictionary へ反映します。

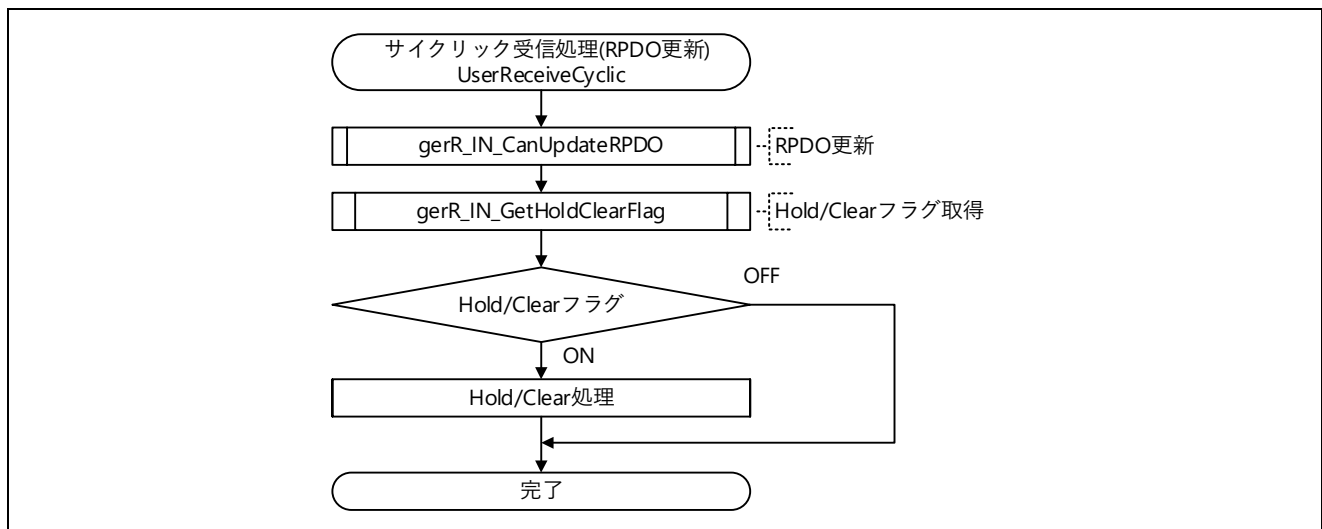


図 5.76 サイクリック受信処理(RPDO 更新)フロー図

定周期処理タスク(TSKID\_PERIODIC)の動作周期>サイクリック通信周期の条件下で RWw を複数回受信した場合、R-IN32M4-CL3 ドライバは定周期処理タスクの動作直前に受信した RWw のみを取得し、RPDO を Object Dictionary へ反映します。

#### ポイント

CANopen 通信時のサイクリック伝送機能は、アプリケーションの規模による負荷変動によって、送受信できる TPDO/RPDO 最大サイズが異なります。また、TPDO/RPDO データサイズによって、TPDO/RPDO を更新する時間も異なります。

#### ①送受信可能サイズ

定周期処理タスクが 200  $\mu$ s 周期で動作した場合で、送受信可能なサイズ(バイト)の目安を示します。  
(データはすべて 2 バイトの Object を割り当てています。)

ユーザプログラム	CCIETSN Class B	CCIETSN Class A	
	1Gbps/100Mbps	1Gbps	100Mbps
サイクリック送信処理(TPDO 更新) / サイクリック受信処理 (RPDO 更新)	128	108	92

#### ②TPDO/RPDO 更新時間

外部マイコン未使用の場合での、更新時間を示します。  
(データはすべて 2 バイトの Object を割り当てています。)

No	TPDO/RPDO データサイズ(バイト)	TPDO/RPDO を更新する時間( $\mu$ s)
1	128	171
2	192	226
3	256	281
4	384	391

### 5.9.3 サイクリック送信処理(TPDO 更新)

PDO マッピング設定に従い、Object Dictionary のデータを RWr へ反映し送信します。

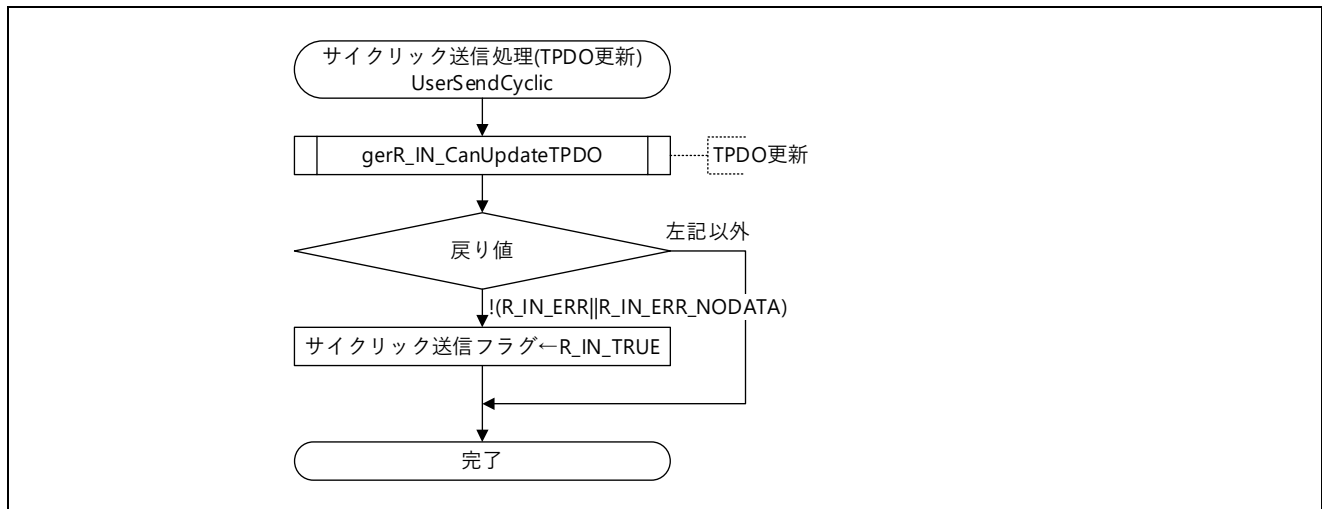


図 5.77 サイクリック送信処理(TPDO 更新)フロー図

定周期処理タスク(TSKID\_PERIODIC)の動作周期>サイクリック通信周期の条件下で TPDO を RWr へ設定すると、タイミングによっては、同じ TPDO データが複数回送信される場合があります。

#### ポイント

CANopen 通信時のサイクリック伝送機能は、アプリケーションの規模による負荷変動によって、送受信できる TPDO/RPDO 最大サイズが異なります。また、TPDO/RPDO データサイズによって、TPDO/RPDO を更新する時間も異なります。

#### ①送受信可能サイズ

定周期処理タスクが 200  $\mu$ s 周期で動作した場合で、送受信可能なサイズ(バイト)の目安を示します。  
(データはすべて 2 バイトの Object を割り当てています。)

ユーザプログラム	CCIETSN Class B	CCIETSN Class A	
	1Gbps/100Mbps	1Gbps	100Mbps
サイクリック送信処理(TPDO 更新) / サイクリック受信処理 (RPDO 更新)	128	108	92

#### ②TPDO/RPDO 更新時間

外部マイコン未使用の場合での、更新時間を示します。  
(データはすべて 2 バイトの Object を割り当てています。)

No	TPDO/RPDO データサイズ(バイト)	TPDO/RPDO を更新する時間( $\mu$ s)
1	128	171
2	192	226
3	256	281
4	384	391



### 5.9.4 SLMP ReadObject 要求コマンド受信処理

マスタ局からの SDO 読出し要求を受信し、応答を送信します。

指定された Index と SubIndex を基に、Object Dictionary の該当オブジェクトに格納されたデータを読み出します。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

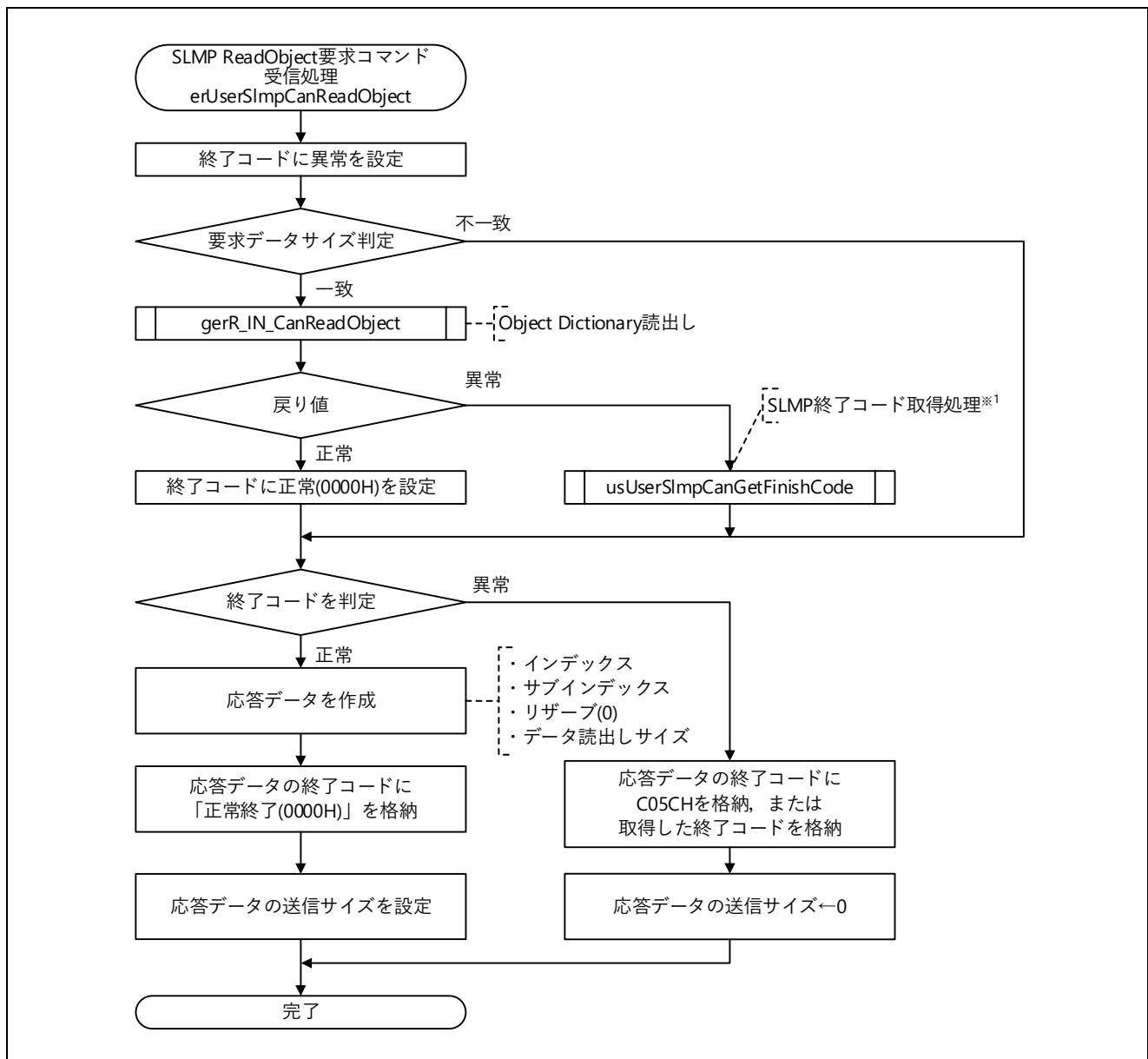


図 5.78 SLMP ReadObject 要求コマンド受信処理フロー図

【注】※1 「usUserSmpCanGetFinishCode」(5.9.10 SLMP 終了コード取得処理)を参照してください。

### 5.9.5 SLMP WriteObject 要求コマンド受信処理

マスタ局からの SDO 書き込み要求を受信し、応答を送信します。

指定された Index と SubIndex を基に、Object Dictionary の該当オブジェクトに指定されたデータを書き込みます。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

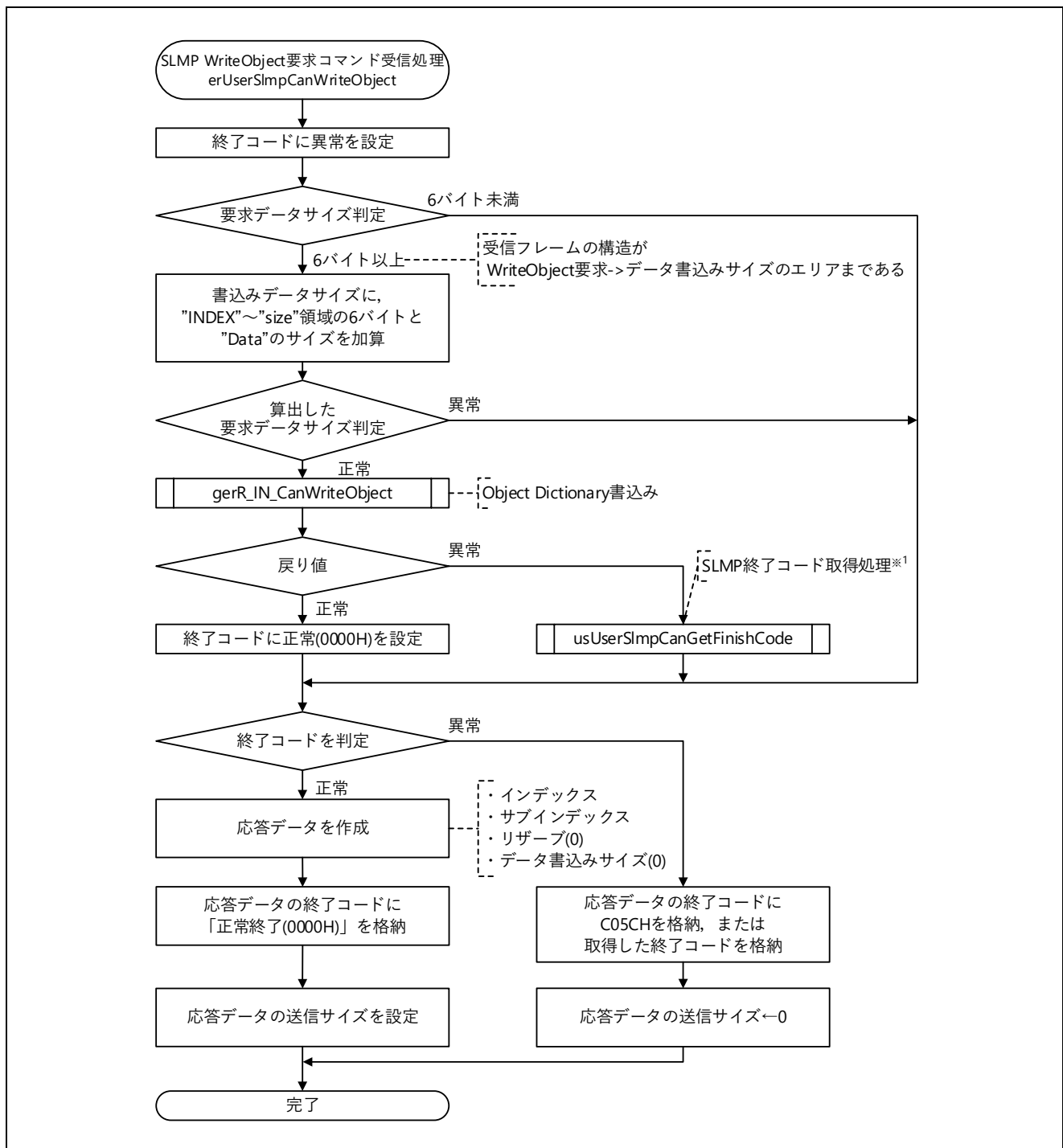


図 5.79 SLMP WriteObject 要求コマンド受信処理フロー図

【注】※1 「usUserSmpCanGetFinishCode」(5.9.10 SLMP 終了コード取得処理)を参照してください。

### 5.9.6 SLMP ObjectSubIDReadBlock 要求コマンド受信処理

マスタ局からの SDO 読出し要求を受信し、応答を送信します。

指定された Index と SubIndex を基に、Object Dictionary の該当オブジェクトから指定したサイズ分データを読み出します。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

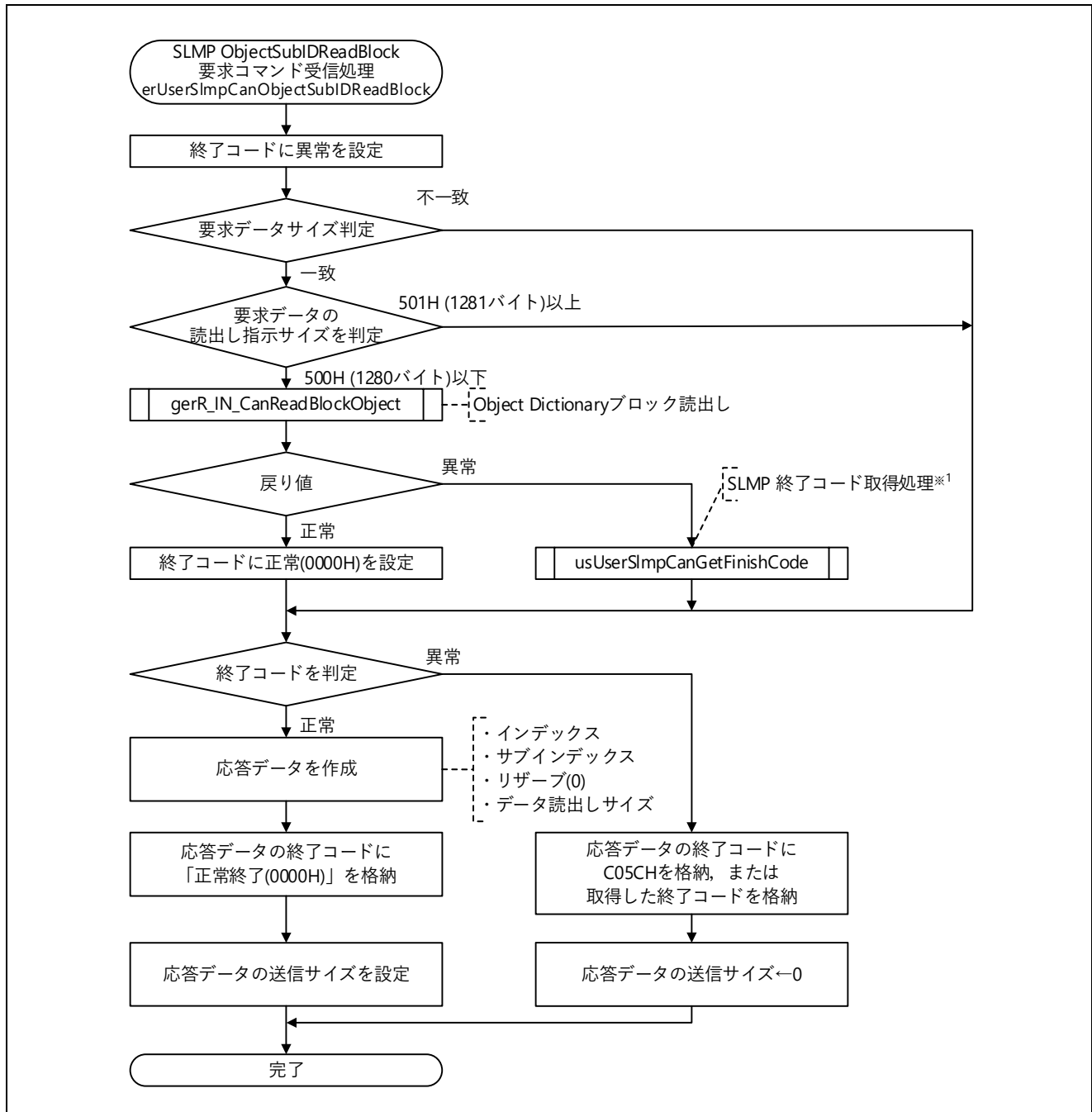


図 5.80 SLMP ObjectSubIDReadBlock 要求コマンド受信処理フロー図

【注】※1 「usUserSmpCanGetFinishCode」(5.9.10 SLMP 終了コード取得処理)を参照してください。

### 5.9.7 SLMP ObjectSubIDWriteBlock 要求コマンド受信処理

マスタ局からの SDO 書き込み要求を受信し、応答を送信します。

指定された Index と SubIndex を基に、Object Dictionary の該当オブジェクトから指定したサイズ分データを書き込みます。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

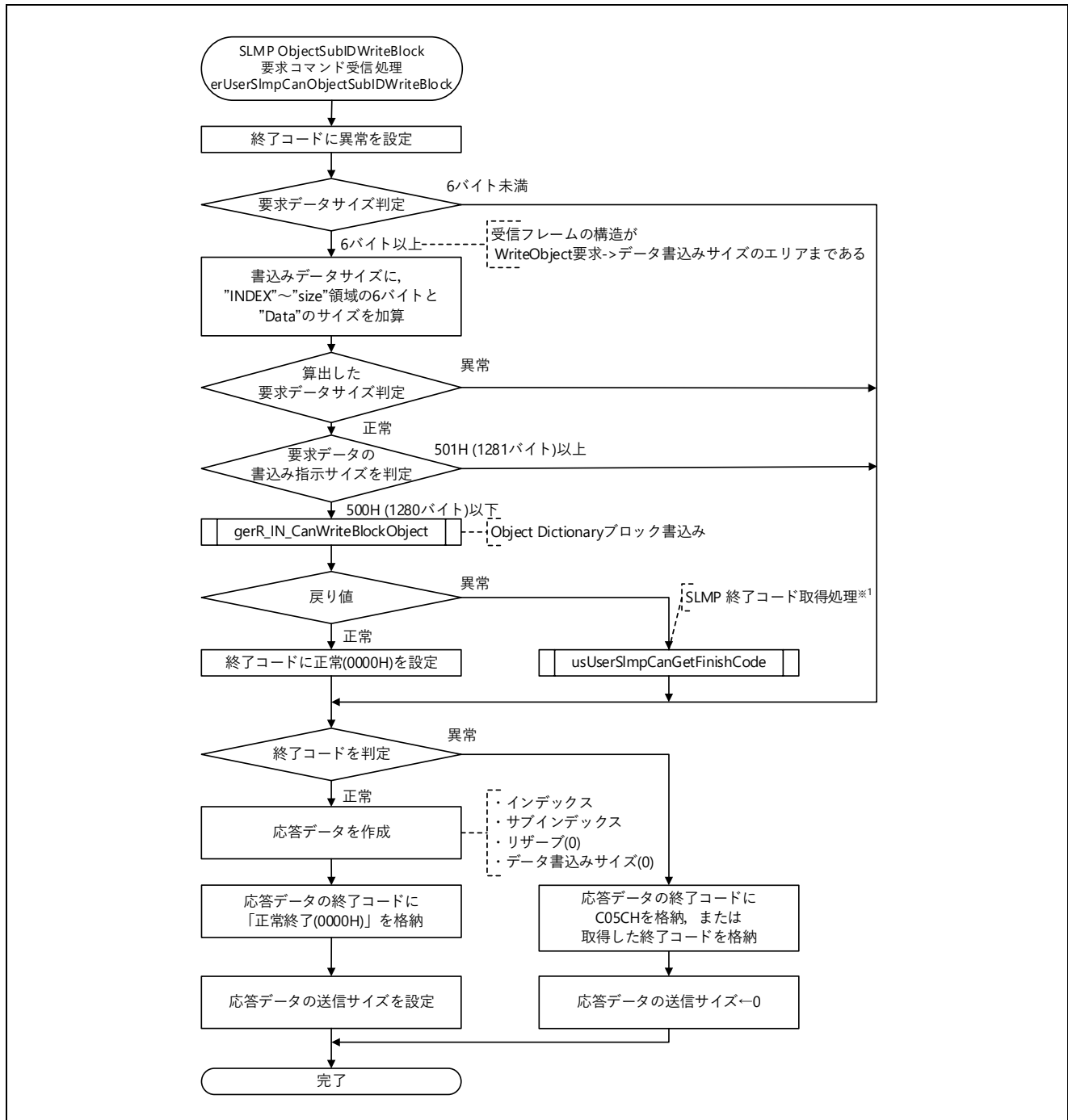


図 5.81 SLMP ObjectSubIDWriteBlock 要求コマンド受信処理フロー図

【注】※1 「usUserSmpCanGetFinishCode」(5.9.10 SLMP 終了コード取得処理)を参照してください。

### 5.9.8 SLMP NMTStateUpload 要求コマンド受信処理

自局の NMTState とマスタ局が最後に指定した NMTState を取得します。  
マスタ局が指定した NMTState がない場合、INIT を取得します。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

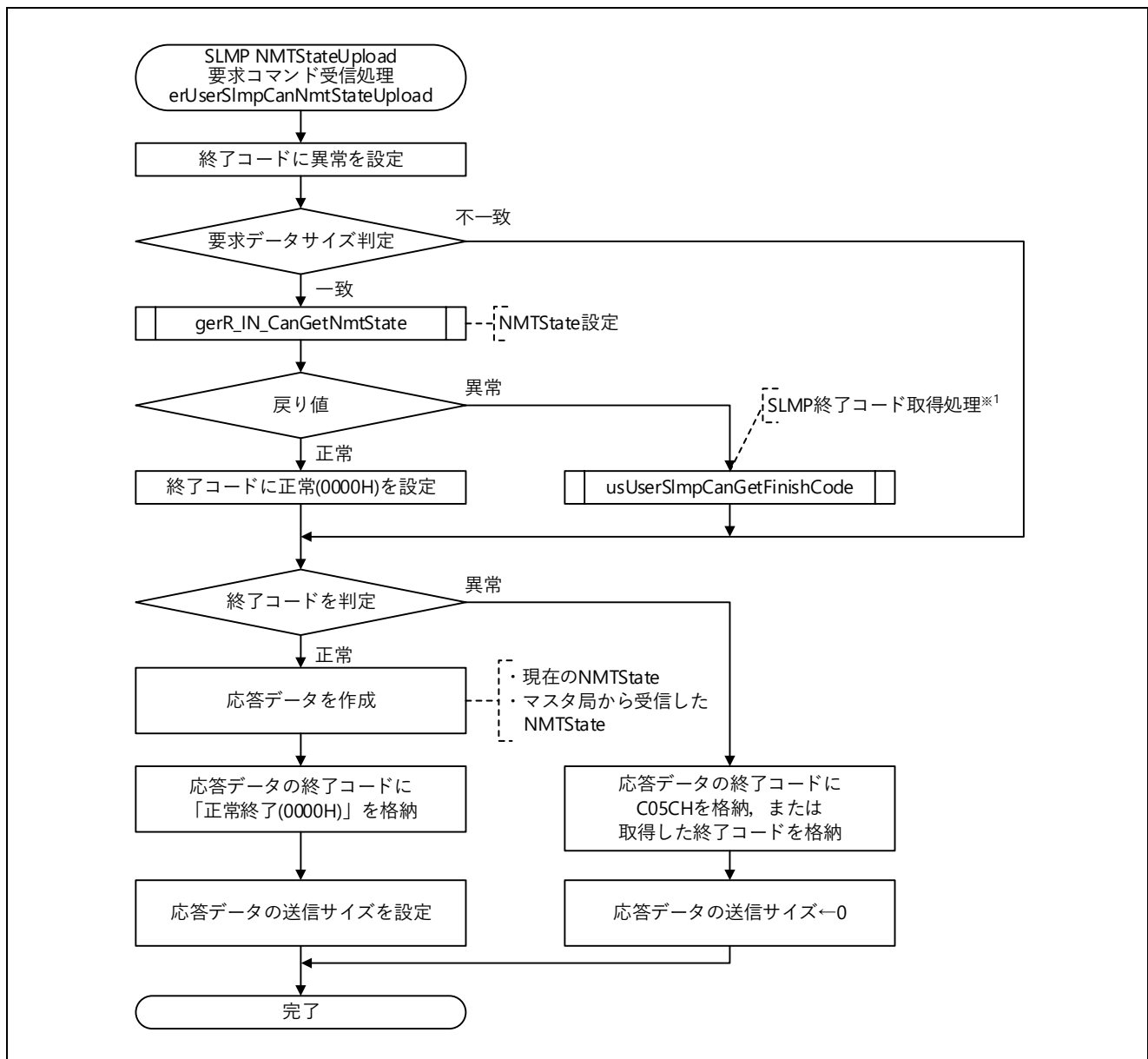


図 5.82 SLMP NMTStateUpload 要求コマンド受信処理フロー図

【注】※1 「usUserSlmpCanGetFinishCode」(5.9.10 SLMP 終了コード取得処理)を参照してください。

### 5.9.9 SLMP NMTStateDownload 要求コマンド受信処理

指定された NMTState を R-IN32M4-CL3 ドライバへ設定します。

本コマンド受信処理は、LMT 種別のフレームを受信します。フレーム種別指定と異なる種別のフレームを受信した場合、異常応答を送信し、受信データを破棄します。受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバが終了コードに C05CH を設定し応答します。そのため、本コマンド受信処理は実行されません。

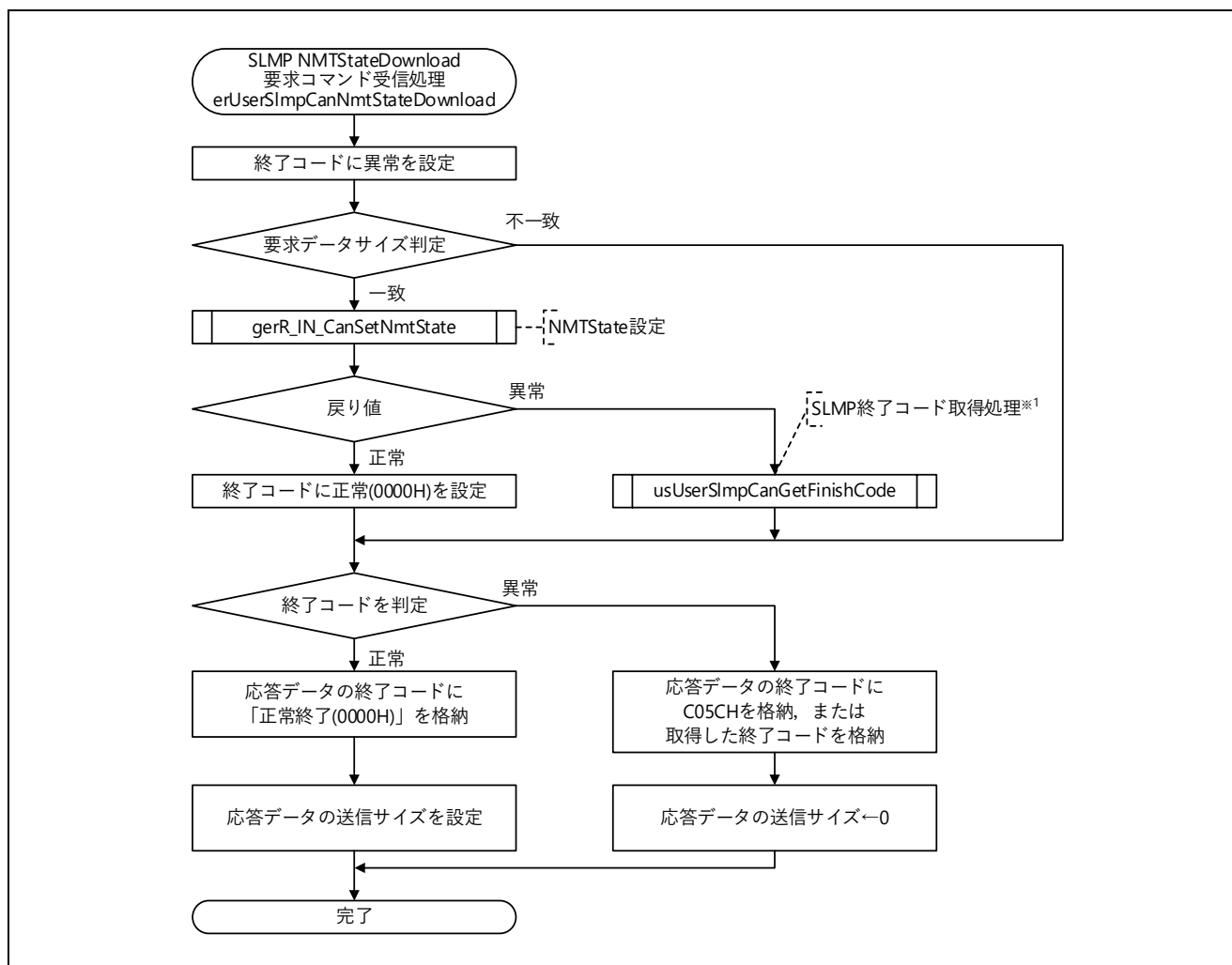


図 5.83 SLMP NMTStateDownload 要求コマンド受信処理フロー図

【注】※1 「usUserSmpCanGetFinishCode」(5.9.10 SLMP 終了コード取得処理)を参照してください。

### 5.9.10 SLMP 終了コード取得処理

指定された SDO アボートコードを SLMP の終了コードへ変換します。

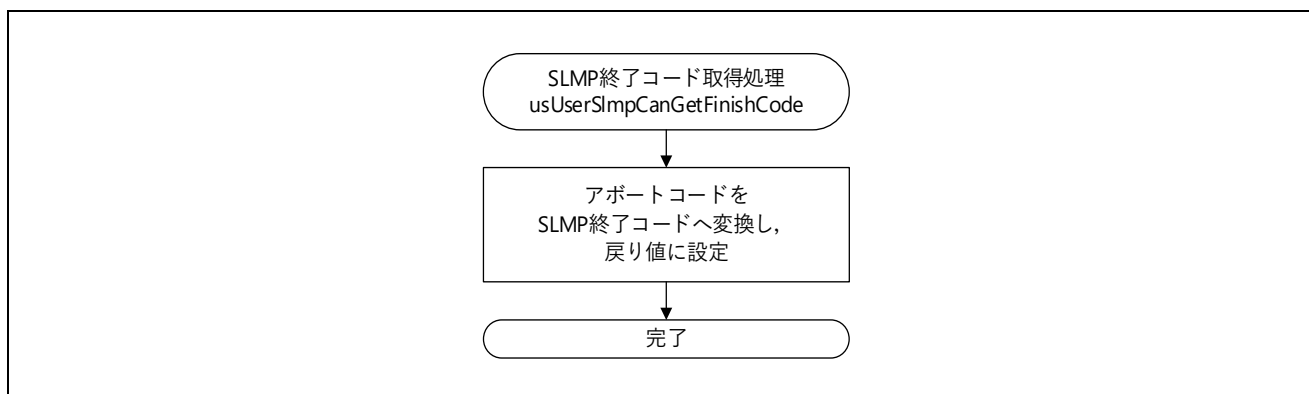


図 5.84 SLMP 終了コード取得処理フロー図

表 5.34 SDO アボートコードと SLMP 終了コードの対応表

No	アボートコード	説明	SLMP 終了コード	説明
1	0000 0000H	エラーなし	CCFFH	オブジェクトのアクセスエラー (エラー検出したが、アボートコードが設定されていない)
2	0503 0000H	トグルビット変更不可	CCFFH	オブジェクトのアクセスエラー
3	0504 0000H	SDO プロトコルタイムアウト	CCFFH	(同上)
4	0504 0001H	無効/不明なクライアント/サーバコマンド指定子	CCFFH	(同上)
5	0504 0005H	メモリ範囲外	CCFFH	(同上)
6	0601 0000H	サポートしていないオブジェクトへのアクセス	CCC7H	オブジェクトへのアクセス許可されていない条件下でオブジェクトへアクセスした
7	0601 0001H	書込み専用オブジェクトへのリードアクセス	CCC8H	ライトオンリーのオブジェクトにリードアクセスした
8	0601 0002H	読出し専用オブジェクトへのライトアクセス	CCC9H	リードオンリーのオブジェクトにライトアクセスした
9	0602 0000H	存在しないオブジェクトへのアクセス	CCCAH	Object Dictionary で定義されていない Index を指定した
10	0604 0041H	PDO マッピング不可 オブジェクトを PDO にマッピングできない	CCCBH	PDO マッピングを許可されていないオブジェクトをマッピングした
11	0604 0042H	PDO マッピングオブジェクト数およびデータ長が PDO のデータ長を超過	CCCCH	PDO マッピングするデータ数及びデータ長の合計がアプリケーションなどで定義された値を超えた
12	0604 0043H	一般的なパラメータ不一致	CCFFH	オブジェクトのアクセスエラー
13	0604 0047H	デバイスの一般的な内部不一致	CCFFH	(同上)
14	0606 0000H	ハードウェアエラー	CCFFH	(同上)
15	0607 0010H	SDO のデータ長が不一致	CCFFH	(同上)
16	0607 0012H	SDO のデータが長すぎる	CCFFH	(同上)
17	0607 0013H	SDO のデータが短すぎる	CCFFH	(同上)
18	0609 0011H	サブインデックスが存在しない	CCD3H	Object Dictionary で定義されていない SubIndex を指定した
19	0609 0030H	無効なパラメータ値(書込みのみ)	CCD4H	要求パラメータが範囲外
20	0609 0031H	書き込まれたパラメータ値が大きすぎる	CCD5H	パラメータ範囲より大きい値を設定した
21	0609 0032H	書き込まれたパラメータ値が小さすぎる	CCD6H	パラメータ範囲より小さい値を設定した
22	0609 0036H	最大値が最小値より小さい	CCFFH	オブジェクトのアクセスエラー
23	0800 0000H	一般的なエラー	CCFFH	(同上)
24	0800 0020H	アプリケーションへの読出し・書込み不可	CCDAH	アプリケーションがデータを転送または格納できない
25	0800 0021H	アプリケーションへの読出し・書込み不可 (ローカル制御による)	CCFFH	オブジェクトのアクセスエラー
26	0800 0022H	アプリケーションへの読出し・書込み不可 (現在デバイス状態による)	CCFFH	(同上)
27	0800 0023H	オブジェクトディクショナリが存在しない	CCFFH	(同上)
28	-	上記以外	CCFFH	(同上)



## 5.9.11 CANopen パラメータ設定処理

パラメータ自動設定機能で受信したパラメータを、CANopen パラメータ変数へ設定します。

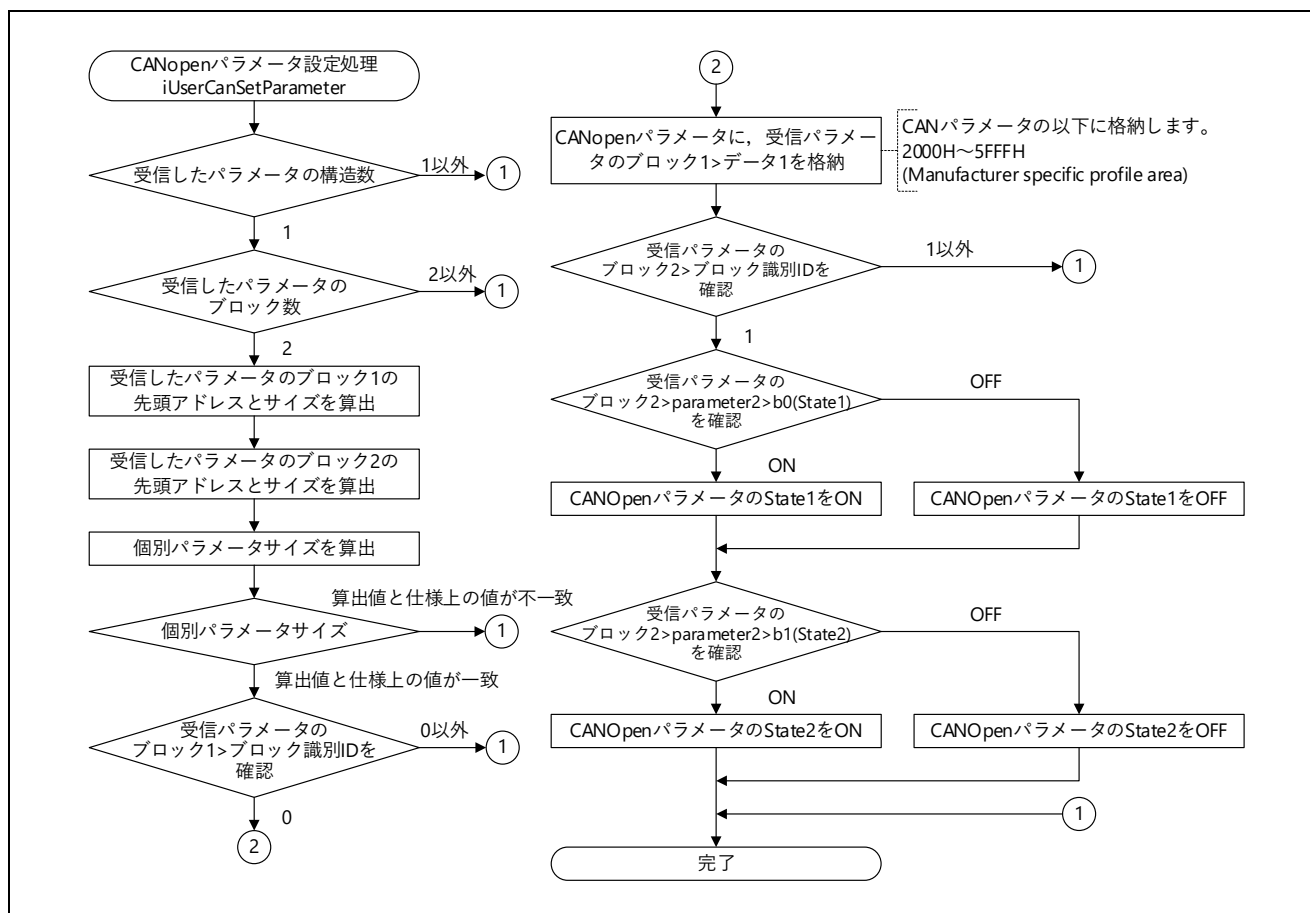


図 5.85 CANopen パラメータ設定処理フロー図

「CSPP」フォルダに同梱しているサンプル CSP+ファイルを使用した場合に、本処理の引数で受け渡される領域の構造を示します。

表 5.35 CANopen のパラメータ構造

No	項目		サイズ (バイト)	設定値	内容
1	オフセット		2	24 (Byte)	ファイルデータ本体部のデータサイズ
2	パラメータ構成		2	0001H	個別パラメータのみ使用
3	個別パラメータサイズ		2	18 (Byte)	「パラメータブロック数」のサイズ+「ブロック」のサイズ
4	パラメータブロック数		2	2	パラメータブロック数
5	ブロック 1	ブロック識別 ID	4	00000000H	ブロックを識別するための ID
6		データ長	2	1 (word)	「データ 1」～「データ n」のサイズ (word)
7		データ 1	2	Parameter1	GX Works3 で設定する「Parameter1」の設定値が格納されています。CANopen パラメータへ格納します。
8	ブロック 2	ブロック識別 ID	4	00000000H	ブロックを識別するための ID
9		データ長	2	1 (word)	「データ 1」～「データ n」のサイズ (word)
10		データ 2	2	Parameter2 b0 : State1 b1 : State2 b2-15 : 未使用	GX Works3 で設定する「Parameter2」の設定値が格納されています。CANopen パラメータへ格納します。

### 5.9.12 Index1010 書込み処理

Object Dictionary に登録し Index1010 に書込みが発生した場合に呼び出される処理です。

CANopen 通信では、Index1010/SubIndex1 に「save (65766173H)」を書き込むと、デバイス上の全てのパラメータを不揮発性メモリへ保存する仕様となっています。(パラメータの保存処理自体はユーザ実装)

R-IN32M4-CL3 適用製品が、例えば多軸一体型サーボアンプのように、通信を行う「基本ユニット(軸 1)」と通信を行わない「増設ユニット(軸 2～)」で構成する機器の場合、軸数に応じて本関数を追加してください。(最大 8 関数まで)

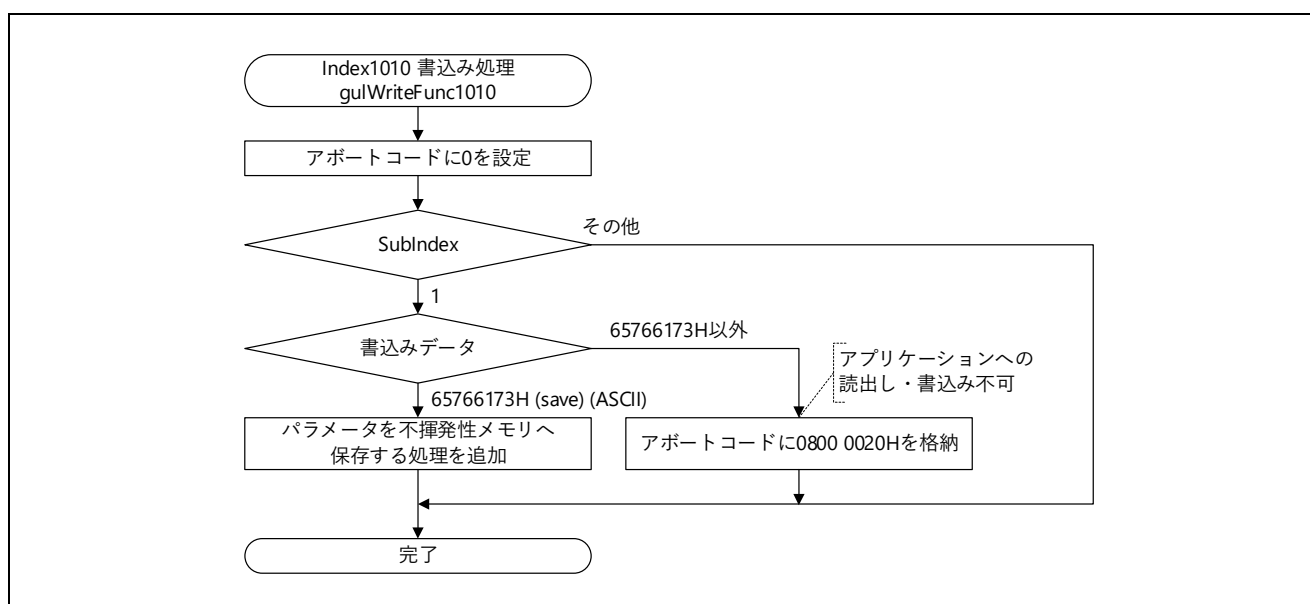


図 5.86 Index1010 書込み処理フロー図

### 5.9.13 Index1011 書込み処理

Object Dictionary に登録し Index1011 に書込みが発生した場合に呼び出される処理です。

CANopen 通信では、Index1011/SubIndex1 に「load (64616F6CH)」を書き込むと、デバイス上の全てのパラメータをデフォルト値に復元する仕様となっています。(パラメータの復元処理自体はユーザ実装)

R-IN32M4-CL3 適用製品が、例えば多軸一体型サーボアンプのように、通信を行う「基本ユニット(軸 1)」と通信を行わない「増設ユニット(軸 2～)」で構成する機器の場合、軸数に応じて本関数を追加してください。(最大 8 関数まで)

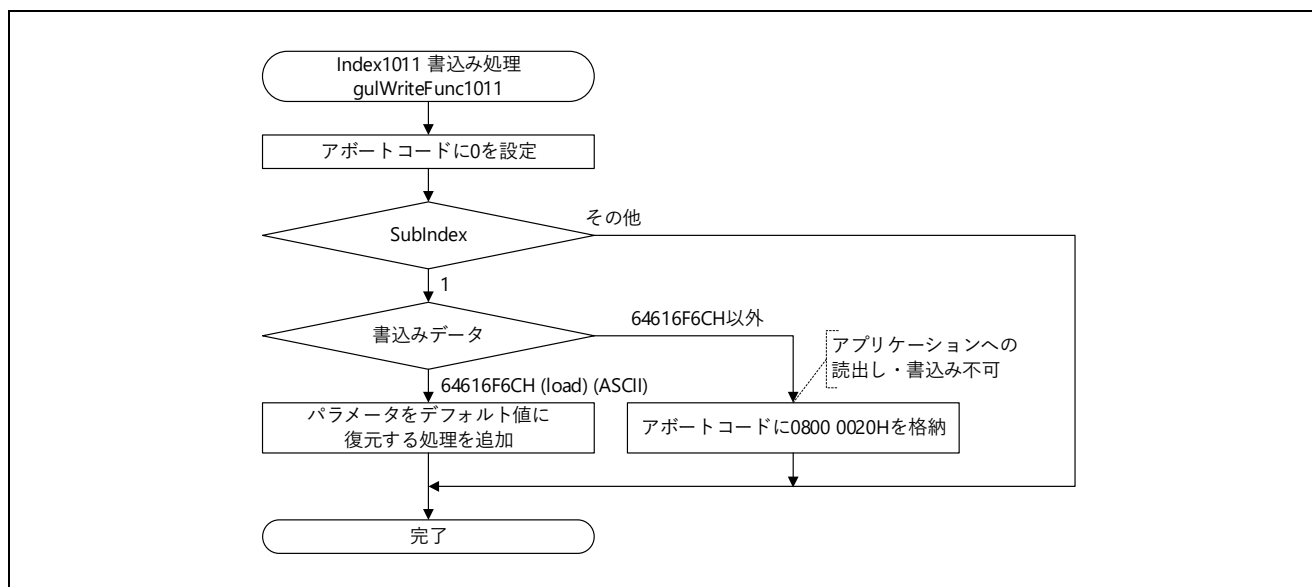


図 5.87 Index1011 書込み処理フロー図

## 5.10 ユーザプログラム詳細(MCU 間 I/F 関連)

### (1) MCU 間 I/F 機能の概要

MCU 間 I/F 機能は、通信用 MCU(R-IN32M4-CL3 内部)と安全用 MCU(外部)間で安全 PDU を交信する機能です。  
GPIO 通信(受信)、GPIO 通信(送信)、シリアル通信(RT DMA 転送)で構成されます。

通信イメージと使用用途を示します。

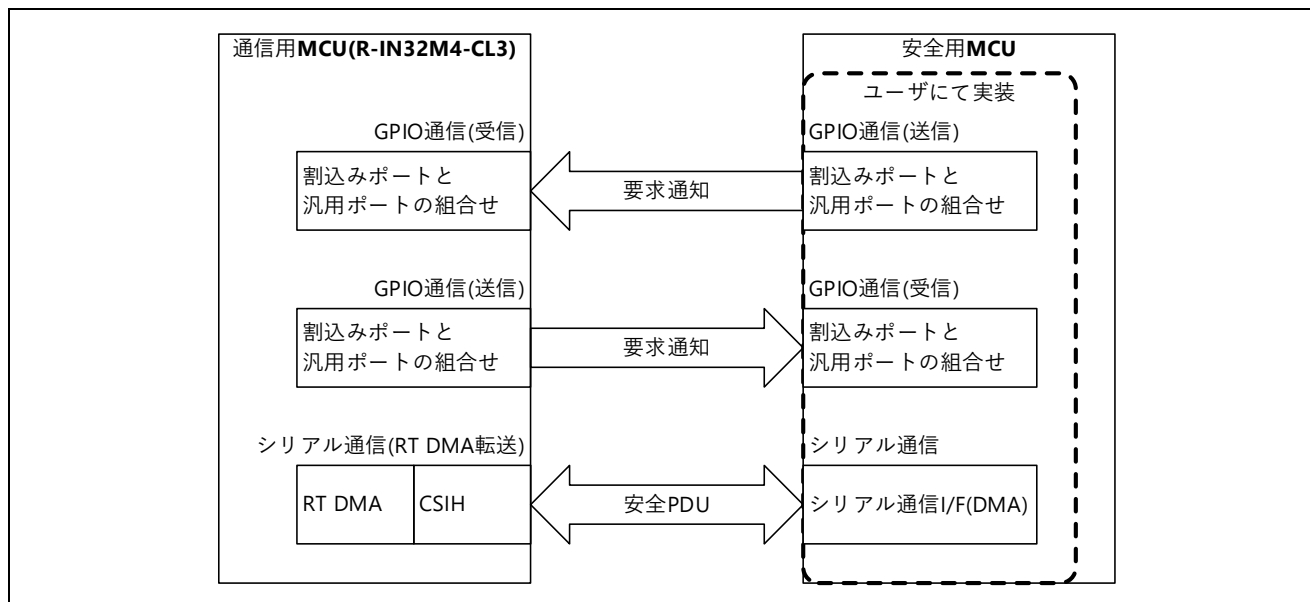


図 5.88 MCU 間 I/F のイメージ

表 5.36 MCU 間 I/F の使用用途一覧

No.	項目	内容
1	GPIO 通信(受信用)	安全用 MCU から通信用 MCU に対し、要求通知を行う際に使用。
2	GPIO 通信(送信用)	通信用 MCU から安全用 MCU に対し、要求通知を行う際に使用。
3	シリアル通信(RT DMA 転送)	安全 PDU の値をシリアル送信、もしくはシリアル受信する際に使用。

## (2) GPIO 通信

GPIO 通信は 1 本の割り込みポートと 11 本の汎用ポートの組合せで構成します。

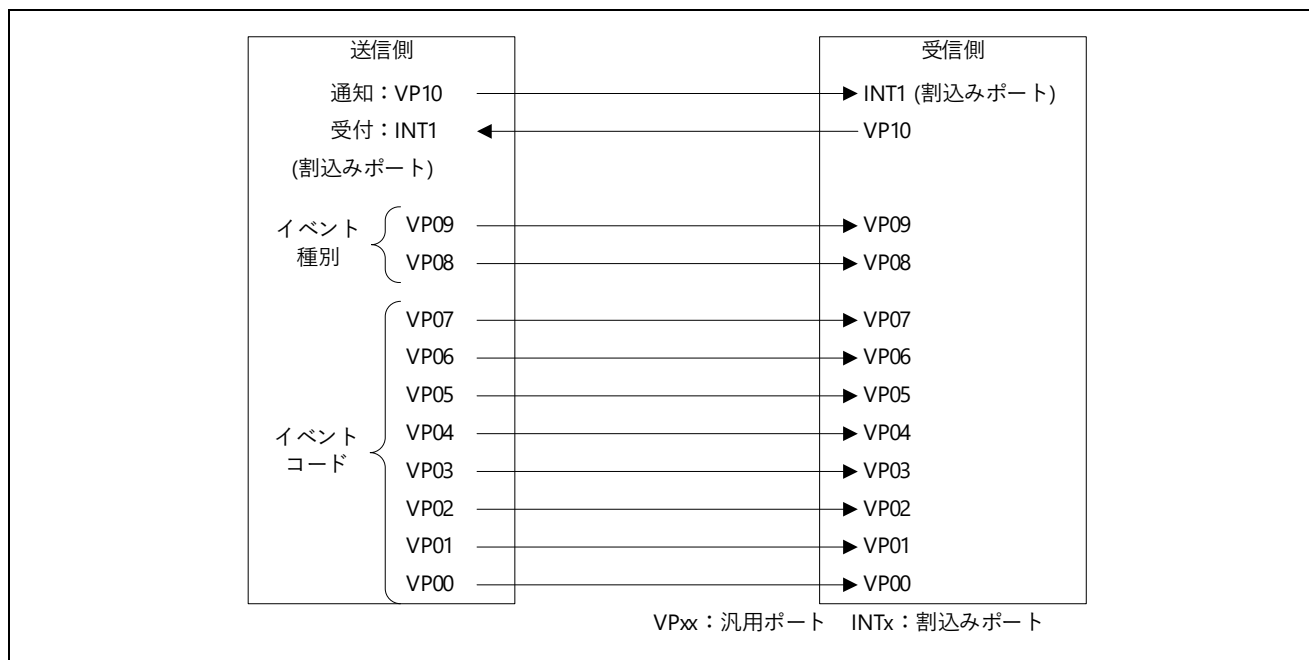


図 5.89 GPIO 通信の構成イメージ

GPIO 通信で使用するポートの割当を以下に記載します。

表 5.37 GPIO 通信のポート割当

信号種別		信号割当て					
		通信用 MCU (R-IN32M4-CL3)	方向	安全用 MCU	安全用 MCU	方向	通信用 MCU (R-IN32M4-CL3)
通知	VP10	任意の汎用ポート	→	任意の割り込みポート	任意の汎用ポート	→	割り込みポート INTPZ15※1
受付	INT1	割り込みポート INTPZ14※1	←	任意の汎用ポート	任意の割り込みポート	←	任意の汎用ポート
イベント 種別	VP09 VP08	任意の汎用ポート	→	任意の汎用ポート	任意の汎用ポート	→	任意の汎用ポート
イベント コード	VP07 VP06 VP05 VP04 VP03 VP02 VP01 VP00	任意の汎用ポート	→	任意の汎用ポート	任意の汎用ポート	→	任意の汎用ポート

【注】※1 割り込みが発生すると GPIO 通信応答受信タスクを起動します。

※2 割り込みが発生すると GPIO 通信受信タスクを起動します。

本シーケンスで使用するイベント種別、イベントコードの値を以下に示します。

表 5.38 内部 MCU(通信用 MCU)が受信するイベント種別、イベントコードの一覧

イベント種別	イベントコード	コマンド
01b	0000 0001b	安全 PDU 転送要求通知(MCU 内部→外部)
01b	0000 0010b	安全 PDU 転送要求通知(MCU 内部←外部)
上記以外		未使用

表 5.39 内部 MCU(通信用 MCU)が送信するイベント種別、イベントコードの一覧

イベント種別	イベントコード	コマンド
01b	0000 0001b	安全 PDU 転送要求通知(MCU 内部→外部)
01b	0000 0010b	安全 PDU 転送要求通知(MCU 内部←外部)
上記以外		未使用

GPIO 通信時のタイミングチャートを以下に記載します。

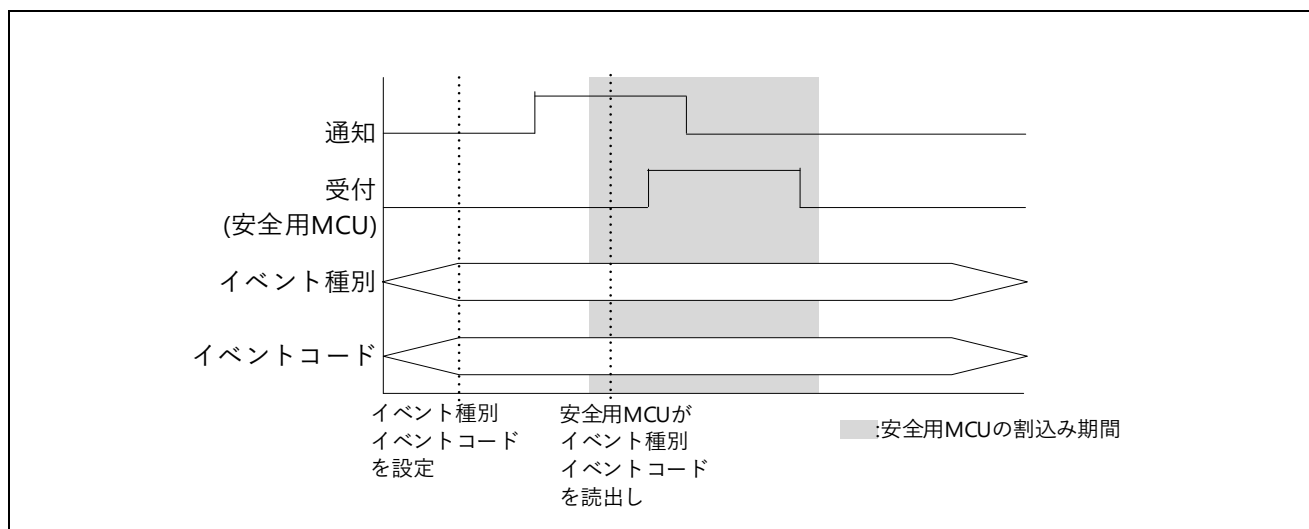


図 5.90 GPIO 通信のタイミング

No.	通信用 MCU の送信処理概要 (要求通知を送信する)
1	イベント種別、イベントコードの汎用ポートに ON/OFF を設定。
2	通知の汎用ポートを OFF→ON→OFF に設定。 受信側は、割り込みポートが OFF→ON することで、要求通知を検出。
3	受付の割り込みポートの OFF→ON を検出。
4	GPIO 通信の終了処理を実施。

No.	通信用 MCU の受信処理概要 (要求通知を受信する)
1	通知の割り込みポートの OFF→ON を検出。
2	汎用ポートからイベント種別、イベントコードを読み出す。
3	受付の汎用ポートを OFF→ON に設定。
4	読み出したイベント種別、イベントコードを基に該当のコマンド処理を実行。
5	受付の汎用ポートを ON→OFF に設定。

## (3) シリアル通信(RT DMA 転送)

シリアル通信には R-IN32M4-CL3 のクロック同期シリアル・インタフェース H を使用します。

クロック同期シリアル・インタフェース H の詳細は、「R-IN32M4-CL3 ユーザズ・マニュアルハードウェア編」を参照してください。

表 5.40 MCU 間 I/F で使用するシリアル通信の通信仕様

通信方式	伝送速度	データ長	DMA	使用割込み
同期式	7.14Mbps	<ul style="list-style-type: none"> <li>・ 8bit</li> <li>・ パリティなし</li> <li>・ StopBit=1</li> </ul>	RT DMA	DMA 転送完了割込み

シリアル通信で送受信するデータ構造の概要を以下に記載します。

安全 PDU の構造の詳細は、CC-Link 協会の「CC-Link IE 安全通信機能仕様書」を参照してください。

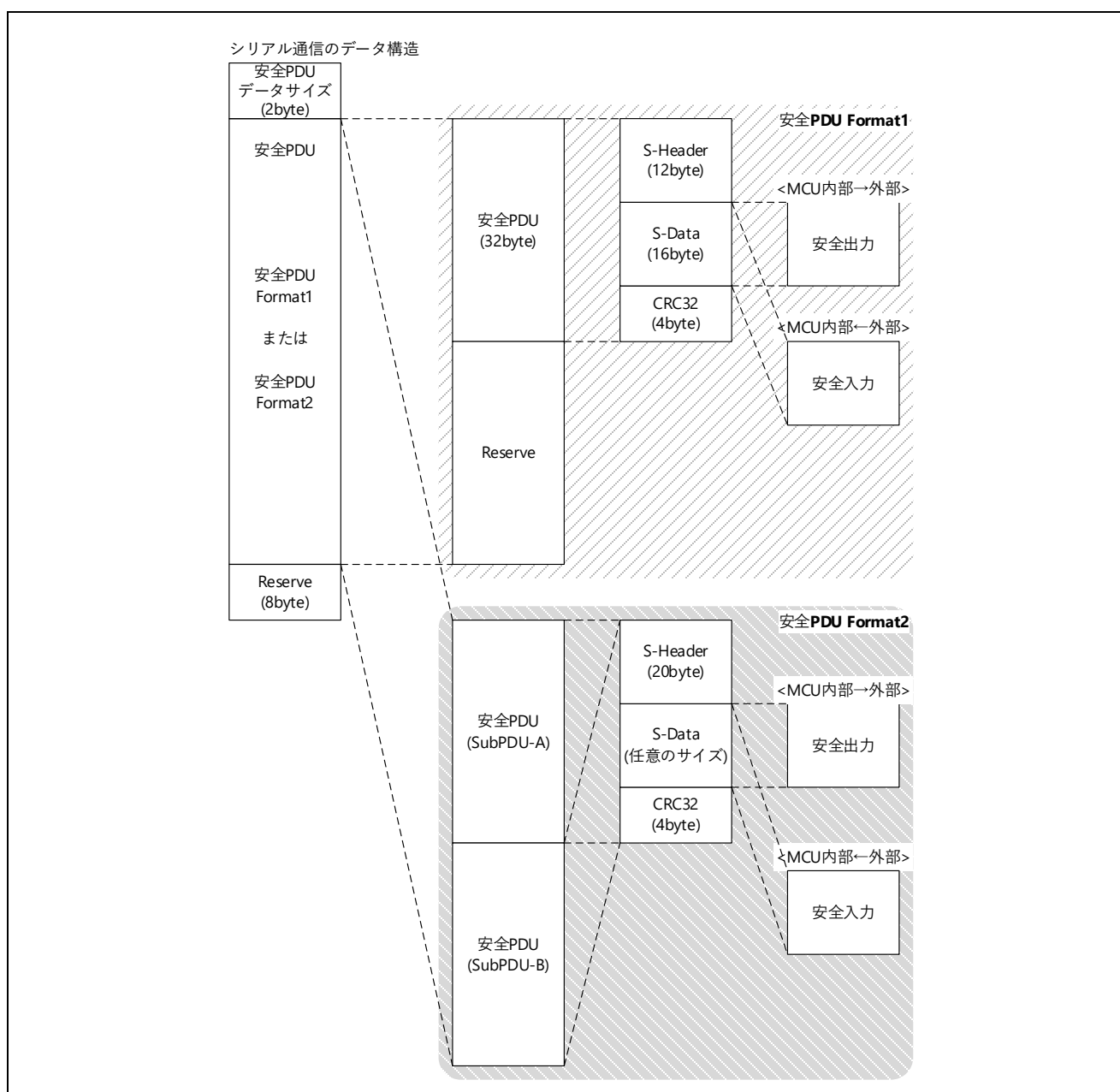


図 5.91 シリアル通信(RT DMA 転送)の安全 PDU 構造(概要)

R-IN32M4-CL3 ドライバに安全 PDU を設定する際は、安全 PDU 全体<sup>※1</sup>を設定してください。

また、R-IN32M4-CL3 ドライバを使用して安全 PDU を取得する際は、安全 PDU 全体<sup>※1</sup>を読み出します。

【注】※1 安全 PDU Format1 の場合 S-Header から CRC32 ままでが対象です。

安全 PDU Format2 の場合 SubPDU-A の先頭から SubPDU-B 末尾までが対象です。

#### (4) MCU 間シーケンス

MCU 間において安全 PDU を送受信する際のシーケンスを以下に示します。

##### (a) 内部 MCU から外部 MCU へ安全 PDU を転送

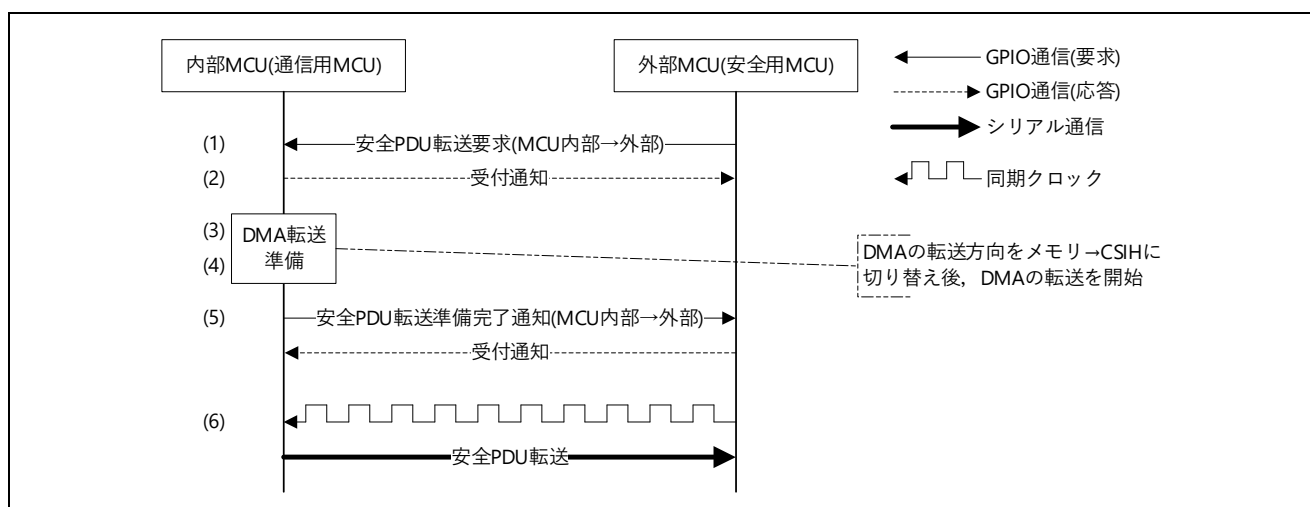


図 5.92 内部 MCU から外部 MCU へ安全 PDU を転送

No.	内部 MCU の処理	参照
1	GPIO 通信(受信)の通知が ON されたことを検出する。	
2	GPIO 通信(受信)で受信したイベント種別、イベントコードを読み出し、受付を ON する。	5.10.2
3	読み出したイベント種別、イベントコードを基に DMA 転送準備を実行する。	5.10.5
4	DMA 転送準備完了後、受付を OFF する。	5.10.3
5	GPIO 通信(送信)を使用し安全 PDU 転送準備が完了したことを通知する。 (イベント種別、イベントコードを設定後、通知を OFF→ON→OFF する)	5.10.4 5.10.7
6	同期クロックを受信することで、No.3 の DMA 転送準備で設定したメモリの値が送信される。	



## (b) 外部 MCU から内部 MCU へ安全 PDU を転送

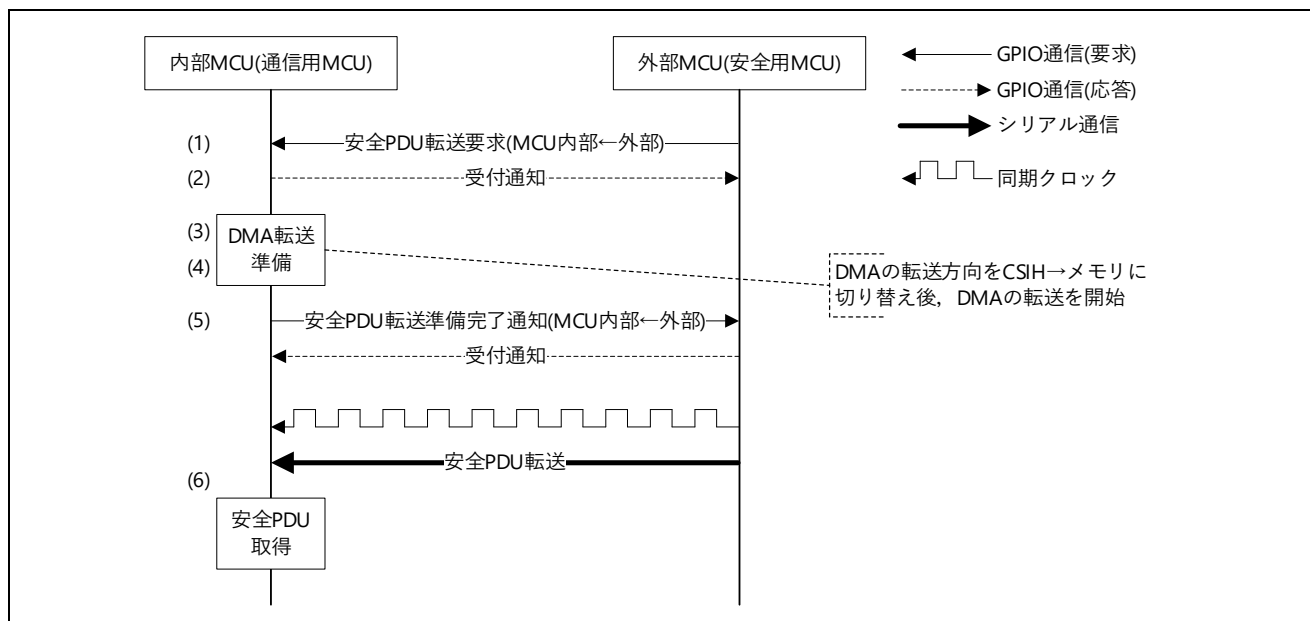


図 5.93 外部 MCU から内部 MCU へ安全 PDU を転送

No.	内部 MCU の処理	参照
1	GPIO 通信(受信)の通知が ON されたことを検出する。	
2	GPIO 通信(受信)で受信したイベント種別、イベントコードを読み出し、受付を ON する。	5.10.2
3	読み出したイベント種別、イベントコードを基に DMA 転送準備を実行する。	5.10.6
4	DMA 転送準備完了後、受付を OFF する。	5.10.3
5	GPIO 通信(送信)を使用し安全 PDU 転送準備が完了したことを通知する。 (イベント種別、イベントコードを設定、通知を OFF→ON→OFF する)	5.10.4 5.10.8
6	安全 PDU を受信完了後、No.3 の DMA 転送準備で設定したメモリから安全 PDU を取得する。	5.10.9

## 5.10.1 MCU 間 I/F 初期化処理

GPIO 通信関数、およびシリアル通信完了時実行関数の関数ポインタを R-IN32M4-CL3 ドライバへ通知します。

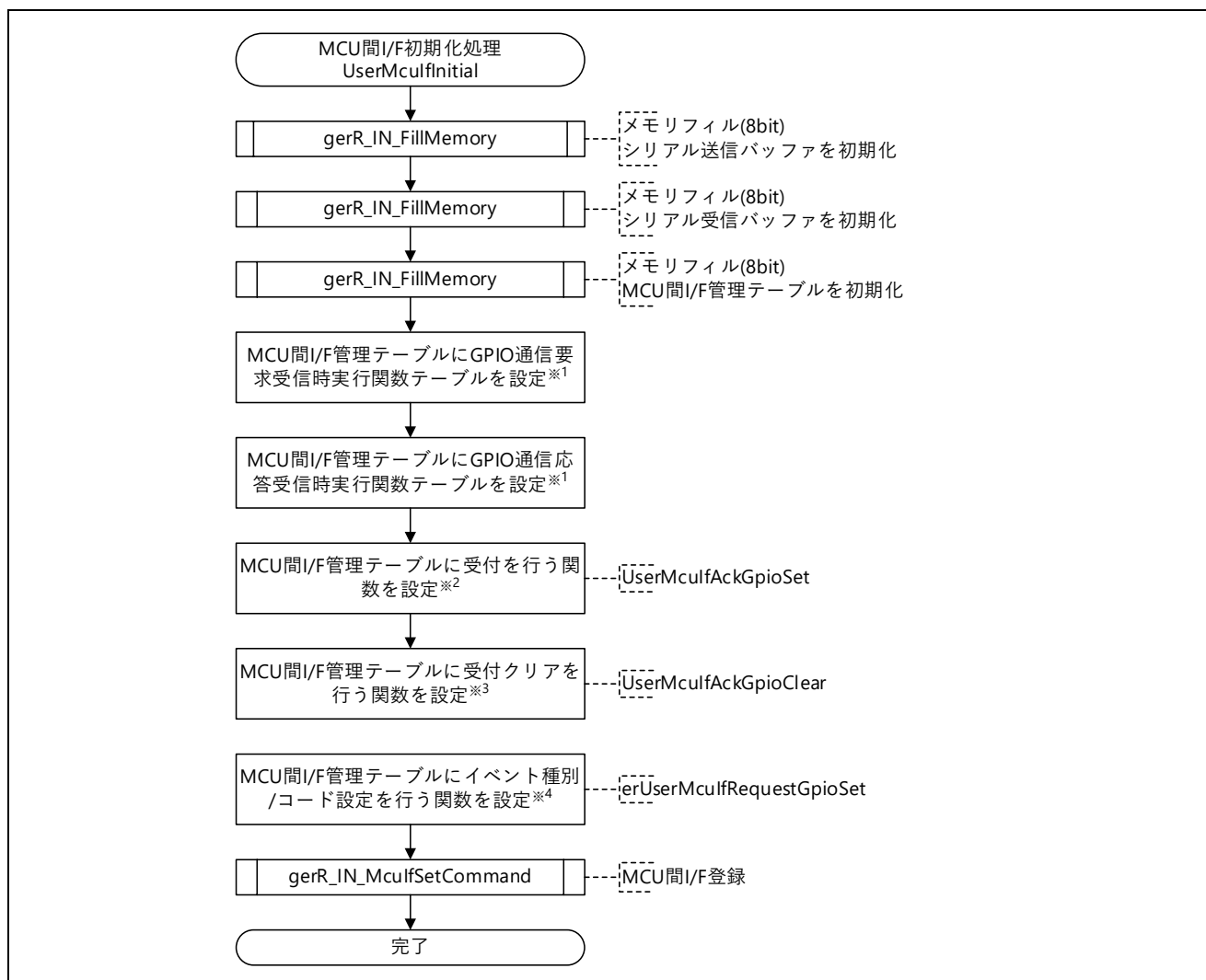


図 5.94 MCU 間 I/F 初期化処理フロー図

【注】※1 詳細は「表 6.42 R\_IN\_MCU\_IF\_GPIO\_MANAGEMENT\_TBL\_T」を参照してください。

※2 UserMculfAckGpioSet は「5.10.2 GPIO 通信受付処理」を参照してください。

※3 UserMculfAckGpioClear は「5.10.3 GPIO 通信受付クリア処理」を参照してください。

※4 erUserMculfRequestGpioSet は「5.10.4 GPIO 通信イベント種別/コード設定処理」を参照してください。

### 5.10.2 GPIO 通信受付処理

ユーザが任意に割り当てた GPIO からイベント種別、イベントコードを読み出し、受付を ON します。  
H/W 構成に合わせてユーザにて実装してください。

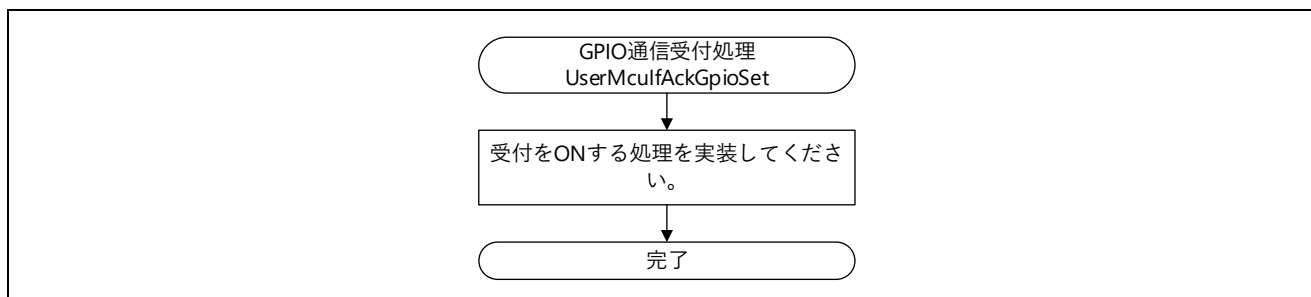


図 5.95 GPIO 通信受付処理フロー図

### 5.10.3 GPIO 通信受付クリア処理

ユーザが任意に割り当てた GPIO から受付を OFF します。  
H/W 構成に合わせてユーザにて実装してください。

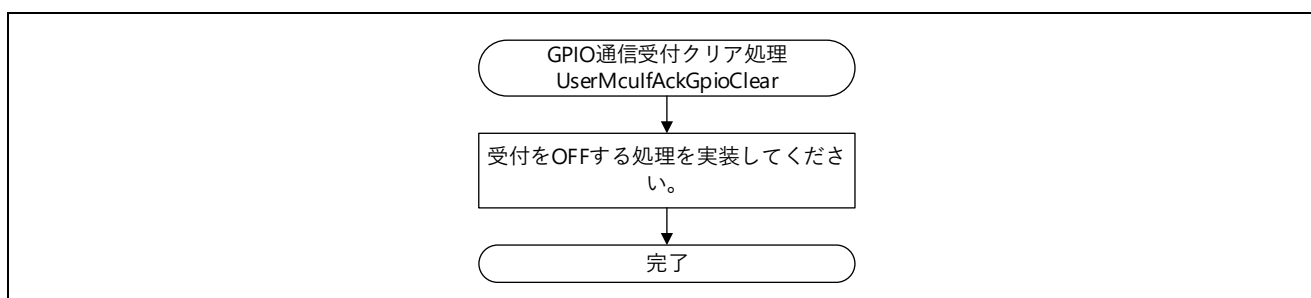


図 5.96 GPIO 通信受付クリア処理フロー図

#### 5.10.4 GPIO 通信イベント種別/コード設定処理

ユーザが任意に割り当てた GPIO からイベント種別、イベントコードを設定し、通知信号を OFF→ON→OFF します。  
H/W 構成に合わせてユーザにて実装してください。

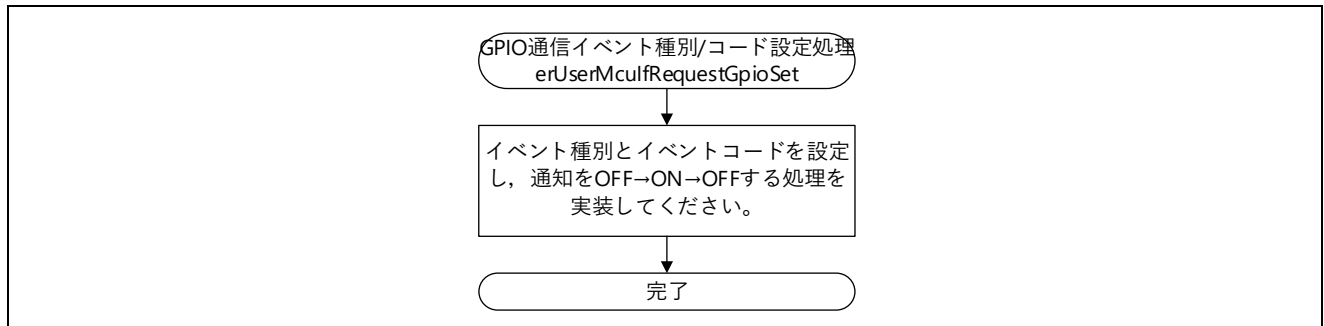


図 5.97 GPIO 通信イベント種別/コード設定処理フロー図

通知を ON→OFF とするタイミングについては、ユーザにて実装する安全用 MCU の処理速度に合わせて変更してください。

#### 5.10.5 安全 PDU 転送処理(MCU 内部→外部)

安全 PDU をシリアル送信用のバッファに書込みます。

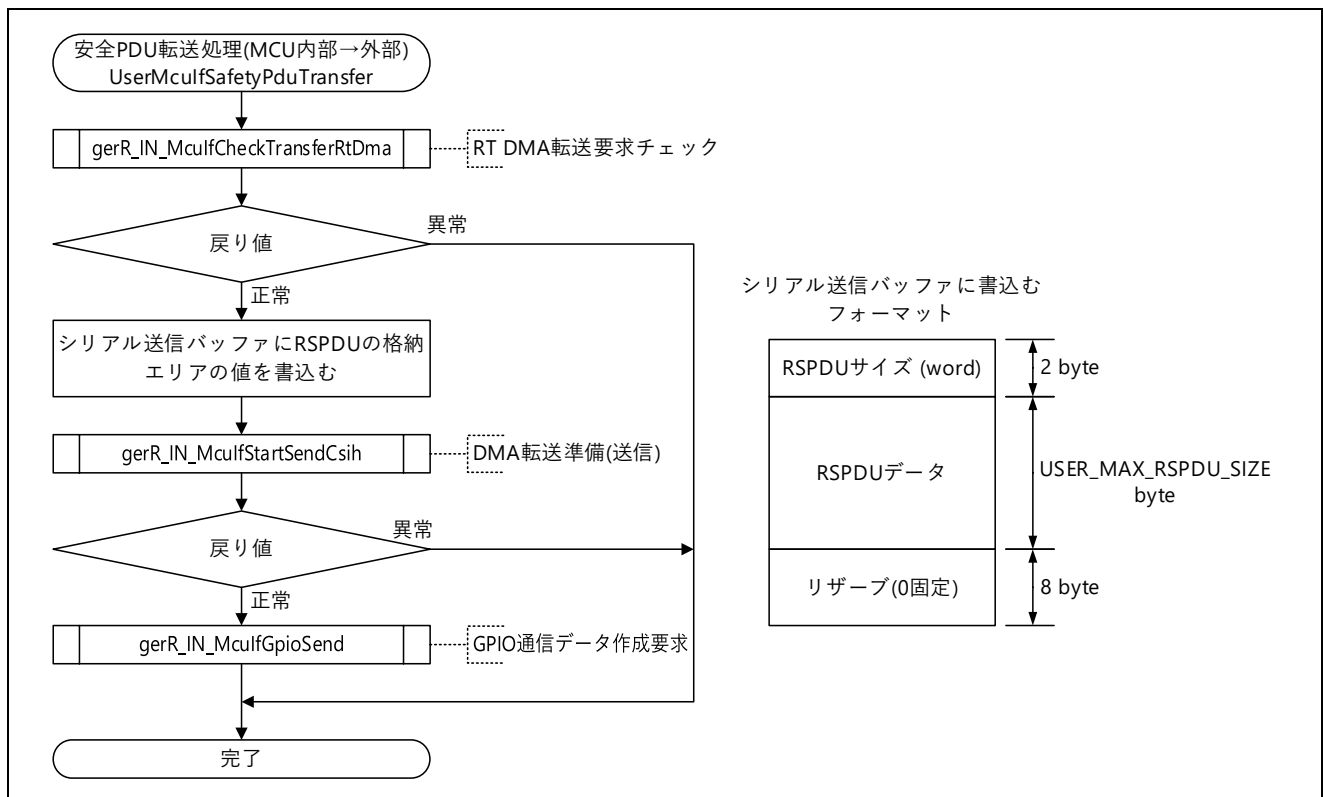


図 5.98 安全 PDU 転送処理(MCU 内部→外部)フロー図

### 5.10.6 安全 PDU 転送処理(MCU 内部←外部)

DMA 転送準備(受信)、GPIO 通信データ作成要求を実行します。

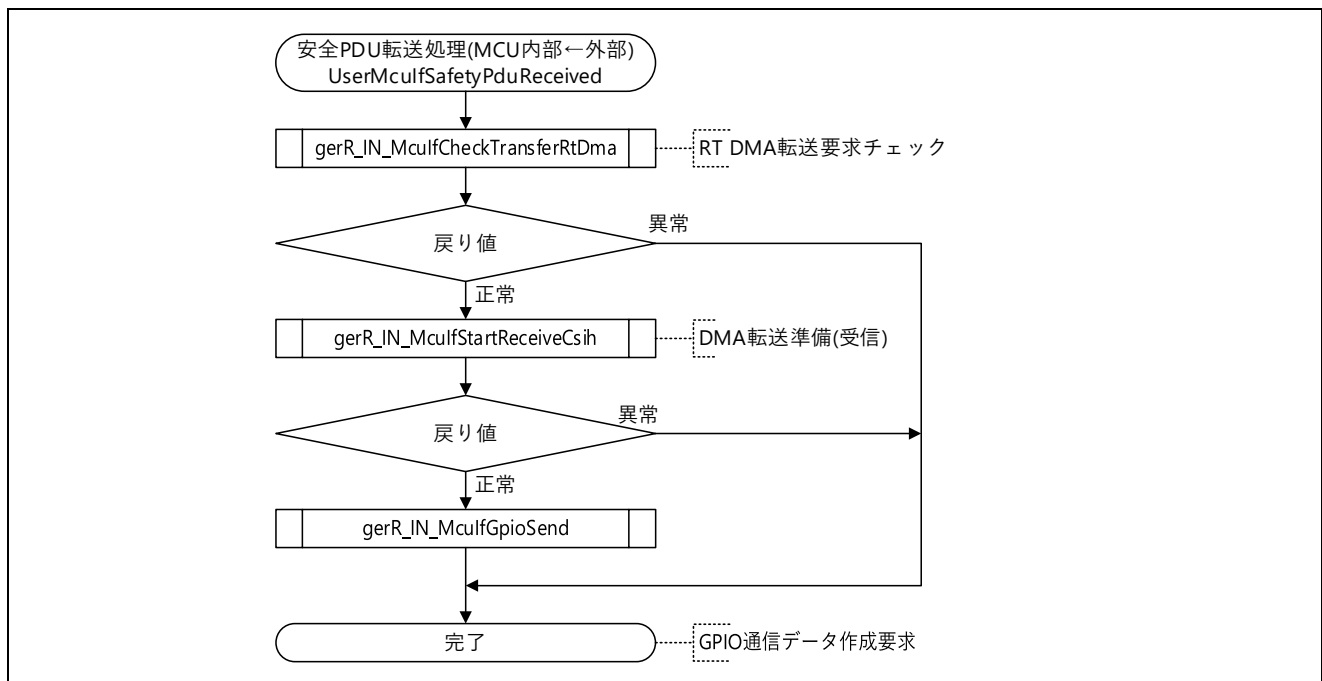


図 5.99 安全 PDU 転送処理(MCU 内部←外部)フロー図

### 5.10.7 安全 PDU 転送準備完了(MCU 内部→外部)受付受信処理

外部 MCU へ安全 PDU 転送準備完了(MCU 内部→外部)を通知後、実行結果を引数で受け付ける関数です。ユーザ任意の処理を実行します。

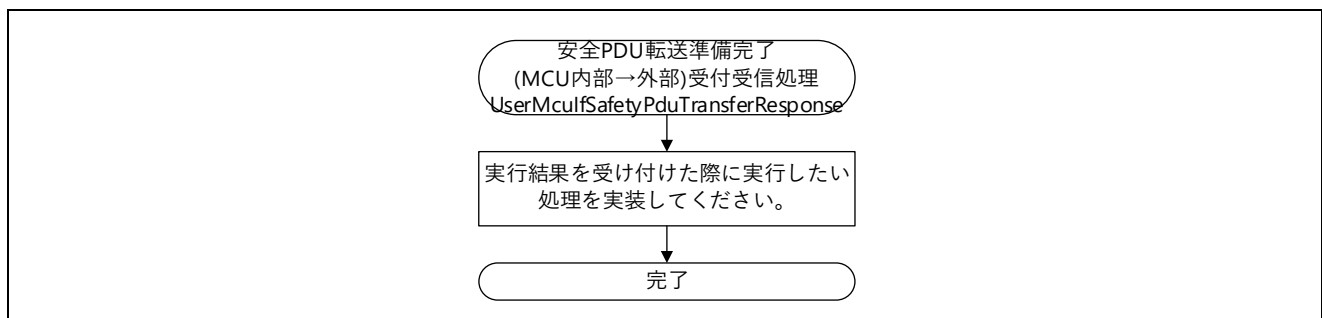


図 5.100 安全 PDU 転送準備完了(MCU 内部→外部)受付受信処理フロー図

GPIO 通信送信の実行結果は下記条件により設定します。

引数	結果	内容
R_IN_OK	正常完了	外部 MCU から受付を検出した。
R_IN_ERR	異常完了	GPIO 通信データ作成要求バッファのフル(16 件以上の登録)を検出、もしくは外部 MCU から受付待ちをされていてタイムアウト※1を検出した。

【注】※1 時間は R\_IN\_MCU\_IF\_RESPONSE\_TIME により定義しています。

### 5.10.8 安全 PDU 転送準備完了(MCU 内部←外部)受付受信処理

外部 MCU へ安全 PDU 転送準備完了(MCU 内部←外部)を通知後、実行結果を引数で受け付ける関数です。ユーザ任意の処理を実行します。

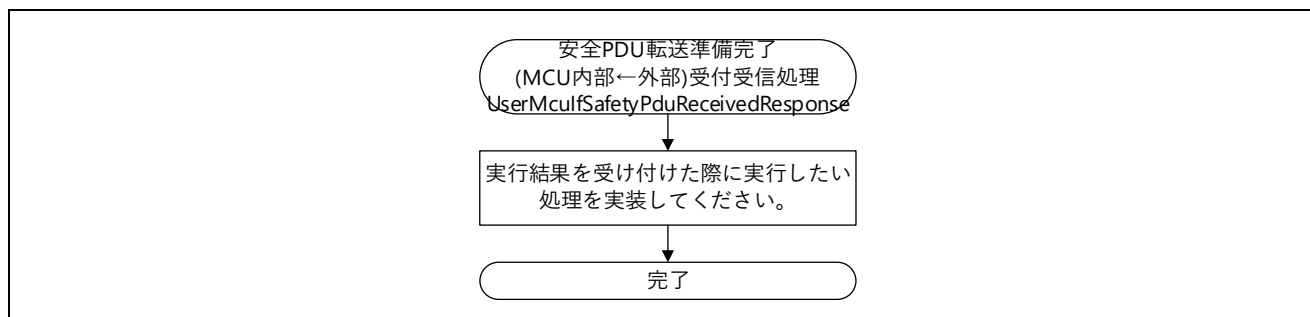


図 5.101 安全 PDU 転送準備完了(MCU 内部←外部)受付受信処理フロー図

GPIO 通信送信の実行結果は下記条件により設定します。

引数	結果	内容
R_IN_OK	正常完了	外部 MCU から受付を検出した。
R_IN_ERR	異常完了	GPIO 通信データ作成要求バッファのフル(16 件以上の登録)を検出、もしくは外部 MCU から受付待ちをされていてタイムアウト※1を検出した。

【注】※1 時間は R\_IN\_MCU\_IF\_RESPONSE\_TIME により定義しています。

## 5.10.9 安全 PDU 転送完了処理(MCU 内部←外部)

シリアル通信の停止と、シリアル受信用のバッファから安全 PDU を読出します。

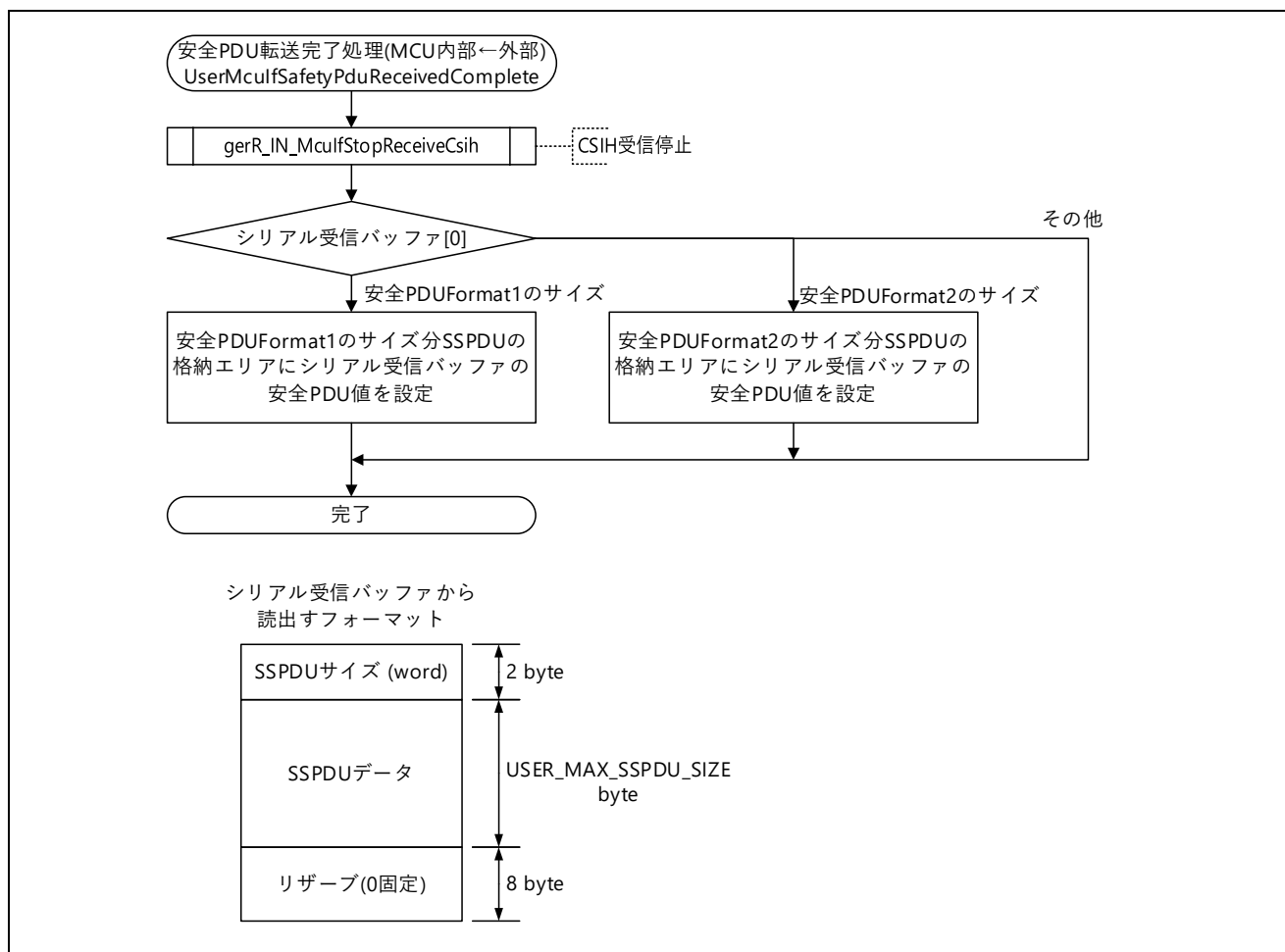


図 5.102 安全 PDU 転送完了処理(MCU 内部←外部)フロー図

## 5.11 ユーザプログラム詳細(H/W テスト関連)

H/W テストは、オンライン(データリンクを行うとき)では実施せずに、オフラインで実施する処理として実装してください。

### 5.11.1 H/W テスト処理(IEEE802.3ab コンプライアンステスト)

IEEE802.3ab コンプライアンステスト用の波形を出力します。

テストモード(テストモード1～4, テスト終了)を引数として、「gerR\_IN\_IEEE\_Test」(6.4.13(1))を実行します。  
gerR\_IN\_IEEE\_Test が PHY レジスタのデータを退避した後、テストモードに従いテストモードデータを PHY レジスタに書き込みますので、書き込み後に測定器で波形出力を確認してください。

表 5.41 テストモード一覧

テストモード	定数名 (値)	内容
テストモード1	R_IN_IEEE_MODE1 (0)	テンプレート、ピーク電圧、ドループ
テストモード2	R_IN_IEEE_MODE2 (1)	ジッタ (ネゴシエーション時に PHY をマスタとして設定)
テストモード3	R_IN_IEEE_MODE3 (2)	ジッタ (ネゴシエーション時に PHY をスレーブとして設定)
テストモード4	R_IN_IEEE_MODE4 (3)	ディストーション、リターン・ロス、コモン・モード電圧
テスト終了	R_IN_IEEE_END (4)	-

#### <注意事項>

- ・「5.2.1 イニシャルタスク」および「5.2.2 アイドルタスク」以外のタスクを起動しないでください。
- ・アイドルタスクの初期化処理(iUserInitializationRoutine)完了後に実行してください。
- ・アイドルタスクの状態管理・トランジェントメイン処理(iUserMainRoutin)内では実施せず、独立した処理として実装してください。



テスト終了時、または次のテストに移行する場合は、ユーザにより次の試験実施フラグを TRUE にすることで、退避している PHY レジスタデータを復元した後、テスト終了または次のテストへ移行します。

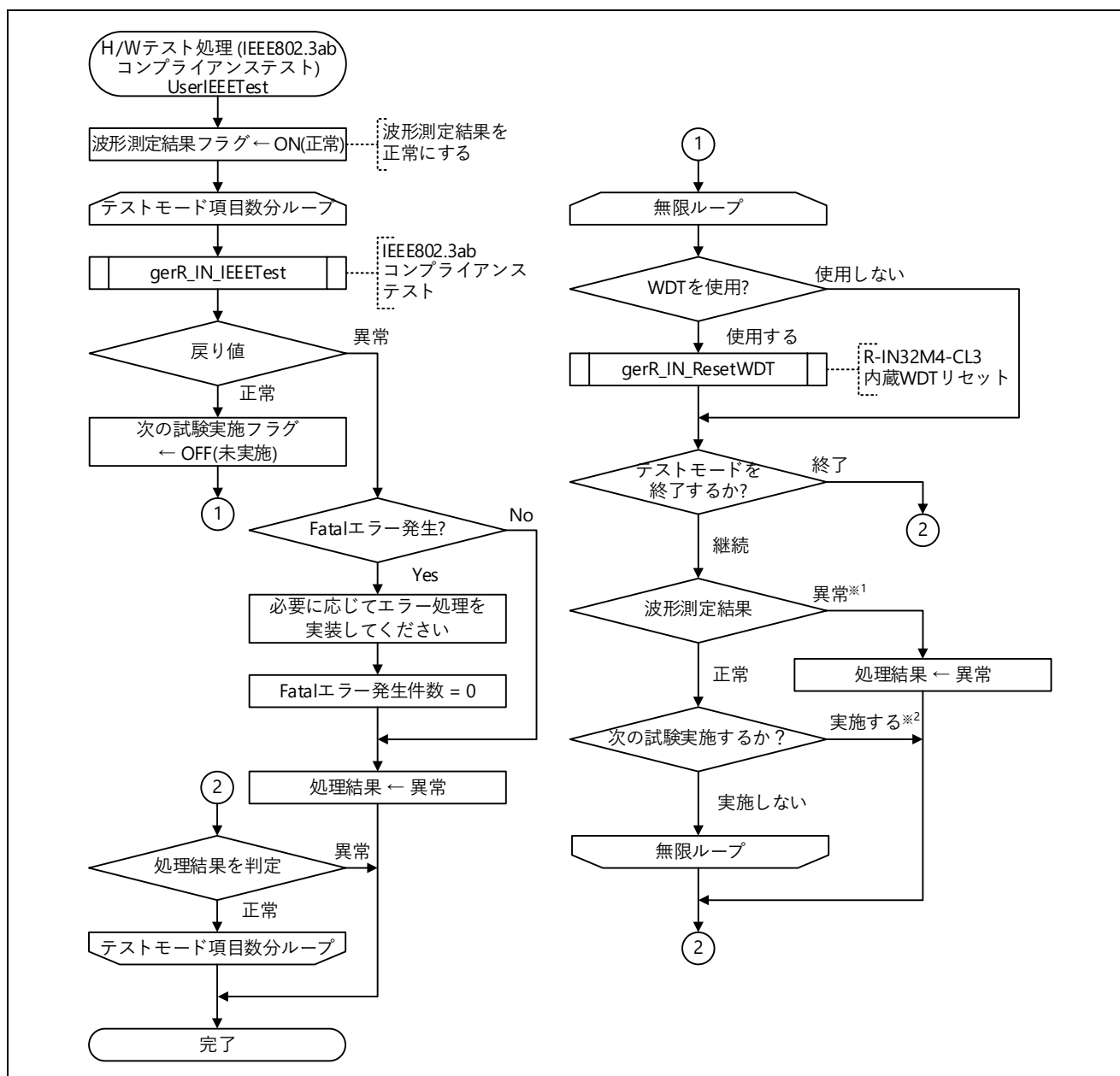


図 5.103 H/W テスト処理(IEEE802.3ab コンプライアンステスト)処理フロー図

【注】※1 測定結果が異常の場合、波形測定結果フラグを異常にする処理が必要です。

※2 次の試験を実施する場合、次の試験実施フラグを実施にする処理が必要です。

### 5.11.2 H/W テスト処理(折り返し通信テスト)

折り返し通信テストを実行します。

本テストを実施する場合、PORT1 と PORT2 間で Ethernet ケーブルを接続してください。

本処理は初期化処理および、通信開始処理後に実行してください。

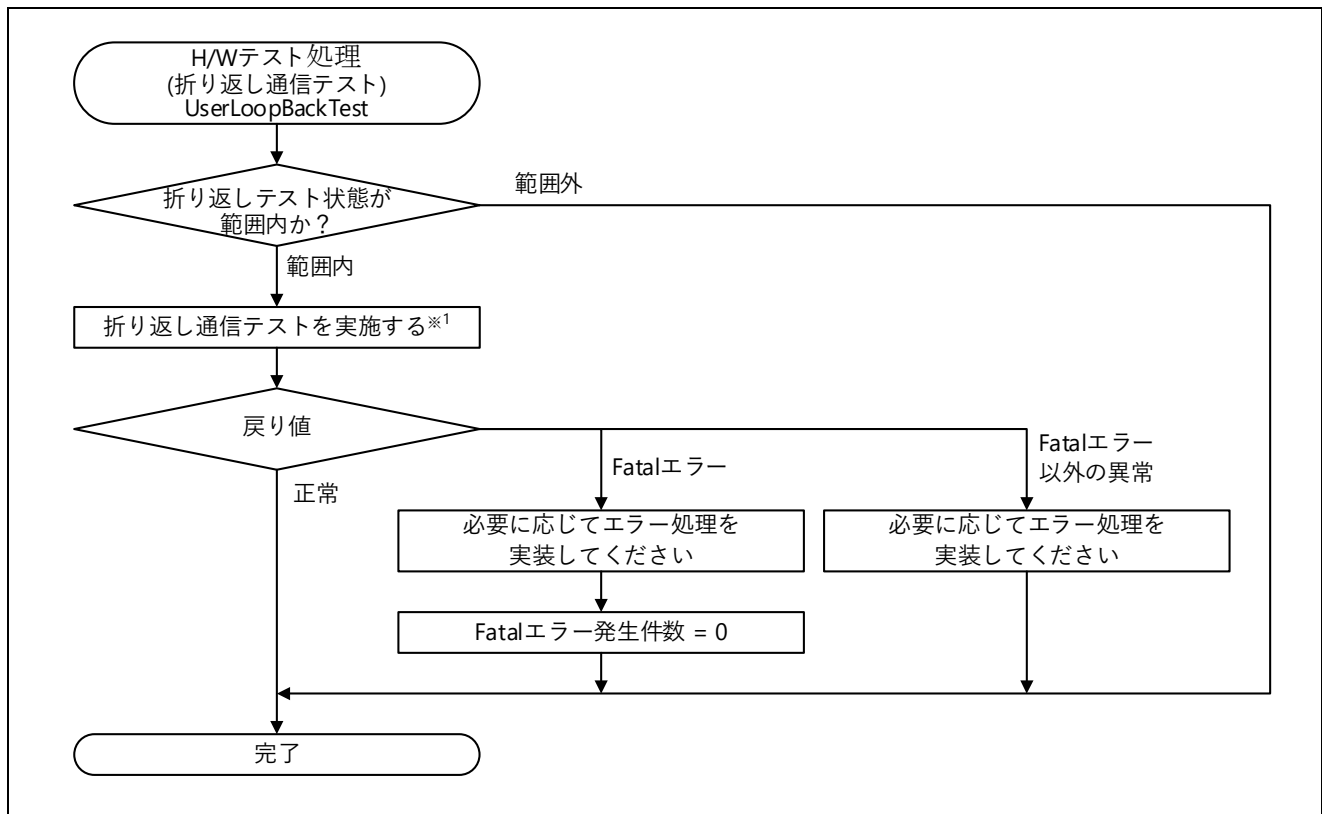


図 5.104 H/W テスト処理(折り返し通信テスト)処理フロー図

【注】※1 各折り返し通信テスト状態の一覧と、状態に対応した処理は、下表を参照してください。

表 5.42 テスト状態の一覧

No	折り返し通信テスト状態	値	処理内容
1	USER_LOOP_BACK_TEST_INITIAL	0	「5.11.3 折り返し通信テスト準備処理」参照
2	USER_LOOP_BACK_TEST_LINK_CHECK	1	「5.11.4 折り返し通信テストリンク状態判定処理」参照
3	USER_LOOP_BACK_TEST_SEND	2	「5.11.5 折り返し通信テストデータ送信処理」参照
4	USER_LOOP_BACK_TEST_RECEIVE	3	「5.11.6 折り返し通信テスト受信結果判定処理」参照
5	USER_LOOP_BACK_TEST_EXIT	4	「5.11.7 折り返し通信テスト完了処理」参照

折り返し通信テスト状態は PORT1 における No.1～4 状態を遷移し、次に PORT2 における No.1～4 状態を遷移します。すべての処理が完了するか、あるいは No.1～4 状態いずれかで異常を検出すると No.5 の状態に遷移します。

### 5.11.3 折り返し通信テスト準備処理

折り返し通信テストにおける異常検出用のレジスタ、PHY レジスタの初期化を行います。

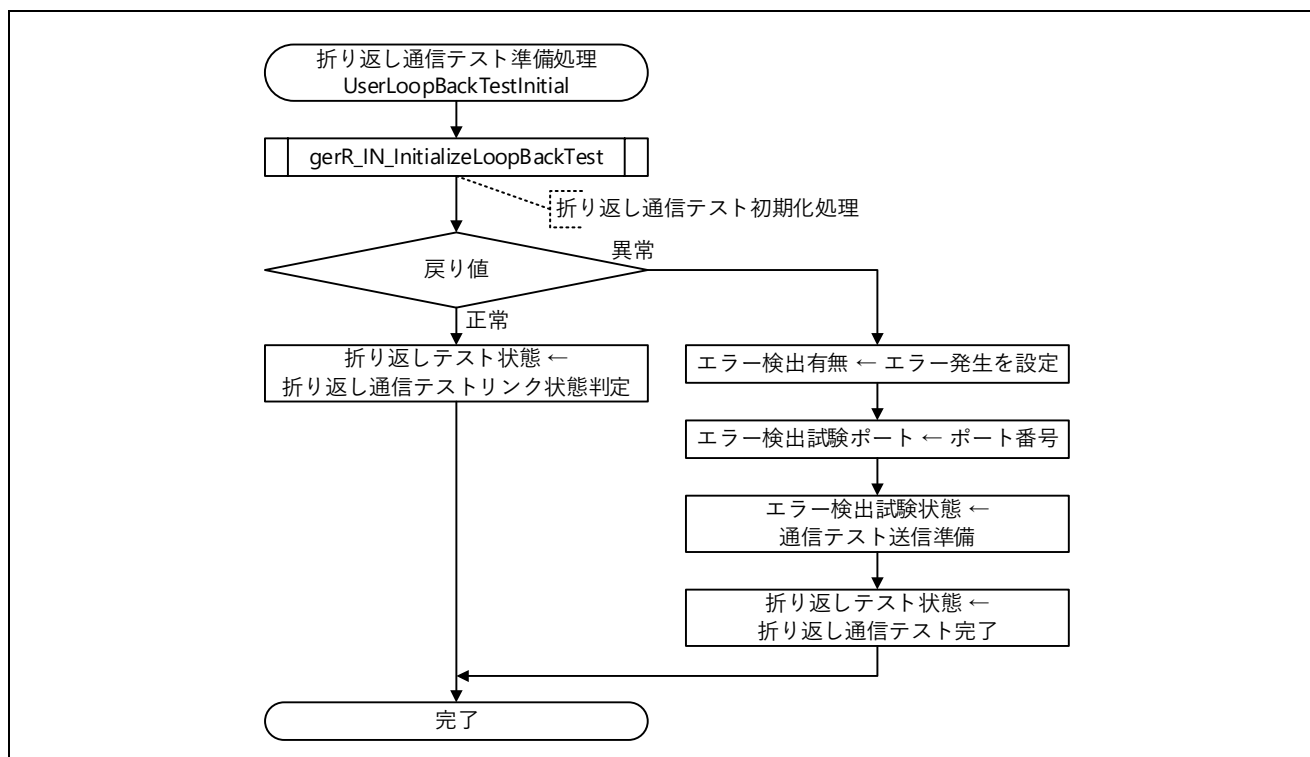


図 5.105 折り返し通信テスト準備処理フロー図

## 5.11.4 折り返し通信テストリンク状態判定処理

折り返し通信テストの通信ポート状態を設定します。

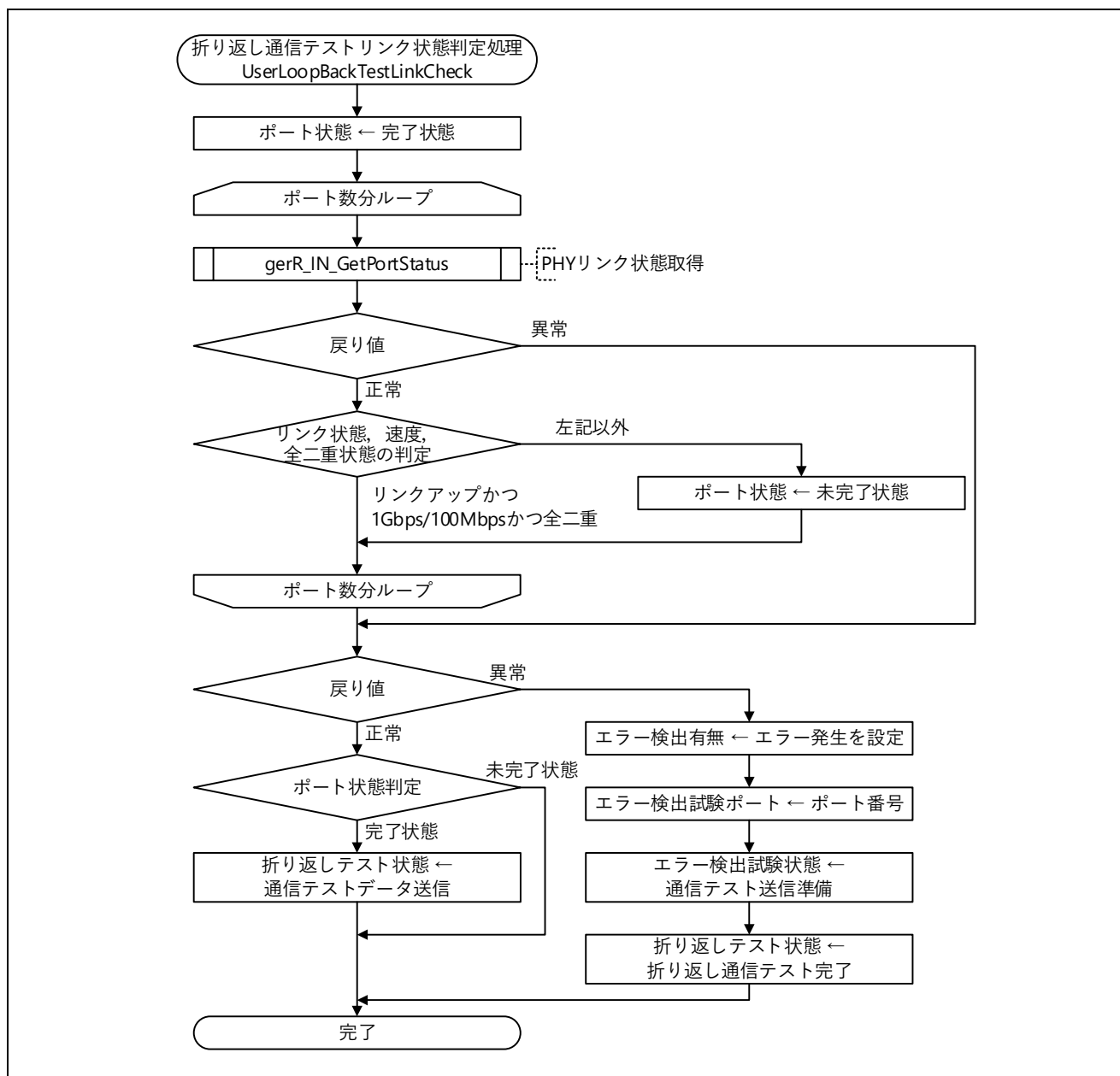


図 5.106 折り返し通信テストリンク状態判定処理フロー図

## 5.11.5 折り返し通信テストデータ送信処理

折り返し通信テストで使用するテストデータを作成、送信します。

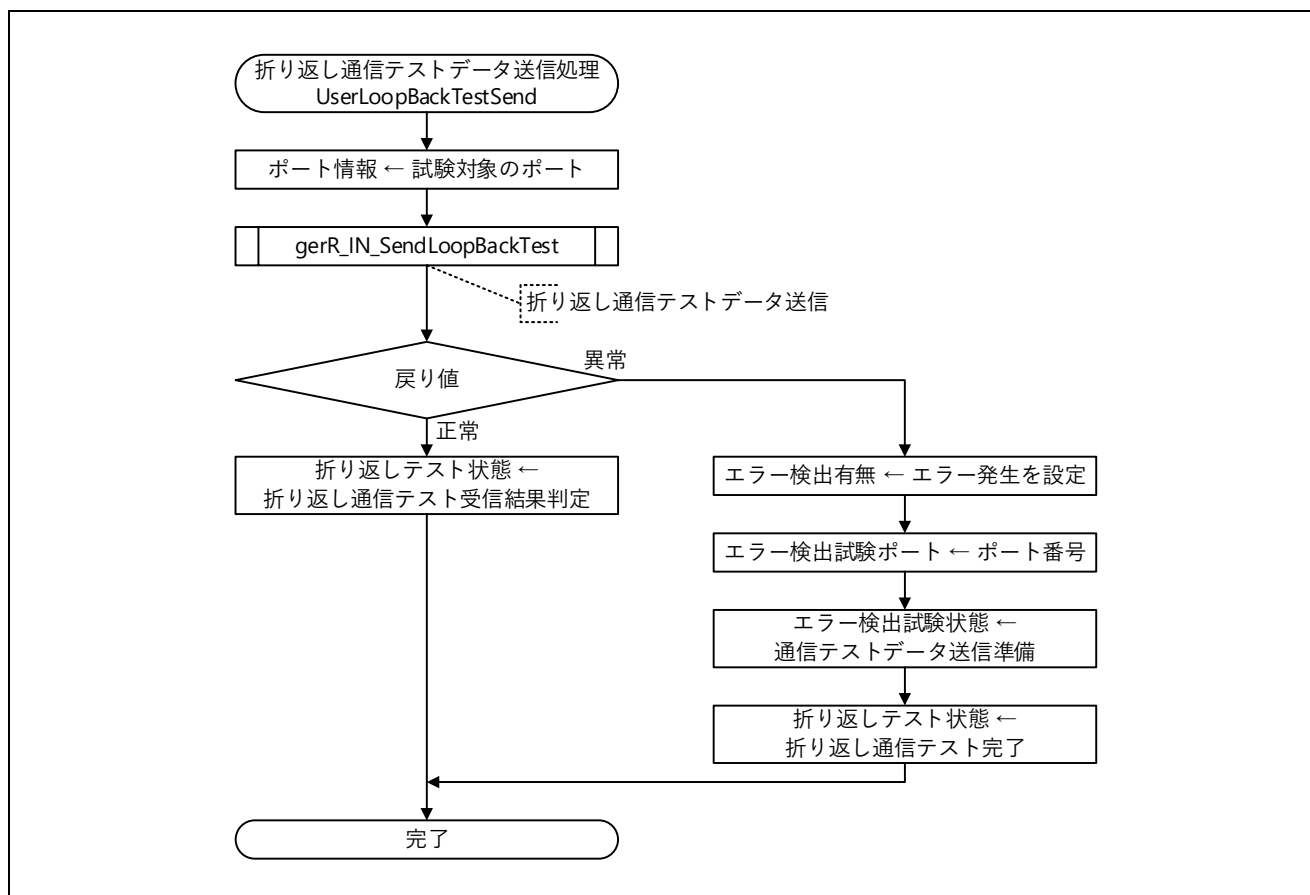


図 5.107 折り返し通信テストデータ送信処理フロー図

## 5.11.6 折り返し通信テスト受信結果判定処理

受信したテストデータやレジスタの値から、テスト結果の妥当性を判定します。

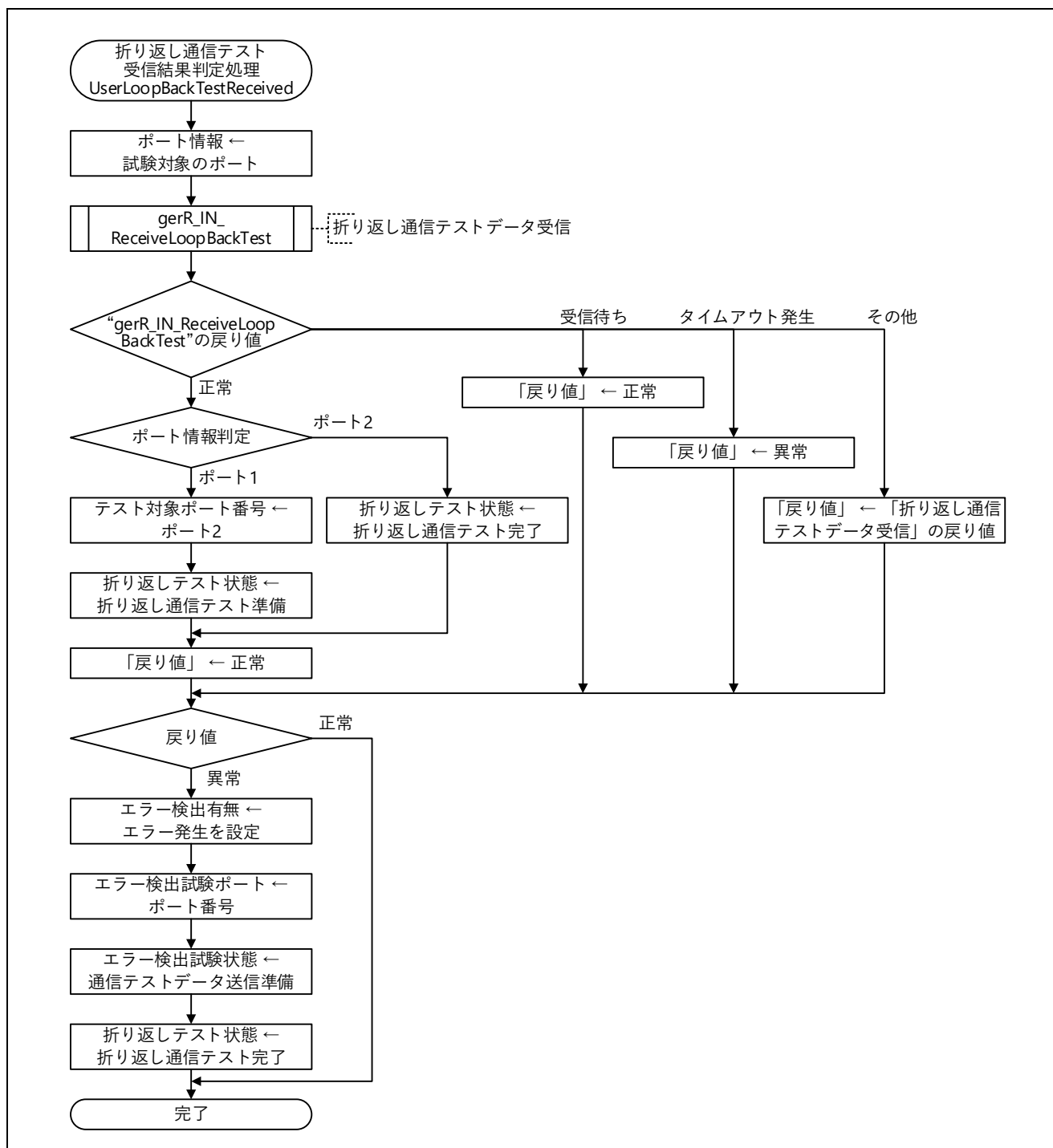


図 5.108 折り返し通信テスト受信結果判定処理フロー図

### 5.11.7 折り返し通信テスト完了処理

ユーザに H/W テスト処理(折り返し通信テスト)の完了を通知します。

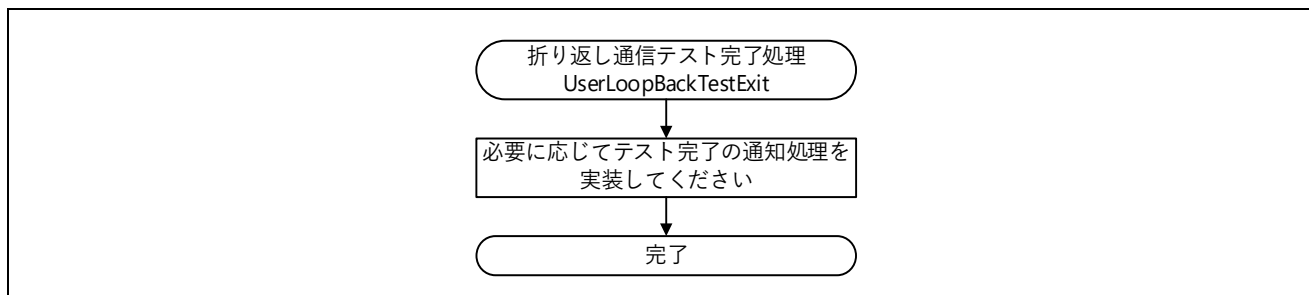


図 5.109 折り返し通信テスト完了処理フロー図

## 6. R-IN32M4-CL3 ドライバの関数仕様

本章は、R-IN32M4-CL3 ドライバを構成する R-IN32M4-CL3 ドライバインタフェース関数と R-IN32M4-CL3 ドライバコールバック関数の仕様について示します。

### 6.1 各関数の概要

#### (1) 概要

各関数の概要と変更の可否を下表に示します。

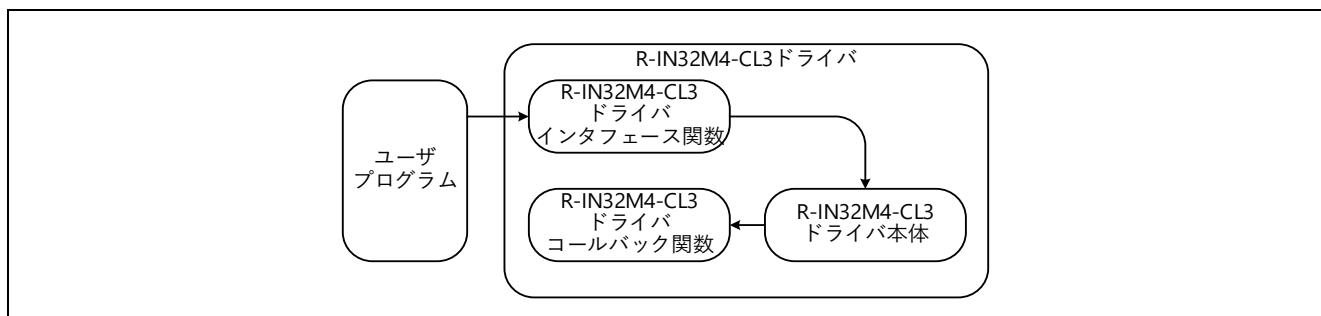


図 6.1 各関数の関連

表 6.1 各関数の概要

プログラム部品名	概要	変更可否
R-IN32M4-CL3 ドライバインタフェース関数	R-IN32M4-CL3 ドライバの機能をユーザプログラムから使用する場合に呼び出す関数。 (ファイル : R_IN_Interface.c)	×
R-IN32M4-CL3 ドライバコールバック関数	ユーザプログラムから R-IN32M4-CL3 ドライバに対してコールバックを要求する際に使用する関数。 R-IN32M4-CL3 ドライバ内で発生するイベントに対するユーザプログラムの処理を記述する。 (ファイル : R_IN32M4_CL3_Callback.c)	○
R-IN32M4-CL3 ドライバ本体	R-IN32M4-CL3 ドライバインタフェース関数に呼び出され、R-IN32M4-CL3 を制御するドライバ部の本体。 (ファイル : R_IN_Interface.c を除いた driver フォルダ以下のファイル)	×

#### (2) 記述仕様

各関数の記述仕様を示します。

表 6.2 ソースコード記述仕様

項目	内容	備考
C 言語規格	ANSI C 準拠	一部、コンパイラメーカー拡張仕様を使用
文字エンコード	ASCII、日本語(Shift_JIS)	-
タブ長	4	-
改行コード	CR+LF	-



## (3) 型定義とエラーコード

R-IN32M4-CL3 ドライバで定義する型とエラーコードを以下に示します。

表 6.3 R-IN32M4-CL3 ドライバの型一覧

No.	定義型	実装
1	VOID	void
2	CHAR	char
3	UCHAR	unsigned char
4	SHORT	short
5	USHORT	unsigned short
6	INT	int
7	UINT	unsigned int
8	LONG	long
9	ULONG	unsigned long
10	ERRCODE	int
11	BOOL	int

表 6.4 R-IN32M4-CL3 ドライバのエラーコード一覧

No.	シンボル	値	内容
1	R_IN_OK	0	正常
2	R_IN_BUSY	1	処理中
3	R_IN_ERR	-1	異常完了(状態異常/不一致)
4	R_IN_ERR_OTHER	-2	(ライブラリ内ドライバで異常発生)
5	R_IN_ERR_OUTOFRANGE	-3	範囲外
6	R_IN_ERR_EMPTY	-4	エンプティ
7	R_IN_ERR_OVERFLOW	-5	オーバフロー
8	R_IN_ERR_NOENTRY	-6	エントリがない
9	R_IN_ERR_NOPERMIT	-7	許可されていない
10	R_IN_ERR_NODATA	-8	データがない
11	R_IN_ERR_STSW	-9	有効な自局状態情報がない
12	R_IN_ERR_BOUNDARY	-10	バウンダリ指定異常
13	R_IN_ERR_SEMAPHORE	-11	セマフォ獲得失敗
14	R_IN_ERR_FATAL	-12	Fatal エラー発生
15	R_IN_ERR_TIMEOUT	-13	タイムアウト

## 6.2 R-IN32M4-CL3 ドライバインタフェース関数一覧

R-IN32M4-CL3 ドライバインタフェース関数の一覧を以下に示します。

表 6.5 R-IN32M4-CL3 ドライバインタフェース関数一覧

機能分類 (参照項)	関数名	関数型	概要
初期設定 (6.4.1 項)	gulR_IN_GetResetStatus	ULONG	リセット状態取得
	gerR_IN_Initialize	ERRCODE	R-IN32M4-CL3 の初期化
	gerR_IN_SetIPAddress	ERRCODE	IP アドレス設定
	gerR_IN_SetIPAddressAndNodeNumber	ERRCODE	IP アドレス・ノード番号設定
	gerR_IN_Start	ERRCODE	R-IN32M4-CL3 通信開始
	gerR_IN_MIBLedTableDefine	ERRCODE	LED ステータス情報 MIB 定義
	gerR_IN_RegistIPAddressFilteringFunction	VOID	IP アドレスフィルタリング処理登録
	gerR_IN_RegistCallback	ERRCODE	コールバック処理登録
ウォッチドッグ・タイマ (6.4.2 項)	gerR_IN_ResetWDT	ERRCODE	R-IN32M4-CL3 内部 WDT リセット
	gerR_IN_SetWDT	ERRCODE	R-IN32M4-CL3 内部 WDT 時限設定
	gerR_IN_DisableWDT	ERRCODE	R-IN32M4-CL3 内部 WDT 無効
	gerR_IN_EnableWDT	ERRCODE	R-IN32M4-CL3 内部 WDT 有効
イベント (6.4.3 項)	gerR_IN_GetEvent	ERRCODE	R-IN32M4-CL3 イベント検出
	gerR_IN_Main	ERRCODE	R-IN32M4-CL3 イベント検出メイン処理
	gerR_IN_RestartEvent	ERRCODE	R-IN32M4-CL3 イベント検出再開
サイクリック 伝送 (6.4.4 項)	gerR_IN_GetMasterNodeStatus	ERRCODE	マスタ局状態取得
	gerR_IN_GetHoldClearFlag	ERRCODE	Hold/Clear フラグ取得
	gerR_IN_GetReceivedCyclicData	ERRCODE	サイクリック受信データ取得(一括)
	gerR_IN_UpdateReceivedCyclicData	ERRCODE	サイクリック受信データ更新
	gerR_IN_FinishReceivedCyclicDataAcquisition	ERRCODE	サイクリック受信データ取得完了
	gerR_IN_GetReceivedCyclicRY	ERRCODE	サイクリック受信データ取得(RY)
	gerR_IN_GetReceivedCyclicRWw	ERRCODE	サイクリック受信データ取得(RWw)
	gerR_IN_GetReceivedCyclicRspdu	ERRCODE	サイクリック受信データ取得(RSPDU)
	gerR_IN_SetSendCyclicData	ERRCODE	サイクリック送信データ設定(一括)
	gerR_IN_FinishSendCyclicDataSeting	ERRCODE	サイクリック送信データ設定完了
	gerR_IN_SetSendCyclicRX	ERRCODE	サイクリック送信データ設定(RX)
	gerR_IN_SetSendCyclicRWr	ERRCODE	サイクリック送信データ設定(RWr)
	gerR_IN_SetSendCyclicSspdu	ERRCODE	サイクリック送信データ設定(SSPDU)
	gerR_IN_GetReceivedCyclicBuffer	ERRCODE	サイクリック受信バッファ取得
	gerR_IN_FinishReceivedCyclicBuffer	ERRCODE	サイクリック受信バッファ取得完了
	gerR_IN_GetSendCyclicBuffer	ERRCODE	サイクリック送信バッファ取得
	gerR_IN_GetSplitSendCyclicBuffer	ERRCODE	サイクリック分割送信バッファ取得
	gerR_IN_FinishSendCyclicBuffer	ERRCODE	サイクリック送信バッファ設定完了
	gvR_IN_SendCyclicFrameClassA	VOID	サイクリックフレーム送信(CC-Link IE TSN Class A)
自局状態設定 (6.4.5 項)	gerR_IN_SetNodeStatus	ERRCODE	自局状態設定

機能分類 (参照項)	関数名	関数型	概要
自局状態取得 (6.4.6 項)	gerR_IN_GetIPAddress	ERRCODE	IP アドレス取得
	gerR_IN_GetCurrentCyclicSize	ERRCODE	マスタ局が指定したサイクリック伝送 サイズ取得
	gerR_IN_GetCyclicStatus	ERRCODE	サイクリック伝送状態取得
	gerR_IN_GetCommunicationStatus	ERRCODE	データリンク状態取得
	gerR_IN_GetPortStatus	ERRCODE	PHY リンク状態取得
	gerR_IN_GetStatisticalInformation	ERRCODE	統計情報取得
	gerR_IN_ClearStatisticalInformation	ERRCODE	統計情報クリア
	guIR_IN_GetNetworkTopology	ULONG	ネットワークトポロジ情報取得
	gerR_IN_GetNodeOperationMode	ERRCODE	自局動作モード取得
	gerR_IN_GetLinkSpeed	ERRCODE	通信速度取得
	gerR_IN_GetIPAddressDuplication	ERRCODE	IP アドレス重複状態取得
LED 制御 (6.4.7 項)	gerR_IN_SetUSER1LED	ERRCODE	LED 点灯制御(USER LED1)
	gerR_IN_SetUSER2LED	ERRCODE	LED 点灯制御(USER LED2)
	gerR_IN_SetRUNLED	ERRCODE	LED 点灯制御(RUN)
	gerR_IN_SetERRLED	ERRCODE	LED 点灯制御(ERR)
	gerR_IN_SetLERR1LED	ERRCODE	LED 点灯制御(L ER1 LED)
	gerR_IN_SetLERR2LED	ERRCODE	LED 点灯制御(L ER2 LED)
	gerR_IN_DisableLED	ERRCODE	LED 点灯機能無効
	gerR_IN_EnableLED	ERRCODE	LED 点灯機能有効
	gerR_IN_UpdateLedStatus	ERRCODE	通信状態表示 LED の更新
	gerR_IN_SetSDRDLEDMode	ERRCODE	SD/RD 点灯モード設定
	gerR_IN_StartTestLED	ERRCODE	LED テスト開始
	gerR_IN_ExecuteTestLED	ERRCODE	LED テスト実行
	gerR_IN_StopTestLED	ERRCODE	LED テスト終了
ネットワーク 時刻 (6.4.8 項)	gerR_IN_GetNetworkTime	ERRCODE	ネットワーク時刻(シリアル値)取得
	gerR_IN_NetworkTimeToDate	ERRCODE	ネットワーク時刻(シリアル値)→時計情 報変換
	gerR_IN_DateToNetworkTime	ERRCODE	時計情報→ネットワーク時刻(シリアル 値)変換
	gerR_IN_GetUnixTime	ERRCODE	ネットワーク時刻(UNIX 時間)取得
	gerR_IN_UnixTimeToDate	ERRCODE	ネットワーク時刻(UNIX 時間)→時計情 報変換
	gerR_IN_DateToUnixTime	ERRCODE	時計情報→ネットワーク時刻(UNIX 時 間)変換
	gerR_IN_SetUnixTimeClassA	ERRCODE	ネットワーク時刻(UNIX 時間)設定(CC- Link IE TSN Class A)
	gvR_IN_SetUnixOffsetTime	VOID	ネットワーク時刻オフセット設定
MDIO アクセ ス (6.4.9 項)	gerR_IN_GetUnixOffsetTime	ERRCODE	ネットワーク時刻オフセット取得
	gerR_IN_EnableMACIPAccess	ERRCODE	MAC IP アクセス許可
	gerR_IN_DisableMACIPAccess	ERRCODE	MAC IP アクセス禁止
	gerR_IN_WritePhy	ERRCODE	PHY 内部レジスタライト
	gerR_IN_ReadPhy	ERRCODE	PHY 内部レジスタリード

機能分類 (参照項)	関数名	関数型	概要
SLMP 送受信 (6.4.10 項)	gerR_IN_ReceivedSmpMain	ERRCODE	SLMP 受信メイン処理
	gvR_IN_ReceivedSmpExecution	VOID	SLMP 受信コマンド実行
	gblR_IN_GetReceiveSmpStatus	BOOL	ユーザ理由による SLMP 受信許可設定 状態取得
	gerR_IN_SetReceiveSmpStatus	ERRCODE	ユーザ理由による SLMP 受信許可設定
	gerR_IN_SetSmpCommand	ERRCODE	SLMP コマンド判定登録
	gerR_IN_SendSmpFrame	ERRCODE	SLMP フレーム送信
	gerR_IN_WriteSmpSendBuffer	ERRCODE	SLMP 送信バッファ書込み
	gblR_IN_GetSmpSendBufferUsed	BOOL	SLMP 送信バッファ使用有無取得
	gerR_IN_SetSmpResponseFrameSendRequest	ERRCODE	SLMP 応答フレーム送信要求
	gvR_IN_ReleaseSmpReceiveFrame	VOID	SLMP 受信バッファ解放要求
SLMP コマンド 実行 (6.4.11 項)	gvR_IN_ExecuteReset	VOID	自局リセット
	gerR_IN_StartSmpRequestTimer	ERRCODE	SLMP 要求待ちタイマ開始
	gerR_IN_StopSmpRequestTimer	ERRCODE	SLMP 要求待ちタイマ停止
	gerR_IN_GetMasterIPAdress	ERRCODE	管理マスタ局 IP アドレス取得
	gerR_IN_CheckIpAdressSmp	ERRCODE	IP アドレスの判定
	gerR_IN_SetNodeIndicationStatus	ERRCODE	NodeIndication ステータス設定
	gvR_IN_RequestIPAdressChanging	VOID	IP アドレス変更要求
	gerR_IN_SetSNMPCommunityName	ERRCODE	SNMP コミュニティ名設定
エラー履歴 (0 項)	gerR_IN_SetErrorHistory	ERRCODE	エラー履歴登録
	gvR_IN_ClearErrorHistory	VOID	エラー履歴クリア
	gvR_IN_ClearErrorHistoryController	VOID	エラー履歴クリア(コントローラ情報)
	gerR_IN_SetErrorHistoryOption	ERRCODE	エラー履歴登録(オプション情報)
	gerR_IN_ClearErrorHistoryOption	ERRCODE	エラー履歴クリア(オプション情報)
H/W テスト (6.4.13 項)	gerR_IN_IEEE8023abTest	ERRCODE	IEEE802.3ab コンプライアンステスト
	gerR_IN_InitializeLoopBackTest	VOID	折り返し通信テスト初期化
	gerR_IN_SendLoopBackTest	ERRCODE	折り返し通信テストデータ送信
	gerR_IN_ReceiveLoopBackTest	ERRCODE	折り返し通信テストデータ受信
汎用共通 (6.4.14 項)	gverR_IN_CopyMemory	ERRCODE	メモリコピー
	gerR_IN_FillMemory	ERRCODE	メモリフィル(8bit)
	gerR_IN_FillMemory16	ERRCODE	メモリフィル(16bit)
	gerR_IN_FillMemory32	ERRCODE	メモリフィル(32bit)
	gerR_IN_EndianShort	ERRCODE	エンディアン変換(USHORT)
	gerR_IN_EndianLong	ERRCODE	エンディアン変換(ULONG)
	gerR_IN_EndianLongLong	ERRCODE	エンディアン変換(ULONGULONG)
	gvR_IN_DisableInt	VOID	割込み禁止
	gvR_IN_EnableInt	VOID	割込み許可
	gvR_IN_DisableDispatch	VOID	ディスパッチ禁止
	gvR_IN_EnableDispatch	VOID	ディスパッチ許可

機能分類 (参照項)	関数名	関数型	概要
ネットワーク同期通信 (6.4.15 項)	gerR_IN_GetSyncDeviationFlag	ERRCODE	時刻同期外れ検出
	gerR_IN_StopAppSyncSignal	ERRCODE	同期信号出力停止
	gerR_IN_SetWdcThreshold	ERRCODE	ウォッチドッグカウンタチェック異常 連続カウンタ閾値設定
	gerR_IN_MakeResponseWdcInformation	ERRCODE	ウォッチドッグカウンタ情報設定応答 データ作成
	gvR_IN_IncrementWdcUL	VOID	ウォッチドッグカウンタインクリメント(送信用)
	gvR_IN_LatchWdcUL	VOID	ウォッチドッグカウンタラッチ値更新 (送信用)
	gerR_IN_GetWdcUL	ERRCODE	ウォッチドッグカウンタ UL 取得
	gerR_IN_CheckWdcDL	ERRCODE	ウォッチドッグカウンタチェック(受信用)
	gvR_IN_ClearSyncDeviationFlag	VOID	時刻同期外れ検出状態クリア
CANopen 通信 (6.4.16 項)	gerR_IN_CanInit	ERRCODE	CANopen 通信機能の初期化
	gerR_IN_CanGetValidObjectDictionaryNumber	ERRCODE	有効 ObjectDictionary 数取得
	gerR_IN_CanGetObjectHandle	ERRCODE	Object Dictionary ハンドル取得
	gerR_IN_CanReadObject	ERRCODE	Object Dictionary 読出し
	gerR_IN_CanWriteObject	ERRCODE	Object Dictionary 書込み
	gerR_IN_CanReadBlockObject	ERRCODE	Object Dictionary ブロック読出し
	gerR_IN_CanWriteBlockObject	ERRCODE	Object Dictionary ブロック書込み
	gerR_IN_CanGetNmtState	ERRCODE	NMTState 取得
	gerR_IN_CanSetNmtState	ERRCODE	NMTState 設定
	gerR_IN_CanUpdateRPDO	ERRCODE	RPDO 更新
	gerR_IN_CanUpdateTPDO	ERRCODE	TPDO 更新
MCU 間 I/F (6.4.17 項)	gvR_IN_SchedulerMain	VOID	スケジューラタスクメイン
	gvR_IN_MculfGpioSendMain	VOID	GPIO 通信送信タスクメイン
	gvR_IN_MculfGpioResponseMain	VOID	GPIO 通信応答受信タスクメイン
	gvR_IN_MculfGpioReceivedMain	VOID	GPIO 通信受信タスクメイン
	gvR_IN_MculfRtDmaCompleteMain	VOID	RT DMA 転送完了タスクメイン
	gvR_IN_MculfInitial	VOID	MCU 間 I/F イニシャル
	gerR_IN_MculfSetCommand	ERRCODE	MCU 間 I/F 登録処理
	gerR_IN_MculfCheckTransferRtDma	ERRCODE	RT DMA 転送要求チェック
	gerR_IN_MculfStartSendCsih	ERRCODE	DMA 転送準備(送信)
	gerR_IN_MculfStartReceiveCsih	ERRCODE	DMA 転送準備(受信)
	gerR_IN_MculfStopReceiveCsih	ERRCODE	CSIH 受信停止
	gerR_IN_MculfStopSendCsih	ERRCODE	CSIH 送信停止
	gerR_IN_MculfCheckGpioState	ERRCODE	GPIO 通信使用状態確認
	gerR_IN_MculfGetGpioState	ERRCODE	GPIO 通信使用状態取得
	gvR_IN_MculfClearGpioState	VOID	GPIO 通信使用状態クリア
	gerR_IN_MculfGpioSend	ERRCODE	GPIO 通信データ作成要求

### 6.3 R-IN32M4-CL3 ドライバのタスク

R-IN32M4-CL3 ドライバにおける、RTOS のコンフィグレーションファイルに記述する生成情報を示します。各種設定値の詳細は、「R-IN32M4-CL3 プログラミング・マニュアル OS 編」を参照してください。

なお、各種設定値は、特別な理由がない限り、変更しないでください。

ユーザにてタスクを新規生成する場合は、「5.2 ユーザプログラムのタスク」に記載する生成情報も確認のうえ、生成してください。

#### (1) タスク一覧

表 6.6 R-IN32M4-CL3 ドライバタスク一覧

No.	タスク名称	タスク ID	T_CTSK 構造体メンバ					
			タスク属性	拡張情報	起動番地 (関数名で記載)	起動時 優先度	スタック サイズ	スタック領域 の先頭番地
1	低優先割込みタスク	TSKID_NX_LOW_INT (33)	TA_HLNG	0000H	vNX_Task_CchiefNx_Low	2	800H	NULL
2	高優先割込みタスク	TSKID_NX_HIGH_INT (34)	TA_HLNG	0000H	vNX_Task_CchiefNx_High	1	400H	NULL
3	非サイクリックフレーム 受信タスク	TSKID_NX_NCYC_RX (35)	TA_HLNG	0000H	vNX_Task_RxNonCycFrame	5	1000H	NULL
4	非サイクリックフレーム 送信タスク	TSKID_NX_NCYC_TX (36)	TA_HLNG	0000H	vNX_Task_TxNonCycFrame	4	1800H	NULL
5	通信ドライバ定期処理 タスク	TSKID_NX_PERIODIC (37)	TA_HLNG	0000H	vNX_Task_ComDriverPeriodic	8	1000H	NULL
6	USnet 定期処理タスク	TSKID_NX_IPCOMM (38)	TA_HLNG	0000H	vNX_Task_IpFrameComm	6	400H	NULL
7	TSN 定期処理タスク	TSKID_NX_TSN (39)	TA_HLNG	0000H	vNX_Task_TSNPeriodic	7	1000H	NULL
8	中継 RAM クリアタスク	TSKID_NX_RLYRAMCLR (40)	TA_TPRI	0000H	vNX_Task_RelayRamClr	11	400H	NULL

#### (2) セマフォ

表 6.7 セマフォの生成情報

No.	名称	ID	セマフォ属性※1	資源数の初期値	最大資源数
1	PTP 提供用セマフォ 1	SEMIC_NX_TSN_1(65)	TA_TFIFO	1	1
2	PTP 提供用セマフォ 2	SEMIC_NX_TSN_2(66)	TA_TFIFO	1	1
3	PHY アクセス用セマフォ	SEMIC_NX_PHYACCESS(67)	TA_TFIFO	1	1
4	中継 RAM クリア用セマフォ	SEMIC_NX_RLYRAMCLR(68)	TA_TFIFO	1	1

【注】※1 ITRON ではセマフォ属性を以下のとおり定義しています。(μITRON4.0 仕様 ver4.03.03 P.75)

TA\_TFIFO(00H) …タスクの待ち行列を FIFO 順に管理する

TA\_TPRI (01H)…タスクの待ち行列をタスクの優先度順に管理する

## (3) Hardware ISR の設定

表 6.8 Hardware ISR の設定

No.	Hardware ISR 名称	対象割り込み番号	例外 番号	割り込み発生時に 自動実行する サービスコール	自動実行する サービスコールの 対象オブジェクト ID
1	非サイクリック受信 DMA 転送完了	24 (汎用 DMAC チャンネル 2 転送完了割り込み)	40	HWISR_SET_FLG	EVFLGID_NX_DMACOMP02
2	非サイクリック送信 DMA 転送完了	25 (汎用 DMAC チャンネル 3 転送完了割り込み)	41	HWISR_SET_FLG	EVFLGID_NX_DMACOMP03
3	通信ドライバ定期処理周期用タイマ	29 (TAUD チャンネル 2 割り込み)	45	HWISR_WUP_TSK	TSKID_NX_PERIODIC
4	IP フレーム通信周期用タイマ	30 (TAUD チャンネル 3 割り込み)	46	HWISR_WUP_TSK	TSKID_NX_IPCOMM
5	非サイクリックフレーム送信周期用タイマ	31 (TAUD チャンネル 4 割り込み)	47	HWISR_WUP_TSK	TSKID_NX_NCYC_TX
6	非サイクリックフレーム受信周期用タイマ	58 (INTPZ11 入力/TAUD チャンネル 5 割り込み)	74	HWISR_WUP_TSK	TSKID_NX_NCYC_RX
7	時刻同期制御周期用タイマ	59 (INTPZ12 入力/TAUD チャンネル 6 割り込み)	75	HWISR_WUP_TSK	TSKID_NX_TSN
8	TSN 高優先割り込み周期用タイマ	111 (TSN 高優先割り込み)	127	HWISR_WUP_TSK	TSKID_NX_HIGH_INT
9	TSN 低優先割り込み周期用タイマ	112 (TSN 低優先割り込み)	128	HWISR_WUP_TSK	TSKID_NX_LOW_INT

## 6.4 R-IN32M4-CL3 ドライバインタフェース関数詳細

R-IN32M4-CL3 ドライバインタフェース関数の使用方法と機能の詳細を示します。

### 6.4.1 初期設定

#### (1) gulR\_IN\_GetResetStatus

機能	リセット状態取得			
呼出し形式	ULONG gulR_IN_GetResetStatus (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_RESET_PWRON(1) : パワーオンリセット R_IN_RESET_SYSTEM(2) : システムリセット			
説明	リセット状態を取得します。 本関数の呼出しは、「gerR_IN_Initialize」(6.4.1(2))を呼び出す前に実施してください。			

#### (2) gerR\_IN\_Initialize

機能	R-IN32M4-CL3 の初期化			
呼出し形式	ERRCODE gerR_IN_Initialize (const UCHAR *puchMACAddress, const R_IN_UNITINFO_T *pstUnitInfo, const R_IN_UNITINIT_T *pstUnitInit)			
引数	型名	変数名	内容	入出力
	const UCHAR*	puchMACAddress	自局 MAC アドレス 12-34-56-78-90-AB の場合、以下のように設定してください。 puchMACAddress [0] : 12H puchMACAddress [1] : 34H puchMACAddress [2] : 56 H puchMACAddress [3] : 78 H puchMACAddress [4] : 90 H puchMACAddress [5] : AB H	入力
	const R_IN_UNITINFO_T*	pstUnitInfo	R-IN32M4-CL3 ユニット情報 詳細は「表 6.9 R_IN_UNITINFO_T 一覧」を参照してください。	入力
	const R_IN_UNITINIT_T*	pstUnitInit	R-IN32M4-CL3 初期設定情報 詳細は「表 6.17 R_IN_UNITINIT_T 一覧」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了(パラメータエラー) R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	R-IN32M4-CL3 の初期化を行います。			



gerR\_IN\_Initialize の引数「R\_IN\_UNITINFO\_T」の構成を以下に示します。

表 6.9 R\_IN\_UNITINFO\_T 一覧

No.	メンバ		概要	設定内容
1	USHORT	usMaxRySize	RY サイズ	自局が通信可能な RY のサイズ(バイト)を 1 バイト単位で設定します。
2	USHORT	usMaxRWwSize	RWw サイズ	自局が通信可能な RWw のサイズ(ワード)を 2 ワード単位で設定します。
3	USHORT	usMaxRspduSize	RSPDU サイズ	自局が通信可能な RSPDU のサイズ(バイト)を 1 バイト単位で設定します。 安全 PDU 送受信時(コンパイルスイッチ「SAFETY_PDU_ENABLE」を有効にした場合)のみ使用します。
4	USHORT	usMaxRxSize	RX サイズ	自局が通信可能な RX のサイズ(バイト)を 1 バイト単位で設定します。
5	USHORT	usMaxRWrSize	RWr サイズ	自局が通信可能な RWr のサイズ(ワード)を 2 ワード単位で設定します。
6	USHORT	usMaxSspduSize	SSPDU サイズ	自局が通信可能な SSPDU のサイズ(バイト)を 1 バイト単位で設定します。 安全 PDU 送受信時(コンパイルスイッチ「SAFETY_PDU_ENABLE」を有効にした場合)のみ使用します。
7	UCHAR	uchMyStationPortTotalNumber	自局ポート数	自局が保有する物理的な CC-Link IE TSN 用のポート数を設定します。 2 または 1 を設定してください。
8	USHORT	usNetVersion	ネットワークのファームウェアバージョン	ファームウェアバージョンは、ユーザが任意に定義してください。
9	USHORT	usNetModelType	ネットワークの機種タイプ	CC-Link 協会に指定された機種タイプ(deviceType)を設定します。詳細は「3.2 ベンダーコードの取得と機種タイプの選択」を参照してください。
10	ULONG	UINetUnitModelCode	ネットワークの型名コード	型名コードは、ユーザが任意に定義したコードです。同一ベンダーコード内で一意になるように管理してください。
11	USHORT	usNetVendorCode	ネットワークのベンダーコード	CC-Link 協会入会時に取得したベンダーコード(vendorCode)を BCD で設定します。詳細は「3.2 ベンダーコードの取得と機種タイプの選択」を参照してください。
12	UCHAR	auchNetUnitModelName[20]	ネットワークの型名名称	型名名称は、ユーザが任意に定義した名称です。同一ベンダーコード内で一意になるように管理してください。(20 バイトの文字列(ASCII コード))
13	UCHAR	auchNetVendorName[32]	ネットワークのベンダー名称	ベンダー名称は、ユーザが任意に(社名やブランド名など)定義してください。(32 バイトの文字列(ASCII コード))

No.	メンバ		概要	設定内容
14	UCHAR	uchNetHwVersion	ネットワークのハードウェアバージョン	ハードウェアバージョンは、ユーザが任意に定義してください。
15	USHORT	usNetDeviceVersion	ネットワークの機器バージョン	機器バージョンは、開発機器が持つ機能のバージョンを示します。開発機器と CSP+ファイルを対応付けるために使用します。
16	BOOL	blInformationFlag	コントローラ情報有無フラグ	本表の No.17～No.22 および No.25～No.28 の有効/無効を設定します。R_IN_FALSE は無効、R_IN_TRUE は有効を表します。通信機能しか持たない場合、コントローラ情報有無フラグを無効にします。
17	UCHAR	uchCtrlVersion	コントローラのファームウェアバージョン	ファームウェアバージョンは、ユーザが任意に定義してください。
18	USHORT	usCtrlModelType	コントローラの機種タイプ	CC-Link 協会に指定された機種タイプ (deviceType)を設定します。詳細は「3.2 ベンダーコードの取得と機種タイプの選択」を参照してください。
19	ULONG	ulCtrlUnitModelCode	コントローラの型名コード	型名コードは、ユーザが任意に定義したコードです。同一ベンダーコード内で一意になるように管理してください。
20	USHORT	usCtrlVendorCode	コントローラのベンダーコード	CC-Link 協会入会時に取得したベンダーコード (vendorCode)を BCD で設定します。詳細は「3.2 ベンダーコードの取得と機種タイプの選択」を参照してください。
21	UCHAR	auchCtrlUnitModelName[20]	コントローラの型名名称	型名名称は、ユーザが任意に定義した名称です。同一ベンダーコード内で一意になるように管理してください。(20 バイトの文字列(ASCII コード))
22	UCHAR	auchCtrlVendorName[32]	コントローラのベンダー名称	ベンダー名称は、ユーザが任意に定義してください。(32 バイトの文字列(ASCII コード))
23	USHORT	usNetExUnitModelCode	ネットワークの拡張型名コード	ネットワークの拡張型名コードは、ユーザが任意に定義してください。
24	UCHAR	auchNetSerialNumber[32]	ネットワークのシリアル番号	シリアル番号は、ユーザが任意に定義してください。(32 バイトの文字列(ASCII コード))
25	UCHAR	uchCtrlDeviceVersion	コントローラの機器バージョン	機器バージョンは、開発機器が持つ機能のバージョンを示します。開発機器と CSP+ファイルを対応付けるために使用します。
26	UCHAR	uchCtrlHwVersion	コントローラのハードウェアバージョン	ハードウェアバージョンは、ユーザが任意に定義してください。
27	UCHAR	auchCtrlSerialNumber[32]	コントローラのシリアル番号	シリアル番号は、ユーザが任意に定義してください。(32 バイトの文字列(ASCII コード))
28	USHORT	usCtrlExUnitModelCode	コントローラの拡張型名コード	拡張型名コードは、ユーザが任意に定義してください。

No.	メンバ		概要	設定内容
29	USHORT	usStationMode	ステーションモード	CSP+の COMM_IF を識別する情報を設定してください。(0000H~FFFFH)
30	BOOL	blOptionInfoFlag	オプション情報有無フラグ	本表の No.31~No.33 の有効/無効を設定します。R_IN_FALSE は無効、R_IN_TRUE は有効を表します。 増設ユニットまたはスライス I/O を未装着時は R_IN_FALSE を設定してください。 増設ユニットまたはスライス I/O を装着時は R_IN_TRUE を設定してください。
31	BOOL	blOptionInfoExtModuleCyclicPDUSelectFlag	増設ユニットサイクリック伝送 PDU 選択フラグ	オプション情報に該当する増設ユニットのサイクリック伝送 PDU 使用有無に応じて設定してください。 R_IN_FALSE : サイクリック伝送 PDU を使用する増設ユニット (例 : サーボアンプや CANopen における増設軸) R_IN_TRUE : サイクリック伝送 PDU を使用しない増設ユニット (例 : スライス I/O)
32	UCHAR	uchNumberOfOption	オプション情報数	増設ユニットまたはスライス I/O の装着数を設定してください。
33	R_IN_OPTIONINFO_T	astOptionInfo[R_IN_OPTIONTABLE_ENTRY_SIZE]	オプション情報テーブル	詳細は「表 6.10 R_IN_OPTIONINFO_T 一覧」を参照してください。
34	R_IN_PTPINFO_T	stPtpInfo	時刻同期関連情報	詳細は「表 6.11 R_IN_PTPINFO_T 一覧」を参照してください。
35	ULONG	ulCorrespondingFunction	対応機能	各機能の搭載有無を設定してください。 詳細は以下を参照してください。 CC-Link IE TSN Class A : 「表 6.12」 CC-Link IE TSN Class B : 「表 6.13」
36	USHORT	usNumberOfStationSubID	局サブ ID 数	局サブ ID のエントリ数を設定してください。(0001H~1000H) 詳細は「表 5.19 局サブ ID に関する設定値」を参照してください。
37	R_IN_WDCINFO_T	stWdcInfo	ウォッチドッグカウンタ情報	ネットワーク同期通信を行う場合に設定してください。 詳細は「表 6.14 R_IN_WDCINFO_T 一覧」を参照してください。
38	ULONG	ulLinkSpeed	通信速度	通信速度を指定してください。2 ポート共通の設定です。 1Gbps : R_IN_SPEED_1G (0) 100Mbps : R_IN_SPEED_100M (1)
39	USHORT	usCCIETSNClass	CC-Link IE TSN Class	CC Link IE TSN Class を設定してください。 CC-Link IE TSN Class A : 0 CC-Link IE TSN Class B : 1
40	USHORT	usCyclicDataUpdatePeriodic	サイクリックデータ更新周期	定周期処理タスクの実行間隔を設定します。 初期値 : 200 (μs)

No.	メンバ		概要	設定内容
41	ULONG	ullCommunicationCycleMin 1G	通信周期最小値 (1Gbps)	<p>通信周期の最小値を設定します。 初期値を変更する場合は、ユーザアプリケーションにおける制御に問題がないことを十分検証してください。</p> <p>(初期値) CC-Link IE TSN Class A : 1000000 (1000μs) CC-Link IE TSN Class B : 31250 (31,25μs)</p> <p>CC-Link IE TSN Class B の場合、初期値より小さい値を設定しないでください。</p>
42	ULONG	ullCommunicationCycleMax 1G	通信周期最大値 (1Gbps)	<p>通信周期の最大値を設定します。 初期値を変更する場合は、ユーザアプリケーションにおける制御に問題がないことを十分検証してください。</p> <p>(初期値) CC-Link IE TSN Class A : 6400000000 (6400ms) CC-Link IE TSN Class B : 10000000 (10ms)</p> <p>(最大値) 15000000000 (15s)以下</p>
43	ULONG	ullCommunicationCycleMin 100M	通信周期最小値 (100Mbps)	<p>通信周期の最小値を設定します。 初期値を変更する場合は、ユーザアプリケーションにおける制御に問題がないことを十分検証してください。</p> <p>(初期値) CC-Link IE TSN Class A : 1000000 (1000μs) CC-Link IE TSN Class B : 500000 (500μs)</p> <p>CC-Link IE TSN Class B の場合、初期値より小さい値を設定しないでください。</p>
44	ULONG	ullCommunicationCycleMax 100M	通信周期最大値 (100Mbps)	<p>通信周期の最大値を設定します。 初期値を変更する場合は、ユーザアプリケーションにおける制御に問題がないことを十分検証してください。</p> <p>(初期値) CC-Link IE TSN Class A : 6400000000 (6400ms) CC-Link IE TSN Class B : 10000000 (10ms)</p> <p>(最大値) 15000000000 (15s)以下</p>
45	USHORT	usCyclicMaxResponseTime	タイムマネージド・ポーリング 方式応答時間	<p>CC-Link IE TSN Class A の場合は、マスタ局からの定期的なポーリングに対して、自局が応答できる間隔を、2<sup>n</sup>(本値)で設定します。定周期処理タスク(vTask_Periodic)の実行間隔の2倍を目安にしてください。</p> <p>初期値 : 9 (29 = 512 μs)</p> <p>CC-Link IE TSN Class B の場合は、0 を設定してください。</p>

表 6.10 R\_IN\_OPTIONINFO\_T 一覧

No.	メンバ		概要	設定
1	ULONG	ulOptionUnitModelCode	オプション 型名コード	型名コードは、ユーザが任意に定義したコードです。同一ベンダーコード内で一意になるように管理してください。
2	USHORT	usOptionExUnitModelCode	オプション 拡張型名コード	拡張型名コードは、ユーザが任意に定義してください。
3	USHORT	usOptionVendorCode	オプション ベンダーコード	CC-Link 協会入会時に取得したベンダーコード (vendorCode) を BCD で設定します。 詳細は「3.2 ベンダーコードの取得と機種タイプの選択」を参照してください。
4	UCHAR	auchOptionSerialNumber[32]	オプション シリアル番号	シリアル番号は、ユーザが任意に定義してください。(32 バイトの文字列(ASCII コード))
5	UCHAR	uchOptionDeviceVersion	オプション 機器バージョン	オプション機器バージョンは、開発機器が持つ機能のバージョンを示します。 開発機器と CSP+ファイルを対応付けるために使用します。
6	UCHAR	auchOptionModelName[20]	オプション 型名名称	型名名称は、ユーザが任意に定義した名称です。同一ベンダーコード内で一意になるように管理してください。
7	UCHAR	auchOptionVendorName[32]	オプション ベンダー名称	ベンダー名称は、ユーザが任意に設定してください。

表 6.11 R\_IN\_PTPINFO\_T 一覧

No.	メンバ		概要	設定内容
1	UCHAR	uchPriority1	グラントマスタ 優先度 1	グラントマスタを決めるための優先度情報です。 0~255 の値をとり、値が小さいほど優先度が高く、255 の場合はグラントマスタになり得ません。 また、0 の値は管理用に予約されていますので、設定することができません。 CC-Link IE TSN Class A : 255 CC-Link IE TSN Class B : 127
2	ULONG	ulClockQuality	クロック品質	クロック品質について設定します。 各情報がとり得る値の詳細は、「IEEE Std 1588-2008」を参照してください。 第 1 オクテット : clockClass (クロック源の性能値を示す) 第 2 オクテット : clockAccuracy (クロックの精度を示す) 第 3-4 オクテット : offsetScaledLogVariance (クロックの分散値を示す) CC-Link IE TSN Class A : FFFFFFFFH CC-Link IE TSN Class B : F8FE436AH
3	UCHAR	uchPriority2	グラントマスタ 優先度 2	グラントマスタを決めるための第 2 優先度情報です。 Priority1 と同じく 0~255 の値をとり、値が小さいほど優先度が高くなります。 また、0 の値は管理用に予約されていますので、設定することができません。 CC-Link IE TSN Class A : 255 CC-Link IE TSN Class B : 128

表 6.12 USER\_CORRESPONDING\_FUNCTION\_CLASS\_A (CC-Link IE TSN Class A)

ビット	概要	設定内容	初期値	備考
0	ローカル機能	0b : 搭載なし 1b : 搭載あり	0b	初期値の変更を禁止します。
1	設定上書き可否	0b : 不可 1b : 許可	1b	初期値の変更を禁止します。
2	ウォッチドッグカウンタ搭載有無	0b : 搭載なし 1b : 搭載あり	0b	初期値の変更を禁止します。
3	バックアップリストア機能搭載有無	0b : 搭載なし 1b : 搭載あり	0b	初期値の変更を禁止します。
4	安全通信使用有無	0b : 使用しない 1b : 使用する	0b	安全通信を行う場合、 「1b」を設定してください。
5	中継フィルタ設定機能有無	0b : 搭載なし 1b : 搭載あり	1b	-
6	ネットワーク同期通信機能有無	0b : 搭載なし 1b : 搭載あり	0b	初期値の変更を禁止します。
7	分岐マスタ機能	0b : 無効	0b	初期値の変更を禁止します。
8	SNMP コミュニティ名設定対応	0b : 未対応 1b : 対応	0b	-
9-15	予約	-	-	-
16	エラー履歴搭載有無	0b : 搭載なし 1b : 搭載あり	1b	-
17	イベント履歴搭載有無 ※1	0b : 搭載なし 1b : 搭載あり	0b	初期値の変更を禁止します。
18	中継送信禁止機能搭載有無	0b : 搭載なし 1b : 搭載あり	0b	初期値の変更を禁止します。
19	中継時の自局情報格納機能搭載有無	0b : 搭載なし 1b : 搭載あり	1b	初期値の変更を禁止します。
20	IEEE802.1Qbv 対応(IEEE802.1AS)有無	0b : 未対応 1b : 対応	0b	初期値の変更を禁止します。
21	IEEE802.1Qbv 対応(IEEE1588)有無	0b : 未対応 1b : 対応	0b	初期値の変更を禁止します。
22	メッシュ接続対応有無	0b : 未対応 1b : 対応	0b	初期値の変更を禁止します。
23	リング接続対応有無	0b : 未対応 1b : 対応	0b	初期値の変更を禁止します。
24	VLAN 対応	0b : 未対応 1b : 対応	1b	-
25	タイムマネージド・ポーリング方式対応	0b : 未対応 1b : 対応	1b	初期値の変更を禁止します。
26-31	予約	-	-	-

【注】※1 本機能は、SLMP イベント履歴取得コマンド(3061H)でイベント履歴をマスタ局へ送信するものを指します。  
R-IN32M4-CL3 適用製品は別の方法(SNMP)でイベント履歴をマスタ局へ送信しているため、本ビットは  
「0b : 搭載なし」としています。

表 6.13 USER\_CORRESPONDING\_FUNCTION (CC-Link IE TSN Class B)

ビット	概要	設定内容	初期値	備考
0	ローカル機能	0b : 搭載なし 1b : 搭載あり	0b	初期値の変更を禁止します。
1	設定上書き可否	0b : 不可 1b : 許可	1b	初期値の変更を禁止します。
2	ウォッチドッグカウンタ搭載有無	0b : 搭載なし 1b : 搭載あり	0b	ネットワーク同期通信を行う場合、 「1b」を設定してください。
3	バックアップリストア機能搭載有無	0b : 搭載なし 1b : 搭載あり	0b	初期値の変更を禁止します。
4	安全通信使用有無	0b : 使用しない 1b : 使用する	0b	安全通信を行う場合、 「1b」を設定してください。
5	中継フィルタ設定機能有無	0b : 搭載なし 1b : 搭載あり	1b	-
6	ネットワーク同期通信機能有無	0b : 搭載なし 1b : 搭載あり	0b	ネットワーク同期通信を行う場合、 「1b」を設定してください。
7	分岐マスタ機能	0b : 無効	0b	初期値の変更を禁止します。
8	SNMP コミュニティ名設定対応	0b : 未対応 1b : 対応	0b	-
9-15	予約	-	-	-
16	エラー履歴搭載有無	0b : 搭載なし 1b : 搭載あり	1b	-
17	イベント履歴搭載有無 ※1	0b : 搭載なし 1b : 搭載あり	0b	初期値の変更を禁止します。
18	中継送信禁止機能搭載有無	0b : 搭載なし 1b : 搭載あり	1b	-
19	中継時の自局情報格納機能搭載有無	0b : 搭載なし 1b : 搭載あり	1b	初期値の変更を禁止します。
20	IEEE802.1Qbv 対応(IEEE802.1AS)有無	0b : 未対応 1b : 対応	1b	-
21	IEEE802.1Qbv 対応(IEEE1588)有無	0b : 未対応 1b : 対応	1b	-
22	メッシュ接続対応有無	0b : 未対応 1b : 対応	0b	初期値の変更を禁止します。
23	リング接続対応有無	0b : 未対応 1b : 対応	1b	-
24	VLAN 対応	0b : 未対応 1b : 対応	1b	-
25	タイムマネージド・ポーリング方式対応	0b : 未対応 1b : 対応	0b	初期値の変更を禁止します。
26-31	予約	-	-	-

【注】※1 本機能は、SLMP イベント履歴取得コマンド(3061H)でイベント履歴をマスタ局へ送信するものを指します。  
R-IN32M4-CL3 適用製品は別の方法(SNMP)でイベント履歴をマスタ局へ送信しているため、本ビットは  
「0b : 搭載なし」としています。



表 6.14 R\_IN\_WDCINFO\_T 一覧

No.	メンバ		内容
1	R_IN_TRN_SPLD_WDC_INFO_T	stTrnSpldWdcInfo	WDC 情報(送信サブペイロード) 詳細は「表 6.15」を参照してください。
2	R_IN_RCV_SPLD_WDC_INFO_T	stRcvSpldWdcInfo	WDC 情報(受信サブペイロード) 詳細は「表 6.16」を参照してください。
3	USHORT	usWdcThresholdDef	WDC チェック異常連続カウンタ閾値 (デフォルト値)
4	USHORT	usWdcIncrement	WDC インクリメント値

表 6.15 R\_IN\_TRN\_SPLD\_WDC\_INFO\_T 一覧

No.	メンバ		内容
1	USHORT	usWdcOffset	WDC UL オフセット RWrr 先頭からのオフセットをバイト単位で指定します。 FFFFH 指定時はチェック無効です。 <sup>※1</sup>

【注】※1 CANopen 通信時は PDO マッピング設定にてオフセットを設定しますので FFFFH を設定してください。

表 6.16 R\_IN\_RCV\_SPLD\_WDC\_INFO\_T 一覧

No.	メンバ		内容
1	USHORT	usWdcOffset	WDC UL オフセット RWw 先頭からのオフセットをバイト単位で指定します。 FFFFH 指定時はチェック無効です。 <sup>※1</sup>

【注】※1 CANopen 通信時は PDO マッピング設定にてオフセットを設定しますので FFFFH を設定してください。



## RY、RWw、RX、RWr、RSPDU、SSPDU サイズの補足

RY、RWw、RX、RWr、RSPDU、SSPDU のサイズ( usMaxRySize, usMaxRWwSize, usMaxRxSize, usMaxRWrSize, usMaxRspduSize, usMaxSspduSize)設定値に合わせて、サイクリック受信/送信データエリア先頭からのオフセットアドレスを変更してください。

オフセットアドレスは、R\_IN32M4\_CL3MemoryAddress.h ファイル内で定義しています。

R\_IN\_MEMORY\_ADDRESS\_RY (初期値 : 0000 0000H)

R\_IN\_MEMORY\_ADDRESS\_RX (初期値 : 0000 0010H)

R\_IN\_MEMORY\_ADDRESS\_RWW (初期値 : 0000 0200H)

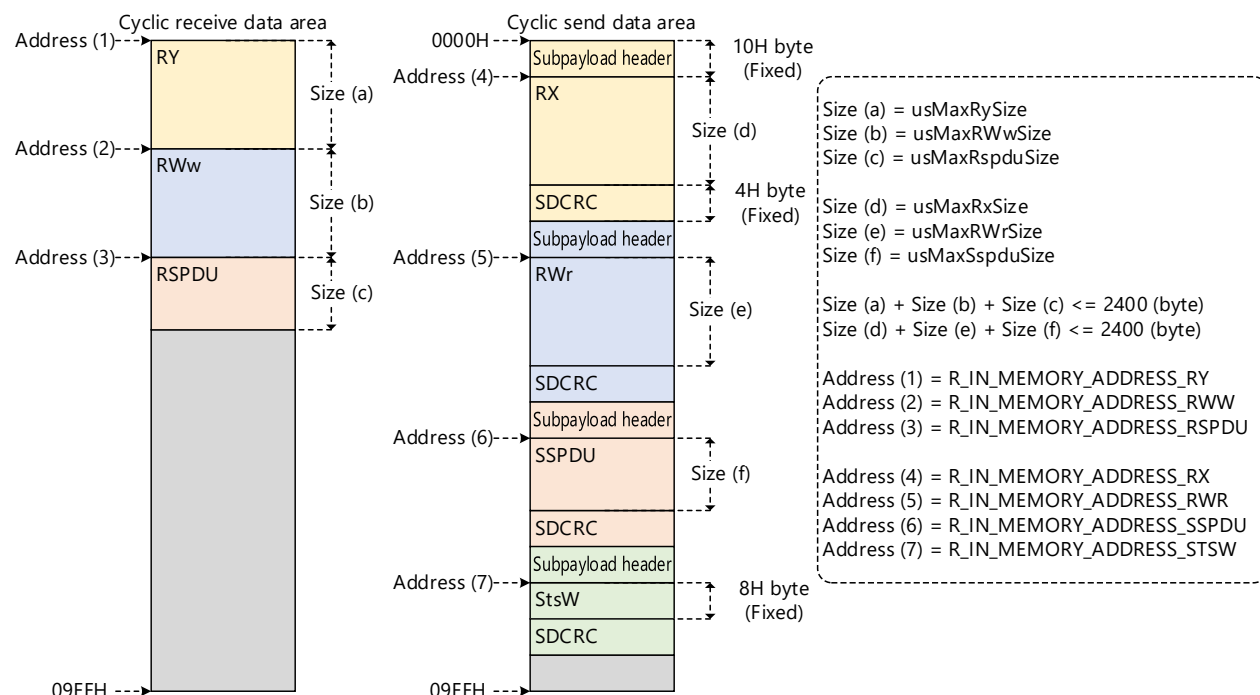
R\_IN\_MEMORY\_ADDRESS\_RWR (初期値 : 0000 0210H)

R\_IN\_MEMORY\_ADDRESS\_RSPDU(初期値 : 0000 0700H)

R\_IN\_MEMORY\_ADDRESS\_SSPDU(初期値 : 0000 0710H)

R\_IN\_MEMORY\_ADDRESS\_STSW (初期値 : 0000 0810H)

RY、RWw、RX、RWr、RSPDU、SSPDU のサイズと、オフセットアドレスの関連イメージを以下に示します。



なお、上記のサイズとオフセットアドレスの設定値に合わせて、CSP+ファイルに記述しているサイズとアドレスを同値に変更してください。

**ネットワークとコントローラの補足****①ネットワークとコントローラの定義**

ネットワーク：自局のうち、R-IN32M4-CL3 および周辺回路で構成した通信機能部分を指します。

コントローラ：自局のうち、ユーザ独自の機能部分(I/O 部分、温調部分、ロボット部分など)を指します。

**②ネットワークの設定**

ネットワークの設定は必須です。コンFORMANCEテストで、以下を確認します。

No.6 ネットワークのファームウェアバージョン

No.8 ネットワークの型名コード

No.7 ネットワークの機種タイプ

No.9 ネットワークのベンダーコード

**③コントローラの設定**

コントローラの設定は任意です。

以下に該当する場合、コントローラを設定してください。(該当しない場合は、コントローラの設定は不要です。)

- ・CSP+ファイルに記述されたベンダーコード/型名コードと、接続しているデバイス局のコントローラの情報を照合したうえで、デバイス局のパラメータ処理・コマンド実行を行いたい場合
- ・R-IN32M4-CL3 適用製品(ネットワーク)はシリーズ製品(コントローラ)などの通信オプション品である場合
- ・コントローラとネットワークの開発メーカーが異なる場合

**機器バージョンの補足****【背景】**

R-IN32M4-CL3 適用製品の S/W バージョン更新などを行う際に、デバイス局パラメータ処理やコマンド実行の追加などの仕様変更を行う場合があります。R-IN32M4-CL3 適用製品の仕様変更を行うと、CSP+ファイルも R-IN32M4-CL3 適用製品の仕様変更に対応して更新する必要があります。

**【機器バージョンの目的】**

仕様変更前と後を識別する情報が機器バージョンであり、各 CSP+ファイルがどの R-IN32M4-CL3 適用製品の仕様に対応しているかを示すために使用されます。

**(a)エンジニアリングツールによる使用目的**

機器バージョンが異なる CSP+ファイルをすべて管理して使い分けることで、使用する R-IN32M4-CL3 適用製品のバージョンに合わせて最適な機能や UI を提供することが可能になります。

**(b)エンドユーザによる使用目的**

CSP+ファイル記載の機器バージョンと、使用する R-IN32M4-CL3 適用製品のバージョンを照らし合わせて、実際に使用する機器用の CSP+ファイルが選択可能になります。

詳細は、「Control & Communication システムプロファイル(CSP+)仕様書」の「DEVICE\_INFO パート」を参照してください。

## オプション情報の補足

R-IN32M4-CL3 適用製品が増設ユニットまたはスライス I/O を装着する場合、装着数に応じてオプション情報に関連する定義を変更する必要があります。

(R-IN32M4-CL3 適用製品が増設ユニットまたはスライス I/O を装着しない場合、変更は不要です。)

オプション情報数 64 の場合で変更例を示します。

No	変更する定義	概要	関連するメンバ	初期値	設定可能範囲	変更例
1	USER_OPTN_INFOFLG	オプション情報有無フラグ	blOptionInfoFlag	R_IN_FALSE	R_IN_TRUE / R_IN_FALSE	R_IN_TRUE
2	USER_OPTN_EXTMODULE_CYCLIC_PDU_SELECTFLAG	増設ユニットサイクリック伝送 PDU 選択フラグ※1	blOptionInfoExtModuleCyclicPDUSelectFlag	R_IN_FALSE	R_IN_TRUE / R_IN_FALSE	R_IN_TRUE
3	USER_NUMBER_OF_OPTION	オプション情報数	uchNumberOfOption	0	0 - 64	64
4	R_IN_OPTIONTABLE_ENTRY_SIZE	オプション情報テーブルエントリ数	astOptionInfo[]	8	0 - 64	64

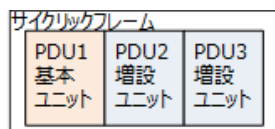
オプション情報を定義することで、MIB の他ユニット情報(オプション情報)に定義した情報が反映されます。

また、GX Works3 の“CC-Link IE TSN 構成画面”で「接続/切断した機器の検出」を実行することで、オプション情報に設定した数分の増設ユニットまたはスライス I/O が自動検出されます。

【注】※1 参考として、増設ユニットがサイクリック伝送 PDU を使用する場合と使用しない場合のサイクリックフレーム構成イメージを以下に示します。

R\_IN\_FALSE

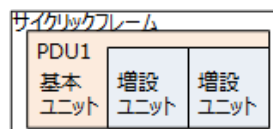
(増設ユニットがサイクリック伝送 PDU を使用する場合)  
基本ユニットが PDU1、PDU2、PDU3 を送受信します。



(例：サーボアンプや CANopen における増設軸)

R\_IN\_TRUE

(増設ユニットがサイクリック伝送 PDU を使用しない場合)  
基本ユニットが増設ユニットのデータを含めて PDU1 を送受信します。



(例：スライス I/O)

gerR\_IN\_Initialize の引数「R\_IN\_UNITINIT\_T」の構成を以下に示します。

表 6.17 R\_IN\_UNITINIT\_T 一覧

No.	メンバ		概要	設定内容
1	BOOL	blHighInterruptUse	高優先割込み使用	高優先割込み機能を使用する場合に“R_IN_TRUE”、使用しない場合“R_IN_FALSE”を設定してください。 “R_IN_TRUE”に設定すると、R-IN32M4-CL3 の高優先割込みが発生したときに RP05 端子が“High”になります。 高優先割込みを使用する場合は、RP05 の兼用機能は使用できません。高優先割込みは、WDT タイムアウトエラーおよび送信異常時に発生します。 高優先割込みの要因が WDT 異常発生であるかを判断するためには、高優先割込み要因レジスタの「自局 WDT 異常発生割込み」が割込み発生の状態であることを確認する必要があります。
2	UCHAR	uchNodeType	局種別	自局の局種別を設定します。 リモート局は“R_IN_NODE_SLAVE” (01H)を設定してください。
3	ULONG	ulSynchronousSignalOutputPulseWidth	同期信号出力パルス幅	ネットワーク同期通信時に同期信号出力機能を使用する(P51 端子を CCI_SYNC として使用する)場合、出力するパルス幅を設定値 n で設定します。(設定値 n : 2~3750) 本設定により、パルス幅は設定値 $n \times 8\text{ns}$ (16ns~30,000ns) で出力されます。同期信号出力機能を使用しない場合、0 を設定してください。

(3) gerR IN SetIPAddress

機能	IP アドレス設定			
呼出し形式	ERRCODE gerR_IN_SetIPAddress (ULONG ulIPAddress, ULONG ulSubnetmask, ULONG ulDefaultGateway)			
引数	型名	変数名	内容	入出力
	ULONG	ulIPAddress	IP アドレス	入力
	ULONG	ulSubnetmask	サブネットマスク	入力
	ULONG	ulDefaultGateway	デフォルトゲートウェイ	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了(ステータス異常) R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	<p>R-IN32M4-CL3 ドライバに IP アドレス関連の初期値を設定します。</p> <p>IP アドレスは ULONG 型で以下のフォーマットで指定します。</p> <div style="text-align: center;"> <pre>           IPアドレス      192.   168.    0.     10                         ↓       ↓       ↓       ↓         MSB            ↓       ↓       ↓       ↓         ulIPAddress [C0] [A8] [00] [0A]                       ↑       ↑       ↑       ↑                     ←1byte→                 </pre> </div> <p>サブネットマスク、およびデフォルトゲートウェイも同様に、ULONG 型の上位バイトから順に、第 1 から第 4 オクテットの値を設定します。</p> <p>IP アドレスは 0.0.0.1～223.255.255.254 の範囲で設定可能です。</p> <p>「gerR_IN_Initialize」(6.4.1(2))を実行した後、「gerR_IN_Start」(6.4.1(5))を実行する前に実行してください。上記以外で実行した場合、R_IN_ERR のエラーとなります。</p> <p>DetectionAck フレームの「ipAddressFourthOctet」に ulIPAddress の第 4 オクテットを設定します。 ipAddressFourthOctet に 00H を設定する(ネットワーク経由の IP アドレス設定を実施する場合)、 「gerR_IN_SetIPAddressAndNodeNumber」(6.4.1(4))を使用してください。</p>			

## (4) gerR\_IN\_SetIPAddressAndNodeNumber

機能	IP アドレス・ノード番号設定			
呼出し形式	ERRCODE gerR_IN_SetIPAddressAndNodeNumber ( ULONG ulIPAddress, ULONG ulSubnetmask, ULONG ulDefaultGateway, UCHAR uchNodeNumber )			
引数	型名	変数名	内容	入出力
	ULONG	ulIPAddress	IP アドレス	入力
	ULONG	ulSubnetmask	サブネットマスク	入力
	ULONG	ulDefaultGateway	デフォルトゲートウェイ	入力
	UCHAR	uchNodeNumber	ノード番号	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 (ステータス異常) R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	<p>R-IN32M4-CL3 ドライバに IP アドレス関連の初期値、およびノード番号を設定します。 IP アドレスは ULONG 型で以下のフォーマットで指定します。</p> <div style="text-align: center;"> <p>IPアドレス 192. 168. 0. 10 MSB ↓ ↓ ↓ ↓ LSB ulIPAddress C0 A8 00 0A ←1byte→</p> </div> <p>サブネットマスク、およびデフォルトゲートウェイも同様に、ULONG 型の上位バイトから順に、第 1 から第 4 オクテットの値を設定します。 IP アドレスは 0.0.0.1~223.255.255.254 の範囲で設定可能です。 「gerR_IN_Initialize」(6.4.1(2))を実行した後、「gerR_IN_Start」(6.4.1(5))を実行する前に実行してください。上記以外で実行した場合、R_IN_ERR のエラーとなります。</p> <p>DetectionAck フレームの「ipAddressFourthOctet」にノード番号を設定します。ノード番号が FFH の場合は、ipAddressFourthOctet に 00H を設定します。</p>			

## (5) gerR\_IN\_Start

機能	R-IN32M4-CL3 通信開始			
呼出し形式	ERRCODE gerR_IN_Start (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	<p>データリンクを開始します。 本関数を呼び出すと R-IN32M4-CL3 の内部 WDT が無効になります。 内部 WDT を使用する場合は、「gerR_IN_EnableWDT」(6.4.2(4))を呼び出してください。 「gerR_IN_Initialize」(6.4.1(2))が完了した後に一度だけ実行してください。 IP アドレスが未設定の場合は、マスタ局からの IP アドレス設定が必要となります。 上記以外で実行した場合、R_IN_ERR のエラーとなります。</p>			

## (6) gvR\_IN\_RegistIPAddressFilteringFunction

機能	アドレスフィルタリング処理登録			
呼出し形式	VOID gvR_IN_RegistIPAddressFilteringFunction (R_IN_IP_FILTERING_FUNCTION fpulFunction)			
引数	型名	変数名	内容	入出力
	R_IN_IP_FILTERING_FUNCTION	fpulFunction	フィルタリング判定処理※1 NULL : IP フィルタリング無効 NULL 以外 : IP フィルタリング有効	-
戻り値	-			
説明	<p>IP アドレスフィルタリングを行なうための判定用の処理を登録します。</p> <p>IP アドレスフィルタリングが必要ない場合は、本関数を実行する必要はありません。</p> <p>本処理では、引数で渡された関数を USNetPlus の「void regist_ip_filtr_func(IPFLTRFUNC pfunc)」（IP アドレスフィルタリング処理登録）を使用して USNetPlus に登録します。</p> <p>登録した関数は、IP フレーム受信時に USNetPlus から呼び出され、受信処理の判定に使用されます。</p> <p>「gerR_IN_Initialize」(6.4.1(2))が完了した後に一度だけ実行してください。</p>			

【注】※1 受信フレームの IP アドレスを元に受信処理を行なうかフレームを破棄するか判定します。詳細を以下に示します。

呼出し形式	ULONG ulFunction (ULONG ulIPAddress)			
引数	型名	変数名	内容	入出力
	ULONG	ulIPAddress	受信フレームの IP アドレス	-
戻り値	<p>1 : 受信処理を行なう</p> <p>1 以外 : 受信処理を行わずフレームを破棄する</p>			

## (7) gerR\_IN\_MIBLedTableDefine

機能	LED ステータス情報 MIB 定義			
呼出し形式	ERRCODE gerR_IN_MIBLedTableDefine (UCHAR uchNumberOfLED, const R_IN_MIBLEDDDEFINE_T *pstMIBLedDefine)			
引数	型名	変数名	内容	入出力
	UCHAR	uchNumberOfLED	LED エントリ数	入力
	const R_IN_MIBLEDDDEFINE_T*	pstMIBLedDefine	LED 定義情報 詳細は「表 6.18」を参照してください。	入力
戻り値	<p>R_IN_OK : 正常完了</p> <p>R_IN_ERR : 異常完了</p> <p>R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)</p>			
説明	<p>MIB で管理する LED 情報のエントリを定義します。</p> <p>本関数を実行することで、「gerR_IN_Start」(6.4.1(5))が完了した後から定周期に最新の LED 状態が MIB に格納されます。</p>			

表 6.18 R\_IN\_MIBLEDDEFINE\_T 一覧

No.	メンバ		内容
1	UCHAR	uchIdentification	LED 種別 R_IN_LED_ID_NOTUSE : 未使用 R_IN_LED_ID_RUN : RUN LED R_IN_LED_ID_USER1 : USER LED 1 R_IN_LED_ID_USER2 : USER LED 2 R_IN_LED_ID_DLINK : DLINK LED R_IN_LED_ID_ERR : ERR LED R_IN_LED_ID_SD_SDRD1 : SD/SDRD1 LED R_IN_LED_ID_RD_SDRD2 : RD/SDRD2 LED R_IN_LED_ID_LER1 : LER 1 LED R_IN_LED_ID_LER2 : LER 2 LED R_IN_LED_ID_LINK1 : LINK 1 LED R_IN_LED_ID_LINK2 : LINK 2 LED R_IN_LED_ID_LINK : LINK LED
3	UCHAR	uchColor	LED 色 R_IN_LED_NOTUSE : 未使用 R_IN_LED_COLOR_GREEN : 緑 R_IN_LED_COLOR_RED : 赤 R_IN_LED_COLOR_ORANGE : 橙 R_IN_LED_COLOR_OTHER : その他



## (8) gerR\_IN\_RegistCallback

機能	コールバック処理登録			
呼出し形式	ERRCODE gerR_IN_RegistCallback (ULONG ulFunctionType, R_IN_CALLBACK_FUNCTION fpulFunction)			
引数	型名	変数名	内容	入出力
	ULONG	ulFunctionType	コールバック処理タイプ 詳細は「表 6.19」を参照してください。	入力
	R_IN_CALLBACK_FUNCTION	fpulFunction	コールバック処理 <sup>※1</sup> NULL : ulFunctionType で指定した処理を無効とする NULL 以外 : ulFunctionType で指定した処理を有とする  typedef ULONG(*R_IN_CALLBACK_FUNCTION) (ULONG, ULONG)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	R-IN32M4-CL3 ドライバからのコールバックに対応する処理を登録します。			

【注】 ※1 : 詳細を以下に示します。

呼出し形式	ULONG ulFunction( ULONG ulParam1, ULONG ulParam2 )
引数	コールバック時の R-IN32M4-CL3 ドライバからの付加情報となります。 詳細は「表 6.19」を参照してください。
戻り値	コールバック処理の実行結果を R-IN32M4-CL3 ドライバへ通知します。 詳細は「表 6.19」を参照してください。
説明	上記の ulFunctionType の指定に従い、該当するタイミングで本関数がコールバックされます。 通知タイミングに従い、任意処理を行ってください。

表 6.19 コールバック処理仕様

No	コールバック処理タイプ (処理名) <参照先>	実行時 ディスパッチ 許可/禁止状態	引数		戻り値
			ulParam1	ulParam2	
1	R_IN_FUNCTIONTYPE_TIMESYNC_COMPLETE (データリンク高速化(時刻同期完了時)) <6.6(5)>	ディスパッチ 禁止	0 固定	0 固定	0 固定
2	R_IN_FUNCTIONTYPE_CYCLICDATA_INITIALIZE (サイクリック送信データ初期化) <6.6(4)>	ディスパッチ 禁止	0 固定	0 固定	0 固定
3	R_IN_FUNCTIONTYPE_CYCLIC_START (データリンク高速化(サイクリック伝送開始時)) <6.6(6)>	ディスパッチ 禁止	0 固定	0 固定	0 固定
4	R_IN_FUNCTIONTYPE_DISCONNECT_PARTWAY_THROUGH (データリンク高速化(途中解列時)) <6.6(7)>	ディスパッチ 禁止	0 固定	0 固定	0 固定
5	R_IN_FUNCTIONTYPE_COMCYLCE_DEFINITION (通信周期判定処理) <6.6(8)>	ディスパッチ 禁止	通信周期 情報の先 頭アドレ ス <sup>※2</sup>	0 固定	0 : 正常 0 以外 : 異常
6	R_IN_FUNCTIONTYPE_SYNC_CYCLIC_COM (同期サイクリック通信処理) <6.6(9)>	ディスパッチ 禁止	0 固定	0 固定	0 固定

【注】 ※2 : 通信周期情報は「表 6.47」を参照してください。

## 6.4.2 ウォッチドッグ・タイマ

## (1) gerR\_IN\_ResetWDT

機能	R-IN32M4-CL3 内部 WDT リセット			
呼出し形式	ERRCODE gerR_IN_ResetWDT (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了			
説明	R-IN32M4-CL3 内部 WDT をリセットします。			

## (2) gerR\_IN\_SetWDT

機能	R-IN32M4-CL3 内部 WDT 時限設定			
呼出し形式	ERRCODE gerR_IN_SetWDT (USHORT usWDTCOUNT)			
引数	型名	変数名	内容	入出力
	USHORT	usWDTCOUNT	WDT 時限設定	入力
戻り値	R_IN_OK : 正常完了			
説明	R-IN32M4-CL3 内部 WDT の時限を設定します。  設定範囲 0000H : 100ms 0001H : 200ms 0002H : 300ms ... 001FH : 3200ms			

## (3) gerR\_IN\_DisableWDT

機能	R-IN32M4-CL3 内部 WDT 無効			
呼出し形式	ERRCODE gerR_IN_DisableWDT (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了			
説明	R-IN32M4-CL3 内部 WDT を無効にします。  リセット直後の R-IN32M4-CL3 は“WDT 有効”になっています。 サンプルコードは「gerR_IN_Start」(6.4.1(5))にて“WDT 無効”にします。 電源 ON から「gerR_IN_Start」を呼び出すまでに時間がかかる(WDT エラーになる)場合、以下いずれかを実施してください。 <ul style="list-style-type: none"> <li>リセット解除後に本関数を呼び出し、“WDT 無効”とする。</li> <li>リセット解除後に「gerR_IN_ResetWDT」(6.4.2(1))を呼び出し、内部 WDT をリセットする。</li> </ul>			

## (4) gerR\_IN\_EnableWDT

機能	R-IN32M4-CL3 内部 WDT 有効			
呼出し形式	ERRCODE gerR_IN_EnableWDT (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了			
説明	R-IN32M4-CL3 内部 WDT を有効にします。			

## 6.4.3 イベント

## (1) gerR\_IN\_GetEvent

機能	R-IN32M4-CL3 イベント検出			
呼出し形式	ERRCODE gerR_IN_GetEvent (R_IN_EVTPRM_INTERRUPT_T* pstEvent)			
引数	型名	変数名	内容	入出力
	R_IN_EVTPRM_INTERRUPT_T*	pstEvent	割込み要因 詳細は「表 6.20」を参照してください。	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	発生している割込みイベントを検出し、検出結果を出力します。 引数に NULL ポインタが指定された場合、R_IN_ERR(異常完了)となります。 割込みイベント検出後は割込み要因をマスクします。			

## (2) gerR\_IN\_Main

機能	R-IN32M4-CL3 イベント検出メイン処理			
呼出し形式	ERRCODE gerR_IN_Main (const R_IN_EVTPRM_INTERRUPT_T* pstEvent)			
引数	型名	変数名	内容	入出力
	const R_IN_EVTPRM_INTERRUPT_T*	pstEvent	割込み要因 詳細は「表 6.20」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	検出したイベント情報にもとづき、R-IN32M4-CL3 のイベント処理を実行します。 引数に NULL ポインタが指定された場合、R_IN_ERR(異常完了)となります。			

表 6.20 R\_IN\_EVTPRM\_INTERRUPT\_T 一覧

共用体	メンバ(1)		メンバ(2)		概要
uniFlag	ULONG	usAll	-	-	-
	Struct	stBit	ULONG	b1ZCommConnect:1	(bit 0) 通信参加
			ULONG	b1ZCommDisconnect:1	(bit 1) 通信解列
			ULONG	b5ZReserve1:5	(bit2-6) 予約
			ULONG	b1ZChangeIPAddress:1	(bit7) IP アドレス更新 <sup>※1</sup>
			ULONG	b24ZReserve2:24	(bit8-31) 予約

【注】※1 自局の IP アドレス第 1～第 3 オクテットがマスタ局のものと異なっている場合、自局の IP アドレス第 1～第 3 オクテットはマスタ局と通信を開始するときにマスタ局の指定した IP アドレスに変更されます。このタイミングのみ、bit 7 が ON します。タイミングについては「図 5.20」を参照してください。

## (3) gerR\_IN\_RestartEvent

機能	R-IN32M4-CL3 イベント検出再開			
呼出し形式	ERRCODE gerR_IN_RestartEvent (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了			
説明	割込み要因をマスクしているものについて、割込み要因のマスクを解除します。			

## 6.4.4 サイクリック伝送

## (1) gerR\_IN\_GetMasterNodeStatus

機能	マスタ局状態取得			
呼出し形式	ERRCODE gerR_IN_GetMasterNodeStatus (BOOL *pblRunSts, BOOL *pblErrSts)			
引数	型名	変数名	内容	入出力
	BOOL*	pblRunSts	マスタ局アプリケーション動作状態 R_IN_TRUE : 実行中 R_IN_FALSE : 停止中	出力
	BOOL*	pblErrSts	マスタ局アプリケーションエラー状態 R_IN_TRUE : エラー発生中 R_IN_FALSE : エラーなし	出力
戻り値	R_IN_OK : 正常完了(マスタ局からのサイクリックフレーム受信あり。) R_IN_ERR : 異常完了(解列中のため、マスタ局からのサイクリックフレームを受信していない。)			
説明	マスタ局から受信したサイクリックフレームよりマスタ局の状態を取得します。 解列中のため、マスタ局からのサイクリックフレームを受信していない場合、引数は以下となります。  pblRunSts R_IN_FALSE pblErrSts R_IN_FALSE			

## (2) gerR\_IN\_GetHoldClearFlag

機能	Hold/Clear フラグ取得			
呼出し形式	ERRCODE gerR_IN_GetHoldClearFlag (BOOL *pblHoldClearFlag)			
引数	型名	変数名	内容	入出力
	BOOL*	pblHoldClearFlag	Hold/Clear フラグ R_IN_TRUE : HOLD/CLEAR 発生中 R_IN_FALSE : HOLD/CLEAR 未発生	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	Hold/Clear フラグを取得します。 「gerR_IN_UpdateReceivedCyclicData」(6.4.4(4))を実行することで最新の情報に更新されます。 Hold/Clear フラグは下記の条件で R_IN_TRUE になります。 ・ 自局のデータリンク状態が解列 ・ マスタ局のアプリケーション状態が異常または停止  受信したサイクリックデータ(RY および RWw)を外部へ出力する機器の場合、R_IN_TRUE 時に、出力を 続行する(Hold)または停止する(Clear)処理を必要に応じて追加してください。 なお、マスタ局と通信を確立するまで間は R_IN_TRUE になります。			

## (3) gerR\_IN\_GetReceivedCyclicData

機能	サイクリック受信データ取得(一括)			
呼出し形式	一般通信時	ERRCODE gerR_IN_GetReceivedCyclicData (VOID *pRyDst, VOID *pRWwDst, BOOL blEnable)		
	安全通信時	ERRCODE gerR_IN_GetReceivedCyclicData (VOID* pRyDst, VOID* pRWwDst, VOID* pRspduDst, BOOL blEnable)		
引数	型名	変数名	内容	入出力
	VOID*	pRyDst	RY 領域 <sup>※1</sup>	出力
	VOID*	pRWwDst	RWw 領域 <sup>※1</sup>	出力
	VOID*	pRspduDst	RSPDU 領域 <sup>※1※2</sup>	出力
	BOOL	blEnable	コピー有効／無効 R_IN_TRUE : 有効 R_IN_FALSE : 無効	入力
戻り値	R_IN_OK : 正常完了(受信データあり) R_IN_ERR : 異常完了(受信データなし)			
説明	<p>マスタ局から受信したサイクリック受信データを pRyDst、pRWwDst および pRspduDst の示すアドレスに格納します。ただし、blEnable が R_IN_FALSE の場合、受信したサイクリック受信データは破棄します。(戻り値は、R_IN_ERR となります。)</p> <p>CC-Link IE TSN Class A 動作時は、blEnable を R_IN_TRUE に設定してください。</p> <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>6.4.4(4) gerR_IN_UpdateReceivedCyclicData</li> <li>6.4.4(5) gerR_IN_FinishReceivedCyclicDataAcquisition</li> <li>6.4.4(6) gerR_IN_GetReceivedCyclicRY</li> <li>6.4.4(7) gerR_IN_GetReceivedCyclicRWw</li> <li>6.4.4(14) gerR_IN_GetReceivedCyclicBuffer</li> <li>6.4.4(15) gerR_IN_FinishReceivedCyclicBuffer</li> </ul>			

【注】 ※1 先頭アドレスは、4 バイト単位としてください。

※2 コンパイルスイッチ「SAFETY\_PDU\_ENABLE」を有効にした場合

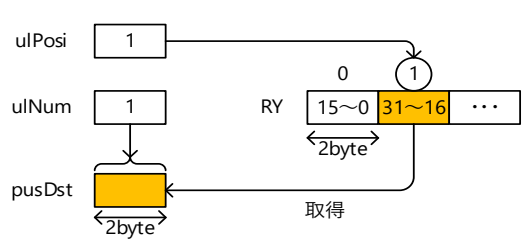
## (4) gerR\_IN\_UpdateReceivedCyclicData

機能	サイクリック受信データ更新			
呼出し形式	ERRCODE gerR_IN_UpdateReceivedCyclicData (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	<p>本関数を実行することで、最新の RY/RWw を以下で取得することができます。</p> <ul style="list-style-type: none"> <li>6.4.4(6) gerR_IN_GetReceivedCyclicRY</li> <li>6.4.4(7) gerR_IN_GetReceivedCyclicRWw</li> </ul> <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>6.4.4(3) gerR_IN_GetReceivedCyclicData</li> <li>6.4.4(14) gerR_IN_GetReceivedCyclicBuffer</li> <li>6.4.4(15) gerR_IN_FinishReceivedCyclicBuffer</li> </ul>			

## (5) gerR\_IN\_FinishReceivedCyclicDataAcquisition

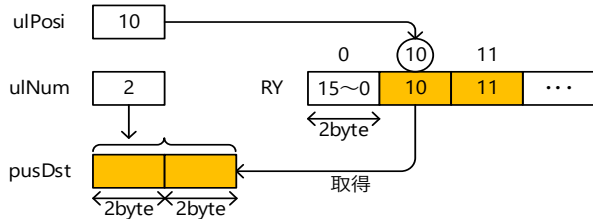
機能	サイクリック受信データ取得完了			
呼出し形式	ERRCODE gerR_IN_FinishReceivedCyclicDataAcquisition (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	<p>サイクリック受信データの取得が完了したことを R-IN32M4-CL3 へ通知します。</p> <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>6.4.4(3) gerR_IN_GetReceivedCyclicData</li> <li>6.4.4(14) gerR_IN_GetReceivedCyclicBuffer</li> <li>6.4.4(15) gerR_IN_FinishReceivedCyclicBuffer</li> </ul>			

## (6) gerR\_IN\_GetReceivedCyclicRY

機能	サイクリック受信データ取得(RY)			
呼出し形式	ERRCODE gerR_IN_GetReceivedCyclicRY (ULONG ulPosi, ULONG ulNum, USHORT *pusDst)			
引数	型名	変数名	内容	入出力
	ULONG	ulPosi	RY の取得開始位置	入力
	ULONG	ulNum	RY の取得データ数(ワード)	入力
	USHORT*	pusDst	RY の格納先領域※1	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(受信データなし) R_IN_ERR_OUTOFRANGE : 異常完了(範囲外) R_IN_ERR_BOUNDARY : 異常完了(バウンダリ異常)			
説明	<p>「6.4.4(4) gerR_IN_UpdateReceivedCyclicData」を実行することで、最新の RY を、pusDst の示すアドレスに格納します。</p>  <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>6.4.4(3) gerR_IN_GetReceivedCyclicData</li> <li>6.4.4(14) gerR_IN_GetReceivedCyclicBuffer</li> <li>6.4.4(15) gerR_IN_FinishReceivedCyclicBuffer</li> </ul>			

【注】※1 先頭アドレスは、2 バイト単位としてください。

## (7) gerR\_IN\_GetReceivedCyclicRWw

機能	サイクリック受信データ取得(RWw)			
呼出し形式	ERRCODE gerR_IN_GetReceivedCyclicRWw (ULONG ulPosi, ULONG ulNum, USHORT *pusDst)			
引数	型名	変数名	内容	入出力
	ULONG	ulPosi	RWw の取得開始位置	入力
	ULONG	ulNum	RWw の取得データ数(ワード)	入力
	USHORT*	pusDst	RWw の格納先領域※1	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(受信データなし) R_IN_ERR_OUTOFRANGE : 異常完了(範囲外) R_IN_ERR_BOUNDARY : 異常完了(バウンダリ異常)			
説明	<p>「6.4.4(4) gerR_IN_UpdateReceivedCyclicData」を実行することで、最新の RWw を、pusDst の示すアドレスに格納します。</p>  <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>6.4.4(3) gerR_IN_GetReceivedCyclicData</li> <li>6.4.4(14) gerR_IN_GetReceivedCyclicBuffer</li> <li>6.4.4(15) gerR_IN_FinishReceivedCyclicBuffer</li> </ul>			

【注】※1 先頭アドレスは、2 バイト単位としてください。

## (8) gerR\_IN\_GetReceivedCyclicRspdu

機能	サイクリック受信データ取得(RSPDU)			
呼出し形式	ERRCODE gerR_IN_GetReceivedCyclicRspdu(USHORT* pusDst)			
引数	型名	変数名	内容	入出力
	USHORT*	pusDst	RSPDU の格納先領域※1	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(受信データなし) R_IN_ERR_BOUNDARY : 異常完了(バウンダリ異常)			
説明	<p>安全マスタ局から受信したサイクリック受信データ(RSPDU の S-Header 以降の値)を pusDst の示すアドレスに格納します。</p> <p>「6.4.4(4) gerR_IN_UpdateReceivedCyclicData」を実行することで、最新の RWw を、pusDst の示す実行することで、最新の通信周期に受信した RSPDU を、pusDst の示すアドレスに格納します。</p> <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>6.4.4(3) gerR_IN_GetReceivedCyclicData</li> <li>6.4.4(14) gerR_IN_GetReceivedCyclicBuffer</li> <li>6.4.4(15) gerR_IN_FinishReceivedCyclicBuffer</li> </ul>			

【注】※1 先頭アドレスは、2 バイト単位としてください。



## (9) gerR\_IN\_SetSendCyclicData

機能	サイクリック送信データ設定(一括)			
呼出し形式	一般通信時	ERRCODE gerR_IN_SetSendCyclicData (const VOID *pRxSrc, const VOID* pRWrSrc, BOOL blEnable)		
	安全通信時	ERRCODE gerR_IN_SetSendCyclicData (const VOID* pRxSrc, const VOID* pRWrSrc, const VOID* pSspduSrc, BOOL blEnable)		
引数	型名	変数名	内容	入出力
	const VOID*	pRxSrc	RX 領域 <sup>※1</sup>	入力
	const VOID*	pRWrSrc	RWr 領域 <sup>※1</sup>	入力
	const VOID*	pSspduSrc	SSPDU 領域 <sup>※1※2</sup>	入力
	BOOL	blEnable	更新有効／無効 R_IN_TRUE : 有効 R_IN_FALSE : 無効	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	<p>pRxSrc、pRWrSrc、および pSspduSrc に示すアドレスに格納されているサイクリック送信データを R-IN32M4-CL3 に設定します。R-IN32M4-CL3 に設定した RX および RWr は次の通信周期で送信します。ただし、blEnable が R_IN_FALSE の場合、サイクリック送信データの設定は行いません。CC-Link IE TSN Class A 動作時は、blEnable を R_IN_TRUE に設定してください。</p> <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>6.4.4(10) gerR_IN_FinishSendCyclicDataSetting</li> <li>6.4.4(11) gerR_IN_SetSendCyclicRX</li> <li>6.4.4(12) gerR_IN_SetSendCyclicRWr</li> <li>6.4.4(16) gerR_IN_GetSendCyclicBuffer</li> <li>6.4.4(17) gerR_IN_GetSplitSendCyclicBuffer</li> <li>6.4.4(18) gerR_IN_FinishSendCyclicBuffer</li> </ul>			

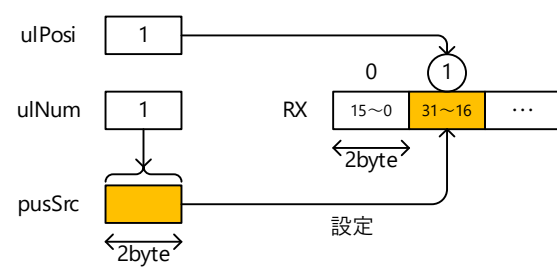
【注】※1 先頭アドレスは、4 バイト単位としてください。

※2 コンパイルスイッチ「SAFETY\_PDU\_ENABLE」を有効にした場合

## (10) gerR\_IN\_FinishSendCyclicDataSetting

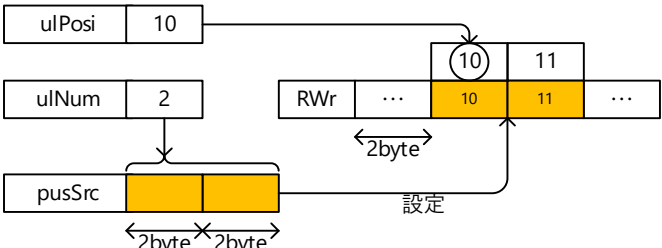
機能	サイクリック送信データ設定完了			
呼出し形式	ERRCODE gerR_IN_FinishSendCyclicDataSetting (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(CCIETSN ClassA 動作時に送信データ設定不要)			
説明	<p>「6.4.4(11) gerR_IN_SetSendCyclicRX」、および「6.4.4(12) gerR_IN_SetSendCyclicRWr」により、サイクリック送信データを設定した場合は、本関数を実行してください。本関数を実行することで、R-IN32M4-CL3 に設定した RX および RWr と、最新の情報に更新した自局状態情報を次の通信周期で送信します。</p> <p>CC-Link IE TSN Class A 動作時に、サイクリックデータを受信していない場合、戻り値で「R_IN_ERR_NODATA」を返します。</p> <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>• 6.4.4(9) gerR_IN_SetSendCyclicData</li> <li>• 6.4.4(16) gerR_IN_GetSendCyclicBuffer</li> <li>• 6.4.4(17) gerR_IN_GetSplitSendCyclicBuffer</li> <li>• 6.4.4(18) gerR_IN_FinishSendCyclicBuffer</li> </ul>			

## (11) gerR\_IN\_SetSendCyclicRX

機能	サイクリック送信データ設定(RX)			
呼出し形式	ERRCODE gerR_IN_SetSendCyclicRX (const USHORT *pusSrc, ULONG ulPosi, ULONG ulNum)			
引数	型名	変数名	内容	入出力
	const USHORT*	pusSrc	RX の設定元領域※1	入力
	ULONG	ulPosi	RX の設定開始位置	入力
	ULONG	ulNum	RX の設定データ数(ワード)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外) R_IN_ERR_BOUNDARY : 異常完了(バウンダリ異常) R_IN_ERR_NODATA : 異常完了(CCIETSN ClassA 動作時に送信データ設定不要)			
説明	<p>pusSrc に示すアドレスに格納されているサイクリック送信データを R-IN32M4-CL3 に設定します。設定した RX は、「gerR_IN_FinishSendCyclicDataSeting」(6.4.4(10))を実行することで、次の通信周期に送信されます。</p>  <p>CC-Link IE TSN Class A 動作時に、サイクリックデータを受信していない場合、戻り値で「R_IN_ERR_NODATA」を返します。</p> <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>6.4.4(9) gerR_IN_SetSendCyclicData</li> <li>6.4.4(16) gerR_IN_GetSendCyclicBuffer</li> <li>6.4.4(17) gerR_IN_GetSplitSendCyclicBuffer</li> <li>6.4.4(18) gerR_IN_FinishSendCyclicBuffer</li> </ul>			

【注】 ※1 先頭アドレスは、2 バイト単位としてください。

## (12) gerR\_IN\_SetSendCyclicRWr

機能	サイクリック送信データ設定(RWr)			
呼出し形式	ERRCODE gerR_IN_SetSendCyclicRWr (const USHORT *pusSrc, ULONG ulPosi, ULONG ulNum)			
引数	型名	変数名	内容	入出力
	const USHORT*	pusSrc	RWr の設定元領域※1	入力
	ULONG	ulPosi	RWr の設定開始位置	入力
	ULONG	ulNum	RWr の設定データ数(ワード)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外) R_IN_ERR_BOUNDARY : 異常完了(バウンダリ異常) R_IN_ERR_NODATA : 異常完了(CCIETSN ClassA 動作時に送信データ設定不要)			
説明	<p>pusSrc に示すアドレスに格納されているサイクリック送信データを R-IN32M4-CL3 に設定します。設定した RWr は、「gerR_IN_FinishSendCyclicDataSeting」(6.4.4(10))を実行することで、次の通信周期に送信されます。</p>  <p>CC-Link IE TSN Class A 動作時に、サイクリックデータを受信していない場合、戻り値で「R_IN_ERR_NODATA」を返します。</p> <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>6.4.4(9) gerR_IN_SetSendCyclicData</li> <li>6.4.4(16) gerR_IN_GetSendCyclicBuffer</li> <li>6.4.4(17) gerR_IN_GetSplitSendCyclicBuffer</li> <li>6.4.4(18) gerR_IN_FinishSendCyclicBuffer</li> </ul>			

【注】 ※1 先頭アドレスは、2 バイト単位としてください。

## (13) gerR\_IN\_SetSendCyclicSspdu

機能	サイクリック送信データ設定(SSPDU)			
呼出し形式	ERRCODE gerR_IN_SetSendCyclicSspdu(const USHORT* pusSrc)			
引数	型名	変数名	内容	入出力
	const USHORT*	pusSrc	SSPDU の設定元領域※1	入力
戻り値	R_IN_OK : 正常完了(受信データあり) R_IN_ERR : 異常完了 R_IN_ERR_BOUNDARY : 異常完了(バウンダリ異常) R_IN_ERR_NODATA : 異常完了(CCIETSN ClassA 動作時に送信データ設定不要)			
説明	pusSrc に示すアドレスに格納されているサイクリック送信データ(SSPDU の S-Header 先頭アドレス)を R-IN32M4-CL3 に設定します。設定した SSPDU は、「gerR_IN_FinishSendCyclicDataSeting」(6.4.4(10))を実行することで、次の通信周期に送信されます。 CC-Link IE TSN Class A 動作時に、サイクリックデータを受信していない場合、戻り値で「R_IN_ERR_NODATA」を返します。  本関数を使用する場合は、以下を使用しないでください。 <ul style="list-style-type: none"> <li>6.4.4(9) gerR_IN_SetSendCyclicData</li> <li>6.4.4(16) gerR_IN_GetSendCyclicBuffer</li> <li>6.4.4(17) gerR_IN_GetSplitSendCyclicBuffer</li> <li>6.4.4(18) gerR_IN_FinishSendCyclicBuffer</li> </ul>			

【注】※1 先頭アドレスは、2 バイト単位としてください。

## (14) gerR\_IN\_GetReceivedCyclicBuffer

機能	サイクリック受信バッファ取得			
呼出し形式	一般通信時	ERRCODE gerR_IN_GetReceivedCyclicBuffer (ULONG* pulRyBuffer, ULONG* pulRWwBuffer)		
	安全通信時	ERRCODE gerR_IN_GetReceivedCyclicBuffer(ULONG* pulRyBuffer, ULONG* pulRWwBuffer, ULONG* pulRspduBuffer)		
引数	型名	変数名	内容	入出力
	ULONG*	pulRyBuffer	RY 格納先アドレス	出力
	ULONG*	pulRWwBuffer	RWw 格納先アドレス	出力
	ULONG*	pulRspduBuffer	RSPDU 格納先アドレス※1	出力
戻り値	R_IN_OK : 正常完了(受信データあり) R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(受信データなし)			
説明	マスタ局から受信したサイクリック受信データの格納先アドレスを取得します。 サイクリック受信データの取得完了後は、「6.4.4(15) gerR_IN_FinishReceivedCyclicBuffer」を実行してください。  本関数を使用する場合は、以下を使用しないでください。 <ul style="list-style-type: none"> <li>6.4.4(3) gerR_IN_GetReceivedCyclicData</li> <li>6.4.4(4) gerR_IN_UpdateReceivedCyclicData</li> <li>6.4.4(5) gerR_IN_FinishReceivedCyclicDataAcquisition</li> <li>6.4.4(6) gerR_IN_GetReceivedCyclicRY</li> <li>6.4.4(7) gerR_IN_GetReceivedCyclicRWw</li> <li>6.4.4(8) gerR_IN_GetReceivedCyclicRspdu</li> </ul>			

【注】※1 コンパイルスイッチ「SAFETY\_PDU\_ENABLE」を有効にした場合

## (15) gerR\_IN\_FinishReceivedCyclicBuffer

機能	サイクリック受信バッファ取得完了			
呼出し形式	ERRCODE gerR_IN_FinishReceivedCyclicBuffer (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了(受信データあり) R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(受信データなし)			
説明	<p>マスタ局から受信したサイクリック受信データの取得が完了したことを R-IN32M4-CL3 へ通知します。</p> <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>6.4.4(3) gerR_IN_GetReceivedCyclicData</li> <li>6.4.4(4) gerR_IN_UpdateReceivedCyclicData</li> <li>6.4.4(5) gerR_IN_FinishReceivedCyclicDataAcquisition</li> <li>6.4.4(6) gerR_IN_GetReceivedCyclicRY</li> <li>6.4.4(7) gerR_IN_GetReceivedCyclicRWw</li> <li>6.4.4(8) gerR_IN_GetReceivedCyclicRspdu</li> </ul>			

## (16) gerR\_IN\_GetSendCyclicBuffer

機能	サイクリック送信バッファ取得			
呼出し形式	一般通信時	ERRCODE gerR_IN_GetSendCyclicBuffer (ULONG* pulRxBuffer, ULONG* pulRWrBuffer)		
	安全通信時	ERRCODE gerR_IN_GetSendCyclicBuffer(ULONG* pulRxBuffer, ULONG* pulRWrBuffer, ULONG* pulSspduBuffer)		
引数	型名	変数名	内容	入出力
	ULONG*	pulRxBuffer	RX 格納先アドレス	出力
	ULONG*	pulRWrBuffer	RWr 格納先アドレス	出力
	ULONG*	pulSspduBuffer	SSPDU 格納先アドレス※1	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(CCIETSN ClassA 動作時に送信データ設定不要)			
説明	<p>マスタ局へ送信するサイクリック送信データの格納先アドレスを取得します。</p> <p>サイクリック送信データ設定完了後は、「6.4.4(18) gerR_IN_FinishSendCyclicBuffer」を実行してください。</p> <p>戻り値が R_IN_OK の場合のみ、引数に RX、RWr、および SSPDU の格納先アドレスが設定されます。</p> <p>CC-Link IE TSN Class A 動作時に、サイクリックデータを受信していない場合、戻り値で「R_IN_ERR_NODATA」を返します。</p> <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>6.4.4(9) gerR_IN_SetSendCyclicData</li> <li>6.4.4(10) gerR_IN_FinishSendCyclicDataSetting</li> <li>6.4.4(11) gerR_IN_SetSendCyclicRX</li> <li>6.4.4(12) gerR_IN_SetSendCyclicRWr</li> <li>6.4.4(13) gerR_IN_SetSendCyclicSspdu</li> </ul> <p>サイクリック送信データが大きく、サイクリック送信フレームが2フレーム必要な場合は「6.4.4(17) gerR_IN_GetSplitSendCyclicBuffer」を使用してください。</p>			

【注】 ※1 コンパイルスイッチ「SAFETY\_PDU\_ENABLE」を有効にした場合

## (17) gerR\_IN\_GetSplitSendCyclicBuffer

機能	サイクリック分割送信バッファ取得			
呼出し形式	ERRCODE gerR_IN_GetSplitSendCyclicBuffer (R_IN_SEND_CYCLIC_BUFFER_T* pstBuffer)			
引数	型名	変数名	内容	入出力
	R_IN_SEND_CYCLIC_BUFFER_T*	pstBuffer	サイクリック送信バッファ 詳細は「表 6.21」を参照してください。	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(CCIETSN ClassA 動作時に送信データ設定不要)			
説明	<p>マスタ局へ送信するサイクリック送信データの格納先アドレスを取得します。 サイクリック送信データ設定完了後は「gerR_IN_FinishSendCyclicBuffer」(6.4.4(18))を実行してください。</p> <p>RX/RWr のサイズを大きくすることでサイクリック送信フレームが2 フレーム構成となる場合に RX/RWr のエリアが分割されます。本関数では、分割されたエリアの先頭アドレスとサイズを取得 することができます。</p> <p>戻り値が R_IN_OK の場合のみ、引数に RX および RWr の格納先アドレスが設定されます。 CC-Link IE TSN Class A 動作時に、サイクリックデータを受信していない場合、戻り値で 「R_IN_ERR_NODATA」を返します。</p> <p>本関数を使用する場合は、以下を使用しないでください。</p> <ul style="list-style-type: none"> <li>6.4.4(9) gerR_IN_SetSendCyclicData</li> <li>6.4.4(10) gerR_IN_FinishSendCyclicDataSeting</li> <li>6.4.4(11) gerR_IN_SetSendCyclicRX</li> <li>6.4.4(12) gerR_IN_SetSendCyclicRWr</li> <li>6.4.4(13) gerR_IN_SetSendCyclicSspdu</li> </ul>			

表 6.21 R\_IN\_SEND\_CYCLIC\_BUFFER\_T 一覧

No	メンバ	内容	
1	stRx	-	
2	[R_IN_SEND_CYCLIC_BUFFER_RX_NUM]	ULONG	ulAddr
3		USHORT	usSize
4	stRWr	-	
5	[R_IN_SEND_CYCLIC_BUFFER_RWR_NUM]	ULONG	ulAddr
6		USHORT	usSize
7	stSspdu	-	
8	[R_IN_SEND_CYCLIC_BUFFER_SSPDU_NUM]	ULONG	ulAddr
9		USHORT	usSize
10	USHORT	usRxNum	RX バッファ数
11	USHORT	usRWrNum	RWr バッファ数
12	USHORT	usSspduNum	SSPDU バッファ数※1

【注】※1 安全 PDU 送受信(コンパイルスイッチ「SAFETY\_PDU\_ENABLE」を有効にした場合)のみ

表 6.22 サイクリック送信バッファ数のマクロ定義

定数名	CANopen 通信以外 のときの値	CANopen 通信時の値
R_IN_SEND_CYCLIC_BUFFER_RX_NUM	2	2
R_IN_SEND_CYCLIC_BUFFER_RWR_NUM	2	R_IN_CAN_MAX_ODTABLE_NUM <sup>※1</sup> + 1
R_IN_SEND_CYCLIC_BUFFER_SSPDU_NUM	2	定義なし

【注】※1 最大 ObjectDictionary 数

## (18) gerR\_IN\_FinishSendCyclicBuffer

機能	サイクリック送信バッファ設定完了			
呼出し形式	ERRCODE gerR_IN_FinishSendCyclicBuffer (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(CCIETSN ClassA 動作時に送信データ設定不要)			
説明	サイクリック送信バッファの設定が完了したことを R-IN32M4-CL3 へ通知し、マスタ局へ送信します。 CC-Link IE TSN Class A 動作時に、サイクリックデータを受信していない場合、戻り値で「R_IN_ERR_NODATA」を返します。  本関数を使用する場合は、以下を使用しないでください。 <ul style="list-style-type: none"> <li>6.4.4(9) gerR_IN_SetSendCyclicData</li> <li>6.4.4(10) gerR_IN_FinishSendCyclicDataSeting</li> <li>6.4.4(11) gerR_IN_SetSendCyclicRX</li> <li>6.4.4(12) gerR_IN_SetSendCyclicRWr</li> <li>6.4.4(13) gerR_IN_SetSendCyclicSspdu</li> </ul>			

## (19) gvR\_IN\_SendCyclicFrameClassA

機能	サイクリックフレーム送信(CC-Link IE TSN Class A)			
呼出し形式	VOID gvR_IN_SendCyclicFrameClassA(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	CC-Link IE TSN Class A 動作時に、設定したサイクリック送信データをマスタ局へ送信します。 本関数を実行することで、実際にサイクリックフレームが送信されます。 CC-Link IE TSN Class B 動作時は、本関数を実行する必要はありません。			



## 6.4.5 自局状態設定

### (1) gerR\_IN\_SetNodeStatus

機能	自局状態設定			
呼出し形式	ERRCODE gerR_IN_SetNodeStatus (ULONG ulErrSts)			
引数	型名	変数名	内容	入出力
	ULONG	ulErrSts	アプリケーションエラー状態 R_IN_ERRSTS_NONE(0) : エラーなし R_IN_ERRSTS_WARNING(1) : 軽度異常 R_IN_ERRSTS_ERROR(2) : 中度異常 R_IN_ERRSTS_FATALERROR(3) : 重度異常	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	サイクリック伝送で送信する情報として、自局状態情報を設定します。 WDT エラーが発生した場合、R-IN32M4-CL3 ドライバによって「重度異常」が設定されます。			

## 6.4.6 自局状態取得

## (1) gerR\_IN\_GetIPAddress

機能	IP アドレス取得			
呼出し形式	ERRCODE gerR_IN_GetIPAddress (ULONG *pulIPAddress, ULONG *pulSubnetmask, ULONG *pulDefaultGateway)			
引数	型名	変数名	内容	入出力
	ULONG*	pulIPAddress	IP アドレス	出力
	ULONG*	pulSubnetmask	サブネットマスク	出力
	ULONG*	pulDefaultGateway	デフォルトゲートウェイ	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	<p>IP アドレス、サブネットマスク、デフォルトゲートウェイの設定値を取得します。</p> <p>IP アドレスが「192.168.3.10」の場合、pulIPAddress、pulDefaultGateway への格納値は「0xC0A8030AH」になります。</p> <p>サブネットマスクが「255.255.255.0」の場合、pulSubnetmask への格納値は「0xFFFFFFFF00H」になります。</p> <p>デフォルトゲートウェイの pulDefaultGateway には、常に「0x00000000H」が格納されます。</p> <p>本関数は、「gerR_IN_Initialize」(6.4.1(2))と「gerR_IN_Start(6.4.1(5))」が完了後に使用してください。</p> <p>完了前に本関数を使用すると IP アドレスが設定されていないため、初期値 0 を取得します。</p>			

## (2) gerR\_IN\_GetCurrentCyclicSize

機能	マスタ局が指定したサイクリック伝送サイズ取得			
呼出し形式	ERRCODE gerR_IN_GetCurrentCyclicSize (R_IN_CYCLIC_SIZE_T *pstCyclicSize)			
引数	型名	変数名	内容	入出力
	R_IN_CYCLIC_SIZE_T*	pstCyclicSize	サイクリック伝送サイズ 詳細は「表 6.23」を参照してください。	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	<p>マスタ局が設定したサイクリック伝送サイズを取得します。</p> <p>「gerR_IN_GetReceivedCyclicData」(6.4.4(3))、および「gerR_IN_SetSendCyclicData」(6.4.4(9))は、本関数で取得するサイズにてサイクリック送受信データの入出力を行います。</p>			

表 6.23 R\_IN\_CYCLIC\_SIZE\_T 一覧

No.	メンバ		内容
1	ULONG	ulRySize	RY サイズ(バイト(オクテット))
2	ULONG	ulRWwSize	RWw サイズ(バイト(オクテット))
3	ULONG	ulRxSize	RX サイズ(バイト(オクテット))
4	ULONG	ulRWrSize	RWr サイズ(バイト(オクテット))
5	ULONG	ulRspduSize	RSPDU サイズ(バイト) <sup>※1</sup>
6	ULONG	ulSspduSize	SSPDU サイズ(バイト) <sup>※1</sup>

【注】※1 安全 PDU 送受信(コンパイルスイッチ「SAFETY\_PDU\_ENABLE」を有効にした場合)のみ

## (3) gerR\_IN\_GetCyclicStatus

機能	サイクリック伝送状態取得			
呼出し形式	ERRCODE gerR_IN_GetCyclicStatus (UCHAR* puchCyclicStatus)			
引数	型名	変数名	内容	入出力
	UCHAR*	puchCyclicStatus	サイクリック伝送状態 00H : 正常交信または電源投入時 02H : 監視時間タイムアップ※1 12H : 自局予約局設定 13H : 自局局番重複	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	サイクリック伝送状態を取得します。			

【注】※1 データリンク開始後、サイクリックデータ未受信が一定回数続き解列した状態です。

マスタ局に RJ71GN11-T2 を使用する場合、GX Works3 のパラメータ設定「解列検出設定」にて、デバイス局を解列とみなすまでの連続通信失敗回数を指定できます。

## (4) gerR\_IN\_GetCommumicationStatus

機能	データリンク状態取得			
呼出し形式	ERRCODE gerR_IN_GetCommumicationStatus (ULONG *pulCommSts)			
引数	型名	変数名	内容	入出力
	ULONG*	pulCommSts	データリンク状態 R_IN_COMMSTS_CYC_DLINK(2) : データリンク中(サイクリック伝送中) R_IN_COMMSTS_CYC_STOP(1) : データリンク中(サイクリック伝送停止中) R_IN_COMMSTS_DISCONNECT(0) : データリンク未実施(解列中)	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	データリンク状態を取得します。データリンク状態により、D LINK LED を点灯/消灯してください。 R_IN_COMMSTS_CYC_DLINK : LED 点灯 R_IN_COMMSTS_CYC_STOP : LED 点滅 R_IN_COMMSTS_DISCONNECT : LED 消灯			

## (5) gerR\_IN\_GetPortStatus

機能	PHY リンク状態取得			
呼出し形式	ERRCODE gerR_IN_GetPortStatus (ULONG ulPort, ULONG *pulLinkStatus, ULONG *pulSpeed, ULONG *pulDuplex)			
引数	型名	変数名	内容	入出力
	ULONG	ulPort	ポート指定 R_IN_PORT1(0) : Port1 R_IN_PORT2(1) : Port2	入力
	ULONG*	pulLinkStatus	リンク状態 R_IN_LINKUP(1) : LinkUp R_IN_LINKDOWN(0) : LinkDown	出力
	ULONG*	pulSpeed	速度 R_IN_SPEED_1G(0) : 1Gbps R_IN_SPEED_100M(1) : 100Mbps R_IN_SPEED_10M(2) : 10Mbps (*pulLinkState が LinkDown のときは、don't care)	出力
	ULONG*	pulDuplex	全二重／半二重 R_IN_DUPLEX_FULL(0) : 全二重 R_IN_DUPLEX_HALF(1) : 半二重 (*pulLinkState が LinkDown のときは、don't care)	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外) R_IN_ERR_SEMAPHORE : 異常完了(セマフォ獲得失敗) R_IN_ERR_FATAL : 異常完了(Fatal エラー発生)			
説明	PHY リンクステータスを取得します。 「gerR_IN_Start」(6.4.1(5))が完了した後に実行してください。			

## (6) gerR\_IN\_GetStatisticalInformation

機能	統計情報取得			
呼出し形式	ERRCODE gerR_IN_GetStatisticalInformation (R_IN_STATISTICS_T *pstStatisticalInformation)			
引数	型名	変数名	内容	入出力
	R_IN_STATISTICS_T*	pstStatisticalInformation	統計情報 詳細は「表 6.24」を参照してください。	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了(パラメータエラー)			
説明	R-IN32M4-CL3 ドライバで保持している MIB 内の統計情報を取得します。 引数が NULL ポインタの場合、R_IN_ERR となります。  統計情報はコンパイルスイッチ「ACCUMULATE_STATISTICS_INFORMATION」を定義することで、累積の格納数を保持することが可能となり、格納数が上限に達した場合は上限値が保持されます。 統計情報の更新処理は更新用のタスクにて実行するため、当該タスクより優先度の低いタスクで本関数を実行するとデータの泣き別れが発生します。 泣き別れを防止するには、更新用のタスクと同じ優先度のタスクで、本関数を実行する必要があります。			

表 6.24 R\_IN\_STATISTICS\_T 一覧

No.	メンバ		内容
1	USHORT	usCyclicReceiveCounter	サイクリック受信カウンタ
2	USHORT	usCyclicReceiveDiscardCounter	サイクリック受信廃棄カウンタ
3	USHORT	usCyclicFrameReceiveCounter	サイクリックフレーム受信カウンタ
4	USHORT	usNonCyclicReceiveCounter	非サイクリック受信カウンタ
5	USHORT	usNonCyclicReceiveDiscardCounter	非サイクリック受信破棄カウンタ
6	USHORT	usNumberOfHecErrorFrame	HEC エラーフレーム数
7	USHORT	usNumberOfDcsErrorFrame	DCS エラーフレーム数
8	USHORT	usNumberOfFcsErrorFrame	FCS エラーフレーム数
9	USHORT	usNumberOfSdcrcErrorFrame	SDCRC エラーフレーム数
10	USHORT	usNumberOfShortPacketFrame	ショートパケットフレーム検出数
11	USHORT	usNumberOfJumboFrame	ジャンボフレーム検出数
12	USHORT	usNumberOfLongPacketFrame	ロングパケット検出数
13	USHORT	usNumberOfFailedCcLinkIePduSize	CC-Link IE TSN PDU 長異常数
14	USHORT	usNumberOfFlagmentErrorFrame	フラグメントエラーフレーム数
15	USHORT	usNumberOfPriorityControlFrame	優先制御フレーム数
16	USHORT	usNumberOfIppFrame	IP フレーム数
17	USHORT	usNumberOfIeee802or1588Frame	IEEE802.1AS/IEEE1588 フレーム数
18	USHORT	usNumberOfLldpFrame	LLDP フレーム数
19	USHORT	usNumberOfSyncFrame	Sync フレーム数

## (7) gerR\_IN\_ClearStatisticalInformation

機能	統計情報クリア			
呼出し形式	ERRCODE gerR_IN_ClearStatisticalInformation (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR_BUSY : クリア要求受付済みまたは処理中			
説明	R-IN32M4-CL3 ドライバが保有する MIB 内の統計情報のクリア指示を行います。 次回統計情報更新時に統計情報がクリアされます。			

## (8) guIR\_IN\_GetNetworkTopology

機能	ネットワークトポロジ情報取得			
呼出し形式	ULONG guIR_IN_GetNetworkTopology(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_NETWORK_TOPOLOGY_UNKNOWN (0) : ネットワークトポロジ未確定 R_IN_NETWORK_TOPOLOGY_OTHER (1) : その他のネットワークトポロジ R_IN_NETWORK_TOPOLOGY_RING (2) : リング接続			
説明	マスタ局から設定された検出トポロジ(単ーリング)の情報を取得します。			

## (9) gerR\_IN\_GetNodeOperationMode

機能	自局動作モード取得			
呼出し形式	ERRCODE gerR_IN_GetNodeOperationMode (R_IN_NODE_OPERATION_MODE_T* pstMode)			
引数	型名	変数名	内容	入出力
	R_IN_NODE_OPERATION_MODE_T*	pstMode	自局動作モード R_IN_ASYNCHRONOUS (0) : 同期しない R_IN_SYNCHRONOUS (1) : 同期する	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	自局(R-IN32M4-CL3)の動作モードを取得します。			

## (10) gerR\_IN\_GetLinkSpeed

機能	通信速度取得			
呼出し形式	ERRCODE gerR_IN_GetLinkSpeed(ULONG* pulLinkSpeed)			
引数	型名	変数名	内容	入出力
	ULONG*	pulLinkSpeed	通信速度 0 : 1Gbps 1 : 100Mbps	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	通信速度の設定値を取得します。 本関数は「gerR_IN_Initialize」(6.4.1(2))完了後に使用してください。			

## (11) gerR\_IN\_GetIPAddressDuplication

機能	IP アドレス重複状態取得			
呼出し形式	ERRCODE gerR_IN_GetIPAddressDuplication(BOOL* pblDuplication)			
引数	型名	変数名	内容	入出力
	BOOL*	pblDuplication	IP アドレス重複 R_IN_FALSE : IP アドレス重複未検出 R_IN_TRUE : IP アドレス重複検出	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	IP アドレス重複状態を取得します。 本関数は「gerR_IN_Initialize」(6.4.1(2))完了後に使用してください。			

## 6.4.7 LED 制御

## (1) gerR\_IN\_SetUSER1LED

機能	LED 点灯制御(USER LED1)			
呼出し形式	ERRCODE gerR_IN_SetUSER1LED (ULONG ulCtrl)			
引数	型名	変数名	内容	入出力
	ULONG	ulCtrl	LED 点灯制御パラメータ 詳細は「表 6.25」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	USER LED1 の点灯制御を行います。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			

## (2) gerR\_IN\_SetUSER2LED

機能	LED 点灯制御(USER LED2)			
呼出し形式	ERRCODE gerR_IN_SetUSER2LED (ULONG ulCtrl)			
引数	型名	変数名	内容	入出力
	ULONG	ulCtrl	LED 点灯制御パラメータ 詳細は「表 6.25」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	USER LED2 の点灯制御を行います。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			

## (3) gerR\_IN\_SetRUNLED

機能	LED 点灯制御(RUN)			
呼出し形式	ERRCODE gerR_IN_SetRUNLED (ULONG ulCtrl)			
引数	型名	変数名	内容	入出力
	ULONG	ulCtrl	LED 点灯制御パラメータ 詳細は「表 6.25」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	RUN LED の点灯制御を行います。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			

## (4) gerR\_IN\_SetERRLED

機能	LED 点灯制御(ERR)			
呼出し形式	ERRCODE gerR_IN_SetERRLED (ULONG ulCtrl)			
引数	型名	変数名	内容	入出力
	ULONG	ulCtrl	LED 点灯制御パラメータ 詳細は「表 6.25」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	ERR LED の点灯制御を行います。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			

表 6.25 LED 点灯制御パラメータ一覧

名称	値	内容
R_IN_LED_OFF	0	LED 消灯
R_IN_LED_ON	1	LED 点灯
R_IN_LED_BLINK_1000	2	LED 点滅(点滅周期 : 1 秒)
R_IN_LED_BLINK_500	3	LED 点滅(点滅周期 : 500m 秒)
R_IN_LED_BLINK_200	4	LED 点滅(点滅周期 : 200m 秒)

## (5) gerR\_IN\_SetLERR1LED

機能	LED 点灯制御(L ER1 LED)			
呼出し形式	ERRCODE gerR_IN_SetLERR1LED (ULONG ulCtrl)			
引数	型名	変数名	内容	入出力
	ULONG	ulCtrl	R_IN_LED_OFF (0) : LED 消灯 R_IN_LED_ON (1) : LED 点灯	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	L ER1 LED の点灯制御を行います。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			

## (6) gerR\_IN\_SetLERR2LE

機能	LED 点灯制御(L ER2 LED)			
呼出し形式	ERRCODE gerR_IN_SetLERR2LED (ULONG ulCtrl)			
引数	型名	変数名	内容	入出力
	ULONG	ulCtrl	R_IN_LED_OFF (0) : LED 消灯 R_IN_LED_ON (1) : LED 点灯	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	L ER2 LED の点灯制御を行います。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			



## (7) gerR\_IN\_DisableLED

機能	LED 点灯機能無効			
呼出し形式	ERRCODE gerR_IN_DisableLED (USHORT usDisable)			
引数	型名	変数名	内容	入出力
	USHORT	usDisable	点灯機能無効 LED 指定 (ON : 無効設定、OFF : 前値保持) Bit 0 : RUN LED 無効 Bit 1 : USER LED2 無効 Bit 3 : ERR LED 無効 Bit 5 : USER LED1 無効 Bit 6 : DLINK LED 無効 Bit 9 : L ER 1 LED 無効 Bit10 : L ER 2 LED 無効 (上記以外 : 未使用)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	LED 点灯機能を無効にします。 引数で規定している未使用 bit に ON : 無効設定を設定すると R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)となります。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			

## (8) gerR\_IN\_EnableLED

機能	LED 点灯機能有効			
呼出し形式	ERRCODE gerR_IN_EnableLED (USHORT usEnable)			
引数	型名	変数名	内容	入出力
	USHORT	usEnable	点灯機能有効 LED 指定 (ON : 有効設定、OFF : 前値保持) Bit 0 : RUN LED 有効 Bit 1 : USER LED2 有効 Bit 3 : ERR LED 有効 Bit 5 : USER LED1 有効 Bit 6 : DLINK LED 有効 Bit 9 : L ER 1 LED 有効 Bit10 : L ER 2 LED 有効 (上記以外 : 未使用)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	LED の点灯機能を有効にします。 引数で規定している未使用 bit に ON : 有効設定を設定すると R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)となります。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			

## (9) gerR\_IN\_UpdateLedStatus

機能	通信状態表示 LED の更新			
呼出し形式	ERRCODE gerR_IN_UpdateLedStatus (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了			
説明	データリンク状態に応じて、RUN、ERR、および D LINK LED の点灯/消灯を行います。 詳細は「表 6.26 D LINK LED の制御」を参照してください。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			

表 6.26 D LINK LED の制御

データリンク状態	D LINK LED
データリンク中(サイクリック伝送中)	点灯
データリンク中(サイクリック伝送停止中)	点滅(500ms 周期)
データリンク未実施(解列中)	消灯

## (10) gerR\_IN\_SetSDRDLEDMODE

機能	SD/RD 点灯モード設定			
呼出し形式	ERRCODE gerR_IN_SetSDRDLEDMODE (ULONG ulMode)			
引数	型名	変数名	内容	入出力
	ULONG	ulMode	SD/RD 点灯モード指定 R_IN_LEDMODE_SDRD1_SDRD2 点灯モード : SDRD1/SDRD2 R_IN_LEDMODE_SD_RD 点灯モード : SD/RD	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	SD/RD の点灯モードを設定します。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			

## (11) gerR\_IN\_StartTestLED

機能	LED テスト開始			
呼出し形式	ERRCODE gerR_IN_StartTestLED (USHORT usTestMode)			
引数	型名	変数名	内容	入出力
	const USHORT	usTestMode	テスト機能有効/無効指定 (ON : 有効設定、OFF : 無効指定) Bit 0 : RUN LED Bit 1 : USER LED 2 Bit 2 : USER LED 1 Bit 3 : D LINK LED Bit 4 : ERR LED Bit 5 : SD/SDRD1 LED Bit 6 : RD/SDRD2 LED Bit 7 : L ER 1 LED Bit 8 : L ER 2 LED Bit 14 : LINK1 LED Bit 15 : LINK2 LED (上記以外 : 未使用)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外) R_IN_ERR_SEMAPHORE : 異常完了(セマフォ獲得失敗) R_IN_ERR_FATAL : 異常完了(Fatal エラー発生)			
説明	LED テストの有効/無効を指定して、機能有効の LED を初期状態(消灯)に設定します。 LED テスト開始時に実行してください。(LED テスト開始～実行～終了の間で本関数の呼び出しは開始時の一度のみとしてください) 関数の戻り値が R_IN_ERR_FATAL の場合、R-IN32M4-CL3 に Fatal エラーが発生しており、ユーザ作成の「gR_IN_CallbackFatalError」(6.6(1))を呼び出しますので R-IN32M4-CL3 の Fatal エラーを取得してください。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			

## (12) gerR\_IN\_ExecuteTestLED

機能	LED テスト実行			
呼出し形式	ERRCODE gerR_IN_ExecuteTestLED (USHORT usTestLed)			
引数	型名	変数名	内容	入出力
	USHORT	usTestLed	強制点消灯指定 (ON : 強制点灯/OFF : 強制消灯) Bit 0 : RUN LED Bit 1 : USER LED 2 Bit 2 : USER LED 1 Bit 3 : D LINK LED Bit 4 : ERR LED Bit 5 : SD/SDRD1 LED Bit 6 : RD/SDRD2 LED Bit 7 : L ER 1 LED Bit 8 : L ER 2 LED Bit 14 : LINK1 LED Bit 15 : LINK2 LED (上記以外 : 未使用)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外) R_IN_ERR_SEMAPHORE : 異常完了(セマフォ獲得失敗) R_IN_ERR_FATAL : 異常完了(Fatal エラー発生)			
説明	LED の強制点灯/消灯を行います。 LED テスト開始～LED テスト終了の間で使用してください。 本関数の戻り値が R_IN_ERR_FATAL の場合、R-IN32M4-CL3 に Fatal エラーが発生しており、ユーザ作成の「gR_IN_CallbackFatalError」(6.6(1))を呼び出しますので R-IN32M4-CL3 の Fatal エラーを取得してください。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			

## (13) gerR\_IN\_StopTestLED

機能	LED テスト終了			
呼出し形式	ERRCODE gerR_IN_StopTestLED (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR_SEMAPHORE : 異常完了(セマフォ獲得失敗) R_IN_ERR_FATAL : 異常完了(Fatal エラー発生)			
説明	LED テスト終了時に実行してください。(LED テスト開始～実行～終了の間で本関数の呼び出しは終了時の一度のみとしてください) 本関数の戻り値が R_IN_ERR_FATAL の場合、R-IN32M4-CL3 に Fatal エラーが発生しており、ユーザ作成の「gR_IN_CallbackFatalError」(6.6(1))を呼び出しますので R-IN32M4-CL3 の Fatal エラーを取得してください。 本関数内でサービスコールのディスパッチ禁止/許可処理を呼び出しているため、他タスクと排他的に処理が行われます。			

## 6.4.8 ネットワーク時刻

## (1) gerR\_IN\_GetNetworkTime

機能	ネットワーク時刻(シリアル値)取得			
呼出し形式	ERRCODE gerR_IN_GetNetworkTime (USHORT* pusSerial)			
引数	型名	変数名	内容	入出力
	USHORT*	pusSerial	ネットワーク時刻(シリアル値) pusSerial[0] : ネットワーク時刻(bit15-0) pusSerial[1] : ネットワーク時刻(bit31-16) pusSerial[2] : ネットワーク時刻(bit47-32)	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	ネットワーク時刻(2000 年 1 月 1 日 0 時 0 分 0 秒を基点とした、15.2587890625 マイクロ秒単位のシリアル値)を取得します。ネットワーク時刻が 2000 年未満である場合、ネットワーク時刻(シリアル値)有効範囲外のエラーを返します。 なお、取得した時刻は UTC 時間であることを留意してください。			

## (2) gerR\_IN\_NetworkTimeToDate

機能	ネットワーク時刻(シリアル値)→時計情報変換			
呼出し形式	ERRCODE gerR_IN_NetworkTimeToDate (R_IN_TIMEINFO_T* pstTimeInfo, const USHORT* pusSerial)			
引数	型名	変数名	内容	入出力
	R_IN_TIMEINFO_T*	pstTimeInfo	時計情報 詳細は「表 6.27」を参照してください。	出力
	const USHORT *	pusSerial	ネットワーク時刻(シリアル値) pusSerial[0] : ネットワーク時刻(bit15-0) pusSerial[1] : ネットワーク時刻(bit31-16) pusSerial[2] : ネットワーク時刻(bit47-32)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	ネットワーク時刻(2000 年 1 月 1 日 0 時 0 分 0 秒を基点とした、15.2587890625 マイクロ秒単位のシリアル値)を、時計情報(西暦・月・日・時・分・秒・ミリ秒・曜日)に変換します。 時計情報が持つミリ秒には、ネットワーク時刻(bit15-0)の値を 0.0152587890625 倍した値(小数点以下切捨て)を設定します。 また、ネットワーク時刻(シリアル値)に 2106 年以降の値を設定された場合は、ネットワーク時刻(シリアル値)有効範囲外のエラーとなります。			

表 6.27 R\_IN\_TIMEINFO\_T 一覧

No.	メンバ		内容
1	USHORT	usYear	西暦 (1970 - 2105)
2	USHORT	usMonth	月 (1-12)
3	USHORT	usDay	日 (1-31)
4	USHORT	usHour	時 (0-23)
5	USHORT	usMin	分 (0-59)
6	USHORT	usSec	秒 (0-59)
7	USHORT	usMsec	ミリ秒 (0-999)
8	USHORT	usWday	曜日 (0 (日) - 6 (土))

## (3) gerR\_IN\_DateToNetworkTime

機能	時計情報→ネットワーク時刻(シリアル値)変換			
呼出し形式	ERRCODE gerR_IN_DateToNetworkTime (const R_IN_TIMEINFO_T* pstTimeInfo, USHORT* pusSerial)			
引数	型名	変数名	内容	入出力
	const R_IN_TIMEINFO_T *	pstTimeInfo	時計情報 詳細は「表 6.27」を参照してください。	入力
	USHORT *	pusSerial	ネットワーク時刻(シリアル値)	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	時計情報(西暦・月・日・時・分・秒・ミリ秒)をネットワーク時刻(2000 年 1 月 1 日 0 時 0 分 0 秒を基点とした、15.2587890625 マイクロ秒単位のシリアル値)に変換します。 ネットワーク時刻(bit15-0)には、時計情報が持つミリ秒の値を 0.0152587890625 で割った値(小数点以下切捨て)を設定します。 ただし、時計情報が 2000 年～2105 年以外の場合は、時計情報有効範囲外のエラーとなります。 R-IN32M4-CL3 ドライバは、上記以外のエラーチェックを行っていませんので、うるう年など年月日に誤りがないようにユーザプログラムにエラー処理を実装してください。			

## (4) gerR\_IN\_GetUnixTime

機能	ネットワーク時刻(UNIX 時間)取得			
呼出し形式	ERRCODE gerR_IN_GetUnixTime (R_IN_UNIX_TIME_T* pstUnixTime)			
引数	型名	変数名	内容	入出力
	R_IN_UNIX_TIME_T*	pstUnixTime	ネットワーク時刻(UNIX 時間) 詳細は「表 6.28」を参照してください。	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	ネットワーク時刻(UNIX 時間(秒、ナノ秒で構成))を取得します。 なお、取得した時刻は UTC 時間であることに留意してください。			

表 6.28 R\_IN\_UNIX\_TIME\_T 一覧

No.	メンバ		内容
1	ULONG	ulNanoSecond	UNIX 時間(ナノ秒)
2	ULONG	ulSecond	UNIX 時間(秒)

## (5) gerR\_IN\_UnixTimeToDate

機能	ネットワーク時刻(UNIX 時間)→時計情報変換			
呼出し形式	ERRCODE gerR_IN_UnixTimeToDate (R_IN_TIMEINFO_T* pstTimeInfo, const R_IN_UNIX_TIME_T* pstUnixTime)			
引数	型名	変数名	内容	入出力
	R_IN_TIMEINFO_T *	pstTimeInfo	時計情報 詳細は「表 6.27」を参照してください。	出力
	const R_IN_UNIX_TIME_T*	pstUnixTime	ネットワーク時刻(UNIX 時間) 詳細は「表 6.28」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	<p>ネットワーク時刻(UNIX 時間(秒、ナノ秒で構成))を時計情報(西暦・月・日・時・分・秒・ミリ秒・曜日)に変換します。</p> <p>UNIX 時間(秒)に 2106 年以降となる値を設定した場合、または UNIX 時間(ナノ秒)に 999,999,992ns より大きい値を設定した場合は、UNIX 時間有効範囲外のエラーとなります。また、時計情報へ正常に変換できるのは、UNIX 時間に 1970 年 1 月 1 日 0 時 0 分 1 秒 0 ミリ秒以降の値を設定したときのみになります。</p> <p>時計情報が持つミリ秒には、ネットワーク時刻(UNIX 時間)が持つナノ秒の値を 1,000,000 で割った値(小数点以下切捨て)を設定します。</p>			

## (6) gerR\_IN\_DateToUnixTime

機能	時計情報→ネットワーク時刻(UNIX 時間)変換			
呼出し形式	ERRCODE gerR_IN_DateToUnixTime (const R_IN_TIMEINFO_T* pstTimeInfo, R_IN_UNIX_TIME_T* pstUnixTime)			
引数	型名	変数名	内容	入出力
	const R_IN_TIMEINFO_T *	pstTimeInfo	時計情報 詳細は「表 6.27」を参照してください。	入力
	R_IN_UNIX_TIME_T *	pstUnixTime	ネットワーク時刻(UNIX 時間) 詳細は「表 6.28」を参照してください。	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	<p>時計情報(西暦・月・日・時・分・秒・ミリ秒)をネットワーク時刻(UNIX 時間(秒、ナノ秒で構成))に変換します。</p> <p>ただし、時計情報が持つ西暦が 1970 年～2105 年以外の場合は、時計情報有効範囲外のエラーとなります。</p> <p>R-IN32M4-CL3 ドライバは、上記以外エラーチェックを行っていませんので、うるう年など年月日に誤りがないようにユーザプログラムにエラー処理を実装してください。</p>			

## (7) gerR\_IN\_SetUnixTimeClassA

機能	ネットワーク時刻(UNIX 時間)設定(CC-Link IE TSN Class A)			
呼出し形式	ERRCODE gerR_IN_SetUnixTimeClassA(R_IN_UNIX_TIME_T* pstUnixTime)			
引数	型名	変数名	内容	入出力
	R_IN_UNIX_TIME_T *	pstUnixTime	ネットワーク時刻(UNIX 時間) 詳細は「表 6.28」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	CC-Link IE TSN Class A の場合に、UNIX 時間としてネットワーク時刻を設定します。 CC-Link IE TSN Class B の場合は使用できません。			

## (8) gvR\_IN\_SetUnixOffsetTime

機能	ネットワーク時刻オフセット設定			
呼出し形式	VOID gvR_IN_SetUnixOffsetTime (LONGLONG lOffsetSec, LONG lOffsetNsec, SHORT sUtcOffsetMin, SHORT sSummerTimeOffsetMin)			
引数	型名	変数名	内容	入出力
	LONGLONG	lOffsetSec	オフセット(秒)	入力
	LONG	lOffsetNsec	オフセット(ナノ秒)	入力
	SHORT	sUtcOffsetMin	UTC オフセット(分)	入力
	SHORT	sSummerTimeOffsetMin	サマータイムオフセット(分)	入力
戻り値	-			
説明	ネットワーク時刻(UNIX 時刻)の補正値を設定します。			

## (9) gerR\_IN\_GetUnixOffsetTime

機能	ネットワーク時刻オフセット取得			
呼出し形式	ERRCODE gerR_IN_GetUnixOffsetTime (LONGLONG* plOffsetSec, LONG* plOffsetNsec, SHORT* psUtcOffsetMin, SHORT* psSummerTimeOffsetMin)			
引数	型名	変数名	内容	入出力
	LONGLONG*	plOffsetSec	オフセット(秒)	入力
	LONG*	plOffsetNsec	オフセット(ナノ秒)	入力
	SHORT*	psUtcOffsetMin	UTC オフセット(分)	入力
	SHORT*	psSummerTimeOffsetMin	サマータイムオフセット(分)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(データなし)			
説明	ネットワーク時刻(UNIX 時刻)の補正値を取得します。			



ネットワーク時刻(UNIX 時間)に対し、オフセットで補正する例を示します。

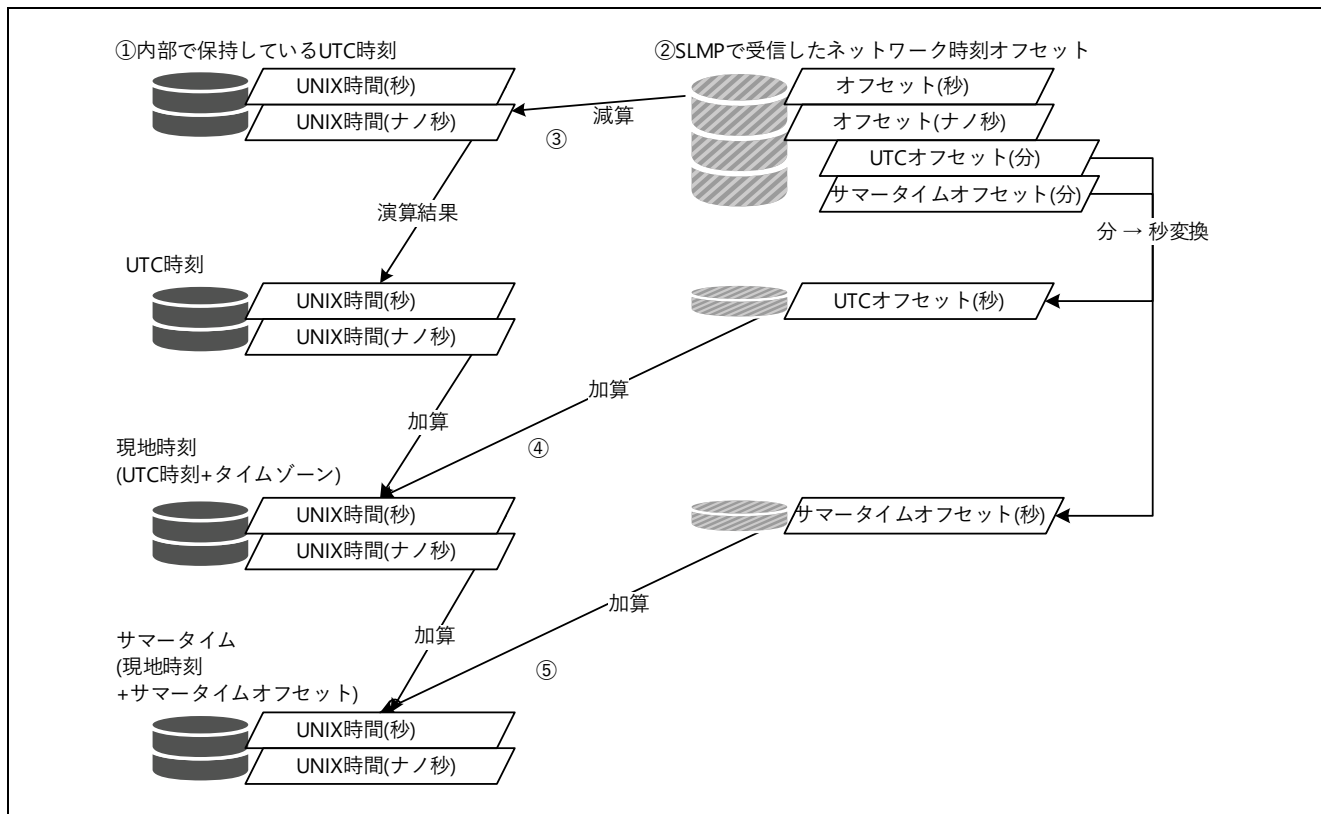


図 6.2 オフセットを使用した時刻補正例

- ① 「gerR\_IN\_GetUnixTime」(ネットワーク時刻(UNIX 時間)取得)を実行し、ネットワーク時刻(UNIX 時間)を取得します。
- ② 「gerR\_IN\_GetUnixOffsetTime」(ネットワーク時刻オフセット取得)を実行し、オフセットを取得します。
- ③ (1)内部で保持している UTC 時刻(UNIX 時間)のナノ秒から、オフセットのナノ秒を減算します。(本演算結果を A とします。)ただし、演算結果により、下記(a),(b),(c)いずれかを行います。

(a) 演算結果 A がマイナスの場合	
1.	ネットワーク時刻(UNIX 時間)の秒を 1 減算 減算前の秒が 0 (1970 年 1 月 1 日 00:00:01 未満)の場合は、補正不可のため処理を中断
2.	演算結果 A にナノ秒の最大値(1000000000)を加算し、ネットワーク時刻(UNIX 時間)のナノ秒に代入します。
(b) 演算結果 A がナノ秒の最大値(1000000000)以上の場合	
1.	ネットワーク時刻(UNIX 時間)の秒を 1 加算
2.	演算結果 A からナノ秒の最大値(1000000000)を減算し、ネットワーク時刻(UNIX 時間)のナノ秒に代入します。
(c) 演算結果 A が上記以外の場合	
1.	処理なし

- (2) 内部で保持している UTC 時刻(UNIX 時間)の秒から、オフセットの秒を減算します。(本演算結果を B とします。)ただし、演算結果がマイナスとなる場合は、補正不可のため処理を中断します。

- 
- ④ マスタ局に同期した UTC 時刻(UNIX 時間)の秒と、UTC オフセットの秒を加算します。その際に、オフセットの UTC オフセットが 8000H 以外か判定します。
- 8000H の場合、オフセットの加算処理は不要です。(8000H : オフセット指定なし)
- 8000H 以外の場合
- (1) UTC オフセットに 60(秒)を乗算
  - (2) ネットワーク時刻(UNIX 時間)の秒に、上記(1)の結果を加算
- ⑤ 現地時刻(UNIX 時間)の秒と、サマータイムオフセットの秒を加算します。その際に、オフセットのサマータイムオフセットが 8000H 以外か判定します。
- 8000H の場合、オフセットの加算処理は不要です。(8000H : オフセット指定なし)
- 8000H 以外の場合
- (1) UTC オフセットに 60(秒)を乗算
  - (2) ネットワーク時刻(UNIX 時間)の秒に、上記(1)の結果を加算

## 6.4.9 MDIO アクセス

## (1) gerR\_IN\_EnableMACIPAccess

機能	MAC_IP アクセス許可			
呼出し形式	ERRCODE gerR_IN_EnableMACIPAccess (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR_SEMAPHORE : セマフォ獲得失敗			
説明	<p>MAC_IP へのアクセスを許可します。</p> <p>本関数内でサービスコールとセマフォ資源の獲得を実行しているため、MAC_IP アクセス許可～MAC_IP アクセス禁止の間は、できるだけ短い間としてください。(ユーザが割込みを使用する場合、MAC_IP アクセス許可～MAC_IP アクセス禁止の間は割込み禁止状態で使用してください。)</p>			

## (2) gerR\_IN\_DisableMACIPAccess

機能	MAC_IP アクセス禁止			
呼出し形式	ERRCODE gerR_IN_DisableMACIPAccess (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR_FATAL : 異常完了(Fatal エラー発生)			
説明	<p>MAC_IP へのアクセスを禁止します。</p> <p>本関数は「gerR_IN_Initialize」(6.4.1(2))以降に実行してください。</p> <p>本関数の戻り値が R_IN_ERR_FATAL の場合、R-IN32M4-CL3 に Fatal エラーが発生しており、ユーザ作成の「gR_IN_CallbackFatalError」(6.6(1))を呼び出しますので R-IN32M4-CL3_Fatal エラーを取得してください。</p>			

## (3) gerR\_IN\_WritePHY

機能	PHY 内部レジスタライト			
呼出し形式	ERRCODE gerR_IN_WritePHY (ULONG ulPort, ULONG ulAddr, ULONG ulData)			
引数	型名	変数名	内容	入出力
	ULONG	ulPort	レジスタライトするポート R_IN_PORT1(0) : Port1 R_IN_PORT2(1) : Port2	入力
	ULONG	ulAddr	PHY レジスタアドレス	入力
	ULONG	ulData	PHY にライトするデータ	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外) R_IN_ERR_FATAL : 異常完了(Fatal エラー発生)			
説明	<p>MDIO にて PHY 内部レジスタをライトします。</p> <p>本関数は「gerR_IN_Initialize」(6.4.1(2))以降に実行してください。</p> <p>「gerR_IN_EnableMACIPAccess」(6.4.9(1))～「gerR_IN_DisableMACIPAccess」(6.4.9(2))の間で使用してください。</p> <p>本関数の戻り値が R_IN_ERR_FATAL の場合、R-IN32M4-CL3 に Fatal エラーが発生しており、ユーザ作成の「gR_IN_CallbackFatalError」(6.6(1))を呼び出しますので R-IN32M4-CL3 の Fatal エラーを取得してください。</p> <p>参考：PHY 内部レジスタリード/ライトにはそれぞれ最小 12 <math>\mu</math> 秒～最大 24 <math>\mu</math> 秒程度かかります。</p>			

## (4) gerR\_IN\_ReadPHY

機能	PHY 内部レジスタリード			
呼出し形式	ERRCODE gerR_IN_ReadPHY (ULONG ulPort, ULONG ulAddr, ULONG* pulData)			
引数	型名	変数名	内容	入出力
	ULONG	ulPort	レジスタリードするポート R_IN_PORT1(0) : Port1 R_IN_PORT2(1) : Port2	入力
	ULONG	ulAddr	PHY レジスタアドレス	入力
	ULONG*	pulData	PHY からリードしたデータ	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了(MDIO コマンド終了待ち異常) R_IN_ERR_OUTOFRANGE : 異常完了(範囲外) R_IN_ERR_FATAL : 異常完了(Fatal エラー発生)			
説明	<p>MDIO にて PHY 内部レジスタをリードします。</p> <p>本関数は「gerR_IN_Initialize」(6.4.1(2))以降に実行してください。</p> <p>「gerR_IN_EnableMACIPAccess」(6.4.9(1))～「gerR_IN_DisableMACIPAccess」(6.4.9(2))の間で使用してください。</p> <p>本関数の戻り値が R_IN_ERR_FATAL の場合、R-IN32M4-CL3 に Fatal エラーが発生しており、ユーザ作成の「gR_IN_CallbackFatalError」(6.6(1))を呼び出しますので R-IN32M4-CL3 の Fatal エラーを取得してください。</p> <p>参考：PHY 内部レジスタリード/ライトにはそれぞれ最小 12 <math>\mu</math> 秒～最大 24 <math>\mu</math> 秒程度かかります。</p>			

## 6.4.10 SLMP 送受信

## (1) gerR\_IN\_ReceivedSmpMain

機能	SLMP 受信メイン処理			
呼出し形式	ERRCODE gerR_IN_ReceivedSmpMain (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	SLMP フレームの受信処理を実行します。			

## (2) gvR\_IN\_ReceivedSmpExecution

機能	SLMP 受信コマンド実行			
呼出し形式	VOID gvR_IN_ReceivedSmpExecution (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	受信フレームから該当のコマンド処理を実行します。			

## (3) gbIR\_IN\_GetReceiveSmpStatus

機能	ユーザ理由による SLMP 受信許可設定状態取得			
呼出し形式	BOOL gbIR_IN_GetReceiveSmpStatus (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_TRUE : 受信許可 R_IN_FALSE : 受信不許可			
説明	ユーザ理由による SLMP 受信許可設定状態を取得します。			

## (4) gerR\_IN\_SetReceiveSmpStatus

機能	ユーザ理由による SLMP 受信許可設定			
呼出し形式	ERRCODE gerR_IN_SetReceiveSmpStatus (BOOL blEnable)			
引数	型名	変数名	内容	入出力
	BOOL	blEnable	受信許可設定	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	ユーザ理由による SLMP 受信許可設定状態に許可/不許可を設定します。			

## (5) gerR\_IN\_SetSmpCommand

機能	SLMP コマンド判定登録			
呼出し形式	ERRCODE gerR_IN_SetSmpCommand (const R_IN_SLMP_EXECUTION_RECEIVE_TBL_T* pstUserSmpReceiveFunctionTable)			
引数	型名	変数名	内容	入出力
	const R_IN_SLMP_EXECUTION_RECEIVE_TBL_T*	pstUserSmpReceiveFunctionTable	SLMP 受信実行関数テーブルポインタ 詳細は「表 6.29」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	引数で取得した SLMP 受信実行関数テーブルポインタを R-IN32M4-CL3 ドライバ本体へ通知します。			

表 6.29 R\_IN\_SLMP\_EXECUTION\_RECEIVE\_TBL\_T 一覧

No.	メンバ		内容
1	USHORT	usRequestNumber	要求コマンド登録数 (IP フレーム)
2	USHORT	usAcyclicDataRequestNumber	要求コマンド登録数 (AcyclicData フレーム)
3	USHORT	usResponseNumber	応答コマンド登録数 (IP フレーム)
4	R_IN_SLMP_FUNCTION_REQUEST_TBL_T*	pstRequest	要求コマンド実行関数テーブルポインタ (IP フレーム) (表 6.30)
5	R_IN_SLMP_FUNCTION_REQUEST_TBL_T*	pstAcyclicDataRequest	要求コマンド実行関数テーブルポインタ (AcyclicData フレーム) (表 6.30)
6	R_IN_SLMP_FUNCTION_RESPONSE_TBL_T*	pstResponse	応答コマンド実行関数テーブルポインタ (IP フレーム) (表 6.33)
7	R_IN_SLMP_RESPONSE_FUNCTION	fpv3Efunction	ST 応答フレーム受信関数ポインタ

表 6.30 R\_IN\_SLMP\_FUNCTION\_REQUEST\_TBL\_T

No.	メンバ		内容
1	USHORT	usCommand	SLMP コマンド
2	USHORT	usSubCommand	SLMP サブコマンド
3	R_IN_SLMP_REQUEST_FUNCTION	fperFunction	<p>コマンド関数ポインタ 受信した SLMP コマンドを処理する関数のポインタです。SLMP コマンドの一覧は「4.2.4 SLMP のコマンドについて」を参照してください。</p> <p>typedef ERRCODE (*R_IN_SLMP_REQUEST_FUNCTION) (VOID* pvRequestData, R_IN_SLMP_RECEIVE_INFORMATION_T* pstRequestInformation, VOID* pvReceiveData, R_IN_SLMP_SEND_INFORMATION_T* pstReceiveInformation);</p> <p>R_IN_SLMP_RECEIVE_INFORMATION_T は、「表 6.31」を参照してください。 R_IN_SLMP_SEND_INFORMATION_T は、「表 6.32」を参照してください。</p>
4	BOOL	blBroadcastSend	ブロードキャスト送信有無
5	ULONG	ulFrameType	<p>フレーム種別指定 Bit 0 : SLMP ST フレーム受信 (0b : 対応しない / 1b : 対応する) Bit 1 : SLMP MT フレーム受信 (0b : 対応しない / 1b : 対応する) Bit 3 : SLMP LMT フレーム受信 (0b : 対応しない / 1b : 対応する) 上記以外 : 0 固定</p>
6	BOOL	blFrameTypeErrorSuppression	<p>フレーム種別指定不一致時の異常応答抑止 0b : 抑止なし(異常応答する) 1b : 抑止あり(無応答)</p>
7	USHORT	usDataLengthErrorFincode	<p>要求データ長異常時の終了コード※1 R_IN_SLMP_FIN_CODE_NORMAL (0000H) : コマンド関数ポインタに登録された該当コマンド関数を実行します。 上記以外 : 設定値を終了コードに設定し R_IN0 ドライバ内でエラー応答します。</p>

【注】※1 受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバにてエラー応答、または該当コマンド関数内でエラー処理を実行するかを設定します。  
該当コマンド関数を実行する際は、要求データ(受信)付属情報(R\_IN\_SLMP\_RECEIVE\_INFORMATION\_T)の SLMP 受信結果にエラーコードを設定します。エラーコードは「表 6.35」を参照してください。

データ長異常として検出される SLMP フレームは、「図 6.3」のような分割番号がないフレーム、あるいはサブコマンド以降のデータが欠損しているフレーム等があります。

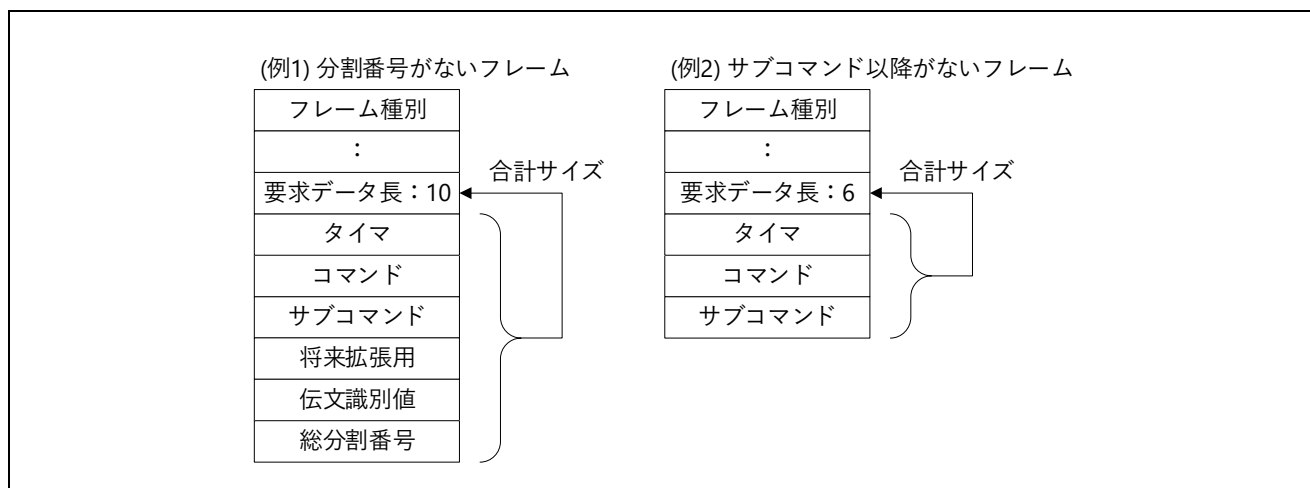


図 6.3 データ長異常の SLMP フレームイメージ

表 6.31 R\_IN\_SLMP\_RECEIVE\_INFORMATION\_T

No.	メンバ		内容
1	USHORT	usRequestDataSize	要求データサイズ
2	USHORT	usFType	フレームの種類
3	USHORT	usSerialNo	シリアル番号
4	USHORT	usReserved2	将来拡張用
5	USHORT	usDstProcNo	要求先局プロセッサ番号
6	UCHAR	uchMultiDropNo	要求先局プロセッササブ番号
7	UCHAR	uchReserved3	将来拡張用
8	USHORT	usLargeNodeNo	要求先局拡大局番号
9	USHORT	usTimer	タイマ
10	UCHAR	uchReserved4	将来拡張用
11	UCHAR	uchReqDataId	伝文識別値
12	USHORT	usDataDevideNum	総分割数
13	USHORT	usDataNumber	分割番号
14	ULONG	ullIPAddress	IP アドレス
15	USHORT	usPhysicalPort	物理ポート番号
16	USHORT	usReceivedResult	SLMP 受信結果※1

【注】※1 詳細は「表 6.35」を参照してください。

表 6.32 R\_IN\_SLMP\_SEND\_INFORMATION\_T

No.	メンバ		内容
1	USHORT	usResponseDataSize	応答データサイズ
2	USHORT	usFinishCode	終了コード



表 6.33 R\_IN\_SLMP\_FUNCTION\_RESPONSE\_TBL\_T

No.	メンバ		内容
1	USHORT	usCommand	SLMP コマンド
2	USHORT	usSubCommand	SLMP サブコマンド
3	R_IN_SLMP_REQUEST_FUNCTION	fperFunction	<p>コマンド関数ポインタ 受信した SLMP コマンドを処理する関数のポインタです。SLMP コマンドの一覧は「表 4.3」を参照してください。</p> <p>typedef VOID (*R_IN_SLMP_RESPONSE_FUNCTION) (const VOID* pvResponseDataOffset, R_IN_SLMP_RECEIVE_RESPONSE_INFORMATION_T* pstReceiveResponseInformation);</p> <p>R_IN_SLMP_RECEIVE_RESPONSE_INFORMATION_T は、「表 6.34」を参照してください。</p>
4	ULONG	ulFrameType	<p>フレーム種別指定 Bit 1 : SLMP MT フレーム受信 (0b : 対応しない / 1b : 対応する) Bit 3 : SLMP LMT フレーム受信 (0b : 対応しない / 1b : 対応する) 上記以外 : 0 固定</p>
5	BOOL	blDataLengthErrorDetection	<p>応答データ長異常時の設定※1 R_IN_TRUE : コマンド関数ポインタに登録された関数を実行します。 R_IN_FALSE : 受信フレームを破棄します。</p>

【注】※1 受信した LMT フレームにサブコマンド以降のデータ長異常を検出した場合、R-IN32M4-CL3 ドライバにてエラー応答、または該当コマンド関数内でエラー処理を実行するかを設定します。  
該当コマンド関数を実行する際は、応答データ(受信)付属情報  
(R\_IN\_SLMP\_RECEIVE\_RESPONSE\_INFORMATION\_T)の SLMP 受信結果にエラーコードを設定します。エラーコードは「表 6.35」を参照してください。

データ長異常として検出されるフレームは、「図 6.3」のように分割識別番号がないフレーム、あるいはサブコマンド以降のデータが欠損しているフレーム等があります。

表 6.34 R\_IN\_SLMP\_RECEIVE\_RESPONSE\_INFORMATION\_T

No.	メンバ	内容
1	USHORT usResponseDataSize	応答データサイズ
2	USHORT usFType	フレームの種別
3	USHORT usSerialNo	シリアル番号
4	USHORT usReserved2	将来拡張用
5	USHORT usDstProcNo	要求先局プロセッサ番号
6	UCHAR uchMultiDropNo	要求先局プロセッササブ番号
7	UCHAR uchReserved3	将来拡張用
8	USHORT usLargeNodeNo	要求先局拡大局番号
9	USHORT usFinishCode	終了コード
10	UCHAR uchReserved4	将来拡張用
11	UCHAR uchResDataId	伝文識別値
12	USHORT usDataDevideNum	総分割数
13	USHORT usDataNumber	分割番号
14	ULONG ullIPAddress	IP アドレス
15	USHORT usPhysicalPort	物理ポート番号
16	USHORT usReceivedResult	SLMP 受信結果※1

【注】※1 詳細は「表 6.35」を参照してください。

表 6.35 SLMP 受信結果のエラーコード一覧

No.	エラーコード	値	内容
1	R_IN_SLMP_RECEIVED_RESULT_NORMAL	0000H	正常受信
2	R_IN_SLMP_RECEIVED_RESULT_ERR_DATALEN	0100H	データ長異常を検出

## (6) gerR\_IN\_SendSlmpFrame

機能	SLMP フレーム送信			
呼出し形式	ERRCODE gerR_IN_SendSlmpFrame (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	SLMP フレームの送信処理を実行します。			

## (7) gerR\_IN\_WriteSlmpSendBuffer

機能	SLMP 送信バッファ書き込み			
呼出し形式	ERRCODE gerR_IN_WriteSlmpSendBuffer (const VOID* pvReceiveBuffer, USHORT usSize, ULONG ullpAddress, USHORT usPort, USHORT usPhysicalPort, VOID* pvSendBuffer)			
引数	型名	変数名	内容	入出力
	const VOID*	pvReceiveBuffer	受信バッファ	入力
	USHORT	usSize	送信サイズ	入力
	ULONG	ullpAddress	送信先 IP アドレス	入力
	USHORT	usPort	送信先ポート番号	入力
	USHORT	usPhysicalPort	物理ポート番号	入力
	VOID*	pvSendBuffer	送信バッファ	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	SLMP フレームを、R-IN32M4-CL3 内送信バッファへ書き込みます。 受信バッファには、送信するフレームに応じて以下のように設定します。 <ul style="list-style-type: none"> <li>SLMP 要求フレームを送信する場合、NULL を設定します。</li> <li>SLMP 応答フレームを送信する場合、受信した SLMP 要求フレームの先頭アドレスを設定します。</li> </ul> 送信先ポート番号には、45238 を設定しないでください。 送信バッファには、送信する SLMP フレームの先頭アドレスを指定します。			

## (8) gblR\_IN\_GetSlmpSendBufferUsed

機能	SLMP 送信バッファ使用有無取得			
呼出し形式	BOOL gblR_IN_GetSlmpSendBufferUsed (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_TRUE : SLMP 送信バッファにデータあり R_IN_FALSE : SLMP 送信バッファにデータなし			
説明	内部で保持している SLMP 送信バッファの使用有無を取得します。			

## (9) gerR\_IN\_SetSlmpResponseFrameSendRequest

機能	SLMP 応答フレーム送信要求			
呼出し形式	ERRCODE gerR_IN_SetSlmpResponseFrameSendRequest (VOID* pvResponseDataTopAddress, R_IN_SLMP_SEND_INFORMATION_T* pstReceiveInformation)			
引数	型名	変数名	内容	入出力
	VOID*	pvResponseDataTopAddress	応答データ先頭アドレス	入力
	R_IN_SLMP_SEND_INFORMATION_T*	pstReceiveInformation	応答データサイズ 詳細は「表 6.32」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	引数で指定された応答データと内部で保持している受信データ(SLMP 要求フレーム)を基に応答フレームを作成します。 本関数は「gvR_IN_CallbackNotifyReceivedSlmp」(6.6(3))から取得するフレームに対し、応答を送信する場合に使用してください。			

## (10) gvR\_IN\_ReleaseSlmpReceiveFrame

機能	SLMP 受信バッファ解放要求			
呼出し形式	ERRCODE gvR_IN_ReleaseSlmpReceiveFrame (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	受信データ(SLMP フレーム)で確保しているエリアを開放します。			

## 6.4.11 SLMP コマンド実行

## (1) gvR\_IN\_ExecuteReset

機能	自局リセット			
呼出し形式	VOID gvR_IN_ExecuteReset (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	自局(R-IN32M4-CL3)をリセットします。			

## (2) gerR\_IN\_StartSmpRequestTimer

機能	SLMP 要求待ちタイマ開始			
呼出し形式	ERRCODE gerR_IN_StartSmpRequestTimer (const R_IN_TIMEOUT_FUNCTION fpvTimeoutFunction , ULONG ulLimitTime , USHORT usTimerId)			
引数	型名	変数名	内容	入出力
	const R_IN_TIMEOUT_ FUNCTION	fpvTimeoutFunction	タイムアウト時実行関数 typedef VOID (* R_IN_TIMEOUT_FUNCTION)(VOID)	入力
	ULONG	ulLimitTime	タイマ時間(ms 単位)	入力
	USHORT	usTimerId	タイマ ID	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	要求伝文に対する応答伝文を送信してから、次の要求伝文受信までの時間を監視するタイマを起動します。本関数は、たとえばバックアップ/リストアなどタイマを使用するコマンドに使用します。			

## (3) gerR\_IN\_StopSmpRequestTimer

機能	SLMP 要求待ちタイマ停止			
呼出し形式	ERRCODE gerR_IN_StopSmpRequestTimer (USHORT usTimerId)			
引数	型名	変数名	内容	入出力
	USHORT	usTimerId	タイマ ID	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	要求伝文に対する応答伝文を送信してから、次の要求伝文受信までの時間を監視するタイマを停止します。本関数は、たとえばバックアップ/リストアなどタイマを使用するコマンドに使用します。			

## (4) gerR\_IN\_GetMasterIPAddress

機能	管理マスタ局 IP アドレス取得			
呼出し形式	ERRCODE gerR_IN_GetMasterIPAddress (ULONG* pulIPAddress, USHORT* pusPhysicalPort)			
引数	型名	変数名	内容	入出力
	ULONG*	pulIPAddress	IP アドレス	出力
	USHORT*	pusPhysicalPort	物理ポート番号	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	接続している管理マスタ局の IP アドレスと物理ポート番号を取得します。			

## (5) gerR\_IN\_CheckIpAddressSimp

機能	IP アドレスの判定			
呼出し形式	ERRCODE gerR_IN_CheckIpAddressSimp (ULONG ulIPAddress, ULONG ulSubnetMask)			
引数	型名	変数名	内容	入出力
	ULONG	ulIPAddress	IP アドレス	入力
	ULONG	ulSubnetMask	サブネットマスク	入力
戻り値	-			
説明	IP アドレスの妥当性を判定します。			

## (6) gerR\_IN\_SetNodeIndicationStatus

機能	NodeIndication ステータス設定			
呼出し形式	ERRCODE gerR_IN_SetNodeIndicationStatus(BOOL blStatus)			
引数	型名	変数名	内容	入出力
	BOOL	blStatus	R_IN_FALSE : インディケータ停止中または非対応 R_IN_TRUE : インディケータ表示中	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	MIB 機器詳細情報のアプリケーション情報「b1: インディケータ表示状態」を設定します。			

## (7) gvR\_IN\_RequestIPAddressChanging

機能	IP アドレス変更要求			
呼出し形式	VOID gvR_IN_RequestIPAddressChanging(ULONG ulIPAddress)			
引数	型名	変数名	内容	入出力
	ULONG	ulIPAddress	IP アドレス	入力
戻り値	-			
説明	<p>自局の IP アドレスを、引数で指定した IP アドレスに変更するよう R-IN32M4-CL3 ドライバに通知します。</p> <p>マスタ局と通信開始前(サイクリック伝送設定前)であれば、通知を受けた R-IN32M4-CL3 ドライバは、自局の IP アドレスを引数で指定した IP アドレスに変更します。</p> <p>マスタ局と通信開始後(サイクリック伝送設定後)の場合、自局の IP アドレスを変更しません。</p> <p>自局の IP アドレス変更結果は「gvR_IN_CallbackIPAddressChangingResult」(6.6(10))により通知されます。本関数を呼び出す前に、「gerR_IN_CheckIpAddressSimp」(6.4.11(5))にて引数の IP アドレスの妥当性を判定してください。</p>			

## (8) gerR\_IN\_SetSNMPCommunityName

機能	SNMP コミュニティ名設定			
呼出し形式	ERRCODE gerR_IN_SetSNMPCommunityName (CHAR* pchCommunityName)			
引数	型名	変数名	内容	入出力
	CHAR*	pchCommunityName	コミュニティ名	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	引数で指定した SNMP コミュニティ名を R-IN32M4-CL3 ドライバに設定します。 本関数を使用するには、コンパイルスイッチで「SNMP_COMMUNITY_NAME_SETTING_ENABLE」を定義してください。			

## 6.4.12 エラー履歴

## (1) gerR\_IN\_SetErrorHistory

機能	エラー履歴登録			
呼出し形式	ERRCODE gerR_IN_SetErrorHistory(R_IN_ERROR_INFORMATION_T* pstErrorInformation)			
引数	型名	変数名	内容	入出力
	R_IN_ERROR_INFORMATION_T*	pstErrorInformation	エラー情報 詳細は「表 6.36」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	<p>ユーザアプリケーションにおけるエラー情報を MIB の現在エラー情報に登録します。登録したエラー履歴は、マスタ局に RJ71GN11-T2 を使用する場合、GX Works3 の CC-Link IE TSN 診断画面に表示されます。</p> <p>増設ユニット/スライス I/O を装着し、コンパイルスイッチ「CURERR_OPTIONINFO_ENABLE」を定義した場合は、基本ユニット(コントローラ情報)におけるエラー情報を登録します。</p> <p>エラー履歴クリアを実行中は本処理を実行しないでください。</p>			

表 6.36 R\_IN\_ERROR\_INFORMATION\_T 一覧

No.	メンバ		内容
1	USHORT	usErrorCode	エラーコード
2	UCHAR	uchErrorDetailSize	エラー詳細サイズ
3	USHORT	ausErrorDetail[10]	エラー詳細情報

## (2) gerR\_IN\_ClearErrorHistory

機能	エラー履歴クリア			
呼出し形式	VOID gvR_IN_ClearErrorHistory(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	<p>エラー履歴の情報をクリアします。</p> <p>増設ユニット/スライス I/O 装着し、コンパイルスイッチ「CURERR_OPTIONINFO_ENABLE」を定義した場合は、増設/スライス I/O 数を含むエラー履歴の全情報をクリアするため、本関数は使用しないでください。</p> <p>コンパイルスイッチ「CURERR_OPTIONINFO_ENABLE」を定義した場合は、以下を使用して、基本ユニット(コントローラ情報)および増設ユニット/スライス I/O (オプション情報)のエラー履歴をクリアしてください。</p> <ul style="list-style-type: none"> <li>・gvR_IN_ClearErrorHistoryController (6.4.12(3))</li> <li>・gerR_IN_ClearErrorHistoryOption (6.4.12(5))</li> </ul> <p>エラー履歴登録を実行中は、本処理を実行しないでください。</p>			



## (3) gvR\_IN\_ClearErrorHistoryController

機能	エラー履歴クリア(コントローラ情報)			
呼出し形式	VOID gvR_IN_ClearErrorHistoryController(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	コントローラ情報のエラー履歴をクリアします。 エラー履歴登録を実行中は、本処理を実行しないでください。			

## (4) gerR\_IN\_SetErrorHistoryOption

機能	エラー履歴登録(オプション情報)			
呼出し形式	ERRCODE gerR_IN_SetErrorHistoryOption (R_IN_ERROR_INFORMATION_T* pstErrorInformation, USHORT usOptNum)			
引数	型名	変数名	内容	入出力
	R_IN_ERROR_INFORMATION_T*	pstErrorInformation	エラー情報 詳細は「表 6.36」を参照してください。	入力
	USHORT	usOptNum	オプション情報番号	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	引数のオプション情報番号とエラー情報（エラーコード、エラー詳細サイズ、エラー詳細情報）をエラー履歴に登録します。 エラー履歴クリア(オプション情報)を実行中は本処理を実行しないでください。			

## (5) gerR\_IN\_ClearErrorHistoryOption

機能	エラー履歴クリア(オプション情報)			
呼出し形式	ERRCODE gerR_IN_ClearErrorHistoryOption(USHORT usOptNum)			
引数	型名	変数名	内容	入出力
	USHORT	usOptNum	オプション情報番号	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	引数のオプション情報番号に該当するエラー履歴の情報をクリアします。 エラー履歴登録(オプション情報)を実行中は本処理を実行しないでください。			

## 6.4.13 H/W テスト

## (1) gerR\_IN\_IEEE\_Test

機能	IEEE802.3ab コンプライアンステスト			
呼出し形式	VOID gerR_IN_IEEE_Test (USHORT usMode)			
引数	型名	変数名	内容	入出力
	USHORT	usMode	テストモード R_IN_IEEE_MODE1(0) : MODE1 R_IN_IEEE_MODE2(1) : MODE2 R_IN_IEEE_MODE3(2) : MODE3 R_IN_IEEE_MODE4(3) : MODE4 R_IN_IEEE_END(4) : テスト終了	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_FATAL : 異常完了(Fatal エラー)			
説明	引数の IEEE802.3ab コンプライアンステストモードに従い、テストモード用の波形出力を PHY に設定します。 本関数の戻り値が R_IN_ERR_FATAL の場合、R-IN32M4-CL3 に Fatal エラーが発生していますので R-IN32M4-CL3 の Fatal エラーを取得してください。			

## (2) gerR\_IN\_InitializeLoopBackTest

機能	折り返し通信テスト初期化			
呼出し形式	ERRCODE gerR_IN_InitializeLoopBackTest(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外) R_IN_ERR_FATAL : 異常完了(Fatal エラー)			
説明	折り返し通信テストを行うための初期化処理を行います。			

## (3) gerR\_IN\_SendLoopBackTest

機能	折り返し通信テストデータ送信			
呼出し形式	ERRCODE gerR_IN_SendLoopBackTest (ULONG ulPort)			
引数	型名	変数名	内容	入出力
	ULONG	ulPort	テスト対象ポート	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	引数で指定されたテスト対象ポートから通信テストデータを送信します。 本テストを実施する場合、Ethernet PORT1 と Ethernet PORT2 間を Ethernet ケーブルで接続してください。			

## (4) gerR\_IN\_ReceiveLoopBackTest

機能	折り返し通信テストデータ受信			
呼出し形式	ERRCODE gerR_IN_ReceiveLoopBackTest (ULONG ulPort)			
引数	型名	変数名	内容	入出力
	ULONG	ulPort	テスト対象ポート	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外) R_IN_ERR_NODATA : 異常完了(データがない) R_IN_ERR_FATAL : 異常完了(Fatal エラー) R_IN_ERR_TIMEOUT : 異常完了(タイムアウト)			
説明	非サイクリックフレーム受信を実施し、送信データと受信データを比較します。			

## 6.4.14 汎用共通

## (1) gverR\_IN\_CopyMemory

機能	メモリコピー			
呼出し形式	ERRCODE gerR_IN_CopyMemory(VOID* pvDestination, const VOID* pvSource, ULONG ulSize)			
引数	型名	変数名	内容	入出力
	VOID*	pvDestination	コピー先データ	出力
	const VOID*	pvSource	コピー元データ	入力
	ULONG	ulSize	コピーデータサイズ(バイト単位)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	32bit/16bit/8bit 単位でメモリのコピーを行います。			

## (2) gerR\_IN\_FillMemory

機能	メモリフィル(8bit)			
呼出し形式	ERRCODE gerR_IN_FillMemory (VOID* pvDestination, UCHAR uchFillData, ULONG ulSize)			
引数	型名	変数名	内容	入出力
	VOID*	pvDestination	フィル先データ	出力
	UCHAR	uchFillData	フィルデータ(00H~FFH)	入力
	ULONG	ulSize	フィル先データサイズ(バイト単位)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	8bit 単位でメモリを指定データで埋めます。			

## (3) gerR\_IN\_FillMemory16

機能	メモリフィル(16bit)			
呼出し形式	ERRCODE gerR_IN_FillMemory16 (VOID* pvDestination, USHORT usFillData, ULONG ulSize)			
引数	型名	変数名	内容	入出力
	VOID*	pvDestination	フィル先データ	出力
	USHORT	usFillData	フィルデータ(0000H~FFFFH)	入力
	ULONG	ulSize	フィル先データサイズ(バイト単位)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	16bit 単位でメモリを指定データで埋めます。			

## (4) gerR\_IN\_FillMemory32

機能	メモリフィル(32bit)			
呼出し形式	ERRCODE gerR_IN_FillMemory32 (VOID* pvDestination, ULONG ulFillData, ULONG ulSize)			
引数	型名	変数名	内容	入出力
	VOID*	pvDestination	フィル先データ	出力
	ULONG	ulFillData	フィルデータ(00000000H~FFFFFFFFH)	入力
	ULONG	ulSize	フィル先データサイズ(バイト単位)	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	32bit 単位でメモリを指定データで埋めます。			

## (5) gerR\_IN\_EndianShort

機能	エンディアン変換(USHORT)			
呼出し形式	ERRCODE gerR_IN_EndianShort(USHORT* pusVal)			
引数	型名	変数名	内容	入出力
	USHORT*	pusVal	変換対象	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	2 バイトの引数をエンディアン変換します。			

## (6) gerR\_IN\_EndianLong

機能	エンディアン変換(ULONG)			
呼出し形式	ERRCODE gerR_IN_EndianLong (ULONG* pulVal)			
引数	型名	変数名	内容	入出力
	ULONG*	pulVal	変換対象	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	4 バイトの引数をエンディアン変換します。			

## (7) gerR\_IN\_EndianLongLong

機能	エンディアン変換(ULONGULONG)			
呼出し形式	ERRCODE gerR_IN_EndianLongLong (ULONGLONG* pullVal)			
引数	型名	変数名	内容	入出力
	ULONGLONG*	pullVal	変換対象	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	8 バイトの引数をエンディアン変換します。			

## (8) gvR\_IN\_DisableInt

機能	割込み禁止			
呼出し形式	VOID gvR_IN_DisableInt (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	<p>割込みプログラムの実行を禁止します。</p> <p>タスクで割込み禁止/割込み許可関数を実行して排他制御を行う場合、割込み禁止/割込み許可関数の前後にディスパッチ禁止/ディスパッチ許可関数を実行して他の優先度が高いタスクにタスク切り替えされないようにしてから割込み禁止/割込み許可関数を実行してください。</p> <p>(例)</p> <ol style="list-style-type: none"> <li>1 ディスパッチ禁止</li> <li>2 割込み禁止</li> <li>3 排他制御対象処理</li> <li>4 割込み許可</li> <li>5 ディスパッチ許可</li> </ol>			

## (9) gvR\_IN\_EnableInt

機能	割込み許可			
呼出し形式	VOID gvR_IN_EnableInt (VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	<p>割込みプログラムの実行を許可します。</p> <p>タスクで割込み禁止/割込み許可関数を実行して排他制御を行う場合、割込み禁止/割込み許可関数の前後にディスパッチ禁止/ディスパッチ許可関数を実行して他の優先度が高いタスクにタスク切り替えされないようにしてから割込み禁止/割込み許可関数を実行してください。</p> <p>(例)</p> <ol style="list-style-type: none"> <li>1 ディスパッチ禁止</li> <li>2 割込み禁止</li> <li>3 排他制御対象処理</li> <li>4 割込み許可</li> <li>5 ディスパッチ許可</li> </ol>			

## (10) gvR\_IN\_DisableDispatch

機能	ディスパッチ禁止			
呼出し形式	VOID gvR_IN_DisableDispatch(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	タスクのスケジューリング(切り替え)を禁止します。 割込み禁止状態で、ディスパッチ禁止/ディスパッチ許可関数を実行しないでください。 ディスパッチ許可関数実行時に、タスク切り替えが発生している場合、SVC 割込みが実行できずにハードフォールト例外が発生します。			

## (11) gvR\_IN\_EnableDispatch

機能	ディスパッチ許可			
呼出し形式	VOID gvR_IN_EnableDispatch(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	タスクのスケジューリング(切り替え)を許可します。 割込み禁止状態で、ディスパッチ禁止/ディスパッチ許可関数を実行しないでください。 ディスパッチ許可関数実行時に、タスク切り替えが発生している場合、SVC 割込みが実行できずにハードフォールト例外が発生します。			

## 6.4.15 ネットワーク同期通信

## (1) gerR\_IN\_GetSyncDeviationFlag

機能	時刻同期外れ検出			
呼出し形式	ERRCODE gerR_IN_GetSyncDeviationFlag(BOOL* pblSyncDeviationFlag)			
引数	型名	変数名	内容	入出力
	BOOL*	pblSyncDeviationFlag	時刻同期外れ検出 R_IN_FALSE : 時刻同期外れ検出なし R_IN_TRUE : 時刻同期外れ検出済み	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	ネットワーク参加/複列からの、時刻同期外れの発生状態を取得します。			

## (2) gerR\_IN\_StopAppSyncSignal

機能	同期信号出力停止			
呼出し形式	ERRCODE gerR_IN_StopAppSyncSignal(USHORT* pusResult)			
引数	型名	変数名	内容	入出力
	USHORT*	pusResult	実行結果 R_IN_SUCCEED : 成功 R_IN_FAIL : 失敗	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	同期信号の出力を停止します。 同期信号の出力を停止することで同期処理タスクが起床されなくなります。 戻り値が R_IN_FAIL の場合、既に同期信号出力が停止済みであることを示します。			

## (3) gerR\_IN\_SetWdcThreshold

機能	ウォッチドッグカウンタチェック異常連続カウンタ閾値設定			
呼出し形式	ERRCODE gerR_IN_SetWdcThreshold(USHORT usWdcThreshold)			
引数	型名	変数名	内容	入出力
	USHORT	usWdcThreshold	連続 WDC 異常回数閾値 詳細は「表 6.37」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_OUTOFRANGE : 異常完了(範囲外)			
説明	SLMP ウォッチドッグカウンタ設定コマンドの要求伝文で指定されたウォッチドッグカウンタチェック異常連続カウンタ閾値を設定します。			

表 6.37 連続 WDC 異常回数閾値の設定内容

ビット	名称	設定値
bit 15	ウォッチドッグカウンタチェック異常連続カウンタ閾値設定有無	0 : 設定なし 1 : 設定あり
bit 14-0	ウォッチドッグカウンタチェック異常連続カウンタ閾値	0000H~0FFFH (0~4095)



## (4) gerR\_IN\_MakeResponseWdcInformation

機能	ウォッチドッグカウンタ情報設定応答データ作成			
呼出し形式	ERRCODE gerR_IN_MakeResponseWdcInformation(VOID* pvResponseData, USHORT* pusDataSize)			
引数	型名	変数名	内容	入出力
	VOID*	pvResponseData	応答データ※1	出力
	USHORT*	pusDataSize	応答データサイズ	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(データなし)			
説明	SLMP ウォッチドッグカウンタ設定コマンド(要求)の応答データを、R-IN32M4-CL3 ドライバが管理している情報を元に作成します。 サブペイロード情報が未確定の状態で行った場合、R_IN_ERR_NODATA のエラーとなります。 CyclicConfigTrnSubPayload フレーム/CyclicConfigRcvSubPayload フレーム受信にてサブペイロード情報が確定した後に呼び出してください。SLMP ウォッチドッグカウンタ情報設定要求コマンドは、CyclicConfigTrnSubPayload フレーム/CyclicConfigRcvSubPayload フレーム受信後にマスタ局から送信されるプロトコル仕様となっていますので、SLMP ウォッチドッグカウンタ情報設定要求コマンド受信時に実行してください。			

【注】※1：応答データの概略構造を以下に示します。詳細は、CC-Link 協会の SLMP 仕様書を参照してください。

No.	応答データ(ResSetWatchdogCounterInfo)の項目		サイズ
1	transmitSubPayloadNum		2
2	transmitSubPayloadInfo1	watchdogCounterExistence	1
3		Reserved	1
4		receiveMemoryAddr	4
5		watchdogCounterUIIndex	2
6		watchdogCounterUISubIndex	1
7		Reserved	1
8		watchdogCounterUIOffset	2
9	transmitSubPayloadInfo2～m (transmitSubPayloadInfo1 同様)		12×(m-1)
10	receiveSubPayloadNum		2
11	receiveSubPayloadInfo1	watchdogCounterExistence	1
12		Reserved	1
13		watchdogCounterDIIndex	2
14		watchdogCounterDISubIndex	1
15		Reserved	1
16		watchdogCounterDIOffset	2
17	receiveSubPayloadInfo2～n (receiveSubPayloadInfo1 同様)		8×(n-1)

m = transmitSubPayloadNum

n = receiveSubPayloadNum

## (5) gvR\_IN\_IncrementWdcUL

機能	ウォッチドッグカウンタインクリメント(送信用)			
呼出し形式	VOID gvR_IN_IncrementWdcUL(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	送信用ウォッチドッグカウンタの正常値をインクリメントします。 1 回の実行で送信用ウォッチドッグカウンタの正常値に加算される値は、「gerR_IN_Initialize」(6.4.1(2))の引数である、R-IN32M4-CL3 ユニット情報(pstUnitInfo)にて設定します。			

## (6) gvR\_IN\_LatchWdcUL

機能	ウォッチドッグカウンタラッチ値更新(送信用)			
呼出し形式	VOID gvR_IN_LatchWdcUL(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	マスタ局へ送信するウォッチドッグカウンタの値(ラッチ値)に、送信用ウォッチドッグカウンタの正常値を設定します。			

## (7) gerR\_IN\_GetWdcUL

機能	ウォッチドッグカウンタ UL 取得			
呼出し形式	ERRCODE gerR_IN_GetWdcUL(BOOL biWdcULValidFlag, USHORT* pusWdcUL)			
引数	型名	変数名	内容	入出力
	BOOL	biWdcULValidFlag	ウォッチドッグカウンタ有効/無効フラグ送信制御フラグ R_IN_FALSE : ウォッチドッグカウンタ無効指定 R_IN_TRUE : ウォッチドッグカウンタ有効指定	入力
	USHORT*	pusWdcUL	ウォッチドッグカウンタ(送信用)	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	「ウォッチドッグカウンタラッチ値更新(送信用)」にてラッチしたウォッチドッグカウンタの値をもとに、サイクリックデータとしてマスタ局へ送信するウォッチドッグカウンタ UL の値を取得します。 ウォッチドッグカウンタ UL を無効とする場合は、ウォッチドッグカウンタ有効/無効フラグ送信制御フラグを R_IN_FALSE とします。			

## (8) gerR\_IN\_CheckWdcDL

機能	ウォッチドッグカウンタチェック(受信用)			
呼出し形式	CANopen 通信未使用時	ERRCODE gerR_IN_CheckWdcDL(USHORT usWdc, USHORT* pusCheckResult)		
	CANopen 通信使用時	ERRCODE gerR_IN_CheckWdcDL(USHORT usObjectDictionaryNo, USHORT usWdc, USHORT* pusCheckResult)		
引数	型名	変数名	内容	入出力
	USHORT	usObjectDictionaryNo	Object Dictionary 番号※1	入力
	USHORT	usWdc	受信ウォッチドッグカウンタ	入力
	USHORT*	pusCheckResult	受信ウォッチドッグカウンタチェック結果 R_IN_WDC_OK : 正常 R_IN_WDC_NOT_COMP : 比較せずに終了 R_IN_WDC_NG_LESS : 異常(閾値の範囲内) R_IN_WDC_NG_OVER : 異常(閾値の範囲外)	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	保持している閾値の情報をもとに、受信したウォッチドッグカウンタをチェックし、結果を返します。			
	チェック結果		状態	
	R_IN_WDC_OK		下記のいずれでもない(受信ウォッチドッグカウンタは正常)	
	R_IN_WDC_NOT_COMP		以下の条件に当てはまる場合 ・ 引数で指定された受信ウォッチドッグカウンタが無効。 ・ 引数で指定された受信ウォッチドッグカウンタは有効だが、前回実行時の受信ウォッチドッグカウンタが無効。 ・ データリンク状態がデータリンク中(サイクリック通信中)ではない。	
	R_IN_WDC_NG_LESS		以下いずれかの状態がウォッチドッグカウンタチェック異常連続カウンタ閾値より少ない回数継続している。 ・ 引数の受信ウォッチドッグカウンタのカウント値が前回実行時の受信ウォッチドッグカウンタのカウント値以下 ・ ウォッチドッグカウンタインクリメント値の2倍を超えて差分がある	
	R_IN_WDC_NG_OVER		以下いずれかの状態がウォッチドッグカウンタチェック異常連続カウンタ閾値以上の回数継続している。 ・ 引数の受信ウォッチドッグカウンタのカウント値が前回実行時の受信ウォッチドッグカウンタのカウント値以下 ・ ウォッチドッグカウンタインクリメント値の2倍を超えて差分がある	

【注】※1 コンパイルスイッチ「TSN\_CAN\_ENABLE」を有効にした場合

## (9) gvR\_IN\_ClearSyncDeviationFlag

機能	時刻同期外れ検出状態クリア			
呼出し形式	VOID gvR_IN_ClearSyncDeviationFlag(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	<p>ネットワーク参加/復列からの時刻同期外れの発生状態をクリアします。</p> <p>時刻同期外れ検出時にエラー登録する場合、エラークリア後に再度同一のエラーが通知されないよう、エラークリアと合わせて本 API を実行してください。</p> <p>時刻同期外れの判定を行う R-IN32M4-CL3 ドライバの TSN 定周期処理タスクより優先度の低いタスクで本 API を実行する場合は、時刻同期外れの判定に用いるデータの泣き別れが発生する可能性があります。</p> <p>本 API 実行後に時刻同期外れを正しく判定できるようにするため、ディスパッチ禁止等の排他処理を行ってください。</p>			

## 6.4.16 CANopen 通信

## (1) gerR\_IN\_CanInit

機能	CANopen 通信機能の初期化			
呼出し形式	ERRCODE gerR_IN_CanInit (const R_IN_CAN_OD_TABLE_T* pstObjectDictionary, USHORT usObjectNumber R_IN_CAN_ERROR_DETAIL_T* pstErrorDetail)			
引数	型名	変数名	内容	入出力
	const R_IN_CAN_OD_TABLE_T*	pstObjectDictionary	Object Dictionary の先頭アドレス 詳細は「表 6.38」を参照してください	入力
	USHORT	usObjectNumber	Object Dictionary の要素数	入力
	R_IN_CAN_ERROR_DETAIL_T*	pstErrorDetail	エラー詳細情報 詳細は「表 6.39」を参照してください	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	CANopen 通信機能の初期化を実行します。登録可能な Object Dictionary の数は「1~8」となります			

表 6.38 R\_IN\_CAN\_OD\_TABLE\_T 一覧

No.	メンバ		内容
1	const R_IN_CAN_OD_T	*pstObjectDictionary	Object Dictionary 先頭アドレス※3
2	USHORT	usObjectDictionaryNum	Object Dictionary 要素数

【注】※3 Object Dictionary 構造体は、「表 6.40 R\_IN\_CAN\_OD\_T 一覧」を参照してください。

表 6.39 R\_IN\_CAN\_ERROR\_DETAIL\_T 一覧

No.	メンバ		内容
1	ULONG	ulAbortCode	アボートコード
2	USHORT	usObjectDictionaryNo	Object Dictionary 番号
3	USHORT	usIndex	Index
4	UCHAR	uchSubIndex	SubIndex

## (2) gerR\_IN\_CanGetValidObjectDictionaryNumber

機能	有効 ObjectDictionary 数取得			
呼出し形式	ERRCODE gerR_IN_CanGetValidObjectDictionaryNumber (USHORT* pusObjectDictionaryNumber)			
引数	型名	変数名	内容	入出力
	USHORT*	pusObjectDictionaryNumber	有効 ObjectDictionary 数	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(有効 ObjectDictionary 数未確定)			
説明	有効な ObjectDictionary の数を取得します。 有効な ObjectDictionary の数は、NMTState が INIT ときに確定します。 有効な ObjectDictionary の数が未確定の間に本関数を実行すると、戻り値で「R_IN_ERR_NODATA」が返ります。			

表 6.40 R\_IN\_CAN\_OD\_T 一覧

No.	メンバ		内容																															
1	USHORT	usIndex	インデックス																															
2	UCHAR	uchMaxSubIndex	最大サブインデックス SubIndex1 以降の数を設定します。SubIndex0 はエントリ数のため含みません。オブジェクトコードが「R_IN_CAN_VAR」の場合は、「0」を設定してください。																															
3	UCHAR	uchObjectCode	下記のオブジェクトコードを設定します。 ・ R_IN_CAN_VAR (07H) : 単一の値 ・ R_IN_CAN_ARRAY (08H) : 同一型の複数の SubIndex を持つ ・ R_IN_CAN_RECORD (09H) : 異なる型の複数の SubIndex を持つ 詳細は、CC-Link 協会の CC-Link IE TSN 仕様書(概要編)を参照してください。																															
4	R_IN_CAN_OBJECT_DATA_T*	pstObjectData	各 SubIndex のデータを定義したテーブルの先頭アドレスを設定します。 uchMaxSubIndex の数が「0」であっても設定が必要です。 「表 6.41」を参照してください。																															
5	const CHAR*	pcName	オブジェクトの名称を格納した領域の先頭アドレスを設定します。 名称を格納した領域の末尾には、終端文字「00H」を設定してください。 文字列は 500H (1280 文字)内で設定してください。																															
6	ULONG	(*pfulRead)()	<p>「gerR_IN_CanReadObject」(6.4.16(4))または 「gerR_IN_CanReadBlockObject」(6.4.16(6))を行った際に実行したい処理がある場合は、任意の関数を作成し登録します。 「gerR_IN_CanReadBlockObject」の場合は、読出し対象の SubIndex 数分実行されます。 NULL の場合は、デフォルトの読出し処理が実行されます。</p> <p>&lt;*pfulRead に登録する関数の形式&gt;</p> <table><tr><td>呼出し形式</td><td colspan="4">ULONG gulReadFunc (USHORT usIndex, UCHAR uchSubindex, ULONG ulSize, UCHAR *puchData )</td></tr><tr><td rowspan="5">引数</td><td>型名</td><td>変数名</td><td>内容</td><td>入出力</td></tr><tr><td>USHORT</td><td>usIndex</td><td>インデックス</td><td>入力</td></tr><tr><td>UCHAR</td><td>uchSubindex</td><td>サブインデックス</td><td>入力</td></tr><tr><td>ULONG</td><td>ulSize</td><td>読出しサイズ Byte)</td><td>入力</td></tr><tr><td>UCHAR</td><td>*puchData</td><td>読出しデータ※4</td><td>出力</td></tr><tr><td>戻り値</td><td colspan="4">アボートコード</td></tr></table> <p>【注】※4 : 読出しデータは読出しサイズ分値を設定してください。設定がない場合、前回値を送信します。</p>	呼出し形式	ULONG gulReadFunc (USHORT usIndex, UCHAR uchSubindex, ULONG ulSize, UCHAR *puchData )				引数	型名	変数名	内容	入出力	USHORT	usIndex	インデックス	入力	UCHAR	uchSubindex	サブインデックス	入力	ULONG	ulSize	読出しサイズ Byte)	入力	UCHAR	*puchData	読出しデータ※4	出力	戻り値	アボートコード			
呼出し形式	ULONG gulReadFunc (USHORT usIndex, UCHAR uchSubindex, ULONG ulSize, UCHAR *puchData )																																	
引数	型名	変数名	内容	入出力																														
	USHORT	usIndex	インデックス	入力																														
	UCHAR	uchSubindex	サブインデックス	入力																														
	ULONG	ulSize	読出しサイズ Byte)	入力																														
	UCHAR	*puchData	読出しデータ※4	出力																														
戻り値	アボートコード																																	
7	ULONG	(*pfulWrite)()	<p>「gerR_IN_CanWriteObject」(6.4.16(5))または、 「gerR_IN_CanWriteBlockObject」(6.4.16(7))を行った際に実行したい処理がある場合は、任意の関数を作成し登録します。 「gerR_IN_CanWriteBlockObject」の場合は、書込み対象の SubIndex 数分実行されます。NULL の場合は、デフォルトの書込み処理が実行されます。</p> <p>&lt;*pfulWrite に登録する関数の形式&gt;</p> <table><tr><td>呼出し形式</td><td colspan="4">ULONG gulWriteFunc (USHORT usIndex, UCHAR uchSubindex, ULONG ulSize, UCHAR *puchData )</td></tr><tr><td rowspan="5">引数</td><td>型名</td><td>変数名</td><td>内容</td><td>入出力</td></tr><tr><td>USHORT</td><td>usIndex</td><td>インデックス</td><td>入力</td></tr><tr><td>UCHAR</td><td>uchSubindex</td><td>サブインデックス</td><td>入力</td></tr><tr><td>ULONG</td><td>ulSize</td><td>書込みサイズ Byte)</td><td>入力</td></tr><tr><td>UCHAR</td><td>*puchData</td><td>書込みデータ</td><td>入力</td></tr><tr><td>戻り値</td><td colspan="4">アボートコード</td></tr></table>	呼出し形式	ULONG gulWriteFunc (USHORT usIndex, UCHAR uchSubindex, ULONG ulSize, UCHAR *puchData )				引数	型名	変数名	内容	入出力	USHORT	usIndex	インデックス	入力	UCHAR	uchSubindex	サブインデックス	入力	ULONG	ulSize	書込みサイズ Byte)	入力	UCHAR	*puchData	書込みデータ	入力	戻り値	アボートコード			
呼出し形式	ULONG gulWriteFunc (USHORT usIndex, UCHAR uchSubindex, ULONG ulSize, UCHAR *puchData )																																	
引数	型名	変数名	内容	入出力																														
	USHORT	usIndex	インデックス	入力																														
	UCHAR	uchSubindex	サブインデックス	入力																														
	ULONG	ulSize	書込みサイズ Byte)	入力																														
	UCHAR	*puchData	書込みデータ	入力																														
戻り値	アボートコード																																	

表 6.41 R\_IN\_CAN\_OBJECT\_DATA\_T 一覧

No.	メンバ		内容
1	USHORT	usDataType	「*pvValue」が示す領域のデータタイプを設定します。
			定義
			設定値
			内容
			R_IN_CAN_INTEGER8
			R_IN_CAN_INTEGER16
			R_IN_CAN_INTEGER32
			R_IN_CAN_UNSIGNED8
2	USHORT	usBitLength	「*pvValue」が示す領域の有効なビット長を設定します。
			定義
			設定値
			R_IN_CAN_INTEGER8
			R_IN_CAN_INTEGER16
			R_IN_CAN_INTEGER32
			R_IN_CAN_UNSIGNED8
			R_IN_CAN_UNSIGNED16
3	UCHAR	uchAccess	SDO のアクセス種別を設定します。
			定義
			設定値
			内容
			R_IN_CAN_READWRITE
			R_IN_CAN_READ
			R_IN_CAN_READ_PREOP
			R_IN_CAN_READ_SAFEOP
			R_IN_CAN_READ_OP
			R_IN_CAN_WRITE
4	UCHAR	uchMapping	PDO のマッピング可否を設定します。
			定義
			設定値
			内容
			R_IN_CAN_NOPDOMAPPING
			R_IN_CAN_RPDOMAPPING
			R_IN_CAN_TPDOMAPPING

No.	メンバ		内容		
5	UCHAR	uchValueInfo	「ユニットタイプ (ulUnitType)」、「初期値 (ulDefaultValue)」、「最小値 (ulMinValue)」、「および「最大値 (ulMaxValue)」を有効にするか設定します。		
			定義	設定値	内容
			R_IN_CAN_VALUEINFO_UNIT TYPE	08H	ユニットタイプのデータが有効になります。
			R_IN_CAN_VALUEINFO_VAL UE	70H	初期値、最小値、最大値の設定が有効になります。
			R_IN_CAN_VALUEINFO_DEF AULTVALUE	10H	初期値の設定が有効になります。
			R_IN_CAN_VALUEINFO_MIN MAXVALUE	60H	最小値、最大値の設定が有効になります。
			R_IN_CAN_VALUEINFO_MINV ALUE	20H	最小値の設定が有効になります。
			R_IN_CAN_VALUEINFO_MAX VALUE	40H	最大値の設定が有効になります。
6	ULONG	ulUnitType	エントリの単位を 4Byte の数値で設定します。 (例 : km/h の場合は 0301 4800H (k : 03、m : 01、h : 48)となります。)		
7	ULONG	ulDefaultValue	初期値を設定します。 ※5		
8	ULONG	ulMinValue	最小値を設定します。 ※5		
9	ULONG	ulMaxValue	最大値を設定します。 ※5		
10	VOID	*pvValue	アプリケーションで使用する変数の先頭アドレスを設定します。 データタイプに「R_IN_CAN_VISIBLESTRING」を設定した場合は、下記の条件を満たしてください。 ・文字列を格納した領域の末尾には、終端文字「00H」を設定する。		

【注】 ※5 設定が不要な場合は、「uchValueInfo」の該当ビットを OFF することで省略可能です。データタイプに「R\_IN\_CAN\_VISIBLESTRING」を設定した場合は、設定不可です。



## (3) gerR\_IN\_CanGetObjectHandle

機能	Object Dictionary ハンドル取得			
呼出し形式	ERRCODE gerR_IN_CanGetObjectHandle (USHORT usObjectDictionaryNo, USHORT usIndex, R_IN_CAN_OD_T** pstObject, R_IN_CAN_ERROR_DETAIL_T* pstErrorDetail)			
引数	型名	変数名	内容	入出力
	USHORT	usObjectDictionaryNo	Object Dictionary 番号	入力
	USHORT	usIndex	Index	入力
	R_IN_CAN_OD_T**	pstObject	Object Dictionary の Index に対応する先頭アドレス 詳細は「表 6.40」を参照してください	出力
	R_IN_CAN_ERROR_DETAIL_T*	pstErrorDetail	エラー詳細情報 詳細は「表 6.39」を参照してください	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	指定された Index を基に、Object Dictionary の該当オブジェクトに対応する先頭アドレスを取得します。			

## (4) gerR\_IN\_CanReadObject

機能	Object Dictionary 読出し			
呼出し形式	ERRCODE gerR_IN_CanReadObject(USHORT usObjectDictionaryNo, USHORT usIndex, UCHAR uchSubIndex, USHORT usRequestDataSize, USHORT* pusDataSize, UCHAR* puchReadData, R_IN_CAN_ERROR_DETAIL_T* pstErrorDetail)			
引数	型名	変数名	内容	入出力
	USHORT	usObjectDictionaryNo	Object Dictionary 番号	入力
	USHORT	usIndex	Index	入力
	UCHAR	uchSubIndex	SubIndex	入力
	USHORT	usRequestDataSize	データ読出し指示サイズ	入力
	USHORT*	pusDataSize	データ読出しサイズ	出力
	UCHAR*	puchReadData	読出しデータ	出力
	R_IN_CAN_ERROR_DETAIL_T*	pstErrorDetail	エラー詳細情報 詳細は「表 6.39」を参照してください	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	<p>指定された Index と SubIndex を基に Object Dictionary の該当オブジェクトに格納されたデータを読み出します。</p> <p>読出し対象オブジェクトのデータタイプが「R_IN_CAN_VISIBLESTRING」の場合</p> <ul style="list-style-type: none"> <li>・データ読出しサイズは、0 または Object Dictionary 該当オブジェクトの有効なビット長(usBitLength)が示すバイトサイズを指定してください。</li> <li>・読出しデータの先頭から終端文字を検出するまで読み出しますので、読出しデータの終端は終端文字で埋めたデータを設定してください。</li> </ul> <p>読出し対象オブジェクトのデータタイプが「R_IN_CAN_VISIBLESTRING 以外」の場合</p> <ul style="list-style-type: none"> <li>・データ読出しサイズは、0 または Object Dictionary 該当オブジェクトのデータタイプ(usDataType)が示すサイズ(バイト)を指定してください。</li> </ul>			

## (5) gerR\_IN\_CanWriteObject

機能	Object Dictionary 書込み			
呼出し形式	ERRCODE gerR_IN_CanWriteObject(USHORT usObjectDictionaryNo, USHORT usIndex, UCHAR uchSubIndex, USHORT usDataSize, UCHAR* puchWriteData, R_IN_CAN_ERROR_DETAIL_T* pstErrorDetail)			
引数	型名	変数名	内容	入出力
	USHORT	usObjectDictionaryNo	Object Dictionary 番号	入力
	USHORT	usIndex	Index	入力
	UCHAR	uchSubIndex	SubIndex	入力
	USHORT	usDataSize	データ書込みサイズ	入力
	UCHAR*	puchWriteData	書込みデータ	入力
	R_IN_CAN_ERROR_DETAIL_T*	pstErrorDetail	エラー詳細情報 詳細は「表 6.39」を参照してください	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	<p>指定された Index と SubIndex を基に Object Dictionary の該当オブジェクトに指定されたデータを書き込みます。</p> <p>書込み対象オブジェクトのデータタイプが「R_IN_CAN_VISIBLESTRING」の場合</p> <ul style="list-style-type: none"> <li>・ SLMP のデータ書込みサイズは Object Dictionary 該当オブジェクトの有効なビット長(usBitLength)が示すサイズ(バイト)を指定してください。</li> <li>・ 文字列の終端以降には終端文字を設定してください。</li> </ul> <p>書込み対象オブジェクトのデータタイプが「R_IN_CAN_VISIBLESTRING 以外」の場合</p> <ul style="list-style-type: none"> <li>・ データ書込みサイズは、Object Dictionary 該当オブジェクトのデータタイプ(usDataType)が示すサイズ(バイト)を指定してください。</li> </ul>			

## (6) gerR\_IN\_CanReadBlockObject

機能	Object Dictionary ブロック 読出し			
呼出し形式	gerR_IN_CanReadBlockObject(USHORT usObjectDictionaryNo, USHORT usIndex, UCHAR uchSubIndex, USHORT usDataSize, UCHAR* puchReadData, R_IN_CAN_ERROR_DETAIL_T* pstErrorDetail);			
引数	型名	変数名	内容	入出力
	USHORT	usObjectDictionaryNo	Object Dictionary 番号	入力
	USHORT	usIndex	Index	入力
	UCHAR	uchSubIndex	SubIndex	入力
	USHORT	usDataSize	データ読出しサイズ	入力
	UCHAR*	puchReadData	読出しデータ	出力
	R_IN_CAN_ERROR_DETAIL_T*	pstErrorDetail	エラー詳細情報 詳細は「表 6.39」を参照してください	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	<p>指定された Index と SubIndex を基に Object Dictionary の該当オブジェクトから指定したサイズ分データを読み出します。</p> <p>SLMP フレームとして読出し可能なサイズは 500H(1280 バイト)です。</p> <p>読出し対象オブジェクトのデータタイプが「R_IN_CAN_VISIBLESTRING」の場合</p> <ul style="list-style-type: none"> <li>・データ読出しサイズは Object Dictionary 該当オブジェクトの有効なビット長(usBitLength)が示すサイズ(バイト)で算出してください。</li> <li>・読出しデータの先頭から終端文字を検出するまで読み出しますので、読出しデータの終端は終端文字で埋めたデータを設定してください。</li> </ul> <p>読出し対象オブジェクトのデータタイプが「R_IN_CAN_VISIBLESTRING 以外」の場合</p> <ul style="list-style-type: none"> <li>・データ読出しサイズは Object Dictionary 該当オブジェクトのデータタイプ(usDataType)が示すサイズ(バイト)で算出してください。</li> </ul>			

## (7) gerR\_IN\_CanWriteBlockObject

機能	Object Dictionary ブロック 書込み			
呼出し形式	ERRCODE gerR_IN_CanWriteBlockObject(USHORT usObjectDictionaryNo, USHORT usIndex, UCHAR uchSubIndex, USHORT usDataSize, UCHAR* puchWriteData, R_IN_CAN_ERROR_DETAIL_T* pstErrorDetail)			
引数	型名	変数名	内容	入出力
	USHORT	usObjectDictionaryNo	Object Dictionary 番号	入力
	USHORT	usIndex	Index	入力
	UCHAR	uchSubIndex	SubIndex	入力
	USHORT	usDataSize	データ書込みサイズ	入力
	UCHAR*	puchWriteData	書込みデータ	入力
	R_IN_CAN_ERROR_DETAIL_T*	pstErrorDetail	エラー詳細情報 詳細は「表 6.39」を参照してください	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	<p>指定された Index と SubIndex を基に Object Dictionary の該当オブジェクトから指定したサイズ分データを書き込みます。</p> <p>SLMP フレームとして書込み可能なサイズは 500H(1280 バイト)です。</p> <p>書込み対象オブジェクトのデータタイプが「R_IN_CAN_VISIBLESTRING」の場合</p> <ul style="list-style-type: none"> <li>SLMP のデータ書込みサイズは Object Dictionary 該当オブジェクトの有効なビット長(usBitLength)が示すサイズ(バイト)で算出してください。</li> <li>文字列の終端以降には終端文字を設定してください。</li> </ul> <p>書込み対象オブジェクトのデータタイプが「R_IN_CAN_VISIBLESTRING 以外」の場合</p> <ul style="list-style-type: none"> <li>データ書込みサイズは、Object Dictionary 該当オブジェクトのデータタイプ(usDataType)が示すサイズ(バイト)で算出してください。</li> </ul>			

## (8) gerR\_IN\_CanGetNmtState

機能	NMTState 取得			
呼出し形式	ERRCODE gerR_IN_CanGetNmtState(USHORT usObjectDictionaryNo, USHORT* pusCurrentNmtState, USHORT* pusMasterRequestedNmtState, R_IN_CAN_ERROR_DETAIL_T* pstErrorDetail)			
引数	型名	変数名	内容	入出力
	USHORT	usObjectDictionaryNo	Object Dictionary 番号	入力
	USHORT*	pusCurrentNmtState	現在の NMTState	出力
	USHORT*	pusMasterRequestedNmtState	マスタ局から受信した NMTState	出力
	R_IN_CAN_ERROR_DETAIL_T*	pstErrorDetail	エラー詳細情報 詳細は「表 6.39」を参照してください	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	<p>現在の NMTState とマスタ局が指定した NMTState を取得します。</p> <p>マスタ局が指定した NMTState がない場合、Init を取得します。</p>			

## (9) gerR\_IN\_CanSetNmtState

機能	NMTState 設定			
呼出し形式	ERRCODE gerR_IN_CanSetNmtState(USHORT usObjectDictionaryNo, USHORT usMasterRequestNmtState, R_IN_CAN_ERROR_DETAIL_T* pstErrorDetail)			
引数	型名	変数名	内容	入出力
	USHORT	usObjectDictionaryNo	Object Dictionary 番号	入力
	USHORT	usMasterRequestedNmtState	マスタ局から受信した NMTState	入力
	R_IN_CAN_ERROR_DETAIL_T*	pstErrorDetail	エラー詳細情報 詳細は「表 6.39」を参照してください	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	指定された NMTState を R-IN32M4-CL3 ドライバへ設定します。			

## (10) gerR\_IN\_CanUpdateRPDO

機能	RPDO 更新			
呼出し形式	ERRCODE gerR_IN_CanUpdateRPDO(R_IN_CAN_ERROR_DETAIL_T* pstErrorDetail)			
引数	型名	変数名	内容	入出力
	R_IN_CAN_ERROR_DETAIL_T*	pstErrorDetail	エラー詳細情報 詳細は「表 6.39」を参照してください	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR_NODATA : 異常完了(受信データなし) R_IN_ERR : 異常完了			
説明	PDO マッピング設定に従い、受信した RWw のデータを Object Dictionary へ反映します。			

## (11) gerR\_IN\_CanUpdateTPDO

機能	TPDO 更新			
呼出し形式	ERRCODE gerR_IN_CanUpdateTPDO(R_IN_CAN_ERROR_DETAIL_T* pstErrorDetail)			
引数	型名	変数名	内容	入出力
	R_IN_CAN_ERROR_DETAIL_T*	pstErrorDetail	エラー詳細情報 詳細は「表 6.39」を参照してください	出力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_ERR_NODATA : 異常完了(CCIETSN ClassA 動作時に送信データ設定不要) R_IN_ERR_PDO_MAPPING : 異常終了(PDO マッピング異常)			
説明	<p>PDO マッピング設定に従い、Object Dictionary のデータを RWr へ反映します。</p> <p>CC-Link IE TSN Class A 動作時にサイクリックデータを受信していない場合、戻り値で「R_IN_ERR_NODATA」を返します。</p> <p>PDO マッピング設定が未完了の場合は 0 データを RWr へ反映し、戻り値で「R_IN_ERR_PDO_MAPPING」を返します。この際に CC Link IE TSN Class A での動作時には、サイクリックフレーム送信フラグを「TRUE」に設定することで、0 データを送信することができます。</p>			

## 6.4.17 MCU 間 I/F

## (1) gvR\_IN\_SchedulerMain

機能	スケジューラタスクメイン			
呼出し形式	VOID gvR_IN_SchedulerMain(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	R-IN32M4-CL3 ドライバの「スケジューラタスク提供タイマ処理」を実行します。			

## (2) gvR\_IN\_MculfGpioSendMain

機能	GPIO 通信送信タスクメイン			
呼出し形式	VOID gvR_IN_MculfGpioSendMain(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	R-IN32M4-CL3 ドライバの「イベント通知送信メイン処理」を実行します。			

## (3) gvR\_IN\_MculfGpioResponseMain

機能	GPIO 通信応答受信タスクメイン			
呼出し形式	VOID gvR_IN_MculfGpioResponseMain(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	R-IN32M4-CL3 ドライバの「イベント通知応答受信処理」を実行します。			

## (4) gvR\_IN\_MculfGpioReceivedMain

機能	GPIO 通信受信タスクメイン			
呼出し形式	VOID gvR_IN_MculfGpioReceivedMain(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	R-IN32M4-CL3 ドライバの「イベント通知受信メイン処理」を実行します。			

## (5) gvR\_IN\_MculfRtDmaCompleteMain

機能	RT DMA 転送完了タスクメイン			
呼出し形式	VOID gvR_IN_MculfRtDmaCompleteMain(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	R-IN32M4-CL3 ドライバの「RT DMA 転送完了タスクメイン処理」を実行します。			

## (6) gvR\_IN\_McufInitial

機能	MCU 間 I/F イニシャル			
呼出し形式	VOID gvR_IN_McufInitial(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	R-IN32M4-CL3 ドライバの下記処理を実行します。 <ul style="list-style-type: none"> <li>・メモリブロック管理機能イニシャル</li> <li>・MCU 間 I/F 機能の初期化</li> <li>・スケジューラタスク提供タイマ機能イニシャル処理</li> </ul>			

## (7) gerR\_IN\_McufSetCommand

機能	MCU 間 I/F 登録処理			
呼出し形式	ERRCODE gerR_IN_MculfSetCommand (R_IN_MCU_IF_GPIO_MANAGEMENT_TBL_T* pstMculfMng, R_IN_MCU_IF_RTDMA_FUNCTION_TBL_T* pstRtDmaFunction)			
引数	型名	変数名	内容	入出力
	R_IN_MCU_IF_GPIO_MANAGEMENT_TBL_T*	pstMculfMng	MCU 間 I/F 管理テーブル 詳細は「表 6.42」を参照してください。	入力
	R_IN_MCU_IF_RTDMA_FUNCTION_TBL_T*	pstRtDmaFunction	シリアル通信完了時実行関数テーブル 詳細は「表 6.45」を参照してください。	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	引数で指定された関数を下位層に展開します。			
	No	R-IN32M4-CL3 ドライバの I/F 関数を登録する関数		実行時に設定する引数
	1	MCU 間通信時のコールバック関数設定		MCU 間 I/F 管理ポインタ
	2	RT DMA 転送完了コールバック関数設定		RT DMA 受信完了時コール関数

表 6.42 R\_IN\_MCU\_IF\_GPIO\_MANAGEMENT\_TBL\_T

No	メンバ		内容
1	USHORT	usRequestNumber	要求コマンド登録数
2	USHORT	usResponseNumber	応答コマンド登録数
3	R_IN_MCU_IF_GPIO_FUNCTION_REQUEST_TBL_T*	pstRequest	GPIO 通信要求受信時の実行関数テーブル 外部 MCU から通知を検出した際に実行する コマンド関数を定義します。 詳細は「表 6.43」を参照してください。
4	CONST R_IN_MCU_IF_GPIO_FUNCTION_RESPONSE_TBL_T*	pstResponse	GPIO 通信応答受信時の実行関数テーブル 外部 MCU に通知を設定した際に実行結果を 受け付ける関数を定義します。 詳細は「表 6.44」を参照してください。
5	CONST R_IN_MCU_IF_ACK_GPIO_SET_FUNCTION	fpvAckGpioSet	GPIO 通信受付処理の関数ポインタ
6	R_IN_MCU_IF_ACK_GPIO_CLEAR_FUNCTION	fpvAckGpioClear	GPIO 通信受付クリア処理の関数ポインタ
7	R_IN_MCU_IF_REQUEST_GPIO_SET_FUNCTION	ferRequestGpioSet	GPIO 通信イベント種別/コード設定処理の関数ポインタ

表 6.43 R\_IN\_MCU\_IF\_GPIO\_FUNCTION\_REQUEST\_TBL\_T

No.	メンバ		内容
1	USHORT	usKind	イベント種別
2	USHORT	usCode	イベントコード
3	R_IN_MCU_IF_REQUEST_FUNCTION	fpvFunction	コマンド関数のポインタ※1

【注】 ※1 デフォルトでは「UserMculfSafetyPduTransfer」および「UserMculfSafetyPduReceived」を定義しています。関数の詳細は「5.10.5」および「5.10.6」を参照してください。

表 6.44 R\_IN\_MCU\_IF\_GPIO\_FUNCTION\_RESPONSE\_TBL\_T

No.	メンバ		内容
1	USHORT	usKind	イベント種別
2	USHORT	usCode	イベントコード
3	R_IN_MCU_IF_RESPONSE_FUNCTION	fpvFunction	コマンド関数のポインタ※1

【注】 ※1 デフォルトでは「UserMculfSafetyPduTransferResponse」および「UserMculfSafetyPduReceivedResponse」を定義しています。関数の詳細は「5.10.7」および「5.10.8」を参照してください。

表 6.45 R\_IN\_MCU\_IF\_RTDMA\_FUNCTION\_TBL\_T

No.	メンバ		内容
1	R_IN_MCU_IF_RTDMA_RESERVED_FUNCTION	fpvReceivedFunction	コマンド関数のポインタ※1
2	R_IN_MCU_IF_RTDMA_SEND_FUNCTION	fpvSendFunction	コマンド関数のポインタ※2

【注】 ※1 デフォルトでは「UserMculfSafetyPduReceivedComplete」を定義しています。関数の詳細は「5.10.9」を参照してください。

※2 デフォルトでは未定義です。(NULL)



## (8) gerR\_IN\_MculfCheckTransferRtDma

機能	RT DMA 転送要求チェック			
呼出し形式	ERRCODE gerR_IN_MculfCheckTransferRtDma(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了 R_IN_BUSY : 処理中			
説明	RT DMA の転送要求をチェックします。			

## (9) gerR\_IN\_MculfStartSendCsih

機能	DMA 転送準備(送信)			
呼出し形式	ERRCODE gerR_IN_MculfStartSendCsih(CONST VOID* pSrc,ULONG ulLen)			
引数	型名	変数名	内容	入出力
	CONST VOID*	pSrc	送信元アドレス	入力
	ULONG	ulLen	送信データ長	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	引数で指定されたエリアをシリアル送信するために、RT DMA と CSIH の設定を行います。			

## (10) gerR\_IN\_MculfStartReceiveCsih

機能	DMA 転送準備(受信)			
呼出し形式	ERRCODE gerR_IN_MculfStartReceiveCsih(VOID* pDst,ULONG ulLen)			
引数	型名	変数名	内容	入出力
	VOID*	pDst	受信データ格納先アドレス	入力
	ULONG	ulLen	受信データ長	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	シリアル受信するデータを引数指定のエリアへ格納するために、RT DMA と CSIH の設定を行います。			

## (11) gerR\_IN\_MculfStopReceiveCsih

機能	CSIH 受信停止			
呼出し形式	ERRCODE gerR_IN_MculfStopReceiveCsih(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	CSIH の受信動作を停止します。			

## (12) gerR\_IN\_McufStopSendCsih

機能	CSIH 送信停止			
呼出し形式	ERRCODE gerR_IN_McufStopSendCsih(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	CSIH の送信動作を停止します。			

## (13) gerR\_IN\_McufCheckGpioState

機能	GPIO 通信使用状態確認			
呼出し形式	ERRCODE gerR_IN_McufCheckGpioState(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 (空き) R_IN_BUSY : 処理中 (使用中)			
説明	GPIO 通信の使用状態を呼び元に返却します。			

## (14) gerR\_IN\_McufGetGpioState

機能	GPIO 通信使用状態取得			
呼出し形式	ERRCODE gerR_IN_McufGetGpioState(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	R_IN_OK : 正常完了 R_IN_BUSY : 処理中 (使用中)			
説明	GPIO 通信の使用状態を「空き」→「使用中」に変更します。			

## (15) gvR\_IN\_McufClearGpioState

機能	GPIO 通信使用状態クリア			
呼出し形式	VOID gvR_IN_McufClearGpioState(VOID)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	GPIO 通信の使用状態を「空き」に設定します。			

## (16) gerR\_IN\_McufGpioSend

機能	GPIO 通信データ作成要求			
呼出し形式	ERRCODE gerR_IN_McufGpioSend(UCHARuchKind ,UCHARuchEvtCode)			
引数	型名	変数名	内容	入出力
	-	-	-	-
戻り値	-			
説明	GPIO 通信送信タスクにメールボックス送信を行います。			

## 6.5 R-IN32M4-CL3 ドライバコールバック関数一覧

R-IN32M4-CL3 ドライバコールバック関数の一覧を以下に示します。

表 6.46 R-IN32M4-CL3 ドライバコールバック関数一覧

No.	機能分類	関数名	関数型	概要
1	Fatal エラー管理	gR_IN_CallbackFatalError	VOID	R-IN32M4-CL3 の Fatal エラー取得
2	SLMP 受信	gerR_IN_CallbackReceivedSimp	ERRCODE	SLMP フレーム取得
3		gvR_IN_CallbackNotifyReceivedSimp	VOID	SLMP フレーム受信通知
4	サイクリック送信 データ初期化	gulR_IN_CallbackInitSendCyclicData	ULONG	サイクリック送信データ初期化
5	データリンク高速化	gulR_IN_CallbackTimeSyncComplete	ULONG	データリンク高速化(時刻同期完了時)
6		gulR_IN_CallbackCyclicStart	ULONG	データリンク高速化(サイクリック伝送開始時)
7		gulR_IN_CallbackDisconnectPartwayThorough	ULONG	データリンク高速化(途中解列時)
8	ネットワーク同期 通信	gulR_IN_CallbackCheckComCycle	ULONG	通信周期判定
9		gulR_IN_CallbackSyncCom	ULONG	同期サイクリック通信処理
10	IP アドレス変更	gvR_IN_CallbackIPAddressChangingResult	VOID	IP アドレス変更結果

## 6.6 R-IN32M4-CL3 ドライバコールバック関数詳細

R-IN32M4-CL3 ドライバコールバック関数の内部処理は、必要に応じてカスタマイズしてください。

R-IN32M4-CL3 ドライバコールバック関数の詳細を以下に示します。

### (1) gR\_IN\_CallbackFatalError

機能	R-IN32M4-CL3 の Fatal エラー取得			
呼出し形式	VOID gR_IN_CallbackFatalError (ULONG ulErrorCode, ULONG ulErrorInfo)			
引数	型名	変数名	内容	入出力
	ULONG	ulErrorCode	Fatal エラーコード	入力
	ULONG	ulErrorInfo	Fatal エラー情報(エラー発生時関数のアドレス)	入力
戻り値	-			
説明	R-IN32M4-CL3 ドライバからの Fatal エラー情報を取得するためのコールバック関数です。			

### (2) gerR\_IN\_CallbackReceivedSlmp

機能	SLMP フレーム取得			
呼出し形式	ERRCODE gerR_IN_CallbackReceivedSlmp (const VOID* pvFrameAddress, USHORT usFrameSize, ULONG ullpAddress, USHORT usPort)			
引数	型名	変数名	内容	入出力
	const VOID*	pvFrameAddress	フレーム先頭アドレス	入力
	USHORT	usFrameSize	フレームサイズ	入力
	ULONG	ullpAddress	送信元 IP アドレス	入力
	USHORT	usPort	送信元ポート番号	入力
戻り値	R_IN_OK : 正常完了 R_IN_ERR : 異常完了			
説明	<p>受信バッファ、受信サイズ、送信元 IP アドレス、送信元ポート番号を基に SLMP フレームの取得有無を判定し、引数の戻り値に設定します。</p> <p>IP フレームを使用した SLMP の場合は、送信元 IP アドレスと送信元ポート番号が設定されます。 AcyclicData フレームを使用した SLMP の場合は、送信元 IP アドレスと送信元ポート番号に 0 が設定されます。</p> <p>SLMP フレームを受信した際、SLMP 実行関数テーブルに登録しているコマンド関数が応答処理を行う前のタイミングで、本関数が呼び出されます。</p> <p>特別な理由（たとえば SLMP フレームを一切取得しないなど）が無い限り、カスタマイズは不要です。</p>			

## (3) gvR\_IN\_CallbackNotifyReceivedSmp

機能	SLMP フレーム受信通知			
呼出し形式	VOID gvR_IN_CallbackNotifyReceivedSmp(const VOID* pvFrameAddress, USHORT usFrameSize, ULONG ullpAddress, USHORT usPort)			
引数	型名	変数名	内容	入出力
	const VOID*	pvFrameAddress	フレーム先頭アドレス	入力
	USHORT	usFrameSize	フレームサイズ	入力
	ULONG	ullpAddress	送信元 IP アドレス	入力
	USHORT	usPort	送信元ポート番号	入力
戻り値	-			
説明	<p>受信した SLMP フレームの先頭アドレス、受信サイズを通知します。</p> <p>IP フレームを使用した SLMP の場合は、送信元 IP アドレスと送信元ポート番号が設定されます。</p> <p>AcyclicData フレームを使用した SLMP の場合は、送信元 IP アドレスと送信元ポート番号に 0 が設定されます。</p> <p>SLMP フレームを受信した際、SLMP 実行関数テーブルにおいて該当のコマンド関数が NULL の場合、本関数が呼び出されます。</p> <p>未対応コマンドの SLMP フレームを受信した場合、必ずしも応答を送信する必要はありません。応答を送信する場合は、「gerR_IN_SetSmpResponseFrameSendRequest」(6.4.10(9))を使用して送信してください。また、処理後は次の SLMP フレームを受信できるように、「gvR_IN_ReleaseSmpReceiveFrame」(6.4.10(10))を実行してください。</p>			

## (4) guR\_IN\_CallbackInitSendCyclicData

機能	サイクリック伝送開始時送信データ初期化			
呼出し形式	ULONG guR_IN_CallbackInitSendCyclicData(ULONG ulParam1, ULONG ulParam2)			
引数	型名	変数名	内容	入出力
	ULONG	ulParam1	パラメータ 1 (0 固定)	入力
	ULONG	ulParam2	パラメータ 2 (0 固定)	入力
戻り値	0 固定			
説明	<p>解列から復列した直後のサイクリック送信データは解列直前のデータが送信されることがあります。任意のデータを送信したい場合に、本コールバック処理内でサイクリック送信データを設定してください。</p> <p>サイクリック送信データを設定する際は、「UserSendCyclic」(5.4.3 サイクリック送信処理)で使用している R-IN32M4-CL3 ドライバインタフェース関数(一括・個別・高速)と同じ関数を使用してください。本コールバック処理は、ディスパッチ禁止状態で実行されます。</p> <p>「gerR_IN_RegistCallback」(6.4.1(8))にて本コールバック処理を登録することで、解列から復列したタイミングで本処理がコールバックされます。</p> <p>(コールバック処理タイプ : R_IN_FUNCTIONTYPE_CYCLICDATA_INITIALIZE (1))</p>			

## (5) gulR\_IN\_CallbackTimeSyncComplete

機能	データリンク高速化(時刻同期完了時)			
呼出し形式	ULONG gulR_IN_CallbackTimeSyncComplete(ULONG ulParam1, ULONG ulParam2)			
引数	型名	変数名	内容	入出力
	ulParam1	ulParam1	パラメータ 1 (0 固定)	入力
	ULONG	ulParam2	パラメータ 2 (0 固定)	入力
戻り値	0 固定			
説明	<p>ユーザタスク(TSKID_PERIODIC)の優先度を低優先割込みタスク(TSKID_NX_LOW_INT)より高く設定することで、データリンクの開始に影響<sup>※1</sup>を及ぼす場合があります。その場合の回避策<sup>※1</sup>として、本コールバック処理を使用してください。</p> <p>なお、タスク優先度がデフォルトのままの場合、データリンクの開始に影響を及ぼすことはありません。  ユーザタスクのデフォルト優先度：3  低優先割込みタスクのデフォルト優先度：2</p> <p>本コールバック処理は、ディスパッチ禁止状態で実行されます。  gulR_IN_RegistCallback(6.4.1(8))にて本コールバック処理を登録することで、時刻同期完了時に、本関数がコールバックされます。  (コールバック処理タイプ：R_IN_FUNCTIONTYPE_TIMESYNC_COMPLETE (0))</p>			

【注】※1：影響と回避策の詳細は、「データリンク高速化の補足」に示す、ユーザタスクの優先度を低優先割込みタスクより高く設定した場合の影響(a)と、その回避策(b)を参照してください。

## (6) gulR\_IN\_CallbackCyclicStart

機能	データリンク高速化(サイクリック伝送開始時)			
呼出し形式	ULONG gulR_IN_CallbackCyclicStart(ULONG ulParam1, ULONG ulParam2)			
引数	型名	変数名	内容	入出力
	ULONG	ulParam1	パラメータ 1 (0 固定)	入力
	ULONG	ulParam2	パラメータ 2 (0 固定)	入力
戻り値	0 固定			
説明	<p>ユーザタスク(TSKID_PERIODIC)の優先度を低優先割込みタスク(TSKID_NX_LOW_INT)より高く設定することで、データリンクの開始に影響<sup>※1</sup>を及ぼす場合があります。その場合の回避策<sup>※1</sup>として、本コールバック処理を使用してください。</p> <p>なお、タスク優先度がデフォルトのままの場合、データリンクの開始に影響を及ぼすことはありません。  ユーザタスクのデフォルト優先度：3  低優先割込みタスクのデフォルト優先度：2</p> <p>本コールバック処理は、ディスパッチ禁止状態で実行されます。  「gulR_IN_RegistCallback」(6.4.1(8))にて本コールバック処理を登録することで、サイクリック伝送開始時に、本関数がコールバックされます。  (コールバック処理タイプ：R_IN_FUNCTIONTYPE_CYCLIC_START (2))</p>			

【注】※1：影響と回避策の詳細は、「データリンク高速化の補足」に示す、ユーザタスクの優先度を低優先割込みタスクより高く設定した場合の影響(a)と、その回避策(b)を参照してください。

## (7) gulR\_IN\_CallbackDisconnectPartwayThrough

機能	データリンク高速化(途中解列時)			
呼出し形式	ULONG gulR_IN_CallbackDisconnectPartwayThrough(ULONG ulParam1, ULONG ulParam2)			
引数	型名	変数名	内容	入出力
	ULONG	ulParam1	パラメータ 1 (0 固定)	入力
	ULONG	ulParam2	パラメータ 2 (0 固定)	入力
戻り値	0 固定			
説明	<p>ユーザタスク(TSKID_PERIODIC)の優先度を低優先割込みタスク(TSKID_NX_LOW_INT)より高く設定することで、データリンクの開始に影響<sup>※1</sup>を及ぼす場合があります。その場合の回避策<sup>※1</sup>として、本コールバック処理を使用してください。</p> <p>なお、タスク優先度がデフォルトのままの場合、データリンクの開始に影響を及ぼすことはありません。  ユーザタスクのデフォルト優先度 : 3  低優先割込みタスクのデフォルト優先度 : 2</p> <p>本コールバック処理は、ディスパッチ禁止状態で実行されます。  「gerR_IN_RegistCallback」(6.4.1(8))にて本コールバック処理を登録することで、イニシャル処理の途中で解列した時に、本関数がコールバックされます。  (コールバック処理タイプ : R_IN_FUNCTIONTYPE_DISCONNECT_PARTWAY_THROUGH (3))</p>			

【注】※1 : 影響と回避策の詳細は、「データリンク高速化の補足」に示す、ユーザタスクの優先度を低優先割込みタスクより高く設定した場合の影響(a)と、その回避策(b)を参照してください。

## 【データリンク高速化の補足】

R-IN32M4-CL3 ドライバは、データリンクを開始するために必要な処理(マスタ局から受信した情報の設定処理や時刻同期処理など)を、低優先割込みタスク(TSKID\_NX\_LOW\_INT)にて行っています。そのため、ユーザタスク(TSKID\_PERIODIC)の優先度を低優先割込みタスクより高く設定することで、自局のデータリンク開始に影響を及ぼす場合があります。

以降に、ユーザタスクの優先度を低優先割込みタスクより高く設定した場合の影響(a)と、その回避策(b)を示します。

## (a) 影響

## ・途中解列が発生しない場合

優先度の高いユーザタスク処理が開始されると、低優先割込みタスク処理は待ち状態となり、ユーザタスク処理の終了後に処理が継続されます。このため、処理開始から終了までの実質的な処理時間が延びることになります。

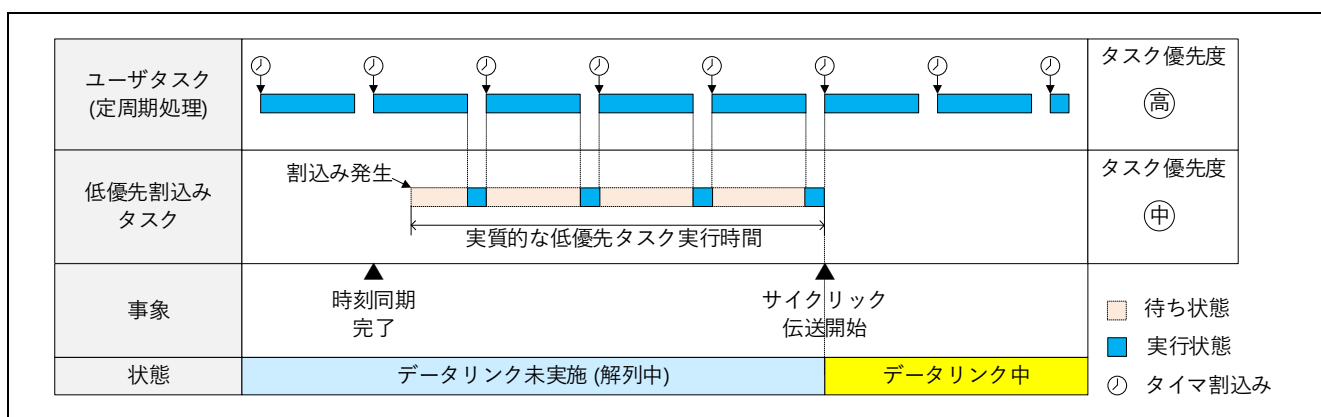


図 6.4 ユーザタスクの優先度が高い場合の問題点(途中解列が発生しない場合)

## ・途中解列が発生した場合

マスタ局は、デバイス局を検出した後、当該局からのサイクリックデータ受信までを監視しています。監視時間のタイムアウトが発生すると、マスタ局により当該局が途中解列したと判断され、マスタ局が初期化シーケンスを先頭から再開します。途中解列発生時は、時刻同期処理、およびマスタ局からのデータリンクに必要な各種設定情報を受信する処理を、改めて行うため、さらにサイクリック伝送開始までの時間が延びます。

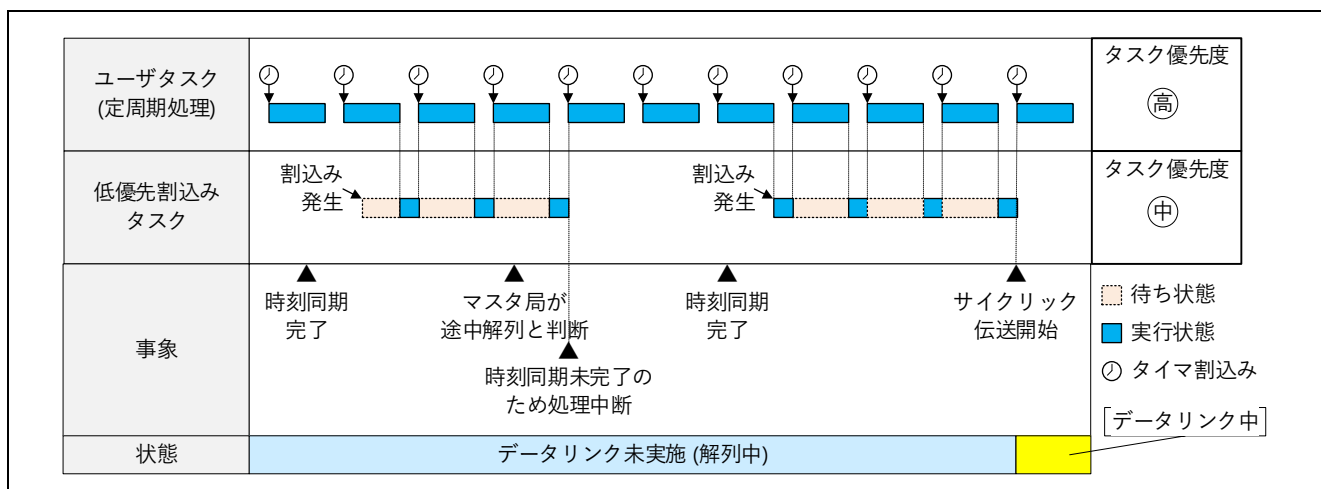


図 6.5 ユーザタスクの優先度が高い場合の問題点(途中解列が発生した場合)



## (b) 回避策

- ・途中解列が発生しない場合

ユーザタスクの優先度を一時的に下げること、低優先割込みタスク実行時におけるユーザタスクの処理は、待ち状態となるため、低優先割込みタスクの処理時間の遅延が発生しません。

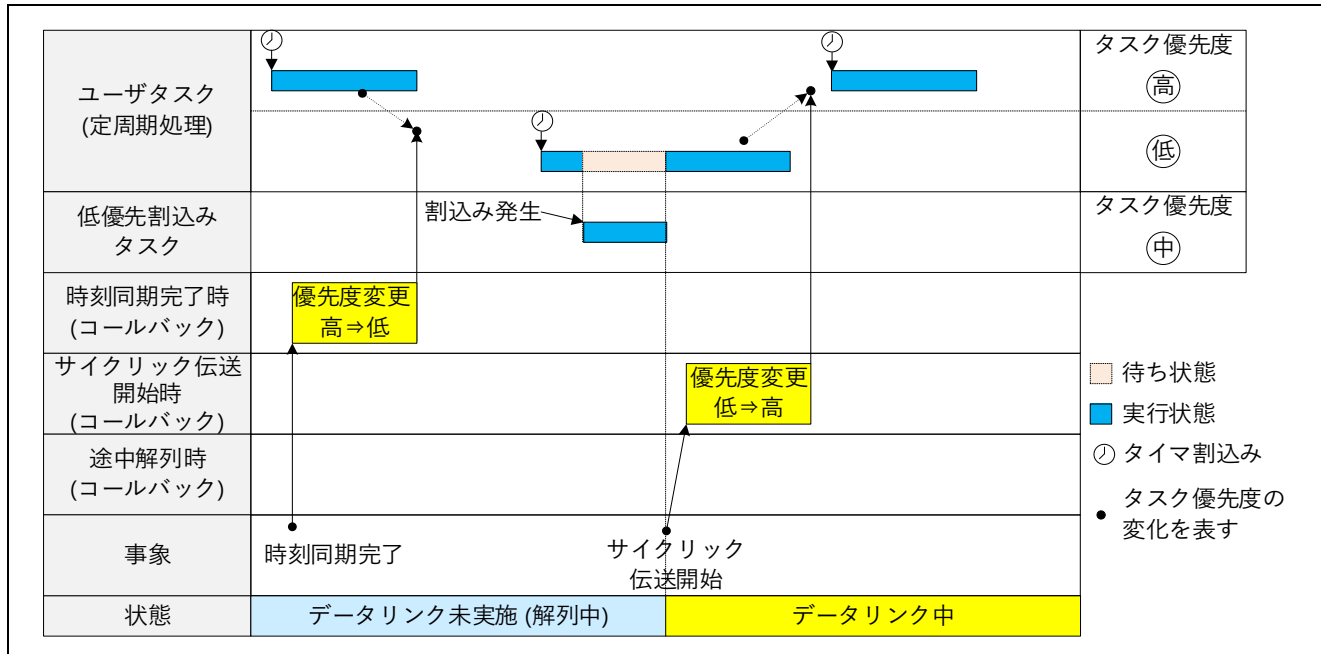


図 6.6 ユーザタスクの優先度を一時的に下げた場合の効果(途中解列が発生しない場合)

【gulR\_IN\_CallbackTimeSyncComplete】

RTOS のサービスコール(chg\_pri)を用いて、優先度変更対象タスクの優先度を、低優先割込みタスクより低い値に一時的に変更します。

【gulR\_IN\_CallbackCyclicStart】

RTOS のサービスコール(chg\_pri)を用いて、一時的に変更したタスク優先度を、再び高優先度に戻します。

・途中解列が発生した場合

時刻同期完了のタイミングでユーザタスクの優先度を一時的に下げた状態で、データリンク開始前に途中解列が発生すると、優先度が下がったまま次の時刻同期完了を待つことになるため、途中解列処理にてタスクの優先度を元に戻します。その後、改めて、次の時刻同期完了からデータリンク開始を待ちます。

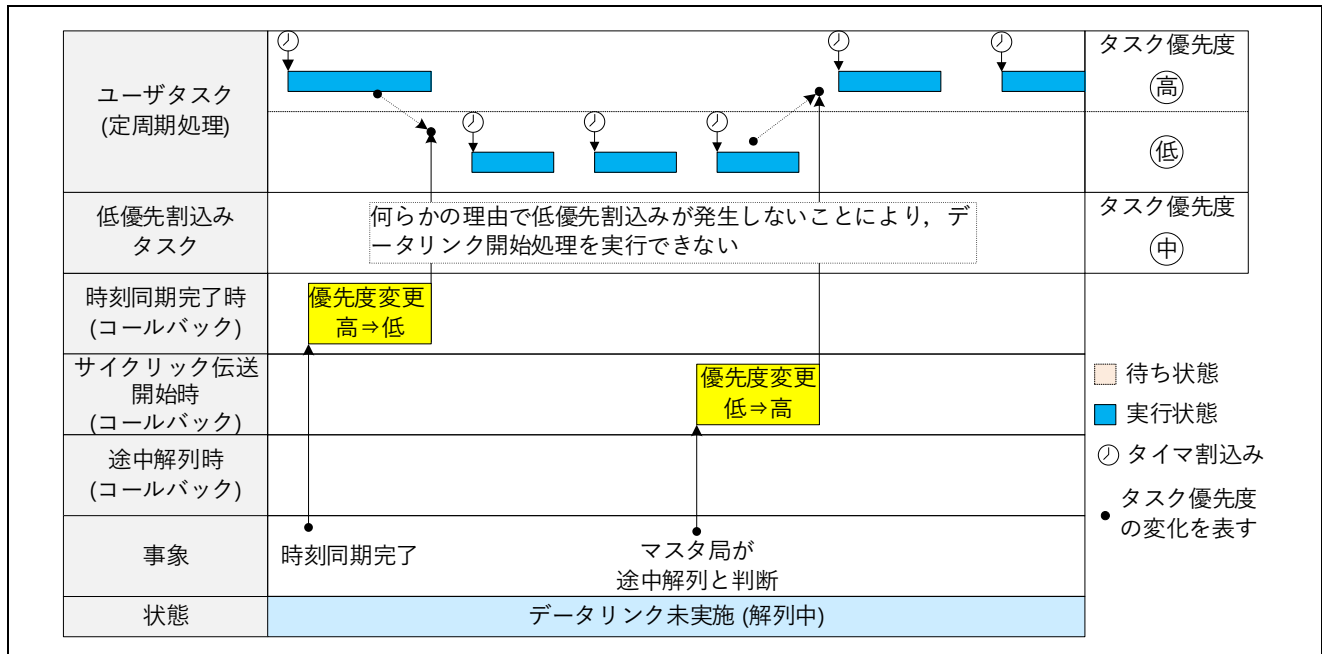


図 6.7 ユーザタスクの優先度を一時的に下げた場合の効果(途中解列が発生した場合)

【gulR\_IN\_CallbackTimeSyncComplete】

RTOS のサービスコール(chg\_pri)を用いて、優先度変更対象タスクの優先度を、低優先割込みタスクより低い値に一時的に変更します。

【gulR\_IN\_CallbackDisconnectPartwayThrough】

RTOS のサービスコール(chg\_pri)を用いて、一時的に変更したタスク優先度を、再び高優先度に戻します。

## (8) gulR\_IN\_CallbackCheckComCycle

機能	通信周期判定			
呼出し形式	ULONG gulR_IN_CallbackCheckComCycle(ULONG ulParam1, ULONG ulParam2)			
引数	型名	変数名	内容	入出力
	ULONG	ulParam1	通信周期情報の先頭アドレス 詳細は「表 6.47」を参照してください。	入力
	ULONG	ulParam2	パラメータ 2 (0 固定)	入力
戻り値	判定結果 0 : 正常 1 : 異常			
説明	<p>CC-Link IE TSN プロトコル初期化フェーズにおいて、マスタ局から通信周期情報を受信し、ネットワーク同期通信が指定されたタイミングで処理が実行されます。通信周期が機器側で対応可能であるかを判断し、判定結果を返します。ユーザアプリケーションで通信周期情報に問題がある場合は、関数の戻り値に 0 以外を返してください。</p> <p>本コールバック処理は、ディスパッチ禁止状態で実行されます。</p> <p>「gerR_IN_RegistCallback」(6.4.1(8))にて本コールバック処理を登録することで、本関数がコールバックされます。ネットワーク同期通信を行わない場合はコールバックされません。</p> <p>(コールバック処理タイプ : R_IN_FUNCTIONTYPE_COMCYLCE_DEFINITION (4))</p>			

表 6.47 R\_IN\_COMCYCLE\_INFO\_T 一覧

No.	メンバ		内容
1	UCHAR	uchTimeslotNum	タイムスロット数
2	UCHAR	uchReserve	リザーブ
3	USHORT	usRepetitionCount	通信周期繰り返し回数
4	ULONGLONG	aulTsStartTime [R_IN_TIMESLOT_MAX]	TS0～TS7 の開始時刻(ns)
5	ULONGLONG	aulTsEndTime [R_IN_TIMESLOT_MAX]	TS0～TS7 の終了時刻(ns)
6	ULONGLONG	ullComCycle	通信周期(ns)

## (9) gulR\_IN\_CallbackSyncCom

機能	同期サイクリック通信処理			
呼出し形式	ULONG gulR_IN_CallbackSyncCom(ULONG ulParam1, ULONG ulParam2)			
引数	型名	変数名	内容	入出力
	ULONG	ulParam1	パラメータ 1 (0 固定)	入力
	ULONG	ulParam2	パラメータ 2 (0 固定)	入力
戻り値	コールバック実行結果 (0 固定)			
説明	<p>ネットワーク同期通信実行時、サイクリック伝送が完了したタイミング(タイムスロット 2 開始のタイミング)で処理が実行されます。受信サイクリックデータに対する処理、次回送信するサイクリックデータの準備などを行います。</p> <p>本コールバック処理は、ディスパッチ禁止状態で実行されます。処理に時間を要する場合は、別タスクで実行するなどの対策を検討してください。</p> <p>「gerR_IN_RegistCallback」(6.4.1(8))にて本コールバック処理を登録することで、本関数がコールバックされます。ネットワーク同期通信を行わない場合はコールバックされません。</p> <p>(コールバック処理タイプ : R_IN_FUNCTIONTYPE_SYNC_CYCLIC_COM (5))</p>			

以下に同期サイクリック通信処理の処理概要を示します。

表 6.48 同期サイクリック通信処理の処理概要

No.	項目	内容
i	送信用 WDC 正常値のラッチ	「gvR_IN_LatchWdcUL」(6.4.15(6))を実行し、同期タイミング処理でインクリメントしている送信用 WDC の正常値をラッチします。
ii	サイクリックデータを受信	「UserReceiveCyclic」(5.4.1)を使用して、サイクリック受信データを取得します。
iii	受信 WDC の検査	受信したサイクリックデータから WDC 値を取り出し、「gerR_IN_CheckWdcDL」(6.4.15(8))にて、受信サイクリックデータの成否を判定します。チェック結果が正常でなければ、受信サイクリックデータを破棄します。 CANopen 通信時は、有効な ObjectDictionary 数分の WDC 値をチェックします。
iv	解列および同期異常のチェック	「gerR_IN_GetSyncDeviationFlag」(6.4.15(1))を実行し、時刻同期外れが発生していないかを確認します。 「gerR_IN_GetCommunicationStatus」(6.4.6(4))を使用し、自局が解列していないかを確認します。時刻同期外れが発生している、または自局解列状態の場合、例えば「gerR_IN_StopAppSyncSignal」(6.4.15(2))を実行することで同期タイミング処理を止めるなど、アプリケーション依存の処理を実行します。  【Ver.1.08J 以前】 StsW(自局状態レジスタ)のアプリケーション情報が「非同期」であるかを判定するため、システム起動後から一度でも「時刻同期外れ検出済み」(R_IN_TRUE)を取得した場合、グローバル変数「時刻同期外れ検出済みフラグ」で保持します。 <sup>※1</sup>  【Ver.1.09K 以降】 StsW(自局状態レジスタ)のアプリケーション情報は、マスタ局とネットワーク同期通信中であれば「同期中」となり、「gerR_IN_StopAppSyncSignal」(6.4.15(2))を実行すると「非同期」となります。 これにより、時刻同期外れ検出時に(gerR_IN_StopAppSyncSignal)(6.4.15(2))を実行しないことで、StsW(自局状態レジスタ)のアプリケーション情報は「同期中」を継続されるようになり、時刻同期外れ検出時でもネットワーク同期通信を継続するようなアプリケーションを実装することができます。
v	WDC UL の作成	「gerR_IN_GetWdcUL」(6.4.15(7))を実行し、サイクリックデータとしてマスタ局へ送信するウォッチドッグカウンタ UL の値を取得します。
vi	送信サイクリックデータ設定	送信サイクリックデータを設定し、作成したウォッチドッグカウンタ UL をサイクリックデータに埋め込みます。
vii	自局状態の送信	自局状態をサイクリックデータにて送信します。
viii	サイクリックデータの送信	「UserSendCyclic」(5.4.3)を使用して、作成したサイクリック送信データを送信します。
ix	自局状態とサイクリック伝送状態の更新	自局の通信状態およびサイクリック伝送状態を更新します。

【注】※1 R-IN32M4-CL3 ドライバ内で作成している StsW(自局状態レジスタ)のアプリケーション情報は、システム起動から一度でも時刻同期外れを検出していると「非同期」となります。一方、「gerR\_IN\_GetSyncDeviationFlag」(6.4.15(1))は、時刻同期外れ検出後に解列→復列が発生すると、「時刻同期外れ検出なし」(R\_IN\_FALSE)として取得されます。

同期サイクリック通信処理(本項)と同期タイミング処理(5.2.5 項)について、サンプルコードで想定する動作とそのタイミングチャートを示します。

(a) 同期サイクリック通信処理・同期タイミング処理の動作タイミングチャート 1

(同期タイミング処理がタイムスロット 1 の区間内に終了するケース)

① 同期サイクリック通信処理の動作

サイクリック送受信に関する一連の処理を実行します。(「表 6.48 同期サイクリック通信処理の処理概要」参照)

このとき、同期タイミング処理でインクリメントしている送信 WDC の値をラッチし、ラッチした値から WDC\_UL を作成し、サイクリックデータに格納します。

② 同期タイミング処理の動作

他のデバイス局とタイミングを揃えたいユーザアプリケーションの処理を実行します。サイクリック受信データをもとにした内部処理、およびサイクリック送信データを作成します。また、同期周期毎に送信 WDC の正常値をインクリメントします。

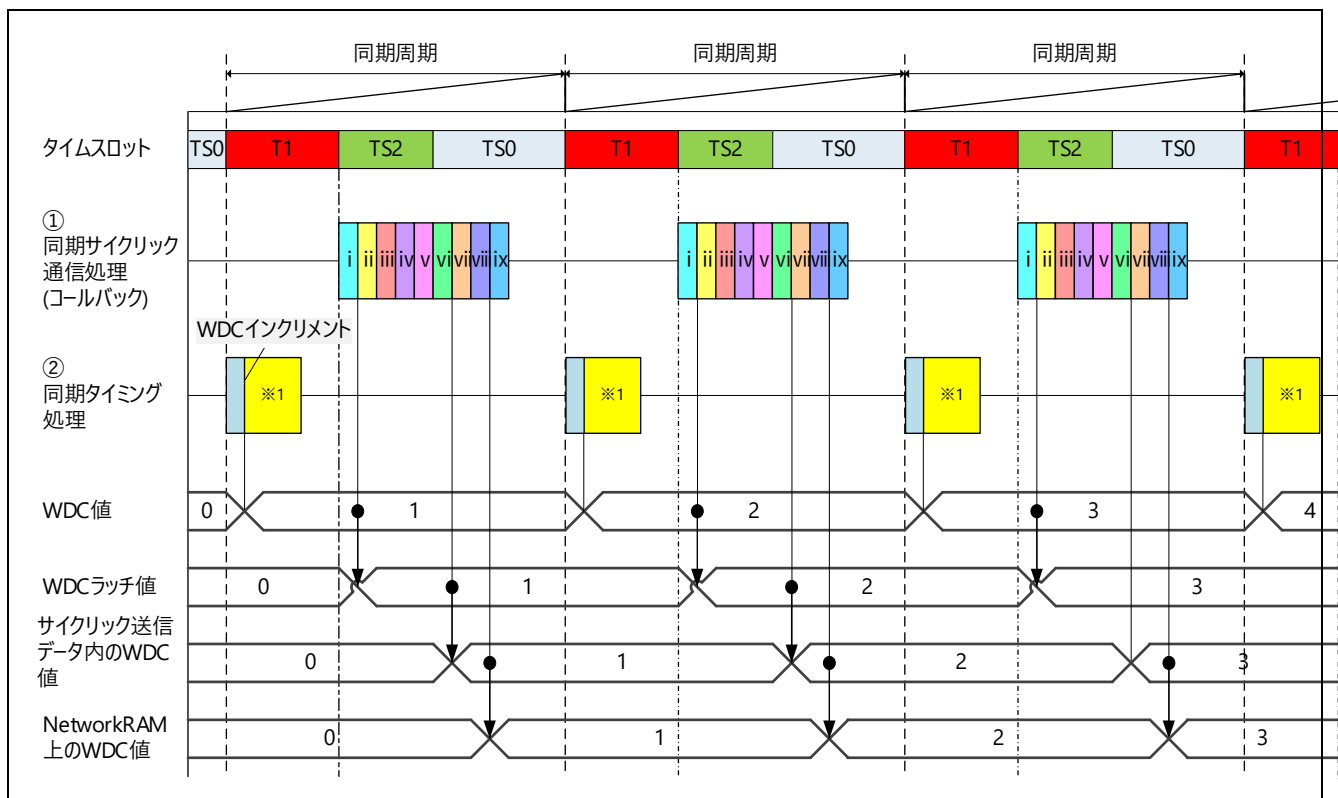


図 6.8 同期サイクリック通信処理・同期タイミング処理の動作タイミングチャート 1

【注】※1 他局とタイミングを揃えたいユーザアプリケーションの処理

【エラー処理の実装】

「gulR\_IN\_CallbackCheckComCycle」(6.6(8))にて通知される通信周期からタイムスロット 1 の時間を引いたものが、同期サイクリック通信処理が実行可能な時間です。この時間を確認し、実行不可な時間が指定されている場合は、「通信周期判定」の実行結果をエラーとしてください。

## (b) 同期サイクリック通信処理・同期タイミング処理の動作タイミングチャート 2

(同期タイミング処理がタイムスロット 1 の区間内に終了しないケース)

タスク起動の優先順位により、タイムスロット 2 開始時にコールバックされる同期サイクリック通信処理を行わず、同期タイミング処理が完了するのを待って、同期サイクリック通信処理を開始します。

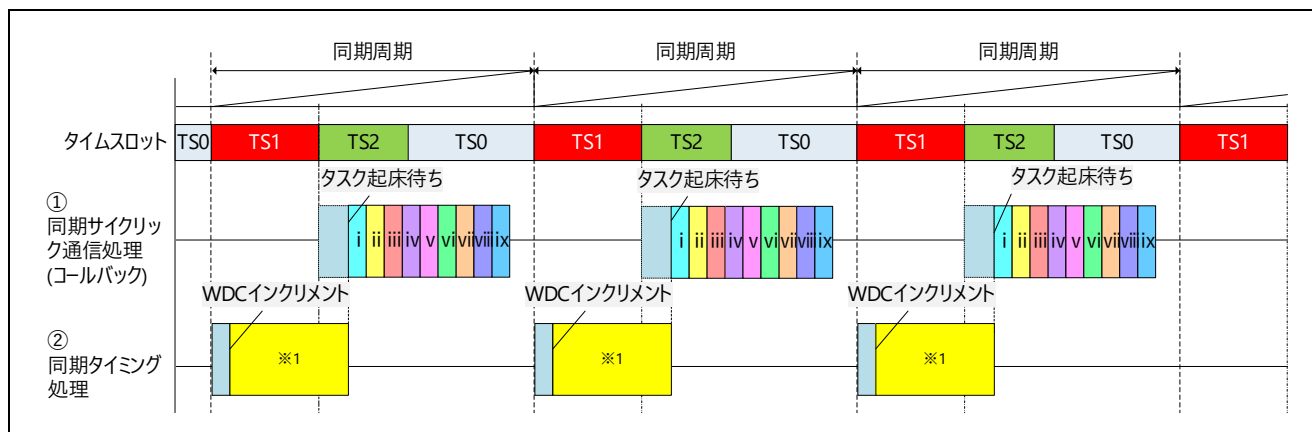


図 6.9 同期サイクリック通信処理・同期タイミング処理の動作タイミングチャート 2

【注】※1 他局とタイミングを揃えたいユーザアプリケーションの処理

図中に示す同期サイクリック通信処理と、同期タイミング処理が行う処理の内容は、「(a) 同期サイクリック通信処理・同期タイミング処理の動作タイミングチャート 1」で示したものと同じです。

## 【エラー処理の実装】

同期タイミング処理がタイムスロット 1 の区間内に終了しない場合、タイムスロット 1 の時間に問題があります。

「gulR\_IN\_CallbackCheckComCycle」(6.6(8))にて通知されるタイムスロット 1 の時間が、同期タイミング処理の実行時間以上であるかのチェックを行います。実行不可な時間が指定されている場合は、「通信周期判定」の実行結果をエラーとしてください。「通信周期判定」でエラーとすることで、データリンク開始を抑制できます。

## (c) 同期サイクリック通信処理・同期タイミング処理の動作タイミングチャート 3

(同期タイミング処理と同期サイクリック通信処理が同期周期内に完了しないケース)

受信ウォッチドッグカウンタが連続して異常を繰り返し、異常回数閾値を越えたとき、「gerR\_IN\_CheckWdcDL」(6.4.15(8))にて受信ウォッチドッグカウンタチェック結果として異常(R\_IN\_WDC\_NG\_OVER)を返します。

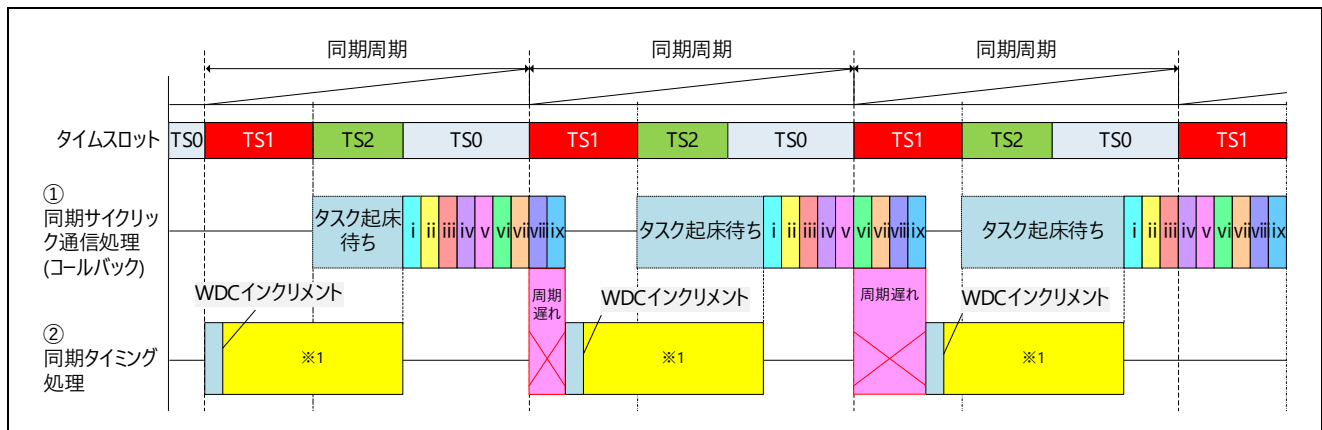


図 6.10 同期サイクリック通信処理・同期タイミング処理の動作タイミングチャート 3

【注】※1 他局とタイミングを揃えたいユーザアプリケーションの処理

図中に示す同期サイクリック通信処理(コールバック)と、同期タイミング処理が行う処理の内容は、「(a) 同期サイクリック通信処理・同期タイミング処理の動作タイミングチャート 1」で示したものと同じです。

## 【エラー処理の実装】

同期サイクリック通信処理と同期タイミング処理の合計時間が同期周期内に完了しない場合、マスタ局から指定された通信周期に問題があります。

「gulR\_IN\_CallbackCheckComCycle」(6.6(8))にて通知される通信周期が、「実行可能な通信周期であるか」のチェックを行います。実行不可な通信周期が指定されている場合は、「通信周期判定」の実行結果をエラーとしてください。「通信周期判定」でエラーとすることで、データリンク開始を抑制できます。

## (10) gvR\_IN\_CallbackIPAddressChangingResult

機能	IP アドレス変更結果												
呼出し形式	VOID gvR_IN_CallbackIPAddressChangingResult(ULONG ulResult)												
引数	型名	変数名	内容	入出力									
	ULONG	ulResult	IP アドレス変更結果 R_IN_OK：自局の IP アドレス変更が正常完了 R_IN_ERR：自局の IP アドレス変更が異常完了	入力									
戻り値	-												
説明	「gvR_IN_RequestIPAddressChanging」(6.4.11(8))を実行後、R-IN32M4-CL3 ドライバでの IP アドレス変更結果が引数にて通知されます。 マスタ局との通信状態により、自局の IP アドレスが変更されない場合(R_IN_ERR)があります。												
	<table><tr><td>変更結果</td><td>自局 IP アドレス</td><td>マスタ局との通信状態</td></tr><tr><td>R_IN_OK</td><td>変更される</td><td>サイクリック伝送設定前</td></tr><tr><td>R_IN_ERR</td><td>変更されない</td><td>サイクリック伝送設定後</td></tr></table>				変更結果	自局 IP アドレス	マスタ局との通信状態	R_IN_OK	変更される	サイクリック伝送設定前	R_IN_ERR	変更されない	サイクリック伝送設定後
	変更結果	自局 IP アドレス	マスタ局との通信状態										
	R_IN_OK	変更される	サイクリック伝送設定前										
	R_IN_ERR	変更されない	サイクリック伝送設定後										
R_IN_ERR の場合には、必要に応じて本処理内にエラー処理を追加し、自局の IP アドレスが変更されなかった旨をユーザへ通知してください。													
「erUserSImpSetIpAddress」(5.6.7 SLMP IP アドレス変更要求コマンド受信処理)にて IP アドレスを不揮発性メモリへ保存する処理を実装している場合は、自局の電源を OFF→ON してください。													
「iUserInitialization」(5.3.1 初期化処理)にて、不揮発性メモリから IP アドレスを読み出して R-IN32M4-CL3 ドライバに設定します。													



改訂記録	R-IN32M4-CL3 ユーザーズ・マニュアル CC-Link IE TSN 編
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2019.11.21	—	初版発行
1.01	2020.04.30	6	「2.3 R-IN32M4-CL3 適用製品の通信仕様」 表 2.2 を修正
		20	「4. R-IN32M4-CL3 適用製品のできること」 表 4.2 追加
		23-25	「4.2.2 SLMP について」 説明を追加、更新
		29-30	「5.1 ユーザプログラム一覧」 項目の追加、概要を更新
		31	「5.2 ユーザプログラムのタスク」 説明を追加
		38	「5.3.1 初期化処理」 図 5.7 を更新
		40-41	「5.3.3 SLMP 受信初期化処理」 「5.3.4 SLMP 要求フレーム作成初期化処理」 「5.3.5 パラメータオペレーション初期化処理」 新規追加
		42-48	「5.4 ユーザプログラム詳細(サイクリック伝送関連)」 説明を追加、更新
		54	「5.5.5 SLMP 受信処理」 説明を更新
		56	「5.5.8 SLMP ST(3E)応答フレーム受信処理」 新規追加
		57-58	「5.6.1 SLMP メモリ読出し要求コマンド受信処理」 「5.6.2 SLMP メモリ書込み要求コマンド受信処理」 図と説明を更新
		60-70	「5.6.4 SLMP メモリ読出し応答コマンド受信処理」～ 「5.6.14 SLMP ネットワーク時刻オフセット配信コマンド受信処理」 図と説明を更新
		71	「5.7.1 H/W テスト処理(IEEE802.3ab コンプライアンステスト)」 図 5.43 を更新
		80-82	「6.2 R-IN32M4-CL3 ドライバインタフェース関数一覧」 表を更新
		83	「6.3 R-IN32M4-CL3 ドライバのタスク」 表を新規追加
		85-87	「6.4.1 初期設定」 表 6.11～表 6.13 を更新
		99-109	「6.4.4 サイクリック伝送」 説明を追加、更新
		130-135	「6.4.10 SLMP 送受信」 説明を更新
		141-144	「6.4.14 汎用共通」 説明を更新

Rev.	発行日	改訂内容	
		ページ	ポイント
1.01	2020.04.30	145	「6.5 R-IN32M4-CL3 ドライバコールバック関数一覧」 表 6.35 を更新
		147	「6.6 R-IN32M4-CL3 ドライバコールバック関数詳細」 説明を追加
1.02	2020.10.31	16-18	「3.4.2 開発環境」～「3.4.4 コンパイルスイッチ」 新規追加
		28-31	「4.2.2 SLMP について」 新規追加
		35-37	「4.5 ネットワーク同期通信」 新規追加
		37-38	「4.6 CANopen 通信」 新規追加
		39-41	「5.1 ユーザプログラム一覧」 各種一覧表を追加・更新
		71-72	「5.5.9 R-IN32M4-CL3 ドライバ内検出 Fatal エラー確認処理」 新規追加
		83	「5.6.11 SLMP パラメータ配信可否チェック処理」 新規追加
		87	「5.6.15 SLMP パラメータデータ書き込み要求処理」 新規追加
		89	「5.6.17 SLMP ウォッチドッグカウンタ情報設定要求コマンド受信処理」 新規追加
		90-102	「5.7 ユーザプログラム詳細(CANopen 通信関連)」 新規追加
		113-115	「6.2 R-IN32M4-CL3 ドライバインタフェース関数一覧」 一覧表を更新
		150	「6.4.6 自局状態取得」 (8), (9)を新規追加
		180-182	「6.4.15 ネットワーク同期通信」 新規追加
		183-192	「6.4.16 CANopen 通信」 新規追加
		196-204	「6.6 R-IN32M4-CL3 ドライバコールバック関数詳細」 (5)～(9)を新規追加
1.05	2021.12.31	10	サンプル CSP+ファイルを追加
		19	コンパイルスイッチを追加
		25	「4. R-IN32M4-CL3 適用製品のできること」 表 4.1 を更新
		29-30	ユーザプログラムが送受信する SLMP コマンドを追加
		39-40	「4.6.1 CANopen 通信時の増設ユニット」新規追加
		40-41	「4.7 SLMP 経由の通信速度設定」新規追加
		43	SLMP コマンド実行関連のユーザプログラム関数を追加
		102-103	「5.7 ユーザプログラム詳細(CANopen 通信関連)」 (4)を新規追加
		129-131	「6.2 R-IN32M4-CL3 ドライバインタフェース関数一覧」を追加・更新

Rev.	発行日	改訂内容	
		ページ	ポイント
1.06	2022.7.29	4	「1.4 CC-Link 協会のご案内」 (1), (2)を更新
		7	「2.3 R-IN32M4-CL3 適用製品の通信仕様」 表 2.2 を更新
		8	「2.3.1 CC-Link IE TSN Class A の動作時の注意事項」 新規追加
		11	2.6 サンプル CSP+ファイルの概要 表 2.5 を更新
		18	「3.4.2 開発環境」 表 3.3 を更新
		19	「3.4.3.2 フラッシュ ROM テーブルの設定」 表 3.4 を更新
		20	「3.4.4 コンパイルスイッチ」 表 3.5 を更新
		26	「4. R-IN32M4-CL3 適用製品のできること」 表 4.1 を更新
		35-37	「4.5 ネットワーク同期通信」 説明を追加
		40-41	「4.7 安全通信時の安全 PDU 送受信」 新規追加
		41	「4.8 SLMP 経由の通信速度・CC-Link IE TSN Class 設定」 新規追加
		43-44	「5.1 ユーザプログラム一覧」 表 5.4 – 表 5.7 を追加・更新
		46-63	「5.2 ユーザプログラムのタスク」 説明を追加・更新
		64-74	「5.4 ユーザプログラム詳細(サイクリック伝送関連)」 説明を追加・更新
		97	「5.6.10 SLMP ネットワーク時刻配信コマンド受信処理」 新規追加
		113-114	「5.8.5 SLMP 機能設定読出し(CC-Link IE TSN Class)要求コマンド受信処理」 「5.8.6 SLMP 機能設定書込み(CC-Link IE TSN Class)要求コマンド受信処理」 新規追加
		134-145	「5.8 ユーザプログラム詳細(MCU 間 I/F 関連)」 新規追加
		156-159	「6.2 R-IN32M4-CL3 ドライバインタフェース関数一覧」 表 6.5 を更新
		163-172	「6.4.1 初期設定」 説明を追加、更新
		183-194	「6.4.4 サイクリック伝送」 説明を追加、更新
		245-249	「6.4.17 MCU 間 I/F」 新規追加

Rev.	発行日	改訂内容	
		ページ	ポイント
1.07	2024.3.29	21	「3.5 IP アドレスの設定方法の検討」を更新
		27	「4. R-IN32M4-CL3 適用製品のできることを更新
		30-31	「4.2.2 SLMP の伝送フレームについて」を更新
		34	「4.3.3 LED の制御」を更新
		35-36	「4.3.5 L ERR. LED の制御」新規追加
		37	「4.5 ネットワーク同期通信」を更新
		45	「4.8 SLMP 経由の通信速度・CC-Link IE TSN Class 設定」を更新
		46	「4.9 デバイス局の IP アドレス設定」新規追加
		69-71	「5.3.1 初期化処理」を更新
		73	「5.3.3 SLMP 受信初期化処理」を更新
		87	「5.5.3 LED 更新処理」を更新
		100-104	「5.6.6 SLMP インディケータ表示要求コマンド受信処理」、 「5.6.7 SLMP IP アドレス変更要求コマンド受信処理」を更新
		111	「5.7.3 SLMP パラメータ配信要否チェック処理」を更新
		116	「5.8 ユーザプログラム詳細(CC-Link IE TSN 機器のパラメータ設定)」を更新
		128	「5.9 ユーザプログラム詳細(CANopen 通信関連)」を更新
		166-168	「6.2 R-IN32M4-CL3 ドライバインタフェース関数一覧」を更新
		173-185	「6.4.1 初期設定」を更新 「(4) gerR_IN_SetIPAddressAndNodeNumber」新規追加
		191-192	「6.4.3 イベント」 「6.4.4 サイクリック伝送」を更新
		205-209	「6.4.6 自局状態取得」を更新 「(11) gerR_IN_GetIPAddressDuplication」新規追加
		225	「6.4.10 SLMP 送受信」を更新
		233	「(6) gerR_IN_SetNodeIndicationStatus」、 「(7) gvR_IN_RequestIPAddressChanging」新規追加
		234	「6.4.12 エラー履歴」を更新
		245	「6.4.15 ネットワーク同期通信」を更新
		260	「6.5 R-IN32M4-CL3 ドライバコールバック関数一覧」を更新
		261-273	「6.6 R-IN32M4-CL3 ドライバコールバック関数詳細」を更新 「(10) gvR_IN_CallbackIPAddressChangingResult」新規追加
1.08	2025.7.1	27	「4. R-IN32M4-CL3 適用製品のできることを更新
		31	「4.2.4 SLMP のコマンドについて」を更新
		40	「4.5.1 ネットワーク同期通信時の同期信号出力」新規追加
		61	「5.2.5 同期タイミング処理タスク」を更新
		76-77	「5.3.5 パラメータオペレーション初期化処理」を更新
		178 187	「6.4 R-IN32M4-CL3 ドライバインタフェース関数詳細」 表 6.9, 表 6.17 を更新
1.09	2025.9.26	8	「2.3.2 サンプルコードのバージョンによる機能の制約」新規追加
		20	「3.4.4 コンパイルスイッチ」を更新
		27	「4 R-IN32M4-CL3 適用製品のできることを更新
		31-32	「4.2.4 SLMP のコマンドについて」を更新
		46	「4.8 SLMP 経由の通信速度・CC-Link IE TSN Class 設定」を更新
		48-50	「4.10 デバイス局のパラメータ設定・コマンド実行」新規追加
		51-52	「4.11 SNMP コミュニティ名設定」新規追加

Rev.	発行日	改訂内容	
		ページ	ポイント
1.09	2025.9.26	54	「5.1 ユーザプログラム一覧」 表 5.4 を更新
		117-119	「5.6.12 SLMP SetCommunityName 要求コマンド受信処理」新規追加
		127	「5.8 ユーザプログラム詳細(SLMP 経由の通信速度・CC-Link IE TSN Class 設定関連)」を更新
		131	「5.8.2 SLMP 機能設定(サポート情報取得)要求コマンド受信処理」を更新
		179-180	「6.2 R-IN32M4-CL3 ドライバインタフェース関数一覧」 表 6.5 を更新
		186, 189-190	「6.4.1 初期設定」 表 6.9, 表 6.12, 表 6.13 を更新
		246	「(8) gerR_IN_SetSNMPCommunityName」新規追加
		259	「(9) gvR_IN_ClearSyncDeviationFlag」新規追加
		268	「(11) gerR_IN_CanUpdateTPDO」を更新
		283 286	「(9) gulR_IN_CallbackSyncCom」 表 6.48, (c) を更新

---

R-IN32M4-CL3

ユーザズ・マニュアル CC-Link IE TSN 編

発行年月日 2025 年 9 月 26 日 Rev.1.09

発行 ルネサス エレクトロニクス株式会社

〒135-0061 東京都江東区豊洲 3-2-24 (豊洲フォレシア)

---

R-IN32M4-CL3

ユーザーズ・マニュアル  
CC-Link IE TSN 編