

R-IN32M4-CL3

ユーザーズ・マニュアル

CC-Link IE Field 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。

資料番号 : R18UZ0071JJ0100

発行年月 : 2019.11.1

ルネサス エレクトロニクス

www.renesas.com

ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器・システムの設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因して、お客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
2. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
3. 本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害に関し、当社は、何らの責任を負うものではありません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を改造、改変、複製等しないでください。かかる改造、改変、複製等により生じた損害に関し、当社は、一切その責任を負いません。
5. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置等

当社製品は、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（原子力制御システム、軍事機器等）に使用されることを意図しておらず、使用することはできません。たとえ、意図しない用途に当社製品を使用したことによりお客様または第三者に損害が生じても、当社は一切その責任を負いません。なお、ご不明点がある場合は、当社営業にお問い合わせください。

6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないように、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。お客様がかかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
9. 本資料に記載されている当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。また、当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途に使用しないでください。当社製品または技術を輸出する場合は、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。
10. お客様の転売等により、本ご注意書き記載の諸条件に抵触して当社製品が使用され、その使用から損害が生じた場合、当社は何らの責任も負わず、お客様にてご負担して頂きますのでご了承ください。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。

注1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社がその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注2. 本資料において使用されている「当社製品」とは、注1において定義された当社の開発、製造製品をいいます。

製品ご使用上の注意事項

ここでは、CMOS デバイスの一般的注意事項について説明します。個別の使用上の注意事項については、本文を参照してください。なお、本マニュアルの本文と異なる記載がある場合は、本文の記載が優先するものとします。

1. 未使用端子の処理

【注意】未使用端子は、本文の「未使用端子の処理」に従って処理してください。

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

2. 電源投入時の処置

【注意】電源投入時は、製品の状態は不定です。

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

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

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

3. リザーブアドレスのアクセス禁止

【注意】リザーブアドレスのアクセスを禁止します。

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

4. クロックについて

【注意】リセット時は、クロックが安定した後、リセットを解除してください。

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

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

○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、SLMPI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フィールド・ネットワークのインテリジェントデバイス局および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。

本製品は、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。

本文中の★印は、本版で改訂された主な箇所を示しています。この"★"をPDF上でコピーして「検索する文字列」に指定することによって、改版箇所を容易に検索できます

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。下記資料番号の末尾****部分は版数です。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	R18UZ0070JJ****
R-IN32M4-CL3 ユーザーズ・マニュアル CC-Link IE Field 編	本マニュアル
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 = 10×10^6 bps

100Mbps = 100×10^6 bps

目次

1. はじめに	1
1.1 序論	1
1.1.1 CC-Link 協会のご案内	1
1.1.2 総称・略称について	3
1.1.3 関連マニュアルについて	5
1.1.4 ご使用の注意事項について	5
1.2 概要	6
1.2.1 R-IN32M4-CL3 の性能仕様	6
1.2.2 サンプルソフトの概要	7
2. 開発前の仕様検討と準備	9
2.1 MACアドレスの取得	10
2.2 ベンダ・コードの取得と機種タイプの選択	10
2.3 局番・ネットワークNo.の書き込み方法の検討	11
2.4 1000BASE-Tコンプライアンス・テストモードの実装検討	13
2.5 エンジニアリング・ツール各種機能への対応準備	14
2.5.1 CC-Link IE フィールド・ネットワーク診断	14
2.5.2 スレーブ局パラメータ処理・コマンド実行	14
2.6 エラー状態、エラー・コードの仕様検討	15
2.6.1 エラー状態の仕様検討	15
2.6.2 リターン・コード（RSTS）と終了コード（End Code）に格納するエラー・コードの仕様検討	15
2.7 リンクデバイスの属性の割り当て検討	16
2.7.1 ダイレクト入出力グループ	16
2.7.2 システム入出力グループ	16
2.7.3 ベンダ入出力グループ	16
2.8 Hold/Clear処理の実装検討	17
2.9 CSP+の作成準備	18
2.9.1 スレーブ局パラメータ処理・コマンド実行	19
2.10 コンフォーマンス・テスト	20
3. R-IN32M4-CL3 の機能	21
3.1 リセット機能	21
3.2 ファストリンクアップ機能	22
3.2.1 ファストリンクアップ機能の有効/無効設定	24

3.2.2	ファストリンクアップ時のシステム構成.....	25
3.2.3	ファストリンクアップ機能の注意事項.....	27
3.3	状態表示機能	28
3.3.1	LED による状態表示.....	28
3.3.2	LED の制御.....	29
3.3.3	User LED の制御	30
3.3.4	LED の点灯機能有効/無効.....	30
3.4	バイパスモード	31
3.5	MIB情報.....	31
3.6	サイクリック伝送機能	32
3.7	トランジェント伝送機能	33
3.7.1	トランジェント伝送のクライアント・サーバ機能について	34
3.7.2	トランジェント伝送のフレームについて.....	34
3.7.3	トランジェント伝送のコマンドについて.....	35
3.8	MyStatus送受信機能.....	36
3.8.1	MyStatus 送信	36
3.8.2	MyStatus 受信	37
3.9	CC-Link IEフィールド・ネットワーク診断機能.....	38
3.9.1	選択局通信状態モニタの LED.....	39
4.	ユーザ・プログラムの作成	40
4.1	ユーザ・プログラム一覧	40
4.2	ユーザ・プログラム詳細（イニシャル・サイクリック伝送関連）	42
4.2.1	メイン処理	42
4.2.2	初期化处理	44
4.2.3	通信開始処理	46
4.2.4	自局エラー処理	47
4.2.5	サイクリック伝送停止処理.....	47
4.2.6	イベント処理	48
4.2.7	マスタ局からの MyStatus およびサイクリック受信処理.....	49
4.2.8	MyStatus 送信処理	51
4.2.9	サイクリック送信処理	51
4.2.10	通信状態更新処理	52
4.2.11	LED 更新処理.....	53
4.2.12	サイクリック伝送状態更新処理.....	53
4.2.13	MIB 情報取得処理.....	54
4.3	ユーザ・プログラム詳細（トランジェント関連）	56
4.3.1	トランジェント伝送の処理概要.....	56
4.3.2	Transient1、Transient2、TransientAck 受信処理.....	58

4.3.3	Transient1 受信データ処理	59
4.3.4	Transient1 受信データ組立て開始処理.....	66
4.3.5	Transient1 受信データ組立て処理	67
4.3.6	SLMP メモリ読出し要求フレーム受信対応処理	68
4.3.7	SLMP メモリ書込み要求フレーム受信対応処理	69
4.3.8	Transient2 受信データ処理	70
4.3.9	Transient2 受信データチェック処理.....	71
4.3.10	Transient2 メモリ書込み要求受信処理.....	72
4.3.11	Transient2 応答フレーム作成処理	73
4.3.12	TransientAck 受信データ処理.....	74
4.3.13	TransientAck フレーム作成処理.....	75
4.3.14	Transient1、Transient2、TransientAck 送信処理.....	76
4.3.15	SLMP メモリ読出し要求フレーム作成処理	79
4.3.16	Transient1 要求送信分割判定処理	80
4.3.17	Transient1 要求フレーム作成処理	81
4.3.18	SLMP メモリ読出し応答受信処理.....	82
4.3.19	Transient2 要求フレーム作成処理	83
4.3.20	Transient2 メモリ読出し要求フレーム作成処理.....	84
4.3.21	Transient2 メモリ読出し応答受信処理.....	85
4.4	ユーザ・プログラム詳細（H/Wテスト関連）	86
4.4.1	H/W テスト（IEEE802.3ab コンプライアンス・テスト）	86
4.4.2	H/W テスト（折り返し通信テスト）	87
5.	R-IN32M4-CL3 ドライバの関数仕様	88
5.1	各関数の概要	88
5.1.1	概要	88
5.1.2	型定義とエラー・コード	89
5.1.3	記述仕様	89
5.2	R-IN32M4-CL3 ドライバ・インタフェース関数一覧.....	90
5.3	R-IN32M4-CL3 ドライバ・インタフェース関数詳細.....	93
5.3.1	初期設定	93
5.3.2	ウォッチドッグタイマ	102
5.3.3	イベント	104
5.3.4	サイクリック伝送	106
5.3.5	自局状態設定	109
5.3.6	自局状態取得	110
5.3.7	LED 制御	114
5.3.8	ネットワーク時刻	117
5.3.9	MDIO アクセス	119

5.3.10	トランジェント受信処理	121
5.3.11	トランジェント要求受信対応.....	123
5.3.12	トランジェント送信フレームヘッダ作成.....	127
5.3.13	トランジェント送信処理	129
5.3.14	H/W テスト	133
5.4	R-IN32M4-CL3ドライバ・コールバック関数一覧.....	135
5.5	R-IN32M4-CL3ドライバ・コールバック関数詳細.....	135
6.	リンクデバイスのシステム領域	138
6.1	システム領域詳細	139
7.	フレーム・フォーマット	142
7.1	共通フォーマット	143
7.2	Transient1フレーム	145
7.3	CC-Link互換トランジェント・フレーム	148
7.3.1	CC-Link 互換トランジェント・フレームの詳細.....	149
7.3.2	コマンドタイプ (CT) 詳細	151
7.3.3	リターン・コード (RSTS) 詳細	151
7.3.4	メモリアクセス情報取得	153
7.3.5	リモート RUN.....	156
7.3.6	リモート STOP.....	157
7.3.7	メモリ読出し	158
7.3.8	メモリ書込み	160
7.3.9	アクセス・コードと属性	162
7.4	SLMPフレーム	164
7.4.1	SLMP フレームの基本フォーマットの詳細	165
7.4.2	終了コード (End Code) 詳細.....	166
7.4.3	SLMP メモリ読出しフレーム	167
7.4.4	SLMP メモリ書込みフレーム	168
7.4.5	TransientAck フレーム.....	169

1. はじめに

1.1 序論

1.1.1 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として、CC-Link 協会発行の下記仕様書があります。CC-Link IE フィールド・ネットワークおよび 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。

表 1.1 CC-Link IE フィールド・ネットワーク仕様書一覧（CC-Link 協会発行）

資料名称	資料番号
CC-Link IE フィールド・ネットワーク仕様書（概要編）	BAP-C2005-001
CC-Link IE フィールド・ネットワーク仕様書（物理層・データリンク層編）	BAP-C2005-002
CC-Link IE フィールド・ネットワーク仕様書（アプリケーション層サービス編）	BAP-C2005-003
CC-Link IE フィールド・ネットワーク仕様書（アプリケーション層プロトコル編）	BAP-C2005-004
CC-Link IE フィールド・ネットワーク仕様書（通信プロファイル編）	BAP-C2005-005
CC-Link IE フィールド・ネットワーク仕様書（実装規約編）	BAP-C2005-006
CC-Link IE フィールド・ネットワーク仕様書（デバイスプロファイル編）	BAP-C2005-007
SLMP 仕様書（概要編）	BAP-C2006-001
SLMP 仕様書（サービス編）	BAP-C2006-002
SLMP 仕様書（プロトコル編）	BAP-C2006-003

(2) CC-Link IE Field ユーティリティについて

マスタ局を PC 上でシミュレートするツールです。マスタ局をお持ちでない場合に、PC があれば簡易的にマスタ局をシミュレートできます。CC-Link 協会のホームページからダウンロードの上、ご活用ください。

表 1.2 CC-Link IE Field ユーティリティ

資料名称/関連ツール	資料番号
CC-Link IE Field ユーティリティ	—
CC-Link IE Field ユーティリティ 操作説明書	—

(3) 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。

表 1.3 CC-Link IE フィールド・ネットワーク コンフォーマンス・テスト仕様書

資料名称	資料番号
CC-Link IE フィールド・ネットワーク インテリジェントデバイス局 コンフォーマンス・テスト仕様書	BAP-C0401-037
CC-Link IE フィールド・ネットワーク リモートデバイス局 コンフォーマンス・テスト仕様書	BAP-C0401-041

(a) Control & Communication システム・プロファイル（CSP+）の作成

コンフォーマンス・テストには、CSP+の確認があります。あらかじめ CSP+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+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。

表 1.4 Control & Communication システム・プロファイル（CSP+）

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	—

(b) 1000BASE-T コンプライアンス・テストの実施

コンフォーマンス・テストには、1000BASE-T コンプライアンス・テスト（IEEE802.3 の仕様に基づいた波形試験）の確認があり、ユーザにて実施していただく必要があります。

1000BASE-T コンプライアンス・テストが実施可能なテストラボがありますので、詳細は、CC-Link 協会までお問い合わせ願います。

(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 : info@cc-link.org
	Web : http://www.cc-link.org/

1.1.2 総称・略称について

本書では、特に明記する場合を除き、下記に示す総称・略称を使用して説明します。

表 1.5 本マニュアルにおける総称/略称

(1/2)

総称/略称	総称/略称の内容
R-IN32M4-CL3	CC-Link IE フィールド・ネットワーク インテリジェントデバイス局・リモートデバイス局用 Gigabit Ethernet PHY (GbE-PHY) 内蔵通信 LSI です。
R-IN32M4-CL3 適用回路	通信 LSI R-IN32M4-CL3 および周辺部品で構成した CC-Link IE フィールド・ネットワークの通信用回路です。
R-IN32M4-CL3 適用製品	本書を参考に製作した、CC-Link IE フィールド・ネットワーク接続対応製品です。
R-IN32M4-CL3 ドライバ	R-IN32M4-CL3 を制御するプログラム群です。 ドライバ・インタフェース関数、ドライバ・コールバック関数、ドライバ本体で構成します。
ユーザ・プログラム	ユーザが作成するアプリケーション・プログラムです。インテリジェントデバイス局またはリモートデバイス局のロジックを確認するための参考用（サンプル・プログラム）のため、カスタマイズする必要があります。
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。ノードは同義語です。
マスタ局	CC-Link IE フィールド・ネットワークを制御する局で、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。
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。サイクリック伝送で、他のスレーブ局の RX、RY、RW _r 、RW _w を受信します。
インテリジェントデバイス局	マスタ局と、ビット単位の入出力信号とワード単位の入出力データをサイクリック伝送する局です。トランジェント伝送も可能です。 他局からのトランジェント伝送（要求）に対して応答を返します。また、他局へトランジェント伝送（要求）を発行します。
リモートデバイス局	マスタ局と、ビット単位の入出力信号とワード単位の入出力データをサイクリック伝送する局です。他局からのトランジェント伝送（要求）に対して応答を返します。
リモート I/O 局	マスタ局と、ビット単位の入出力信号を、サイクリック伝送する局です。
スレーブ局	マスタ局以外の局（ローカル局、インテリジェントデバイス局、リモートデバイス局、リモート I/O 局）の総称です。
自局	本書を基に開発するインテリジェントデバイス局またはリモートデバイス局です。
他局	自局以外の局です。
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。ノード番号は同義語です。
サイクリック伝送	リンクデバイス (RX/R _Y /RW _r /RW _w) 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通信方式です。
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通信方式です。
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。

表 1.5 本マニュアルにおける総称／略称

(2/2)

総称／略称	総称／略称の内容
リモート入力 (RX)	スレーブ局からマスタ局にビット単位で入力される情報です。
リモート出力 (RY)	マスタ局からスレーブ局にビット単位で出力される情報です。
リモートレジスタ (RW _r)	スレーブ局からマスタ局に 16 ビット単位 (1 ワード) で入力される情報です。
リモートレジスタ (RW _w)	マスタ局からスレーブ局に 16 ビット単位 (1 ワード) で出力される情報です。
リンクデバイス	RX、RY、RW _r 、RW _w の総称です。
デバイス	シーケンサ CPU ユニットが内部に持っているデバイス (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。
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。
バトンパス	ネットワークにデータを送信するための、送信権データ (トークン) 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。
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。
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。
ユーザ	本書をもとに CC-Link ファミリ接続対応製品を開発・販売するメーカーです。ベンダ、パートナー・メーカーと同義です。
エンド・ユーザ	ユーザが開発した CC-Link ファミリ接続対応製品の購入・使用者です。
GX Works2/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。
マスタ・ローカルユニット	RJ71GF11-T2 形、QJ71GF11-T2 形、または LJ71GF11-T2 形 CC-Link IE フィールド・ネットワーク・マスタ・ローカルユニットの略称です。 また、CC-Link IE フィールド・ネットワーク機能使用時の RJ71EN71 形、RnENCPU 形ユニットの総称です。
Hold/Clear 処理	サイクリック伝送において、サイクリック伝送停止または解列になった場合、それまで正常に受信したデータ (RY) を保持またはクリアする処理です。
RIRD	マスタ・ローカルユニットの専用命令 JP.RIRD、GP.RIRD の略称です。
RIWT	マスタ・ローカルユニットの専用命令 JP.RIWT、GP.RIWT の略称です。
CSP+	Control & Communication システム・プロファイルの略称です。

1.1.3 関連マニュアルについて

CC-Link IE フィールド・ネットワークの用語、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>)

表 1.6 CC-Link IE フィールド・ネットワーク関連マニュアル

関連マニュアル名称	マニュアル番号 (形名コード)
【MELSEC iQ-R Ethernet/CC-Link IE ユーザーズ・マニュアル（スタートアップ編）】 Ethernet、CC-Link IE コントローラネットワーク、CC-Link IE フィールド・ネットワークの仕様、 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。	SH (名) - 081252 (13J2A7)
【MELSEC iQ-R CC-Link IE フィールド・ネットワーク ユーザーズ・マニュアル（応用編）】 CC-Link IE フィールド・ネットワークの機能、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。	SH (名) - 081255 (13J2B7)
【MELSEC-Q CC-Link IE フィールドネットワークマスタ・ローカルユニット ユーザーズ・マニュアル】 CC-Link IE フィールド・ネットワークおよび MELSEC-Q シリーズマスタ・ローカルユニットの仕 様、運転までの手順、システム構成、設置、設定、機能、プログラミング、トラブル・シューティ ングについて記載しています。	SH (名) - 080916 (13J244)
【MELSEC-L CC-Link IE フィールド・ネットワークマスタ・ローカルユニット ユーザーズ・マニュアル】 CC-Link IE フィールド・ネットワークおよび MELSEC-Q シリーズマスタ・ローカルユニットの仕 様、運転までの手順、システム構成、設置、設定、機能、プログラミング、トラブル・シューティ ングについて記載しています。	SH (名) - 080971 (13J261)
【MELSEC-L CC-Link IE フィールド・ネットワークヘッド ユニット ユーザーズ・マニュアル】 ヘッドユニットの仕様、運転までの手順、システム構成、設置、設定、トラブル・シューティング について記載しています。	SH (名) - 080918 (13J245)
【SLMP リファレンス・マニュアル】 外部機器から SLMP 対応機器に対して、データの読出し、書き込みを行うプロトコル (SLMP) につ いて記載しています。	SH (名) - 080931 (13JD25)

1.1.4 ご使用の注意事項について

- (1) 本書の記載事項、仕様などは、お断りなしに変更することがありますので、ご了承ください。
- (2) 本書にて解説するサンプルコードは、インテリジェントデバイス局・リモートデバイス局用 GbE-PHY
内蔵通信 LSI R-IN32M4-CL3 開発用のものです。サンプルコードは本資料の使用例を示すものであり、
ルネサスエレクトロニクスがその動作を保証するものではありません。
- (3) ルネサスエレクトロニクスの WEB サイトからダウンロードした情報の知的財産権その他の権利は、ル
ネサスエレクトロニクスに帰属します。上記の情報を、ルネサスエレクトロニクスの許諾を得ずに転載
すること、インテリジェントデバイス局またはリモートデバイス局の開発目的以外で複製することは禁
じられています。ただし、サンプルコードをコンパイルした後の生成物（オブジェクトコードなど）に
配布の制限は設けません。

1.2 概要

本書は、CC-Link IE フィールド・ネットワーク インテリジェントデバイス局、リモートデバイス局用 GbE-PHY 内蔵通信 LSI R-IN32M4-CL3 を使用して、CC-Link IE フィールド・ネットワークのインテリジェントデバイス局またはリモートデバイス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。

- ・ユーザ・プログラムの作成（「4. ユーザ・プログラムの作成」参照）
- ・R-IN32M4-CL3 ドライバの関数仕様（「5. R-IN32M4-CL3 ドライバの関数仕様」参照）

1.2.1 R-IN32M4-CL3 の性能仕様

R-IN32M4-CL3 の CC-Link IE フィールド・ネットワークに関する性能仕様を以下に示します。

表 1.7 CC-Link IE フィールド・ネットワークの性能仕様

項目		内容	
局種別		インテリジェントデバイス局またはリモートデバイス局	
局番		1～120	
ネットワーク No.		1～239	
通信速度		1Gbps	
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	
接続ケーブル		1000BASE-T の規格を満たす Ethernet ケーブル （カテゴリ 5e 以上、二重シールド付・STP、ストレートケーブル）	
最大局間距離		100m	
総延長距離		ライン接続時：12000m（マスタ局 1 台およびスレーブ局 120 台接続時） スター接続時：システム構成による リング接続時：12100m（マスタ局 1 台およびスレーブ局 120 台接続時）	
カスケード接続段数		最大 20 段	
サイクリック伝送		インテリジェントデバイス局	リモートデバイス局
1 局あたりの 最大リンク点数	RX	最大 2048 点（2048 ビット）、256 バイト	最大 128 点（128 ビット）、16 バイト
	RY	最大 2048 点（2048 ビット）、256 バイト	最大 128 点（128 ビット）、16 バイト
	RWr	最大 1024 点（1024 ワード）、2048 バイト	最大 64 点（64 ワード）、128 バイト
	RWw	最大 1024 点（1024 ワード）、2048 バイト	最大 64 点（64 ワード）、128 バイト
トランジェント伝送		インテリジェントデバイス局	リモートデバイス局
クライアント機能		あり	なし
サーバ機能		あり	あり
データサイズ ^注		2048 バイト	1024 バイト
専用命令		サンプルコードをカスタマイズすることで RIRD、RIWT を受信可能	
その他機能		－	
診断機能		CC-Link IE フィールド・ネットワーク診断機能あり	
一時エラー無効局		あり	
ファストリンクアップ		あり	

注. 送受信するトランジェント・フレームが 1518 バイトを超える場合、トランジェント・フレームを分割して送受信します。

1.2.2 サンプルソフトの概要

(1) サンプルソフトの概要

サンプルソフトの概要を以下に示します。

表 1.8 ファイル概要

フォルダ名		内容
cie_intelligent_device	sample driver	インテリジェントデバイス局用サンプルコード 以下のプログラム部品でサンプルコードを構成 ・ ユーザ・プログラム ・ 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 ・ R-IN32M4-CL3 ドライバ・コールバック関数
	IAR	IAR Embedded Workbench 用ファイル (コンパイラ設定、リンク設定、ビルド設定など)
cie_remote_device	sample driver	リモートデバイス局用サンプルコード、IAR Embedded Workbench 用ファイル (インテリジェントデバイス局と同様)
	IAR	

サンプルコードは、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されています。

注意. サンプルコードは、CC-Link IE フィールド・ネットワークに関して（通信機能）のみ記述しています。

- (1) ユーザ・プログラムは、ユーザが作成するアプリケーション・プログラムです。インテリジェントデバイス局またはリモートデバイス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。（「4. ユーザ・プログラムの作成」を参照）
- (2) 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。
（「5.2 R-IN32M4-CL3 ドライバ・インタフェース関数一覧」を参照）
- (3) 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。
（「5.4 R-IN32M4-CL3 ドライバ・コールバック関数一覧」を参照）
- (4) 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。カスタマイズは不要です。

(2) システム構成

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、OS ドライバ・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。

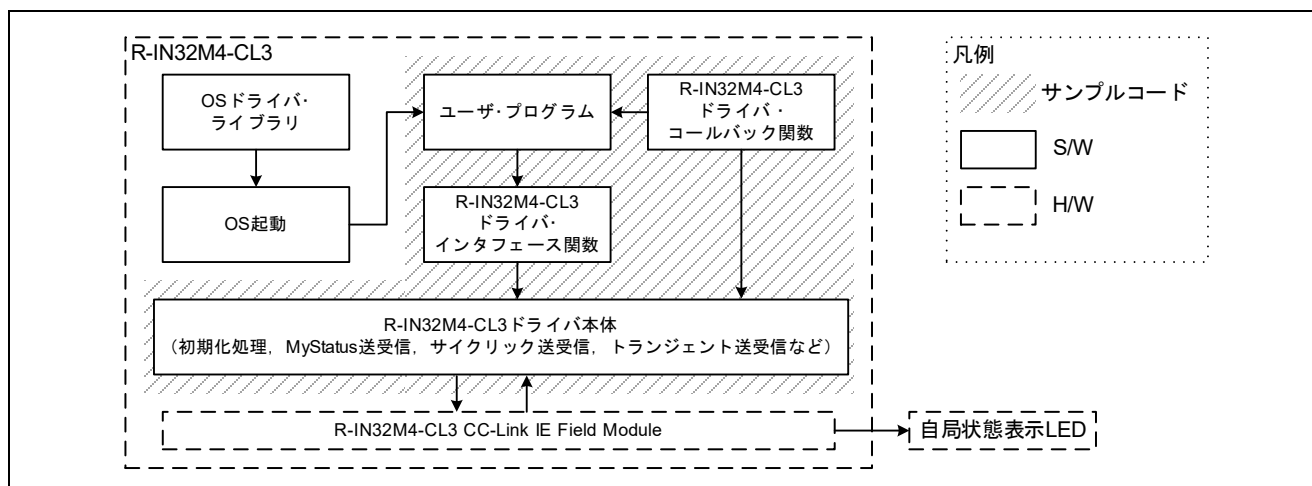


図 1.1 S/W 構成概略図

2. 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。

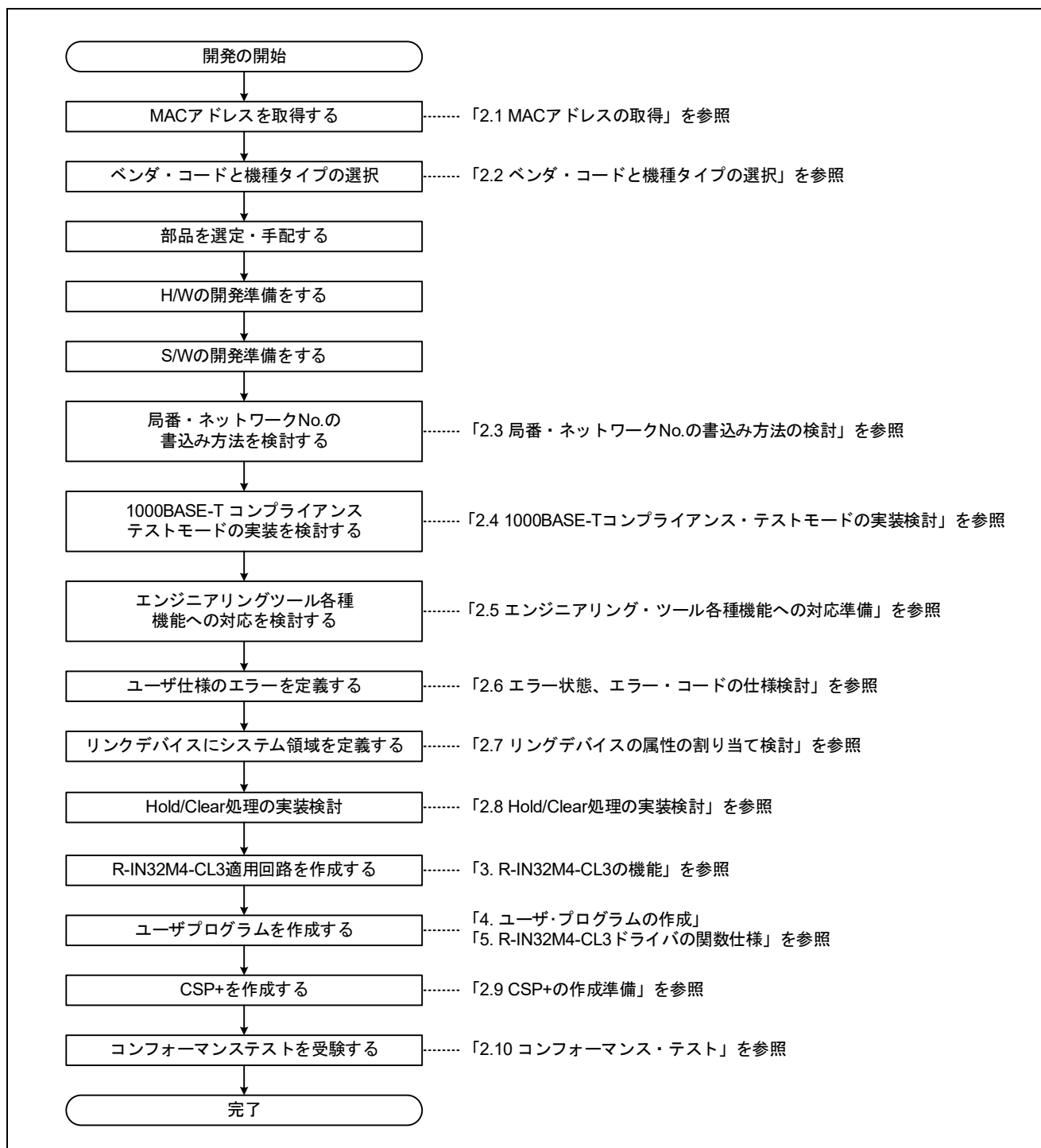


図 2.1 開発の工程例

2.1 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

445 Hoes Lane Piscataway, NJ 08854 USA

Phone : +1 (732) 465-6481

Fax : +1 (732) 562-1571

Web : <https://standards.ieee.org/products-services/regauth/oui/> (2019 年 11 月現在)

2.2 ベンダ・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・コードと機種タイプの登録が必要です。ベンダ・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 2.1 ベンダ・コードと機種タイプ

項目	内容
ベンダ・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) 注
機種タイプ (deviceType)	CC-Link IE フィールド・ネットワーク仕様書 (デバイスプロファイル編) 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。

注. ID 番号が 123-456-7890 の場合、ベンダ・コードは 5678 となります。

2.3 局番・ネットワーク No.の書き込み方法の検討

自局をデータリンクさせるためには、局番およびネットワーク No.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、局番およびネットワーク No.の書き込み方法をあらかじめ検討してください。

例として、H/W スイッチで設定する方法、または開発製品のエンジニアリング・ツール（周辺機器）から設定する方法などがあります。

どちらの方法もユーザ・プログラムの `iUserInitialization`（「4.2.2 初期化処理」参照）において、ドライバ・インタフェース関数の局番およびネットワーク No.設定

（「5.3.1(3)gerR_IN32_SetNodeAndNetworkNumber」参照）を使用します。

参考として、局番およびネットワーク No.の書き込み方法とイメージを以下に例示します。

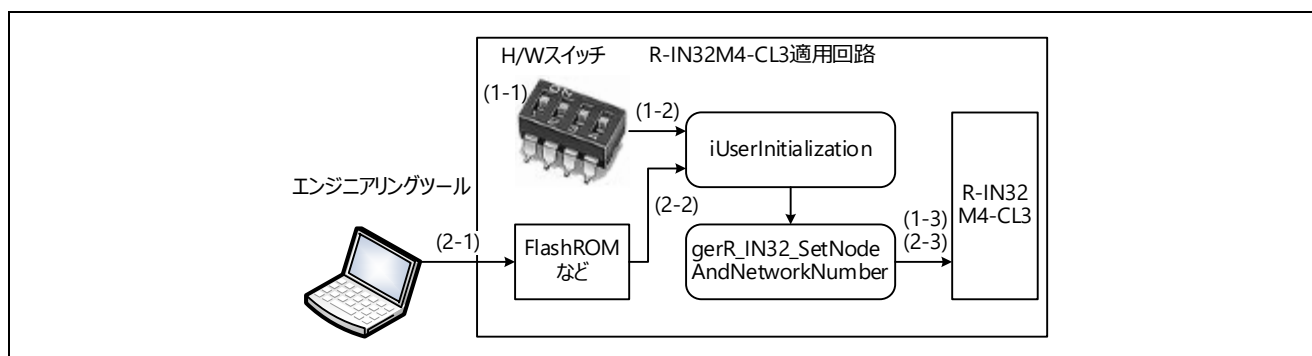


図 2.2 局番及びネットワーク No.の書き込みイメージ

表 2.2 例 1 : H/W スイッチを使用する方法

手順	内容
1-1	H/W スイッチを使用して局番およびネットワーク No.を設定します。
1-2	ユーザ・プログラムの iUserInitialization が、H/W スイッチの現在値を読み出してドライバ・インタフェース関数 gerR_IN32_SetNodeAndNetworkNumber の引数へセットします。 ユーザ・プログラムの iUserInitialization には、H/W スイッチの現在値を読み出す処理を記述していませんので、ユーザの仕様に合わせて読み出す処理を追加してください。
1-3	ドライバ・インタフェース関数の gerR_IN32_SetNodeAndNetworkNumber が、引数の値を R-IN32M4-CL3 に書き込みます。

H/W スイッチは、局番およびネットワーク No.の値域に対応したスイッチを選定してください。

表 2.3 H/W スイッチの範囲

項目	値域
局番	01h~78h (1~120)
ネットワーク No.	01h~EFh (1~239)

表 2.4 例 2 : エンジニアリング・ツールを使用する方法

手順	内容
2-1	エンジニアリング・ツールなどを使用して、局番およびネットワーク No.データを FlashROM などへ書き込みます。
2-2	ユーザ・プログラムの iUserInitialization が、FlashROM に書込まれたデータを読み出して、ドライバ・インタフェース関数 gerR_IN32_SetNodeAndNetworkNumber の引数へセットします。 ユーザ・プログラムの iUserInitialization には、FlashROM に書込まれたデータを読み出す処理を記述していませんので、ユーザの仕様に合わせて読み出し処理を追加してください。
2-3	ドライバ・インタフェース関数の gerR_IN32_SetNodeAndNetworkNumber が、引数の値を R-IN32M4-CL3 に書き込みます。

2.4 1000BASE-T コンプライアンス・テストモードの実装検討

CC-Link IE フィールド・ネットワーク接続対応製品は 1000BASE-T に準拠していますので、IEEE802.3ab の仕様に基づいた 1000BASE-T コンプライアンス・テストを実施する必要があります。

備考. 1000BASE-T コンプライアンス・テストの結果は、CC-Link 協会コンFORMANCE・テストで確認します。

1000BASE-T コンプライアンス・テストは、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。

ユーザが試験中の任意のタイミングでテスト用波形を切り替えられる機能や処理を、R-IN32M4-CL3 適用製品に実装することを検討してください。

例として、H/W スイッチで切り替える方法、または R-IN32M4-CL3 適用製品のエンジニアリング・ツール（周辺機器）などから切り替える方法などがあります。

どちらの方法も、ユーザ・プログラムの UserIEEETest（「4.4.1 H/W テスト（IEEE802.3ab コンプライアンス・テスト）」参照）から、R-IN32M4-CL3 ドライバ・インタフェース関数の IEEE802.3ab コンプライアンス・テスト gerR_IN32_IEEEtest を呼び出します。

表 2.5 例：1000BASE-T コンプライアンス・テスト・モード切替え方法

手順	内容
(1)	R-IN32M4-CL3適用製品の外部（H/Wスイッチなど）から、通常運転時のモード（オンラインモード）以外の「コンプライアンス・テスト・モード（オフラインモード）」に切り替えられる機能を実装する。
(2)	R-IN32M4-CL3適用製品の外部（H/Wスイッチなど）から、R-IN32M4-CL3ドライバ・インタフェース関数 gerR_IN32_IEEEtestの引数「MODE1～4」を指定できる処理を実装する。

2.5 エンジニアリング・ツール各種機能への対応準備

エンジニアリング・ツールを使うことで、「CC-Link IE フィールド・ネットワーク診断」と「スレーブ局パラメータ処理・コマンド実行」を実行することができます。R-IN32M4-CL3 適用製品（スレーブ局）の仕様として、エンジニアリング・ツール機能に対応するかをあらかじめ検討してください。

2.5.1 CC-Link IE フィールド・ネットワーク診断

エンジニアリング・ツールにより、CC-Link IE フィールド・ネットワークの状態をグラフィカルに表示できます。

詳細は「3.9 CC-Link IE フィールド・ネットワーク診断機能」を参照してください。

2.5.2 スレーブ局パラメータ処理・コマンド実行

エンジニアリング・ツールにより、R-IN32M4-CL3 適用機器のパラメータ設定とコマンドをプログラミングすることなく実施できます。

詳細は「2.9.1 スレーブ局パラメータ処理・コマンド実行」を参照してください。

なお、上記 2 つの機能はトランジェント伝送（SLMP フレーム）で行います。そのため、R-IN32M4-CL3 適用製品（スレーブ局）は、マスタ局からの SLMP 要求フレームに対して応答する必要があります。SLMP フレームの送受信を実装するかをあらかじめ検討してください。

表 2.6 エンジニアリング・ツールの機能

No.	エンジニアリング・ツールの機能	開発機器に必要な事項
1	CC-Link IE フィールド・ネットワーク診断	・ SLMP フレームの要求受信・応答送信処理
	a 選択局通信状態モニタ	
	b 交信テスト	
	c ケーブルテスト	
2	スレーブ局パラメータ処理・コマンド実行	・ CSP+を「2.9 CSP+の作成準備」で示す図の範囲③まで記述する。 ・ SLMP フレームの要求受信・応答送信処理

SLMP フレームの送受信については「4.3 ユーザ・プログラム詳細（トランジェント関連）」以降を参照してください。

（サンプルコードに SLMP フレームの送受信処理を記述していますのでご活用ください。）

2.6 エラー状態、エラー・コードの仕様検討

R-IN32M4-CL3 適用製品に異常が発生したとき、エラー状態を他局へ通知します。またトランジェント伝送において、要求フレームを異常受信したとき、エラー・コードを要求元へ通知します。R-IN32M4-CL3 適用製品の仕様に応じて、エラー状態およびエラー・コードの仕様をあらかじめ検討してください。

エラー状態の仕様は、「2.6.1 エラー状態の仕様検討」を参照してください。トランジェント伝送におけるエラー・コードの仕様は、「2.6.2 リターン・コード (RSTS) と終了コード (End Code) に格納するエラー・コードの仕様検討」を参照してください。

【トランジェント伝送以外の異常に対するエラー・コードについて】

R-IN32M4-CL3 の H/W 異常に関するエラー・コードとユーザ・アプリケーション部[※]異常のエラー・コードがあります。

注. ユーザ独自の機能部分 (I/O 部分、センサ部分、温調部分など)

R-IN32M4-CL3のH/W異常のエラー・コードは、R-IN32M4-CL3ドライバが検出しますが他局へ通知しません。検出後のエラー処理は任意です。（「5.5(1) gR_IN32_CallbackFatalError」参照）

ユーザ・アプリケーション部異常のエラー・コードは、他局へ通知しません。定義と実装は任意です。

2.6.1 エラー状態の仕様検討

R-IN32M4-CL3 ドライバが自局のエラー状態を MyStatus フレームのアプリケーション詳細エラー状態に格納して、他局へ通知します。（「3.8 MyStatus 送受信機能」参照）

R-IN32M4-CL3 適用製品の仕様に応じた、軽度/中度/重度以上の状態を定義してください。エラー状態を定義する際の参考として、シーケンサ CPU ユニットのエラー状態の分類を以下に示します。

- ・軽度異常：バッテリー異常など CPU ユニットが演算を続行するエラー
- ・中度異常：WDT エラーなど CPU ユニットが演算を停止するエラー（エラー・コード：1300～10000）
- ・重度異常：RAM 異常などの CPU ユニットが演算を停止するエラー（エラー・コード：1000～1299）
（ハードウェアの故障が疑われるなどの中度異常より、より重篤な異常を重度異常とする）

2.6.2 リターン・コード (RSTS) と終了コード (End Code) に格納するエラー・コードの仕様検討

R-IN32M4-CL3 適用製品が Transient2 要求フレームを異常受信した場合、要求元にエラー発生と原因を通知するために、応答フレームのリターン・コード (RSTS) にエラー・コードを格納して送信します。

ユーザが開発製品のユーザーズ・マニュアルでリターン・コードを確認・処置できるように、R-IN32M4-CL3 適用製品の仕様に応じたエラー・コードリターン・コード (RSTS) に格納してください。詳細は「7.3.3 リターン・コード (RSTS) 詳細」を参照してください。

R-IN32M4-CL3 適用製品が SLMP 要求フレームを異常受信した場合、Transient2 要求フレームと同様に、R-IN32M4-CL3 適用製品の仕様に応じたエラー・コードを終了コード (End Code) に格納してください。詳細は、「7.4.2 終了コード (End Code) 詳細」を参照してください。

2.7 リンクデバイスの属性の割り当て検討

リンクデバイスは、3つの属性グループに分類されます。R-IN32M4-CL3 適用製品の保有するリンクデバイスを、どの属性グループに割り当てるかをあらかじめ仕様を検討してください。（詳細は、CC-Link IE フィールド・ネットワーク仕様書（デバイスプロファイル編）を参照してください。）

2.7.1 ダイレクト入出力グループ

ダイレクト入出力グループは、リンクデバイスを特定の機能に使用しないで汎用入出力として使用します。（例えば、マスタ・ローカルユニットやリモート I/O ユニットなどの入出力に相当します。）

2.7.2 システム入出力グループ

システム入出力グループは、リンクデバイスをマスタ局と自局間のインタロックや自局の状態通知として使用します。（詳細は、「6. リンクデバイスのシステム領域」参照してください。）

表 2.7 リンクデバイスの属性定義例（システム領域）

リンクデバイス	名称	リンクデバイス	名称
RX07	ワーニング状態フラグ	—	—
RX08	イニシャルデータ処理要求フラグ	RY08	イニシャルデータ処理完了フラグ
RX09	イニシャルデータ設定完了フラグ	RY09	イニシャルデータ設定要求フラグ
RX0A	エラー状態フラグ	RY0A	エラーリセット要求フラグ
RX0B	リモート Ready	—	—

2.7.3 ベンダ入出力グループ

ベンダ入出力グループは、リンクデバイスを任意に定義して使用します。

表 2.8 リンクデバイスの属性定義例（インバータの場合）

リンクデバイス	名称	リンクデバイス	名称
RX10	正回転状態/停止状態	RY10	正回転指令/停止指令
RX11	逆回転状態/停止状態	RY11	逆回転指令/停止指令
RW00	出力周波数状態	RW00	出力周波数設定値

2.8 Hold/Clear 処理の実装検討

Hold/Clear 処理は、R-IN32M4-CL3 適用製品が外部へ出力制御を行なう製品の場合、マスタ局アプリケーションが停止/異常、またはデータリンクから解列したなどの理由でサイクリック伝送を停止したときに、出力を続行する（Hold）または出力を停止する（Clear）処理です。

以下の注意事項を考慮したうえで、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、Hold/Clear 処理の実装を検討してください。

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、「4.2.7 マスタ局からの MyStatus およびサイクリック受信処理」を参照してください。

マスタ局アプリケーションの状態は、MyStatus フレームを受信することで監視することができます。MyStatus フレームで得られるマスタ局アプリケーションの情報は、「3.8.2 MyStatus 受信」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、「4.2.10 通信状態更新処理」を参照してください。

注意. スレーブ局（自局）が受信するサイクリックデータは、R-IN32M4-CL3 ドライバ（gerR_IN32_GetReceivedCyclicData）で取得します。

取得するサイクリックデータは、マスタ局アプリケーションの動作/エラー状態、またはデータリンク状態によって内容が異なります。

取得するサイクリックデータの詳細は、「4.2.7 マスタ局からの MyStatus およびサイクリック受信処理」と、「4.2.10 通信状態更新処理」を参照してください。

2.9 CSP+の作成準備

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+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 フィールド・ネットワークの全局を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+の作成は、「CSP+作成支援ツール」をご利用ください。

インテリジェントデバイス局またはリモートデバイス局として CSP+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+は必ず作成してください。R-IN32M4-CL3 適用製品の仕様として、エンジニアリング・ツールのどの機能（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。

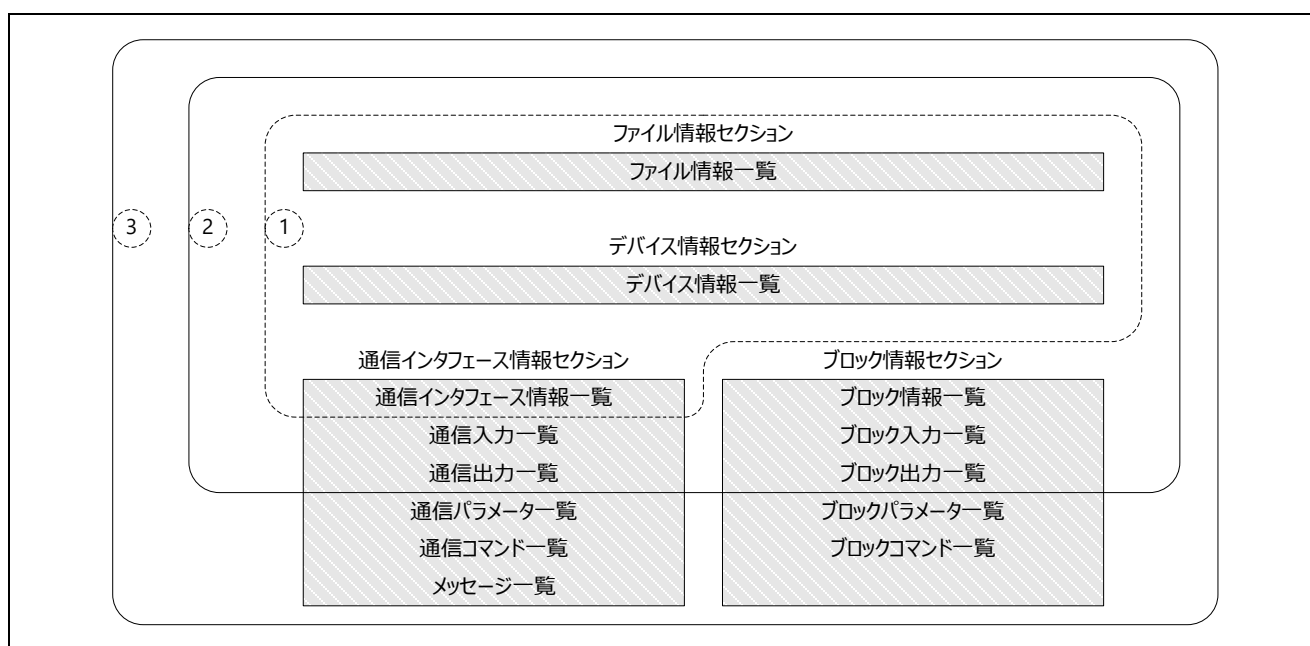


図 2.3 CSP+ファイルのセクション構成

範囲	内容	作成要否
①	CC-Link 協会コンフォーマンス・テストにおいて、必須項目確認のために必要な情報 【GX Works2、GX Works3 の場合】 CC IE Field 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	スレーブ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	スレーブ局のパラメータ処理・コマンド実行をするために必要な情報 ^注 【GX Works2、GX Works3 の場合】 CC IE Field 構成ウィンドウから CC-Link IE フィールド・ネットワーク接続対応製品のパラメータを簡単に設定できます。	任意

注. 詳細は「2.9.1 スレーブ局パラメータ処理・コマンド実行」を参照してください。

2.9.1 スレーブ局パラメータ処理・コマンド実行

R-IN32M4-CL3 適用製品がスレーブ局パラメータ処理・コマンド実行に対応することで、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。

【スレーブ局パラメータ処理の例】

三菱電機リモート I/O ユニット (NZ2GF2B1-***) の場合、以下のパラメータをプログラミングすることなく設定することができます。

名称	初期値	読出値	書込値	設定範囲	単位
局単位パラメータ					
☒ 入力応答時間設定	5:10ms	5:10ms	5:10ms		
☒ 出力HOLD/CLEAR設定	0: CLEAR	0: CLEAR	0: CLEAR		
☒ サイクリックデータ更新監視時間...	0	0	0	0~20	x100ms
☒ モード切換	9: 自動判定...	9: 自動判定...	9: 自動判定...		
☒ イニシャル動作設定	0: イニシャル...	0: イニシャル...	0: イニシャル...		
ユニット単位パラメータ					
☒ 同期入力タイミング取得機能					
同期入力タイミング取得機能...	0: 無効	0: 無効	0: 無効		
☒ 入力OFFディレイ設定					
入力OFFディレイ設定 X0	0	0	0	0~150000	x400us
入力OFFディレイ設定 X1	0	0	0	0~150000	x400us

図 2.4 CC IE Field 構成ウィンドウによるスレーブ局パラメータ処理例

スレーブ局パラメータ処理・コマンド実行は、以下の事項を満たすことで実現できます。

- ・CSP+を「2.9 CSP+の作成準備」で示す図の範囲③まで記述する。
- ・CSP+で記述した SLMP フレームの疎受信処理を R-IN32M4-CL3 適用製品に実装する。

2.10 コンフォーマンス・テスト

コンフォーマンス・テストは、CC-Link IE フィールド・ネットワーク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 フィールド・ネットワーク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・テスト仕様書を入手し、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 フィールド・ネットワーク接続対応製品は、推奨品として「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。

3. R-IN32M4-CL3 の機能

3.1 リセット機能

リセット機能は R-IN32M4-CL3 適用製品の CPU 部および GbE-PHY 部などを初期化する機能です。

本書に記載する「パワー・オン・リセット」および「システム・リセット」は、以下の発生要因によるリセットを指します。

【パワー・オン・リセット】

- ・PONRZ 端子入力によるリセット (R-IN32M4-CL3 適用製品内部 RAM の初期化処理を含む)

【システム・リセット】

- ・HOTRESETZ 端子入力によるリセット (内部 PLL を除く)
- ・システム・リセット・レジスタ (SYSRESET) によるリセット (HOTRESETZ 端子入力同等)

詳細は関連資料の「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。

3.2 ファストリンクアップ機能

ファストリンクアップ機能は、R-IN32M4-CL3 適用製品の電源 ON からリンクアップまでの時間を短縮する機能です。

GbE-PHY のオートネゴシエーション機能（自動的に最適な通信速度や通信モードを選択する機能）などを無効にすることにより、リンクアップ時間を短縮することができます。

使用例として、産業用ロボットのアーム先端ツールの交換機構（ツールチェンジャ）を使用したシステムなどがあります。

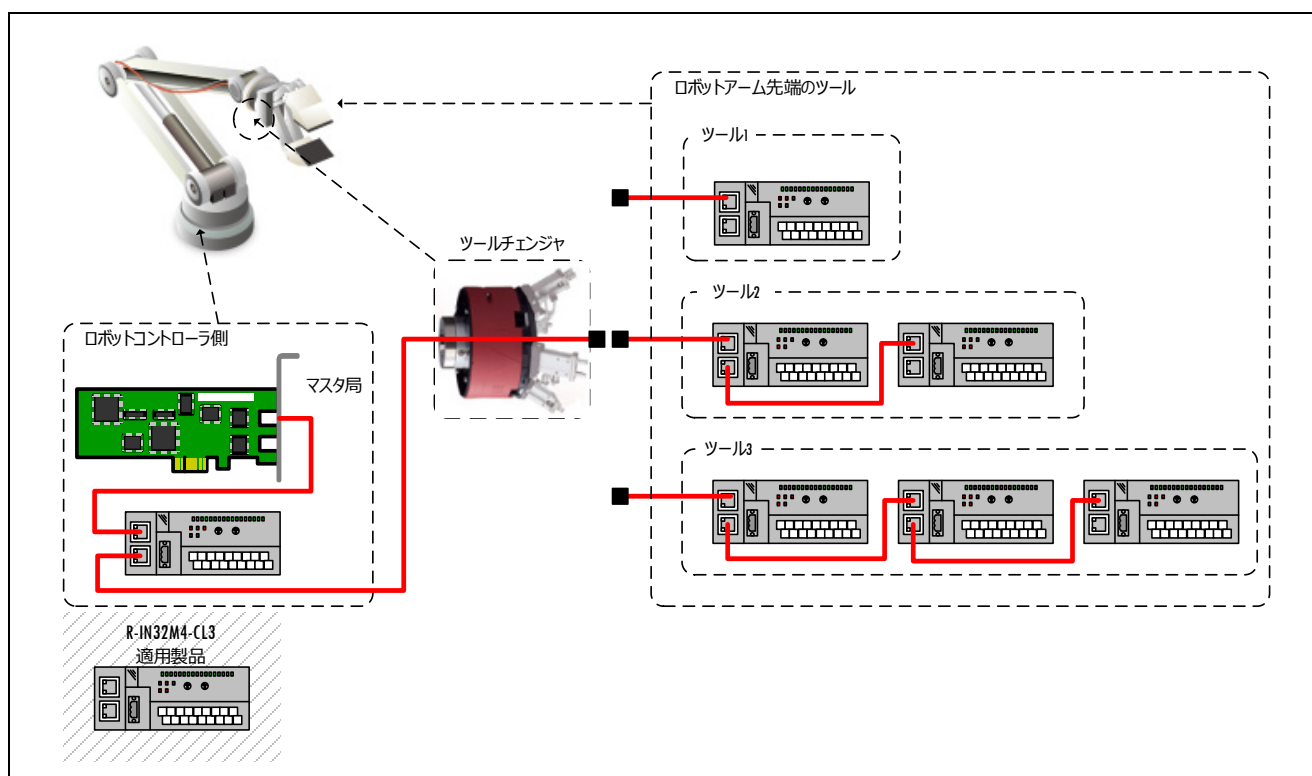


図 3.1 ファストリンクアップ機能使用例（産業用ロボットアームの場合）

本機能を実現するためには、以下の事項が必要です。

【H/W】

ファストリンクアップ機能有効/無効を切り替えるための H/W スイッチが必要です。詳細は「3.2.1 ファストリンクアップ機能の有効/無効設定」を参照してください。

【S/W】

ユーザプログラムの iUserInitialization（「4.2.2 初期化処理」参照）に、上述の H/W スイッチの値を読み出して有効/無効を判別する処理を追加してください。

（H/W スイッチを使用しない場合、R-IN32M4-CL3 適用製品のエンジニアリング・ツール（周辺機器）などからファストリンクアップ機能有効/無効を書き込む処理を追加してください。）

有効の場合、R-IN32M4-CL3 の初期化時にファストリンクアップの設定（PHY 設定初期値）を行う必要があります。「5.3.1(2) gerR_IN32_Initialize」、および「表 5.7R_IN32_UNITINIT_T 一覧」の「PHY 設定初期値」を参照してください。

【注意事項】

本機能は、自局ポートと隣接する他局ポートの双方が機能を有効に設定することで、そのポート間のリンクアップ時間が短縮されます。機能無効のポートあるいは機能未対応局のポートと、機能有効のポートを接続してもリンクアップできません。

3.2.1 ファストリンクアップ機能の有効/無効設定

ファストリンクアップ機能を実装する場合、ポートごとに本機能を有効/無効に設定する方法が必要です。

隣接する局がファストリンクアップ機能に未対応、もしくは無効の場合リンクアップできません。そのため、隣接する局が本機能に未対応、または対応しているか不明である場合を考慮して、エンドユーザが簡単にファストリンクアップ機能を有効/無効に切り替える方法が必要です。

参考として、R-IN32M4-CL3 適用回路にファストリンクアップ有効/無効設定スイッチを実装する場合を例示します。ここで示す設定スイッチはサンプルソフトの `get_board_sw` 関数で状態取得できる 8 ビットの汎用 H/W スイッチです。

表 3.1 ファストリンクアップ有効/無効設定スイッチ

ビット	名称	内容
0	F LINK P1	ON : PORT1 のファストリンクアップ機能有効、 OFF : PORT1 のファストリンクアップ機能無効
1	F LINK P2	ON : PORT2 のファストリンクアップ機能有効、 OFF : PORT2 のファストリンクアップ機能無効
2	- (Reserve)	将来拡張
3	- (Reserve)	将来拡張
4	- (Reserve)	将来拡張
5	- (Reserve)	将来拡張
6	- (Reserve)	将来拡張
7	- (Reserve)	将来拡張

表 3.2 ファストリンクアップ機能の有効/無効設定方法

手順	内容
(1)	ファストリンクアップ有効/無効設定スイッチに、PORT1、PORT2 それぞれの有効/無効を設定します。
(2)	ユーザプログラムの <code>iUserInitialization</code> で、 <code>get_board_sw</code> 関数により H/W スイッチの状態を取得します。
(3)	ユーザプログラムの <code>iUserInitialization</code> で、ドライバ・インタフェース関数 <code>gerR_IN32_Initialize</code> の PHY 設定初期値（「表 5.8 R_IN32_PHY_SETTING_T 一覧」参照）を以下のように設定し、R-IN32M4-CL3 を初期化します。 【手順(2)の取得値のビット 0】 ・ ON（有効）の場合は、PORT1 の MDI 設定を強制 MDI、1000BASE-T Clock 設定を強制マスタに設定 ・ OFF（無効）の場合は、PORT1 の MDI 設定、1000BASE-T Clock 設定をオートに設定 【手順(2)の取得値のビット 1】 ・ ON（有効）の場合は、PORT2 の MDI 設定を強制 MDI-X、1000BASE-T Clock 設定を強制スレーブに設定 ・ OFF（無効）の場合は、PORT2 の MDI 設定、1000BASE-T Clock 設定をオートに設定
(4)	R-IN32M4-CL3 ドライバは手順(3)の設定に従いリンクアップします。

【注意事項】

本機能の有効/無効の状態は、R-IN32M4-CL3 適用製品起動時の上記スイッチの状態により決定されます。R-IN32M4-CL3 適用製品起動後にファストリンクアップ有効/無効設定スイッチを変更しても、本機能の有効/無効の状態は変更されません。起動後に行った設定変更を反映するためには、電源の ON→OFF→ON が必要です。

3.2.2 ファストリンクアップ時のシステム構成

ファストリンクアップ機能を使用する場合のシステム構成を、産業用ロボットアームの場合で例示します。

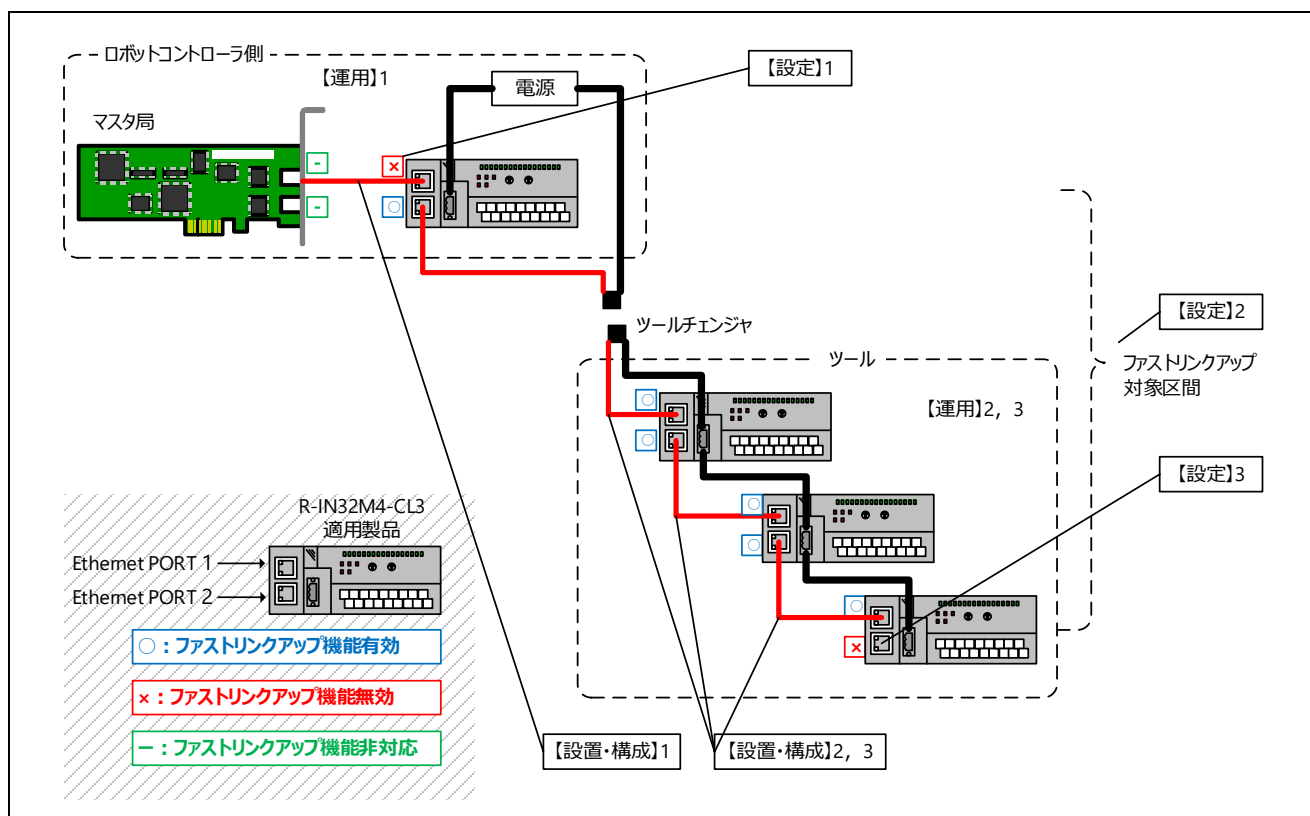


図 3.2 ファストリンクアップ時のシステム構成（産業用ロボットアームの場合）

【設置・構成】

1. ロボットコントローラ側 R-IN32M4-CL3 適用製品の PORT1 をマスタ局に接続、PORT2 をツール側に接続してください。

R-IN32M4-CL3 適用製品は、1000BASE-T のクロック設定において PORT1 をマスタ、PORT2 をスレーブに固定します。また、クロック設定は、スレーブを先、マスタを後の順に行います。クロック設定を PORT2（スレーブ）から開始させることで、ツールとのリンクアップ時間を短縮させます。

2. R-IN32M4-CL3 適用製品同士をファストリンクアップ接続する場合、上流局の PORT2 と下流局の PORT1 とを接続してください。

上述のとおり、R-IN32M4-CL3 適用製品は、1000BASE-T のクロック設定において PORT1 をマスタ、PORT2 をスレーブに固定します。そのため、ファストリンクアップ機能が有効であっても PORT1 同士または PORT2 同士を接続した場合、リンクアップしません。

3. ツール内の R-IN32M4-CL3 適用製品はライン接続してください。

ライン接続以外の伝送路形式によってはリンクアップの時間が遅くなる場合があります。また、接続台数もリンクアップ時間の要因になります。

【設定】

1. ファストリンクアップ機能非対応局（マスタ局）と接続する R-IN32M4-CL3 適用製品の PORT 1 は、ファストリンクアップ機能を無効に設定してください。
ファストリンクアップ機能有効のポートをファストリンクアップ機能非対応のポート、または無効のポートと接続した場合、リンクアップしません。
2. ファストリンクアップ対象区間に含まれるすべての R-IN32M4-CL3 適用製品のポートは、ファストリンクアップ機能を有効に設定してください。
3. R-IN32M4-CL3 適用製品をツール終端とした場合、その未接続ポートはファストリンクアップ機能を無効に設定してください。

【運用】

1. ロボットコントローラ側のマスタ局と R-IN32M4-CL3 適用製品は、常時電源 ON で運用してください。
2. ツールの切り替え後、ツール内すべての R-IN32M4-CL3 適用製品の電源を同時に ON してください。
3. ツール切替え後の再接続（ケーブル接続と電源 ON）は、マスタ局がツール側の解列（切離し）を検知^注した後に行ってください。

解列を検知する前に接続した場合、リンクアップまでの時間が遅くなる場合があります。

R-IN32M4-CL3 内蔵 GbE-PHY は、ツール切り離し後から解列を検知するまでに約 0.4～0.8 秒かかります。解列検知前に接続（電源 ON）すると、正常なタイミングで解列の検知ができないため、リンクアップが遅れます。

注. マスタ局の各局データリンク状態（SW00B0～SW00B7）を監視することで、解列を検知できます。

3.2.3 ファストリンクアップ機能の注意事項

ファストリンクアップ機能を使用した場合、短縮する時間は R-IN32M4-CL3 適用回路およびユーザプログラムに依存します。

下記の項目が短くなるように設計してください。

- ・リセット解除までの時間
- ・ブートメモリからの F/W のロード時間
- ・ファストリンクアップ関連の F/W の設定時間

サンプルソフトではファストリンクアップ機能として、「表 3.2 ファストリンクアップ機能の有効/無効設定方法」で例示したファストリンクアップ設定の他に、ブートメモリからの F/W のロード時間の短縮を目的として、リードモードをファスト・リードとしたシリアルフラッシュの読み出しを適用しています。サンプルソフトのファストリンクアップ機能を使用する場合は、ビルド構成で“SFlash Boot FastLinkUp”を選択して下さい。

3.3 状態表示機能

3.3.1 LED による状態表示

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。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。

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 3.3 LED 状態表示一覧

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
	RD		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
	SD		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
	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、並びにループバックの状態を表示します。 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を実施中
		消灯	正常なデータを受信、またはループバックを未実施
	User LED1、2		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を実施中
		消灯	正常なデータを受信、またはループバックを未実施
Ethernet PORT2 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を実施中
		消灯	正常なデータを受信、またはループバックを未実施

3.3.2 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 があります。

H/W で制御する LED は、R-IN32M4-CL3 の CPU 部または GbE-PHY 部が、自局の状態に応じて該当する LED を点灯/消灯します。S/W による制御が不要です。

S/W で制御する LED は、ユーザ・プログラムから呼び出す R-IN32M4-CL3 ドライバ・インタフェース関数が、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は、「5.3.7 LED 制御」を参照してください。）

以下に LED の制御元と、リセット/エラー時の LED 制御の一覧を示します。

表 3.4 LED 制御一覧表

種別	LED 名称	出力信号名	H/S 区分	制御元	パワーマスター・ リセット ¹	シブスター・ リセット ¹	内部 WDT エラー ² 外部 WDT エラー ² 自局エラー ²
自局状態表示	RUN	CCI_RUNLEDZ	H/S	CPU 部/ドライバ	消灯	消灯	消灯
	RD	CCI_RDLEDZ	H	CPU 部	消灯	—	—
	SD	CCI_SDLEDZ	H	CPU 部	消灯	—	—
	D LINK	CCI_DLINKLEDZ	H/S	CPU 部/ドライバ	消灯	消灯	消灯
	ERR.	CCI_ERRLEDZ	H/S	CPU 部/ドライバ	消灯	消灯	点灯
	L ERR.	—	H/S	CPU 部/ドライバ	—	—	—
	User LED1	RP20	S	ドライバ	消灯	消灯	消灯
	User LED2	RP21	S	ドライバ	消灯	消灯	消灯
Ethernet PORT1 状態表示	LINK	PHY0_LED0	H	GbE-PHY 部	消灯	消灯	—注3
	L ER	CCI_LERR0LEDZ	H/S	CPU 部/ドライバ	消灯	消灯	消灯
Ethernet PORT2 状態表示	LINK	PHY1_LED0	H	GbE-PHY 部	消灯	消灯	—注3
	L ER	CCI_LERR1LEDZ	H/S	CPU 部/ドライバ	消灯	消灯	消灯

備考. H : H/W で制御、S : S/W で制御、H/S : H/W と S/W で制御

ドライバ : R-IN32M4-CL3 ドライバ・インタフェース関数「5.3.7 LED 制御」参照

注 1. リセットの詳細は「3.1 リセット機能」を参照してください。

2. ユーザ・アプリケーションの都合によるエラーです。

詳細は「4.2.4 自局エラー処理」を参照してください。

3. バイパスモード状態であれば点灯します。

3.3.3 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 UserLED 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_IN32_SetUSER1LED`、および `gerR_IN32_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「5.3.7 LED 制御」を参照してください。

3.3.4 LED の点灯機能有効/無効

下表の LED は、点灯機能の有効/無効を切り替えることができます。

表 3.5 LED の有効/無効の切り替え可能な LED

LED 名称	内容
RUN	動作状態表示
ERR	エラー状態表示
D LINK	データリンク状態表示
User LED 1、User LED 2	ユーザが定義した状態表示
L ER 1、L ER 2	Ethernet PORT1、2 受信データエラー状態表示

LED の点灯機能有効/無効は、必要に応じてユーザで仕様を決定してください。

備考.（例）Ethernet PORT1、2 の L ER.LED は、リンクダウンした場合にも点灯を続けることがあるため、リンクダウン中は無効に切り替える。

LED の点灯機能を無効にするには、`gerR_IN32_DisableLED` 関数を使用します。

LED の点灯機能を有効にするには、`gerR_IN32_EnableLED` 関数を使用します。

`gerR_IN32_DisableLED` 関数および `gerR_IN32_EnableLED` 関数の詳細は、「5.3.7 LED 制御」を参照してください。

3.4 バイパスモード

バイパスモードは、ライン型/リング型での接続時に、システム・リセットを実施した場合や、通信に影響を与えるエラーが発生した場合でも、自局からの下流局の通信に影響を与えないようネットワークの接続（リンクアップ）を維持する機能です。下記の要因発生時に R-IN32M4-CL3 が自局をバイパスモードに移行します。

【バイパスモードになる要因】

- ・システム・リセット（「3.1 リセット機能」を参照してください）
- ・WDT エラー（内部 WDT エラー、または CCI_WDTIZ（P12 端子兼用）入力時）
- ・自局エラー（ユーザ・アプリケーションの都合によるエラーです。

詳細は「4.2.4 自局エラー処理」を参照してください。）

3.5 MIB 情報

MIB 情報は、R-IN32M4-CL3 が Ethernet PORT1、PORT2 のフレーム受信回数やエラーフレーム受信回数などを集計した、通信状態を管理する情報です。

ユーザ・プログラムは、MIB 情報を利用することで、自局 Ethernet PORT の通信状態を把握することができます。MIB 情報の詳細については、「4.2.13 MIB 情報取得処理」を参照してください。

3.6 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。

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。

サイクリックデータの流れを以下に例示します。

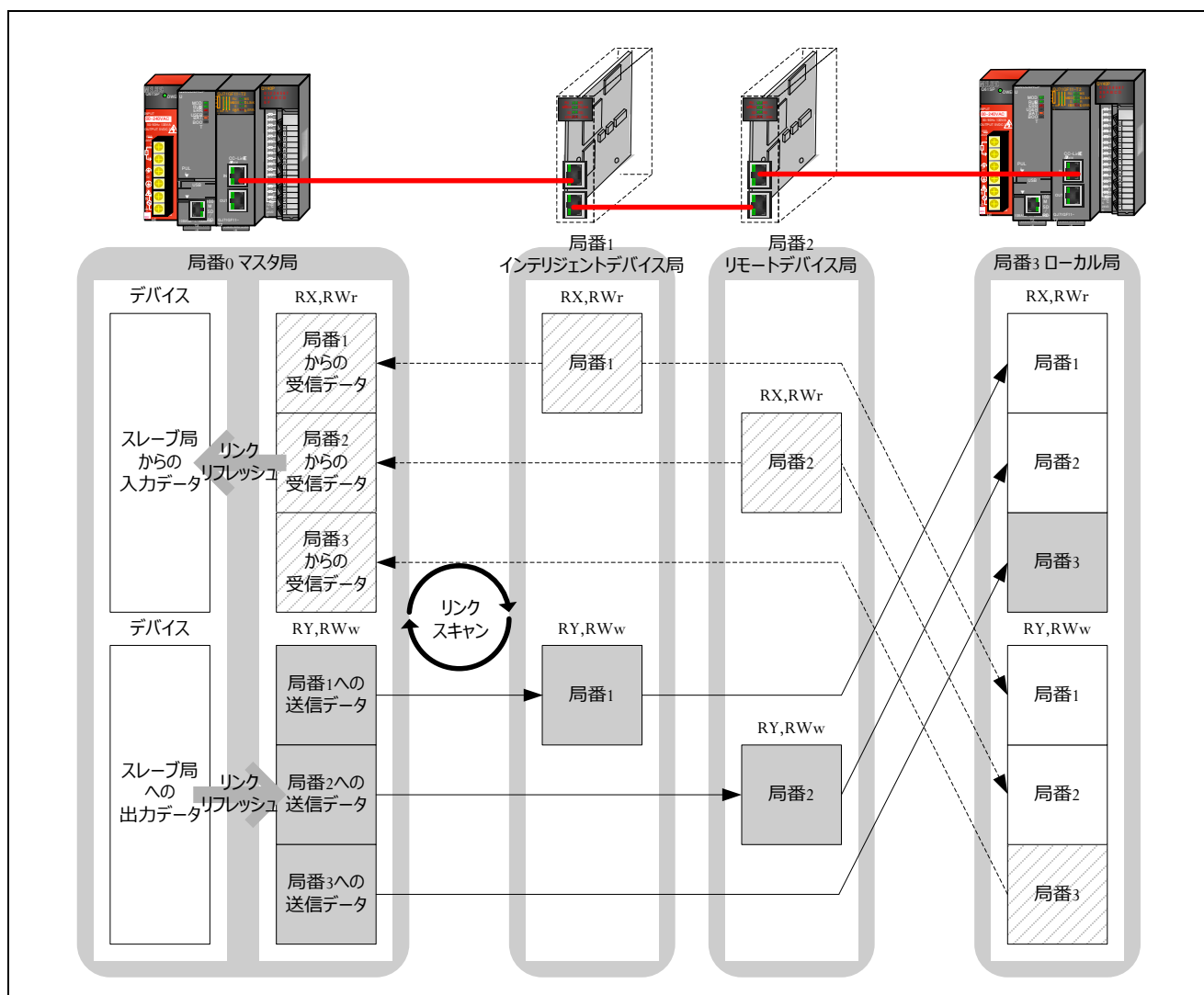


図 3.3 サイクリックデータの流れ

サイクリック伝送の受信は「4.2.7 マスタ局からの MyStatus およびサイクリック受信処理」、送信は「4.2.9 サイクリック送信処理」を参照してください。

3.7 トランジェント伝送機能

トランジェント伝送は、多極からの更新要求があったとき、または自局が更新要求したとき^注に、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。

注. R-IN32M4-CL3 適用製品のエンジニアリング・ツール（周辺機器）が読出し/書き込み命令などを行ったとき

トランジェント伝送を使うことで、サイクリック伝送よりも簡単に送受信できる場合があります。

- ・ 自局/他局リンクデバイス点数を超える大容量データを読み出す/書き込む場合
- ・ 自局/他局のリンクデバイスに汎用データ（エラー履歴、パラメータ設定値など）の送受信エリアがない場合

読み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。

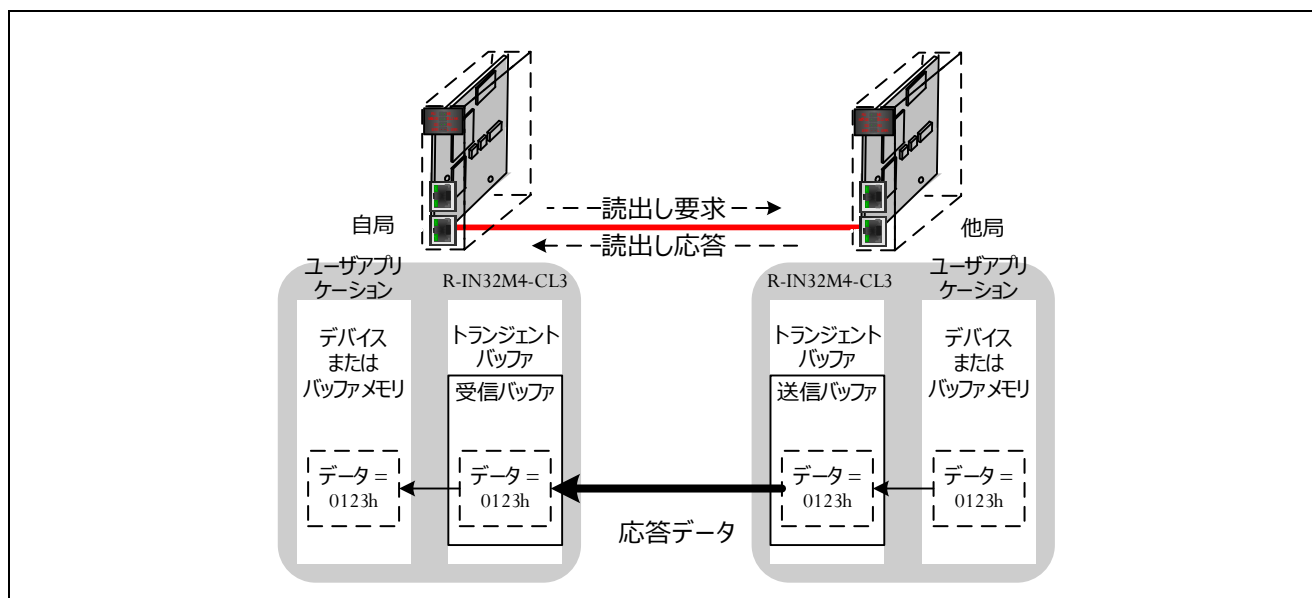


図 3.4 トランジェントデータの流れ

トランジェント伝送の処理は、「4.3 ユーザ・プログラム詳細（トランジェント関連）」以降を参照してください。

注意. トランジェント伝送機能を使用するために、初期設定において bITransientReceiveEnable（トランジェント受信機能）を R_IN32_TRUE に設定してください。（「表 5.7 R_IN32_UNITINIT_T 一覧」の No.6 を参照）

3.7.1 トランジェント伝送のクライアント・サーバ機能について

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。

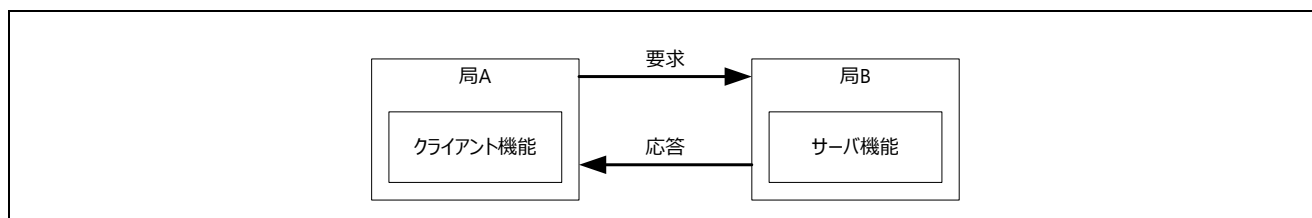


図 3.5 トランジェント クライアント/サーバ機能

インテリジェントデバイス局はクライアント機能とサーバ機能を実装することができます。

リモートデバイス局はサーバ機能のみ実装することができます。

クライアント・サーバ機能の処理概要は、「4.3.1 トランジェント伝送の処理概要」を参照してください。

3.7.2 トランジェント伝送のフレームについて

R-IN32M4-CL3 がサポートするトランジェント伝送のフレーム一覧、およびフレームごとの送受信処理の実装要否を表 3.6 に示します。

表 3.6 トランジェント伝送のフレーム一覧と実装要否

No.	フレーム名称 ^{注1}	フレーム種別 (Ftype)		データ種別 (DataType)		データサブ種別 (DataSubType)		実装 要否
1	CC-Link IEフィールド 固有トランジェント	22h	Transient1	07h	CC-Link IE フィールド固有	0002h	システム固有	必須
2	SLMP	22h	Transient1	05h	ネットワーク共通	0002h	SLMP	任意
3	CC-Link互換トランジェント	25h	Transient2	04h	CC-Link互換	—	—	任意
4	TransientAck	23h	TransientAck	注2	注2	注2	注2	任意

注 1. 本書に限り、各フレームを上記名称で記載します。

2. 受信フレームより抽出します。

1. CC-Link IE フィールド固有トランジェントは、マスタ局がスレーブ局の情報収集とネットワークを管理するために使用するフレームです。
2. SLMP は、エンジニアリング・ツールを使った拡張機能（CC-Link IE フィールド・ネットワーク診断、スレーブ局パラメータ処理・コマンド実行など）で使用するフレームです。
3. CC-Link 互換トランジェントは、主にユーザ製品間の通信で使用するフレームです。CC-Link のトランジェント・フレームと互換性のあるフレームです。
4. TransientAck は、Transient1 および Transient2 フレームを受信したとき、送信元へ確認応答するために使用します。

3.7.3 トランジェント伝送のコマンドについて

R-IN32M4-CL3 適用製品がサポートするトランジェント伝送のコマンド一覧、およびコマンドごとのクライアント・サーバ機能の実装要否を下表に示します。リモートデバイス局は、下表のコマンドのクライアント機能が不要です。

表 3.7 トランジェント伝送のコマンド一覧と実装要否

フレーム名称	コマンド種別	インテリジェントデバイス局		リモートデバイス局	備考
		クライアント機能	サーバ機能	サーバ機能	
CC-Link IE フィールド固有 トランジェント	ノード情報配信	×	△	×	注 1、6
	統計情報取得	×	△	△	注 2、6
	ノード詳細情報取得	×	◎	◎	注 3、6
	オプション情報取得	×	○	○	注 4、6
SLMP	選択局情報取得	×	○	○	注 5、6、7
	交信テスト	×	○	○	
	ケーブルテスト	×	○	○	
	リモートリセット	×	△	△	
	メモリ読出し	△	△	△	—
	メモリ書込み	△	△	△	—
CC-Link 互換 トランジェント	メモリアクセス情報取得	△	△	△	注 8
	RUN	△	△	△	—
	STOP	△	△	△	—
	メモリ読出し	△	△	△	RIRD 相当
	メモリ書込み	△	△	△	RIWT 相当

備考. ◎：必須 ○：推奨 △：任意 ×：不要

注 1. ノード情報配信は、マスタ局が局番と MAC アドレスとの対応を示した情報をマルチキャスト配信するコマンドです。インテリジェントデバイス局が他局へトランジェント要求を送信する場合に、ノード情報を使用します。なお、本コマンドに対する TransientAck および応答は不要です。配信されるノード情報を受信する処理のみ必要です。

2. 統計情報取得は、マスタ局がスレーブ局の Ethernet PORT1、2 に関するエラー情報を収集するコマンドです。

3. ノード詳細情報取得は、マスタ局がスレーブ局の固有情報を収集するコマンドです。

ノード詳細情報は、「表 5.6 R_IN32_UNITINFO_T 一覧」を参照してください。

4. オプション情報取得は、マスタ局がスレーブ局のオプション有無を確認するコマンドです。

オプション情報は、SLMP フレーム送受信および CC-Link IE フィールド・ネットワーク診断などの CC-Link IE フィールド・ネットワークの拡張機能に対応していることを示す情報です。

5. 「3.9 CC-Link IE フィールド・ネットワーク診断機能」を参照してください。

6. R-IN32M4-CL3 ドライバ・インタフェース関数（「5.3.11 トランジェント要求受信対応」参照）が、受信処理を行います。

7. CC-Link IE フィールド・ネットワーク診断に必要

8. アクセス・コードを使用する場合に必要

3.8 MyStatus 送受信機能

MyStatus 送受信は、R-IN32M4-CL3 が MyStatus フレームを定期的に送受信する機能です。

MyStatus はネットワークに接続する局の状態通知に使用されます。MyStatus を送受信することで自局の状態をマスタ局へ通知およびマスタ局の状態を監視することができます。

3.8.1 MyStatus 送信

ユーザ・プログラム UserSendMyStatus（「4.2.8 MyStatus 送信処理」参照）において、ユーザが下表の情報（自局の状態）を R-IN32M4-CL3 ドライバ・インタフェース関数 gerR_IN32_SetNodeStatus（自局状態設定）の引数へセットすることで、自局の状態をマスタ局へ通知することができます。

表 3.8 MyStatus 送信に関する情報

No.	項目	内容
1	アプリケーション詳細動作状態	ユーザ・アプリケーションの動作状態を格納する。 0000h：アプリケーション詳細動作状態通知未サポート 0001h：アプリケーションが停止中 0002h：アプリケーションが動作中 0003h：アプリケーション実体が存在しない 上記以外：未使用
2	アプリケーション詳細エラー状態	ユーザアプリケーションエラー発生時のエラー状態を格納する。 0000h：エラーなし 0001h：軽度異常 0002h：中度異常 0003h：重度異常 上記以外：未使用

3.8.2 MyStatus 受信

ユーザ・プログラム UserReceiveCyclic（「4.2.7 マスタ局からの MyStatus およびサイクリック受信処理」参照）において、ユーザが下表の情報（マスタ局の状態）を R-IN32M4-CL3 ドライバ・インタフェース関数 gerR_IN32_GetMasterNodeStatus（マスタ局状態取得）の指定したアドレスから読み出すことで、マスタ局の状態を監視することができます。

表 3.9 MyStatus 受信に関する情報

No.	項目	内容
1	マスタ局アプリケーション動作状態	マスタ局のアプリケーションの動作状態が格納される。 0b : アプリケーションが停止中 1b : アプリケーションが動作中
2	マスタ局アプリケーションエラー状態	マスタ局のアプリケーションのエラー状態が格納される。 0b : エラーなし 1b : エラーあり

備考. 三菱電機製マスタ局の場合、マスタ局のアプリケーションの状態に、シーケンサ CPU ユニットの以下の状態が格納されます。

1. マスタ局アプリケーション動作状態

【アプリケーションが停止中】

シーケンスプログラムの演算停止

（RUN/STOP スイッチが「STOP」で停止中のとき、または中度・重度エラー発生中のとき）

【アプリケーションが実行中】

シーケンスプログラムの演算実行

（RUN/STOP スイッチが「RUN」で運転中のとき）

2. マスタ局アプリケーションエラー状態

【エラーなし】

エラーがない場合、および

バッテリエラーなどの CPU ユニットが演算を続行するエラー（軽度異常）

【エラーあり】

WDT エラーなどの CPU ユニットが演算を停止するエラー（中度異常）、および

ハードウェアの故障などの CPU ユニットが演算を停止するエラー（重度異常）

3.9 CC-Link IE フィールド・ネットワーク診断機能

CC-Link IE フィールド・ネットワーク診断機能は、エンジニアリング・ツールを使って、CC-Link IE フィールド・ネットワークの状態をグラフィカルに表示する機能です。異常箇所、異常原因と処置方法、イベント履歴がエンジニアリング・ツール上で確認することができます。機能の詳細はマスタ・ローカルユニットのユーザーズ・マニュアルを参照してください。

本機能は、マスタ局からの SLMP フレームの要求に応答することで、CC-Link IE フィールド・ネットワーク診断画面に R-IN32M4-CL3 適用製品が表示されます。また、各種のテスト・操作を実行することができます。

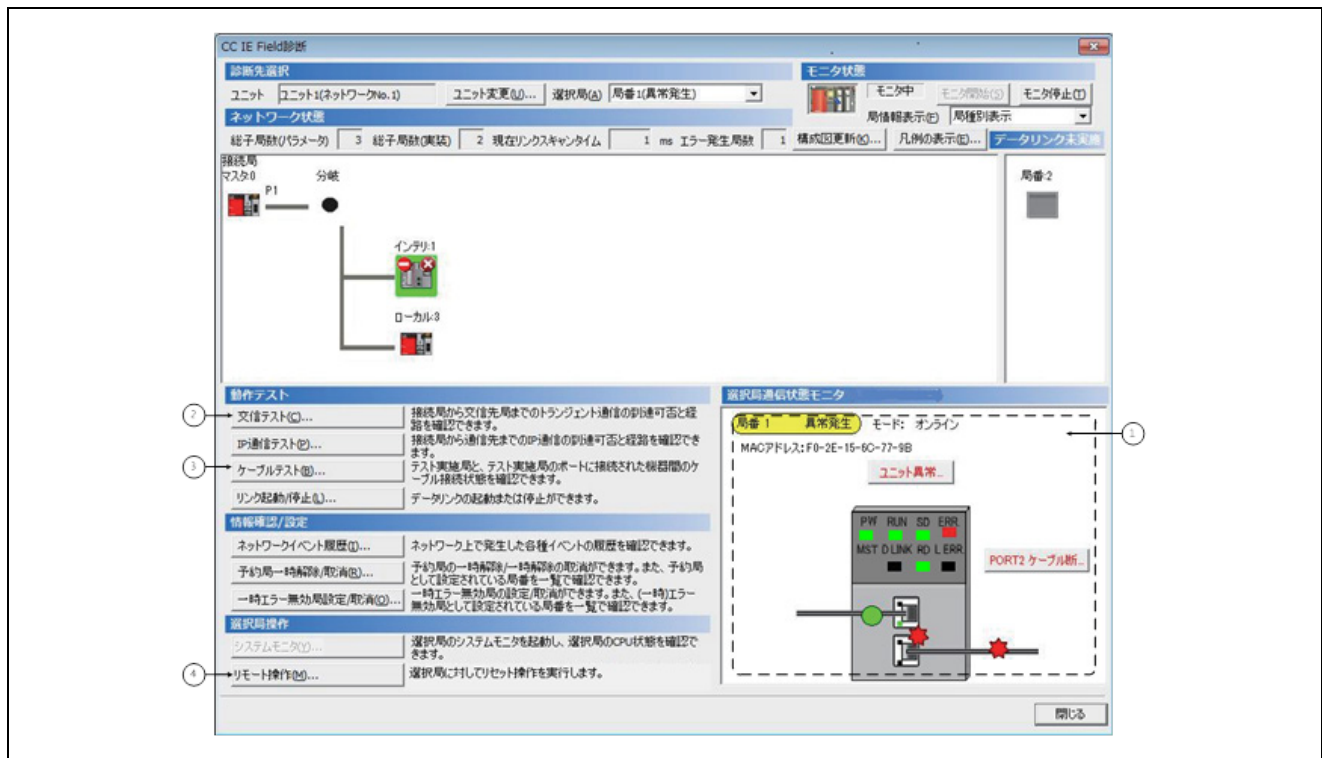


図 3.6 診断画面の表示/操作箇所（GX Works2 の場合）

表 3.10 診断画面の表示/操作箇所と SLMP 要求との対応

No.	項目	内容	SLMP 要求フレーム (Command)
1	選択局通信状態モニタ	選択局の状態と異常内容を表示します。	選択局通信状態取得要求 (3119h)
2	交信テスト	自局から交信先までのトランジェント伝送の通信経路をテストします。	交信テスト要求 (3040h)
3	ケーブルテスト	ケーブルの断線や未接続をテストします。	ケーブルテスト要求 (3050h)
4	リモート操作	R-IN32M4-CL3 適用製品の状態を H/W スイッチ操作なしでリセットします。	リモートリセット要求 (1006h)

【SLMP 要求フレームの応答】

ユーザ・プログラムの UserHandleReceivedTransient1（「4.3.3 Transient1 受信データ処理」参照）にて、R-IN32M4-CL3 ドライバ・インタフェース関数（「5.3.11 トランジェント要求受信対応」参照）が、該当する SLMP フレームの応答処理（要求フレーム受信対応処理）を行います。

サンプルコードには表 3.10 No.1～4 の処理を記述しています。（表 3.10 No.1～3 の実装を推奨します。）

3.9.1 選択局通信状態モニタの LED

UserHandleReceivedTransient1（「4.3.3 Transient1 受信データ処理」参照）にて LED 情報を作成し、選択局通信状態取得要求に応答することで、選択局通信状態モニタに自局の LED 状態を表示させることができます。

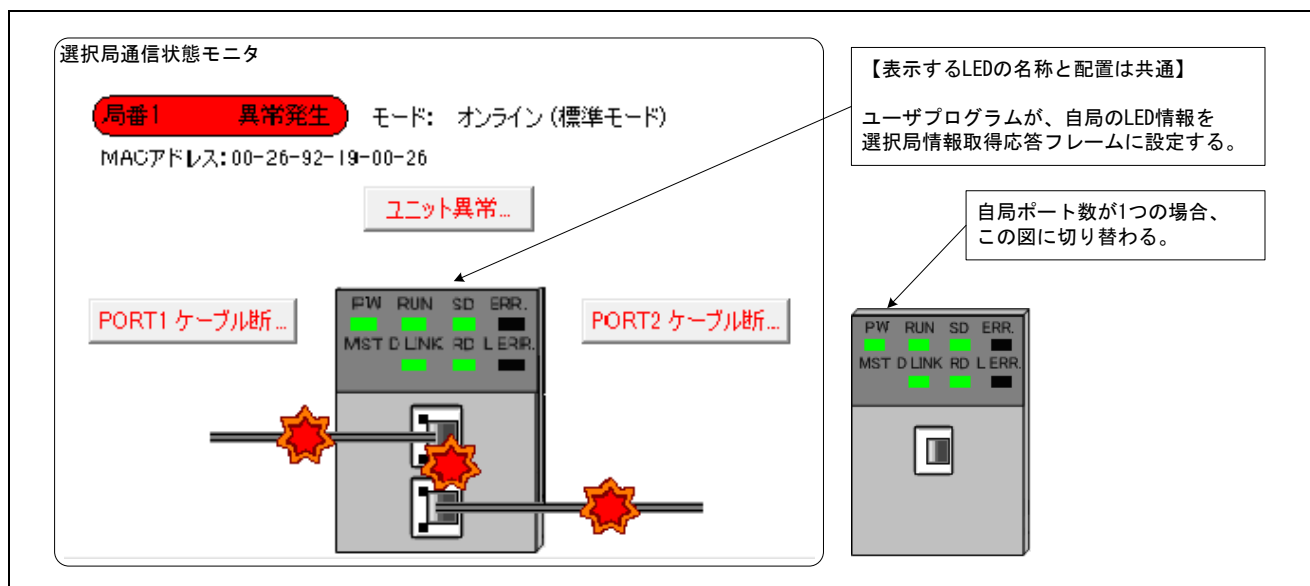


図 3.7 選択局通信状態モニタ表示例

【選択局通信状態モニタの LED 使用例】

エンド・ユーザのトラブル・シューティングにおいて、R-IN32M4-CL3 適用製品の LED 状態が目視できないなどの場合に、CC-Link IE フィールド・ネットワーク診断を使うことで LED 状態を確認することができます。

【表示できる LED】

選択局通信状態モニタに表示できる LED 名称^注と LED 配置は、図 3.7 のとおりです。

注. PW、RUN、SD、ERR.、MST（未使用のためグレーアウト）、D LINK、RD、L ERR.

LED 情報の作成の詳細は、「4.3.3(3) 選択局情報取得フレームの LED について」を参照してください。

4. ユーザ・プログラムの作成

本章は、ユーザ・プログラムの処理概要とカスタマイズについて示します。

ユーザ・プログラムは、インテリジェントデバイス局またはリモートデバイス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。ユーザの仕様に応じてカスタマイズしてください。

4.1 ユーザ・プログラム一覧

「¥cie_intelligent_device¥root¥Japanese¥sample」、または「¥cie_remote_device¥root¥Japanese¥sample」フォルダ以下のユーザ・プログラムの一覧を示します。

また、インテリジェントデバイス局（ID 局）とリモートデバイス局（RD 局）の実装可否を示します。

注意 1. サンプルソフトのフォルダ構成およびファイル名は変更となる可能性があります。

2. R_IN32M4_Transient.c ファイルでは、コマンドごとのサンプル処理として、Transient2 メモリ読出し/書き込み、および SLMP メモリ読出し/書き込みを記述しています。

上記以外のコマンドを実装する場合は、「7 フレーム・フォーマット」と、関連マニュアル「SLMP リファレンス・マニュアル（SH（名）-080931）」を参照して、コマンドごとの処理を追加してください。

表 4.1 ユーザ・プログラム一覧（イニシャル・サイクリック伝送関連）

(1/2)

No.	ファイル	項目	機能	実装可否	
				ID 局	RD 局
1	R_IN32M4_sample.c	iUserMainRoutine	メイン処理	○	
2		iUserInitialization	初期化処理	○	
3		iUserStart	通信開始処理	○	
4		UserForceStop	自局エラー処理	△	
5		UserStopCyclic	サイクリック伝送停止処理	△	
6		iUserExecuteMain	イベント処理	○	
7		UserReceiveCyclic	マスタ局からの MyStatus およびサイクリック受信処理	○	
8		UserSendMyStatus	MyStatus 送信処理	○	
9		UserSendCyclic	サイクリック送信処理	○	
10		UserUpdateStatus	通信状態更新処理	○	
11		UserUpdateLed	LED 更新処理	○	
12		UserGetCyclicStatus	サイクリック伝送状態更新処理	△	
13		UserGetMIB	MIB 情報取得処理	△	

備考. ○ : 必須、△ : 任意、× : 不要

表 4.2 ユーザ・プログラム一覧（トランジェント伝送関連）

(2/2)

No.	ファイル	項目	機能	実装要否	
				ID 局	RD 局
1	R_IN32M4_Transient.c	UserReceiveTransient	Transient1、Transient2、TransientAck 受信処理	○	
2		UserHandleReceivedTransient1	Transient1 受信データ処理	○	
3		UserStartMakingReceivedTransient1	Transient1 受信データ組立て開始処理	○	
4		blUserMakeReceivedTransient1	Transient1 受信データ組立て処理	○	
5		erUserHandleReceivedMemReadRequest	SLMP メモリ読出し要求フレーム受信 対応処理	△	
6		erUserHandleReceivedMemWriteRequest	SLMP メモリ書き込み要求フレーム受信 対応処理	△	
7		UserHandleReceivedTransient2	Transient2 受信データ処理	△	
8		erUserCheckReceivedTransient2	Transient2 受信データチェック処理	△	
9		erUserHandleReceivedTransient2_ RequestSetMemory	Transient2 メモリ書き込み要求受信処理	△	
10		ulUserSetTransient2_Response	Transient2 応答フレーム作成処理	△	
11		UserHandleReceivedTransientAck	TransientAck 受信データ処理	○	
12		blUserSetTransientAck	TransientAck フレーム作成処理	○	
13		UserSendTransient	Transient1、Transient2、TransientAck 送信処理	○	
14		erUserSetSimpMemRead_Request	SLMP メモリ読出し要求フレーム作成 処理	△	×
15		pvUserJudgeTransient1Divide	Transient1 要求送信分割判定処理	△	×
16		erUserSetTransient1_Requset	Transient1 要求フレーム作成処理	△	×
17		erUserHandleReceivedMemReadResponse	SLMP メモリ読出し応答受信処理	△	×
18		UserSetTransient2_Request	Transient2 要求フレーム作成処理	△	×
19		ulUserSetTransient2_RequestGetMemory	Transient2 メモリ読出し要求フレーム 作成	△	×
20		erUserHandleReceivedTransient2_ ResponseGetMemory	Transient2 メモリ読出し応答受信処理	△	×

表 4.3 ユーザ・プログラム一覧（H/W テスト関連）

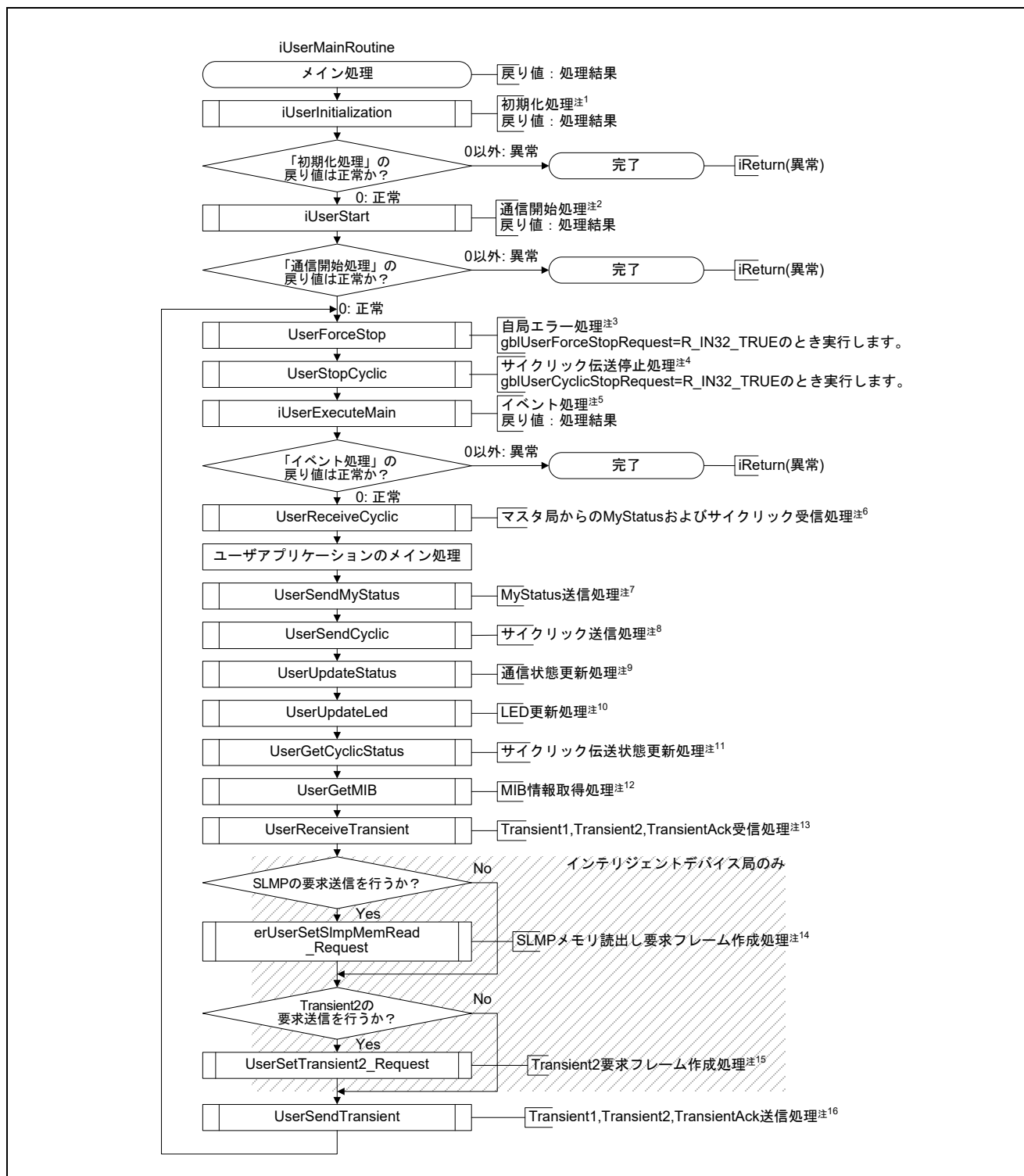
No.	ファイル	項目	機能	実装要否	
				ID 局	RD 局
1	R_IN32M4_HWTest.c	UserIEEETest	H/W テスト (IEEE802.3ab コンプライアンス・テスト)	○	
2		UserLoopBackTest	H/W テスト（折り返し通信テスト）	△	

備考. ○：必須、△：任意、×：不要

4.2 ユーザ・プログラム詳細（イニシャル・サイクリック伝送関連）

4.2.1 メイン処理

メイン処理の概略フローを示します。



- 注 1. 詳細は、「4.2.2 初期化处理」を参照してください。
2. 詳細は、「4.2.3 通信開始処理を参照してください。
3. 詳細は、「4.2.4 自局エラー処理」を参照してください。
4. 詳細は、「4.2.5 サイクリック伝送停止処理」を参照してください。
5. 詳細は、「4.2.6 イベント処理」を参照してください。
6. 詳細は、「4.2.7 マスタ局からの MyStatus およびサイクリック受信処理」を参照してください。
7. 詳細は、「4.2.8 MyStatus 送信処理」を参照してください。
8. 詳細は、「4.2.9 サイクリック送信処理」を参照してください。
9. 詳細は、「4.2.10 通信状態更新処理」を参照してください。
10. 詳細は、「4.2.11 LED 更新処理」を参照してください。
11. 詳細は、「4.2.12 サイクリック伝送状態更新処理」を参照してください。
12. 詳細は、「4.2.13 MIB 情報取得処理」を参照してください。
13. 詳細は、「4.3.2 Transient1、Transient2、TransientAck 受信処理」を参照してください。
14. 詳細は、「4.3.15 SLMP メモリ読出し要求フレーム作成処理」を参照してください。
15. 詳細は、「4.3.19 Transient2 要求フレーム作成処理」を参照してください。
16. 詳細は、「4.3.14 Transient1、Transient2、TransientAck 送信処理」を参照してください。

4.2.2 初期化処理

R-IN32M4-CL3 の初期化、R-IN32M4-CL3 内蔵 WDT の有効/無効、および局番/ネットワーク No.を設定します。

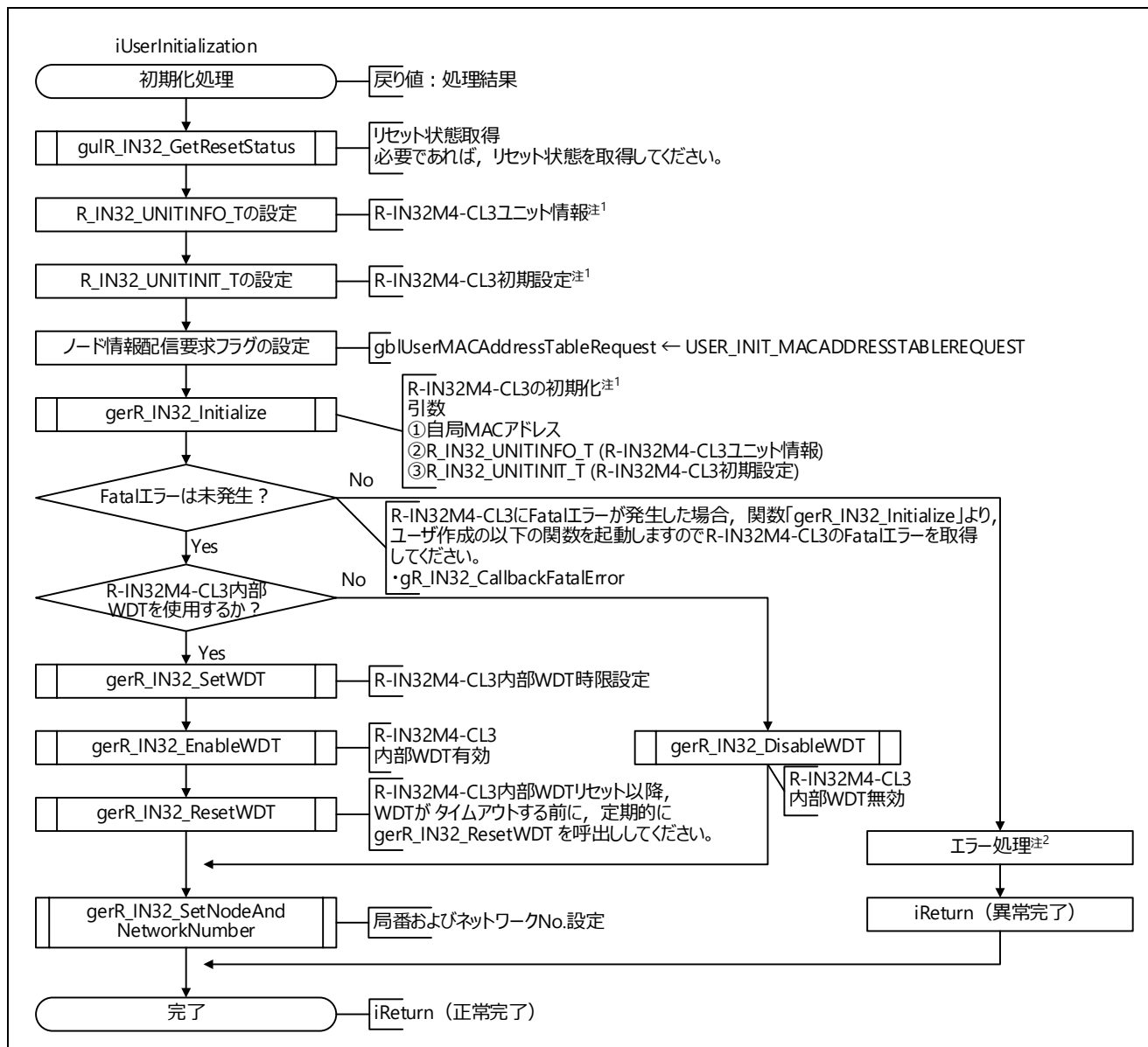


図 4.2 初期化処理 フロー図

注 1. 詳細は、「5.3.1(2) gerR_IN32_Initialize」を参照してください。

2. 例えば自局エラー処理 UserForceStop を起動し、自局をバイパスモードに設定するなどの処理を追加してください。

【gblUserMACAddressTableRequest について】

gblUserMACAddressTableRequest（ノード情報配信要求フラグ）は、ノード情報配信フレームを受信処理するか/否かの判定に使用します。（「4.3.3 Transient1 受信データ処理」参照）

- ・ノード情報を受信する場合（トランジェント要求を送信する場合）
R_IN32_UNITINIT_T の blMACAddressTableRequest（ノード情報配信の要求初期値）と共に gblUserMACAddressTableRequest（ノード情報配信要求フラグ）に R_IN32_TRUE をセットしてください。
- ・ノード情報を受信しない場合（トランジェント要求を送信しない場合）
R_IN32_UNITINIT_T の blMACAddressTableRequest（ノード情報配信の要求初期値）と共に gblUserMACAddressTableRequest（ノード情報配信要求フラグ）に、R_IN32_FALSE をセットしてください。

4.2.3 通信開始処理

R-IN32M4-CL3 に通信開始を指示する処理です。

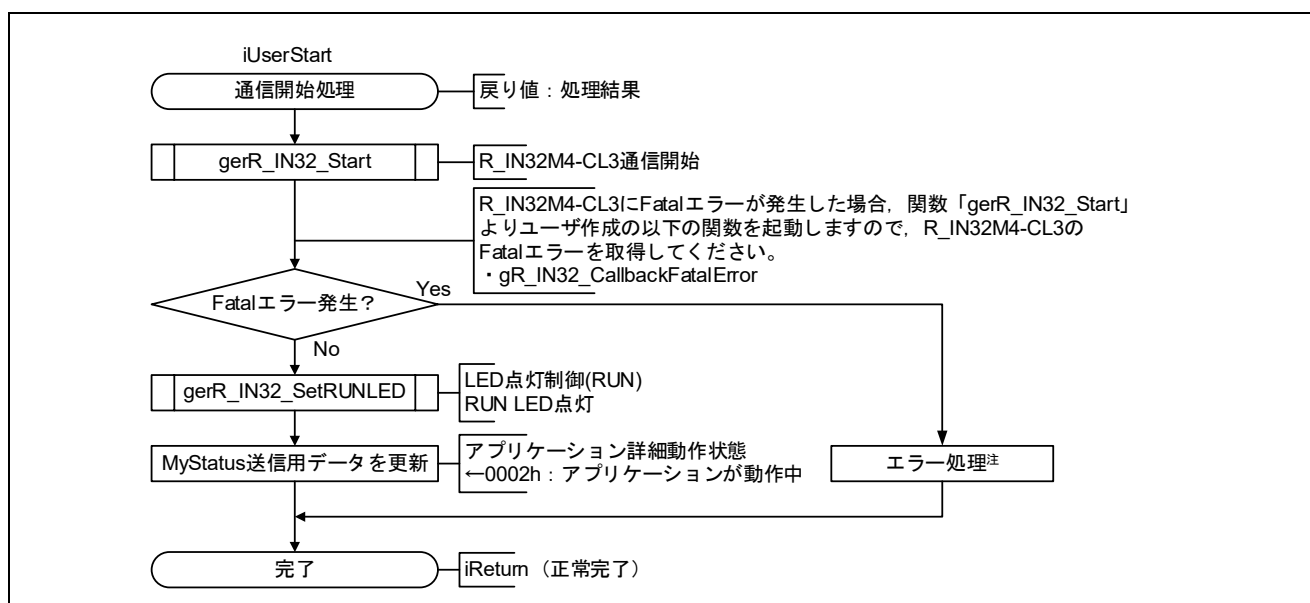


図 4.3 通信開始処理 フロー図

注. 例えば自局エラー処理 `UserForceStop` を呼び出し、自局をバイパスモードに設定するなどの処理を追加してください。

4.2.4 自局エラー処理

ユーザ・プログラムで（ユーザが定義した）エラーが発生したとき、自局をエラー状態にします。（本処理はユーザの任意で実装する処理です。）

gerR_IN32_ForceStop を呼び出すと、R-IN32M4-CL3 は自局エラーとなるとバイパスモードになります。バイパスモードでは、ポートに入った通信フレームを R-IN32M4-CL3 で受信せず、もう一方のポートにそのままフォワードします。

自局エラーを解除する場合はパワー・オン・リセットまたはシステム・リセットを行う必要があります。

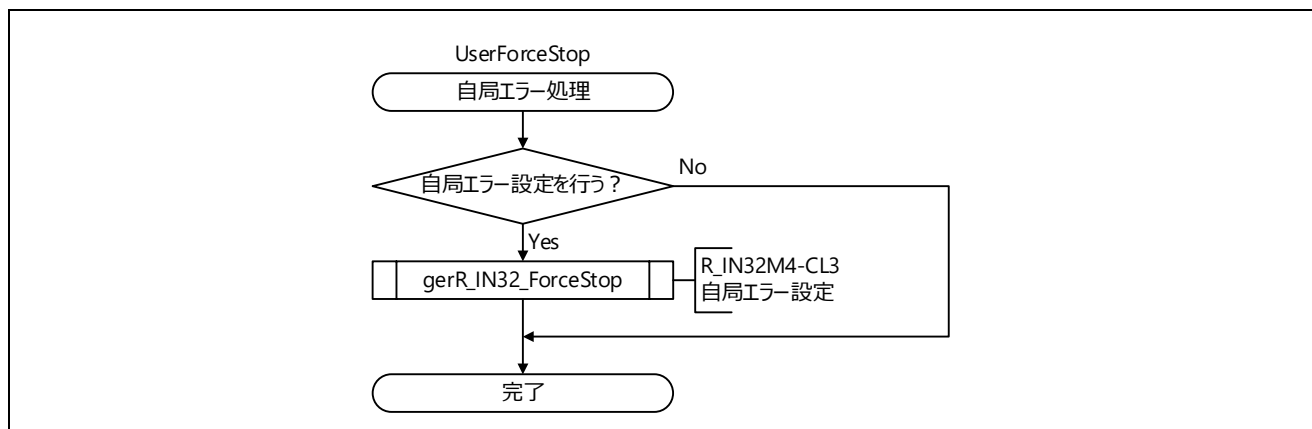


図 4.4 自局エラー処理 フロー図

4.2.5 サイクリック伝送停止処理

ユーザ・アプリケーション側の都合によりサイクリック伝送の停止と再開を制御します。（本処理はユーザの任意で実装する処理です。）

サイクリック伝送を停止させても、トランジェント伝送は可能です。（トークンパッシングは継続します。）

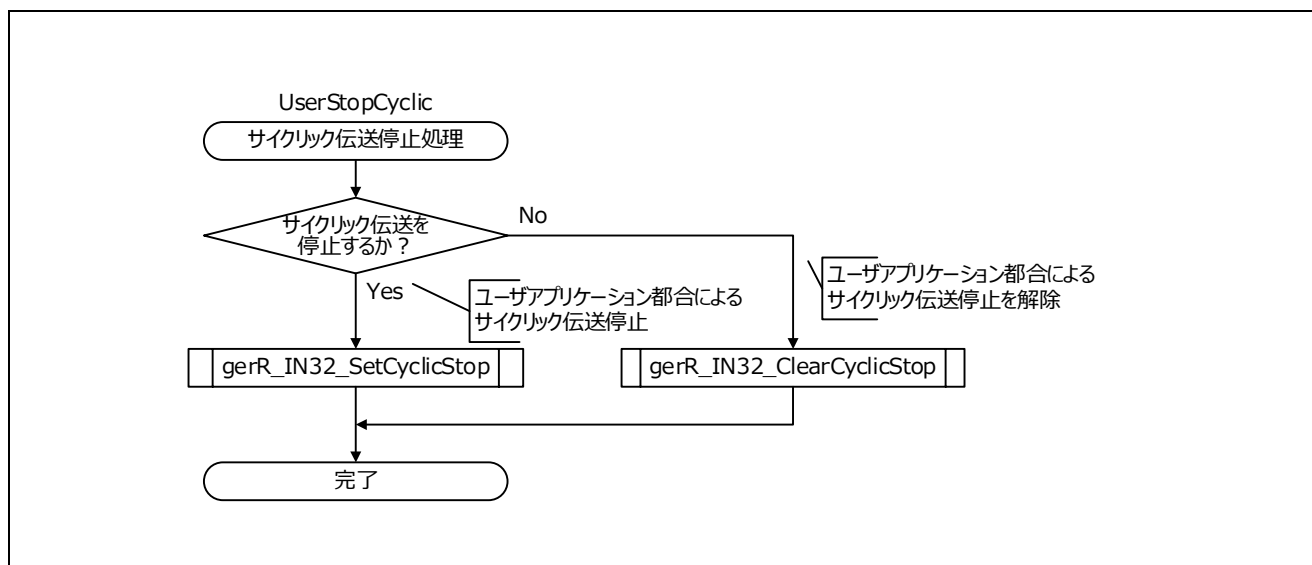


図 4.5 サイクリック伝送停止処理 フロー図

4.2.6 イベント処理

R-IN32M4-CL3 イベント（割込要因）を検出し、イベントの処理と MIB 情報の更新を行います。

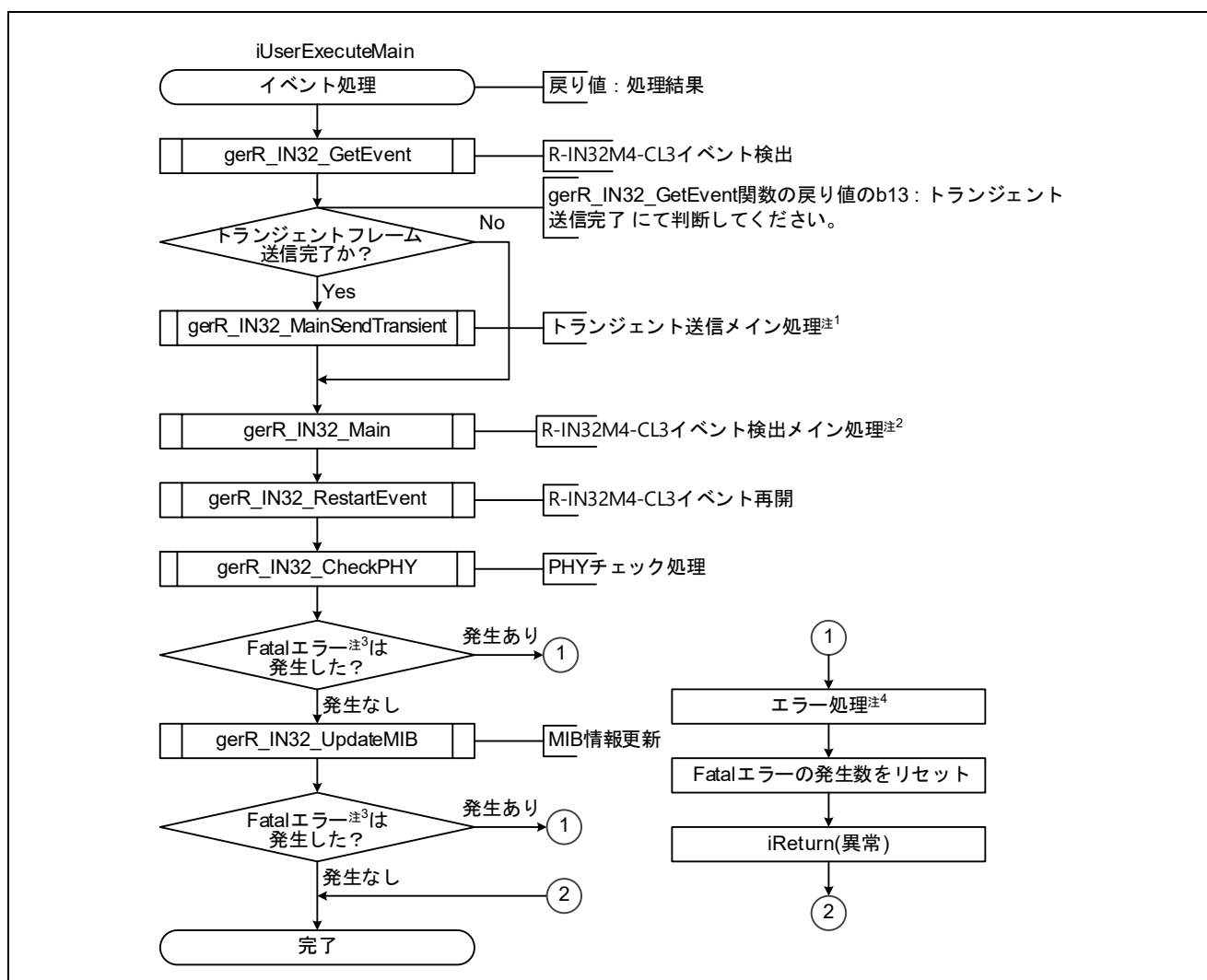


図 4.6 イベント処理 フロー図

注 1. 本関数より、ユーザ作成の以下の関数を起動し、対象の送信ディスクリプタの状態（送信結果）を通知します。

- ・ gerR_IN32_CallbackTransientSendingComplete

2. マスタ局からのコマンドを受信した場合、関数 gerR_IN32_Main より、ユーザ作成の以下の関数を起動しますのでコマンドを取得してください。

- ・ gerR_IN32_CallbackCommandFromMaster

3. R-IN32M4-CL3 に Fatal エラーが発生した場合、関数 gerR_IN32_UpdateMIB より、ユーザ作成の以下の関数を起動しますので R-IN32M4-CL3 の Fatal エラーを取得してください。

- ・ gR_IN32_CallbackFatalError

4. 例えば自局エラー処理 UserForceStop を起動し、自局をバイパスモードに設定するなどの処理を追加してください。

4.2.7 マスタ局からの MyStatus およびサイクリック受信処理

受信した MyStatus フレームからマスタ局の状態の取得と、サイクリック受信フレームからサイクリックデータを取得します。ユーザ・プログラムは、MyStatus フレームで得たマスタ局の状態（マスタ局が停止中またはエラー発生中など）に応じて、「Hold/Clear 処理」を実行してください。

Hold/Clear 処理は、R-IN32M4-CL3 適用製品が外部へ出力制御を行なう製品の場合、マスタ局のアプリケーションが停止/異常、またはデータリンクから解列したなどの理由でサイクリック伝送を停止したときに、出力を続行する（Hold）または出力を停止する（Clear）処理です。

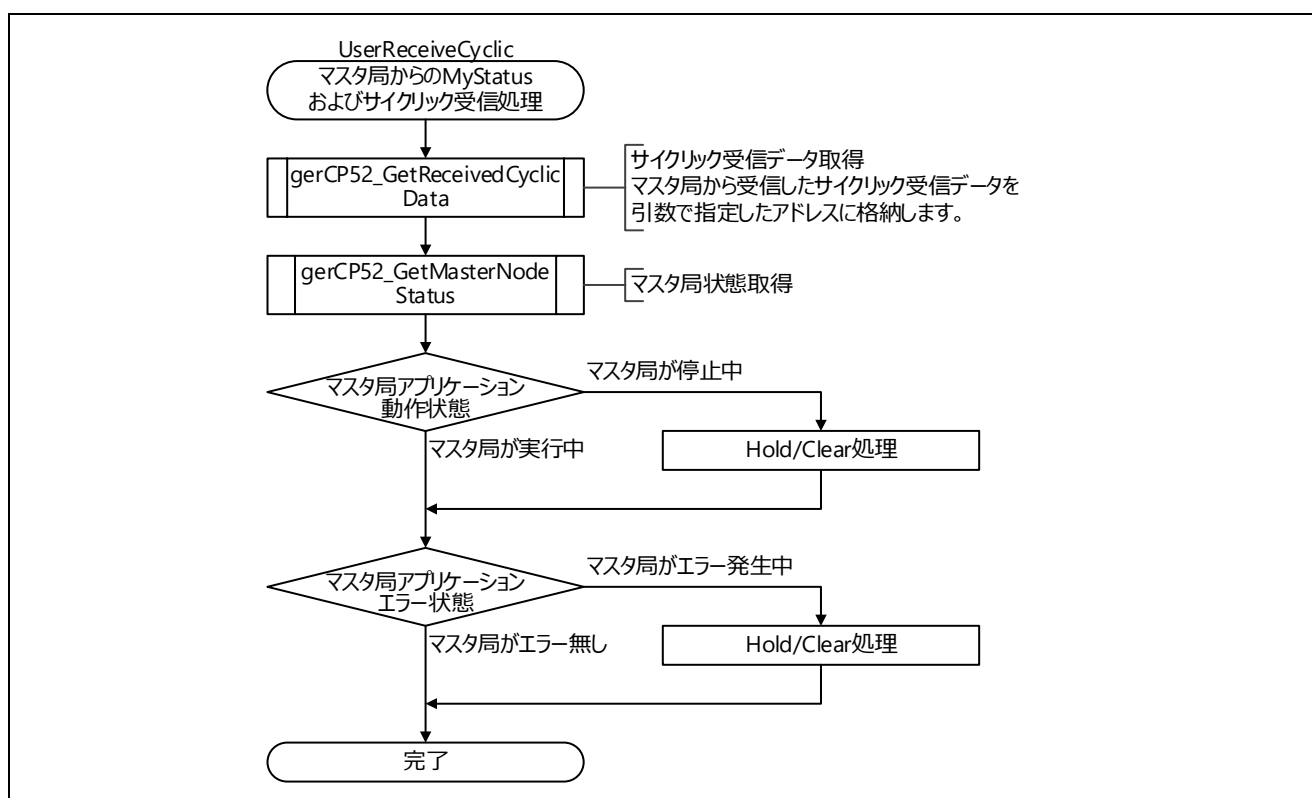


図 4.7 マスタ局からの MyStatus およびサイクリック受信処理 フロー図

注意. フェールセーフとして、以下①および②を考慮して Hold/Clear 処理を実装してください。

① マスタ局が送信するサイクリックデータについて

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。

（三菱電機製マスタ局の場合、“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。）

スレーブ局（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。

② マスタ局アプリケーションの状態による R-IN32M4-CL3 ドライバが取得するサイクリックデータについて

スレーブ局（自局）が受信するサイクリックデータは、R-IN32M4-CL3 ドライバ

（`gerR_IN32_GetReceivedCyclicData`）で取得します。

取得するサイクリックデータは、マスタ局アプリケーションの動作/エラー状態によって内容が異なります。

マスタ局アプリケーション		R-IN32M4-CL3 ドライバが取得する サイクリックデータ
動作状態	エラー状態	
実行中	エラーなし	マスタ局が“現在”送信しているサイクリックデータ
停止中	エラーなし	
実行中	エラーあり [※]	取得しない （引数で指定したアドレスには、マスタ局アプリケーションにエラーが発生する“前”のサイクリックデータが残っている）
停止中	エラーあり	

注. 三菱電機製マスタ局のシーケンサ CPU ユニットのの場合、“実行中かつエラーあり”の状態は、存在しません。

4.2.8 MyStatus 送信処理

MyStatus フレームを作成します。作成したフレームは R-IN32M4-CL3 によって自動的に送信されます。

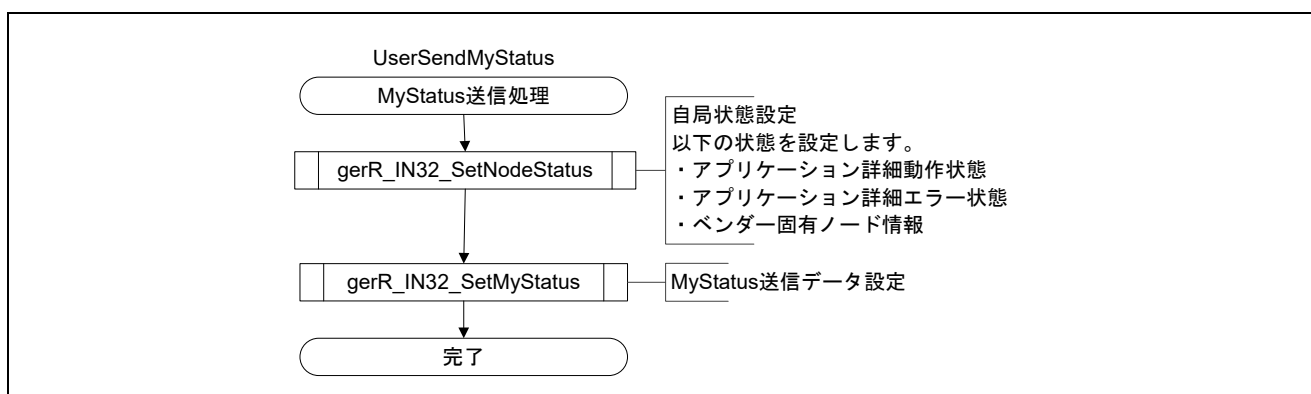


図 4.8 MyStatus 送信処理 フロー図

4.2.9 サイクリック送信処理

サイクリック送信データ (RX・RW_r) を送信します。

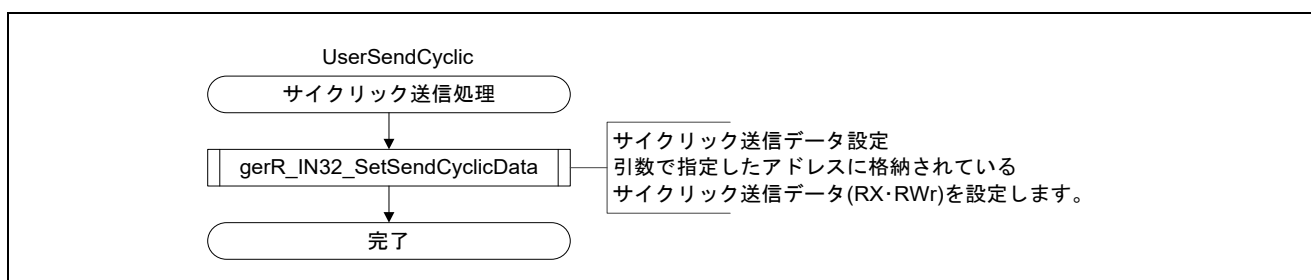


図 4.9 サイクリック送信処理 フロー図

4.2.10 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た Hold/Clear 処理と ERR LED 制御フラグ設定を行います。

ERR. LED 制御フラグは、「4.2.11 LED 更新処理」において、ERR. LED の点灯制御に使用します。

Hold/Clear 処理は、R-IN32M4-CL3 適用製品が外部へ出力制御を行なう製品の場合、マスタ局アプリケーションが停止/異常、またはデータリンクから解列したなどの理由でサイクリック伝送を停止したときに、出力を続行する (Hold) または出力を停止する (Clear) 処理です。

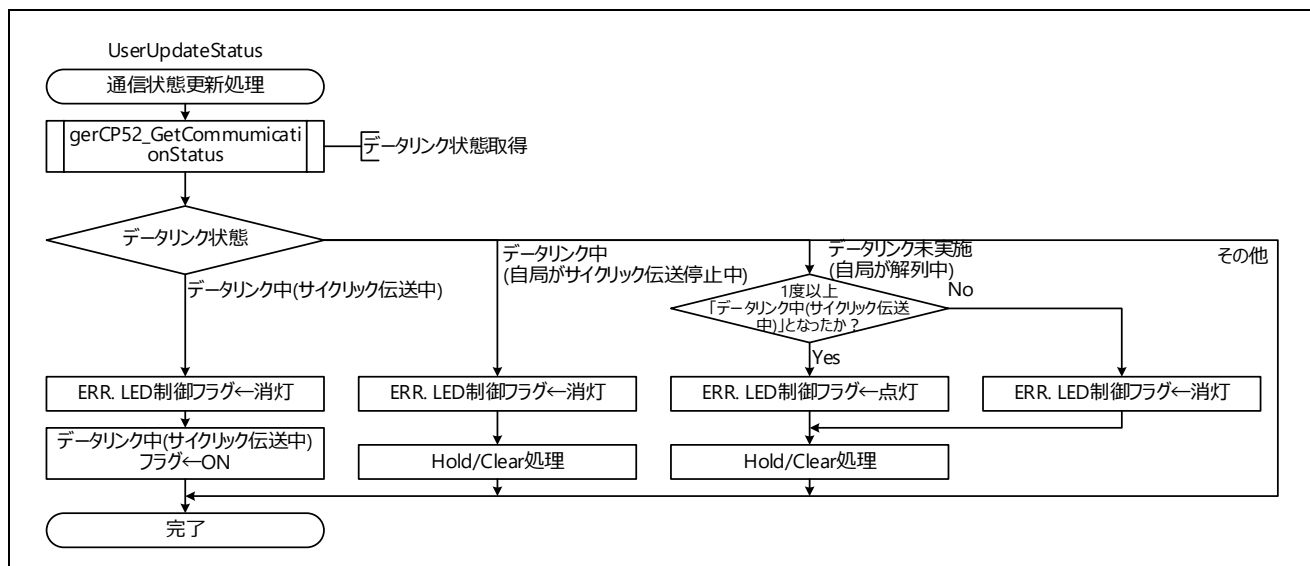


図 4.10 通信状態更新処理 フロー図

注意. 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。

データリンク状態による R-IN32M4-CL3 ドライバが取得するサイクリックデータについて
スレーブ局（自局）が受信するサイクリックデータは、R-IN32M4-CL3 ドライバ
（gerR_IN32_GetReceivedCyclicData）で取得します。
取得するサイクリックデータは、データリンク状態によって内容が異なります。

データリンク状態	R-IN32M4-CL3 ドライバが取得するサイクリックデータ
データリンク未実施 (自局が解列中)	取得しない (引数で指定したアドレスには、自局が解列する“前”のサイクリックデータが残っている)
データリンク中 (自局がサイクリック伝送停止中) 注	マスタ局が“現在”送信しているサイクリックデータ

注. スレーブ局は RY、RWw を受信し、RX、RWr を送信しない

4.2.11 LED 更新処理

自局のデータリンク状態に応じた D LINK LED と L ER LED の点灯/消灯/点滅の制御を、R-IN32M4-CL3 ドライバ・インタフェース関数を呼び出して行います。

また、ERR. LED 制御フラグの状態に応じて、ユーザ・アプリケーションの都合による ERR LED の点灯/消灯の制御を行います。なお、ERR. LED 制御フラグは、「4.2.10 通信状態更新処理」で設定されていますが、ユーザ・アプリケーションの都合でフラグの値を変更することができます。

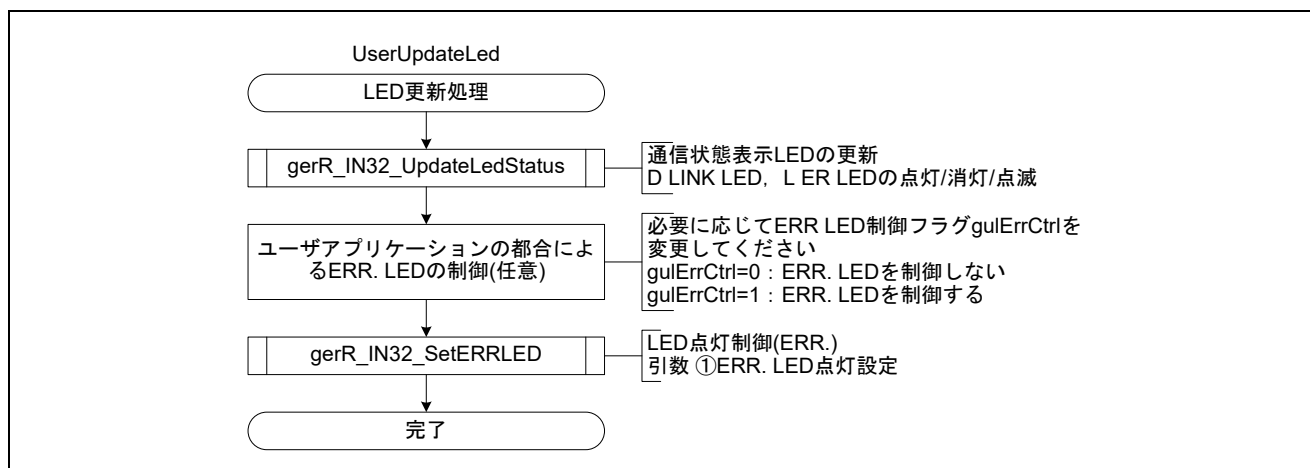


図 4.11 LED 更新処理 フロー図

4.2.12 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。（本処理はユーザの任意で実装する処理です。）

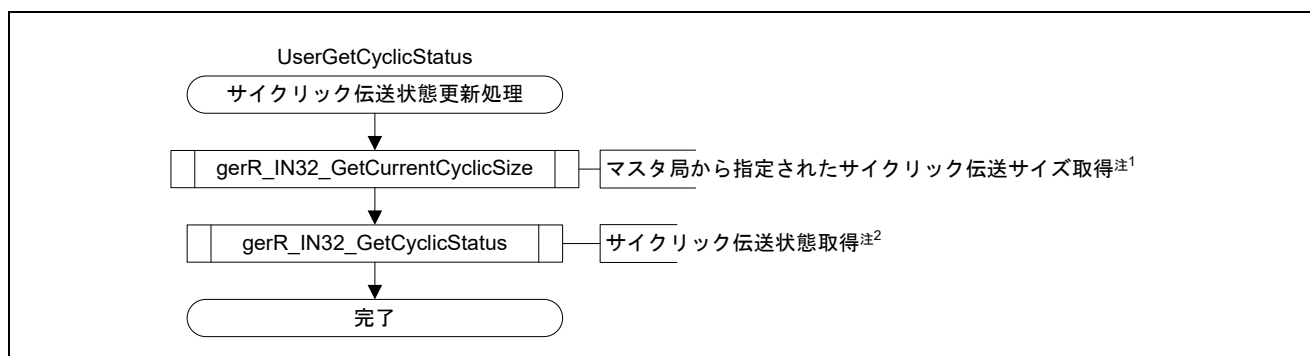


図 4.12 サイクリック伝送状態更新処理 フロー図

注 1. サイクリック伝送は、R-IN32M4-CL3 がマスタ局から指定されたサイクリック伝送サイズにて行います。「マスタ局から指定されたサイクリック伝送サイズ」を把握したい場合のみ関数 `gerR_IN32_GetCurrentCyclicSize` にて取得してください。

2. サイクリック伝送は R-IN32M4-CL3 ドライバが処理するため、ユーザ・プログラムが「サイクリック伝送状態」を取得し、状態に応じた処理をする必要はありません。「サイクリック伝送状態」を把握したい場合のみ関数 `gerR_IN32_GetCyclicStatus` にて取得してください。

4.2.13 MIB 情報取得処理

MIB 情報を取得します。（本処理はユーザの任意で実装する処理です。）

MIB 情報は以下の処理などに利用できます。

備考. (例) ユーザ・プログラムがフレームの送受信状態を監視して、正常/異常の頻度に応じた通知・警告する処理

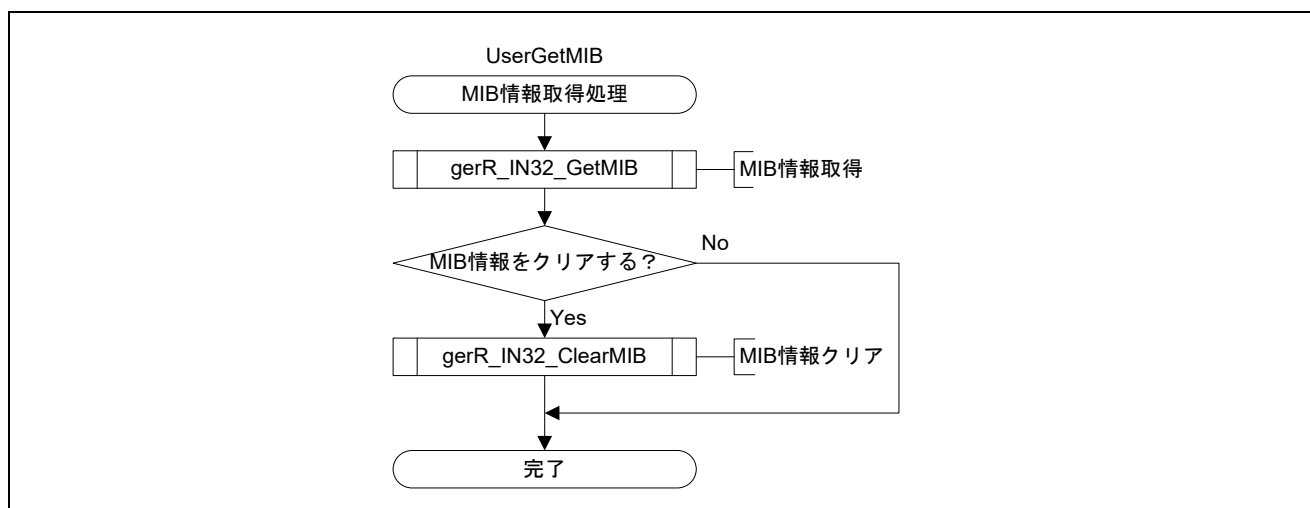


図 4.13 MIB 情報取得処理 フロー図

注意. MIB 情報は非公開情報ですので、エンド・ユーザに公開しないでください。

(1) リング制御部の MIB 情報一覧

表 4.4 リング制御部の MIB 情報一覧表

No.	MIB 情報	内容
1	HEC エラーフレーム数	受信フレームの HEC エラーをカウントする。
2	DCS/FCS エラーフレーム数	受信フレームの DCS/FCS エラーをカウントする。
3	アンダーサイズエラーフレーム数	受信フレームサイズが 28 バイト未満の場合にカウントする。
4	フォワードフレーム数	フォワードしたフレーム数をカウントする。
5	上位層転送フレーム数	上位層に転送したフレーム数をカウントする。
6	フォワードバッファフル破棄フレーム数	フォワードバッファフルにより、破棄されたフレーム数をカウントする。
7	上位層転送バッファフル破棄フレーム数	上位層への転送バッファフルにより、破棄されたフレーム数をカウントする。

(2) MAC IP 部の MIB 情報一覧

表 4.5 MAC IP 部の MIB 情報一覧表

No.	MIB 情報	内容
1	受信フレーム数	異常フレームも含めすべてのフレーム受信にてカウントアップする。 異常フレーム：FCS エラー、アンダーサイズ、オーバーサイズフレーム
2	送信フレーム数	送信フレーム数をカウントアップする。
3	受信アンダーサイズフレーム数	受信フレームサイズが 64 バイト未満の場合にカウントアップする。
4	受信オーバーサイズフレーム数	受信フレームサイズが 1518 バイトを超える場合にカウントアップする。
5	受信フレーム FCS エラー数	受信フレームが FCS エラーの場合カウントアップする。
6	受信フレームフラグメントエラー数	受信フレームがフラグメントエラーの場合カウントする。 フラグメントエラー：64 バイト未満かつ FCS エラーのフレーム
7	最小 IFG フレーム検出数	最小インターフレームギャップ（IFG）内にフレームを検出した際カウントアップする。
8	受信フレーム SFD 以下検出数	受信フレームが SFD までのフィールドで終了し、有効なフレームと認識できなかったフレームをカウントアップする。
9	受信コードエラー数	GMII の受信データエラーを検出した場合（RECV_*_ERR=1 ^注 ）、カウントアップする。 アイドル中（RECV_*_DV =1 ^注 ）に RECV_*_ERR ^注 が複数回 1 となっても、カウントは 1 回になる。
10	受信不正キャリアエラー数	アイドル中に不正キャリアが発生した場合カウントアップする。 アイドル中に複数回の不正キャリアが発生した場合でもカウントは 1 回とカウントする。
11	受信キャリア拡張エラー数	アイドル中にキャリア拡張が発生した場合カウントアップする。 アイドル中に複数回のキャリア拡張が発生した場合でもカウントは 1 回とカウントする。

注. 表中の「*」はワイルド文字です。（A：PORT0、B：PORT1）

(3) その他の MIB 情報一覧

表 4.6 その他の MIB 情報一覧表

No.	MIB 情報	説明
1	リンクダウン数（ポート 1）	ポート 1 のリンクダウン数をカウントアップする。
2	リンクダウン数（ポート 2）	ポート 2 のリンクダウン数をカウントアップする。
3	マスタウォッチタイマエラー数	マスタウォッチタイマのタイムアウト回数をカウントアップする。
4	サイクリック受信フレーム数	受信したサイクリック受信フレーム数をカウントアップする。
5	トランジェント受信フレーム数	受信したトランジェント受信フレーム数をカウントアップする。
6	トランジェント受信フレーム破棄数	破棄したトランジェント受信フレーム数をカウントアップする。

4.3 ユーザ・プログラム詳細（トランジェント関連）

4.3.1 トランジェント伝送の処理概要

【自局がサーバのとき】

クライアントからの要求フレームに対して、サーバが応答フレームを送信する処理手順のイメージを示します。

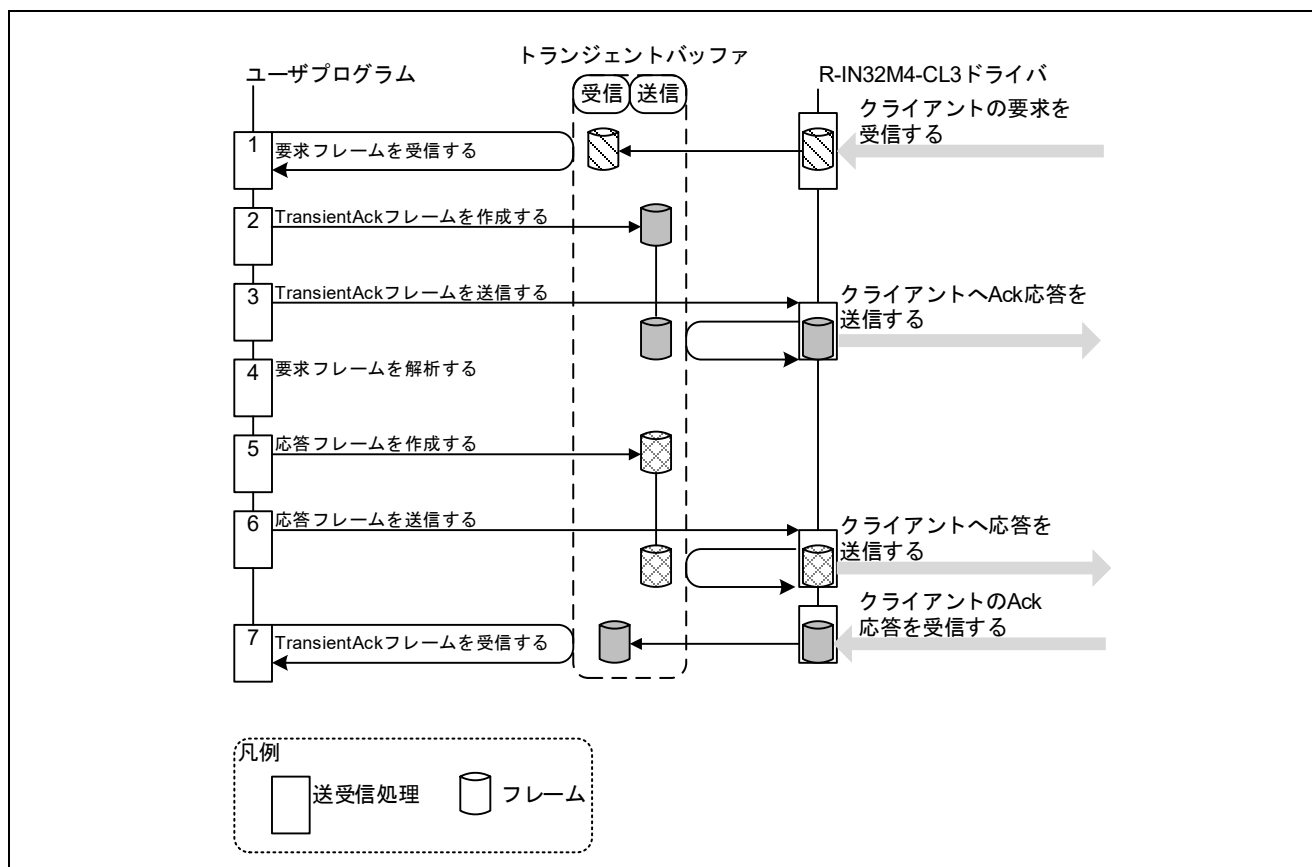


図 4.14 サーバの処理手順

No.	処理	参照項
1	Transient フレームを受信します	4.3.2 Transient1、Transient2、TransientAck 受信処理
2	TransientAck フレームを作成します	4.3.13 TransientAck フレーム作成処理
3	TransientAck フレームを送信します	4.3.14 Transient1、Transient2、TransientAck 送信処理
4	受信した要求フレームを解析し、フレームごとの処理を行います	4.3.3 Transient1 受信データ処理 4.3.8 Transient2 受信データ処理
5	コマンドに応じて応答フレームを作成します	4.3.6 SLMP メモリ読出し要求フレーム受信対応処理 4.3.7 SLMP メモリ書込み要求フレーム受信対応処理 4.3.10 Transient2 メモリ書込み要求受信処理
6	応答フレームを送信します	4.3.14 Transient1、Transient2、TransientAck 送信処理
7	TransientAck フレームを受信します	4.3.12 TransientAck 受信データ処理

【自局がクライアントのとき】

クライアントが要求フレームを送信して、サーバからの応答フレームを受信する処理手順のイメージを示します。

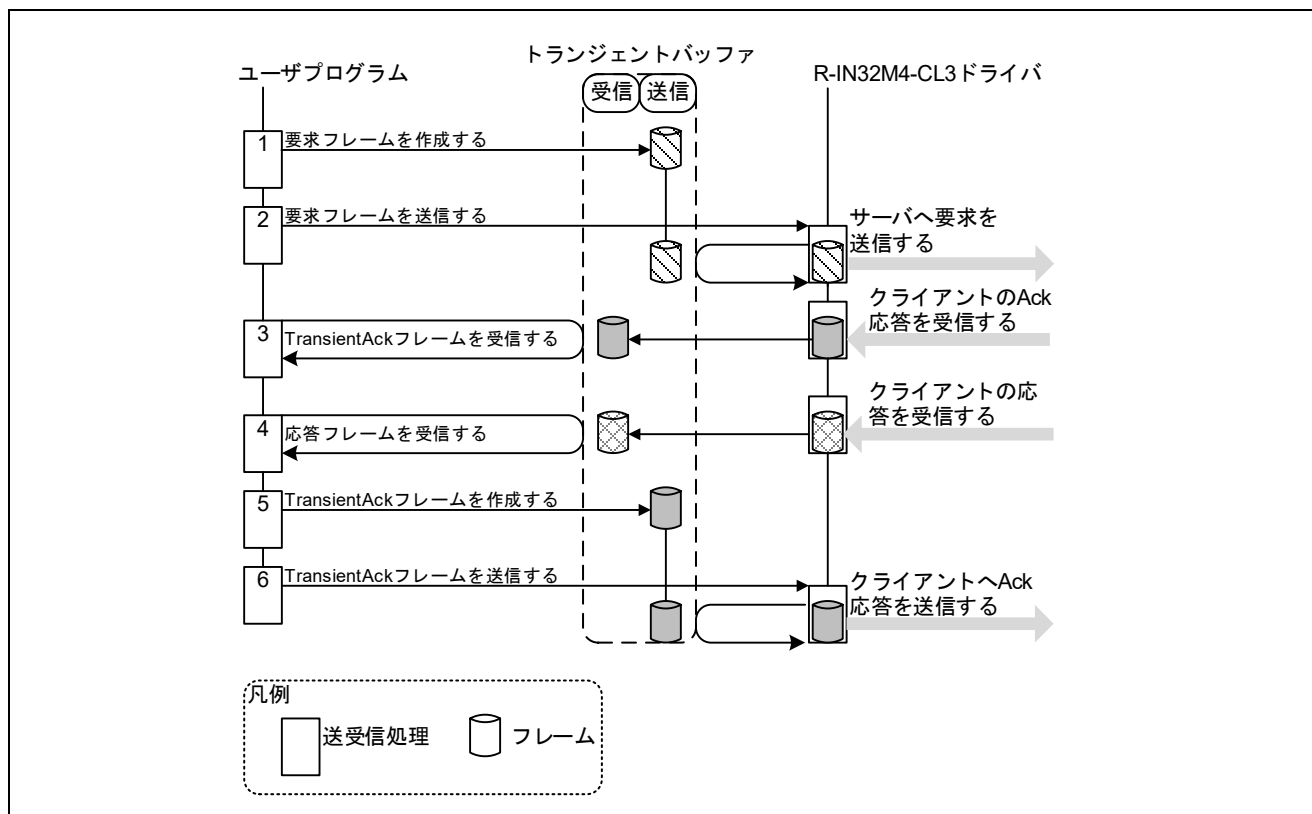


図 4.15 クライアントの処理手順

No.	処理	参照項
1	要求フレームを作成します ^注	4.3.15 SLMP メモリ読出し要求フレーム作成処理 4.3.19 Transient2 要求フレーム作成処理
2	要求フレームを送信します	4.3.14 Transient1、Transient2、TransientAck 送信処理
3	TransientAck フレームを受信します	4.3.2 Transient1、Transient2、TransientAck 受信処理 4.3.12 TransientAck 受信データ処理
4	応答フレームを受信します	4.3.3 Transient1 受信データ処理 4.3.18 SLMP メモリ読出し応答受信処理 4.3.8 Transient2 受信データ処理 4.3.21 Transient2 メモリ読出し応答受信処理
5	TransientAck フレームを作成します	4.3.13 TransientAck フレーム作成処理
6	TransientAck フレームを送信します	4.3.14 Transient1、Transient2、TransientAck 送信処理

注. コマンドごとのサンプル処理として、Transient2 メモリ読出し、および SLMP メモリ読出しをサンプルコードに記述しています。上記以外のコマンドは、ユーザ・プログラムをカスタマイズして実装してください。

4.3.2 Transient1、Transient2、TransientAck 受信処理

Transient1、Transient2、TransientAck フレームを受信し、データを処理します。

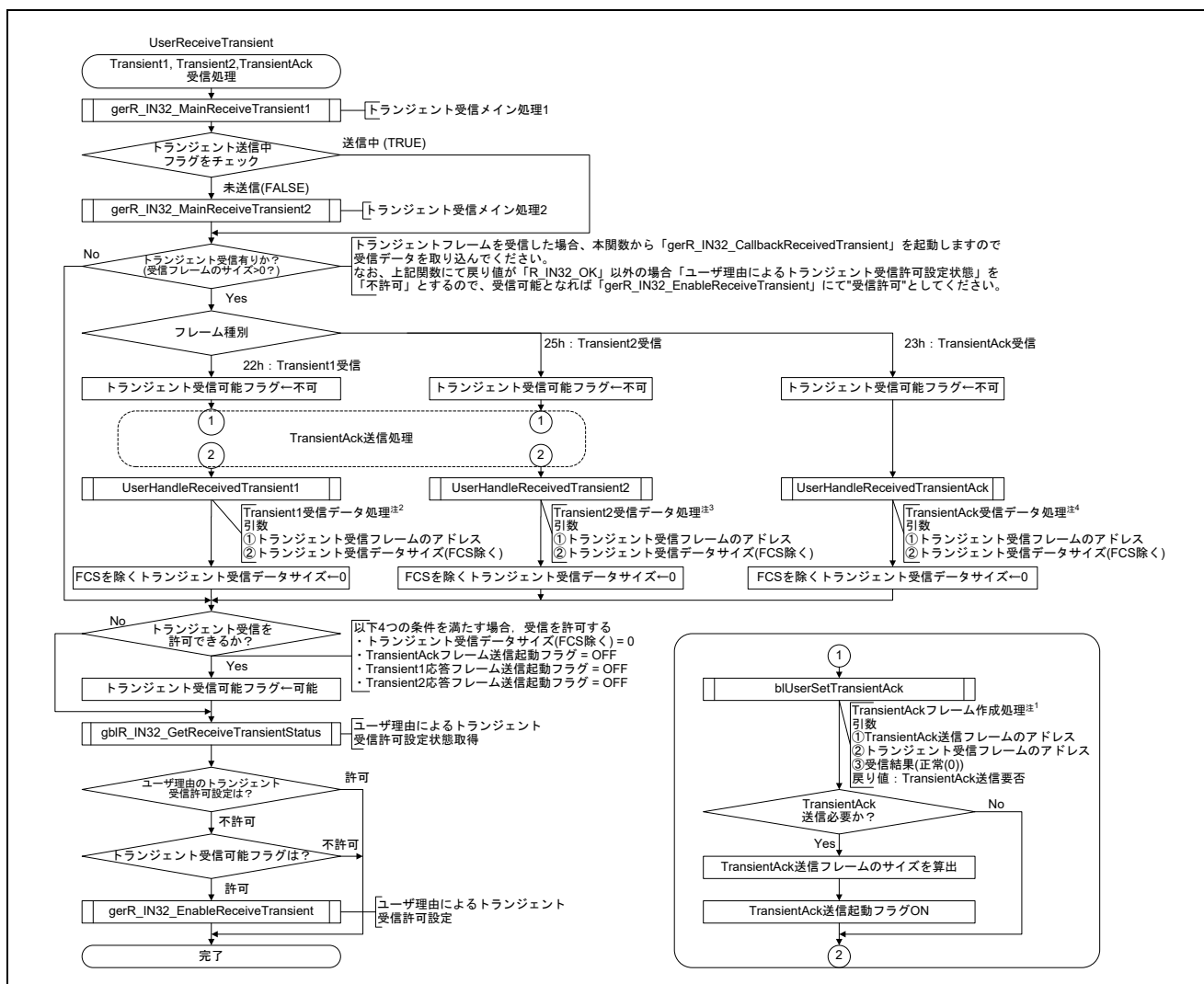


図 4.16 Transient1、Transient2、TransientAck 受信処理フロー図

注 1. 詳細は、「4.3.13 TransientAck フレーム作成処理」を参照してください。

2. 詳細は、「4.3.3 Transient1 受信データ処理」を参照してください。

3. 詳細は、「4.3.8 Transient2 受信データ処理」を参照してください。

4. 詳細は、「4.3.12 TransientAck 受信データ処理」を参照してください。

4.3.3 Transient1 受信データ処理

受信した Transient1 フレームを解析し、解析結果に応じた処理を行います。また、Transient1 フレームを分割して受信した場合はデータ組立て処理を行います。

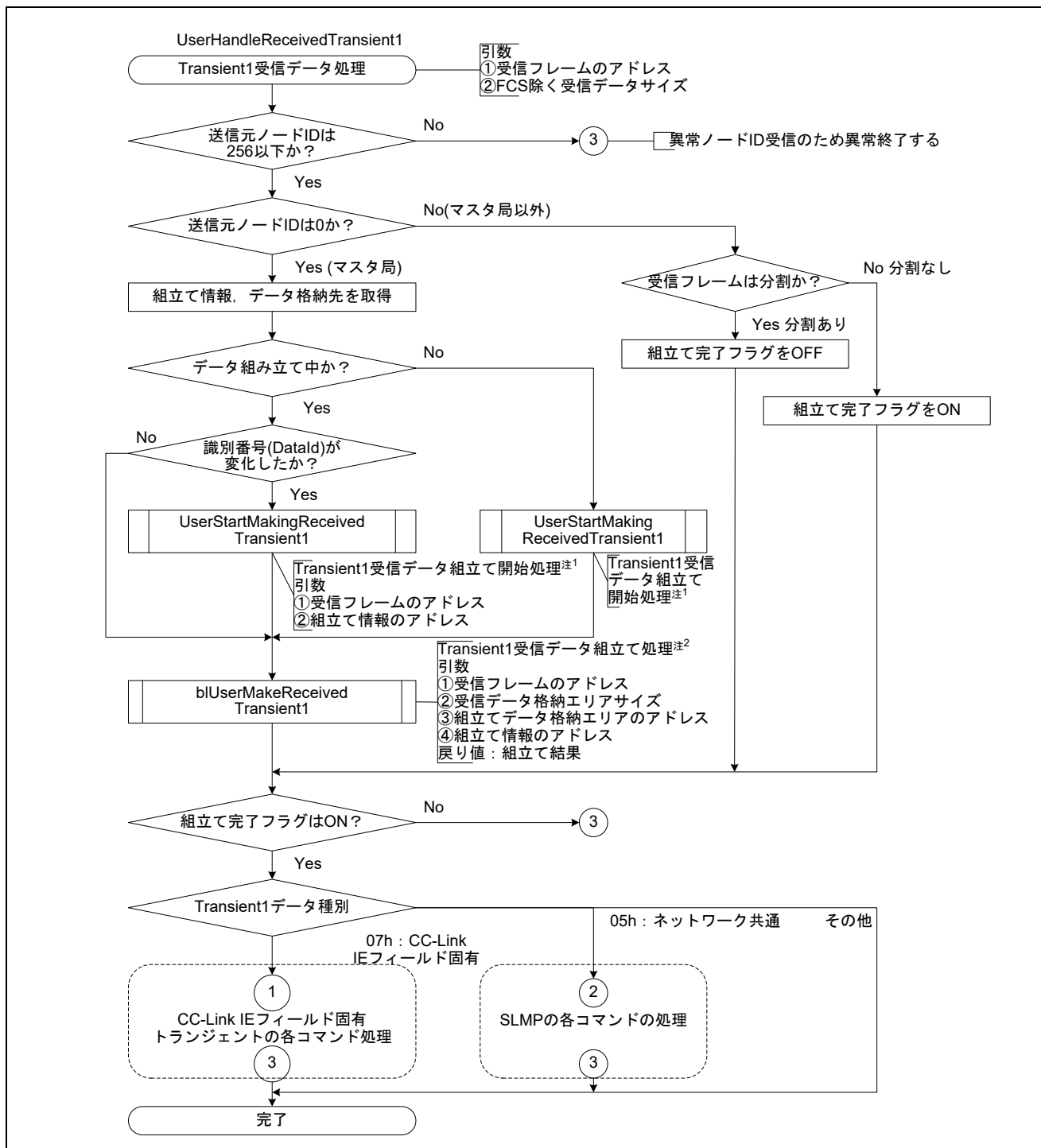


図 4.17 Transient1 受信データ処理 フロー図 (1/4)

注 1. 詳細は、「4.3.4 Transient1 受信データ組立て開始処理」を参照してください。

2. 詳細は、「4.3.5 Transient1 受信データ組立て処理」を参照してください。

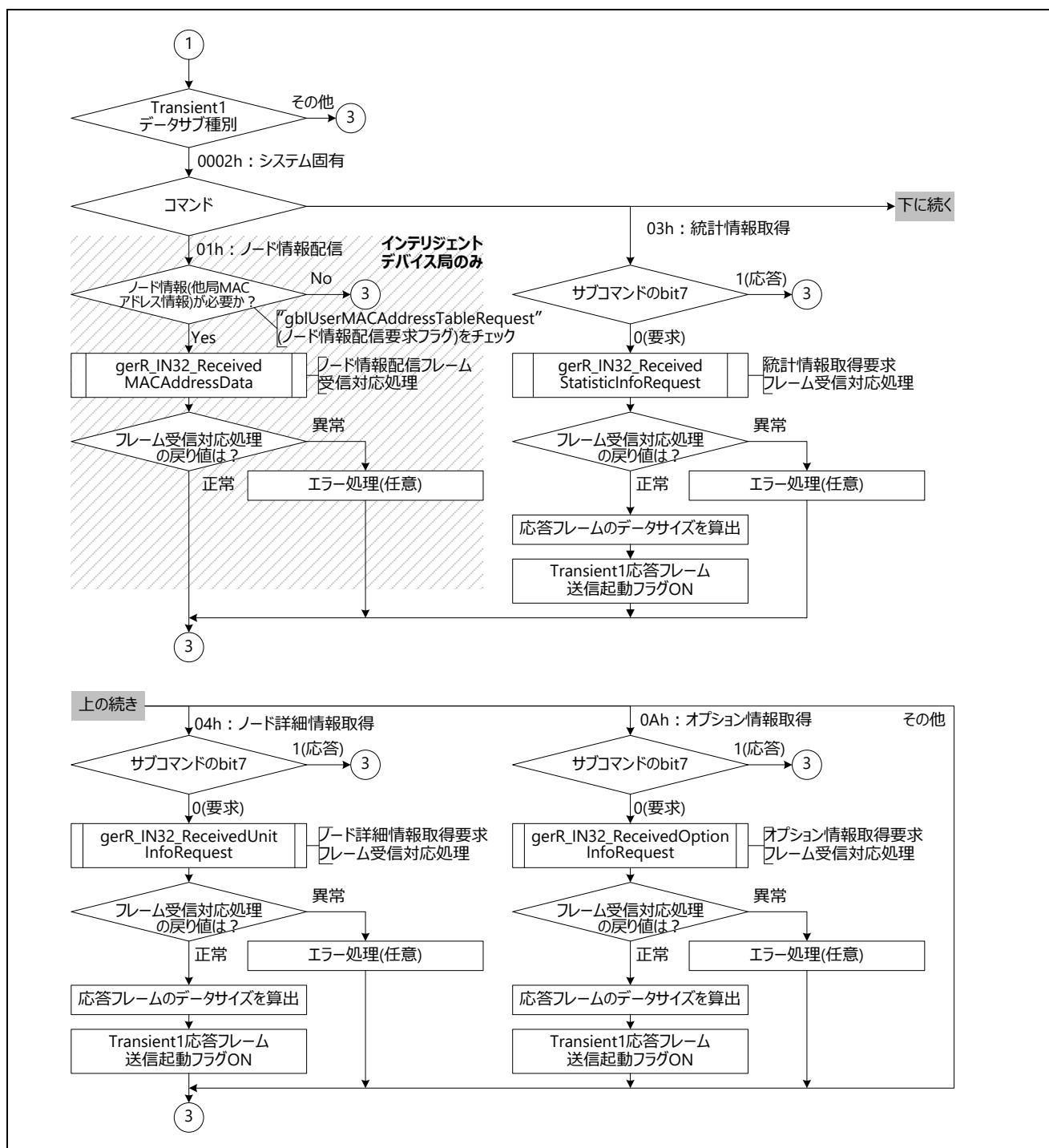


図 4.17 Transient1 受信データ処理 フロー図 (2/4)

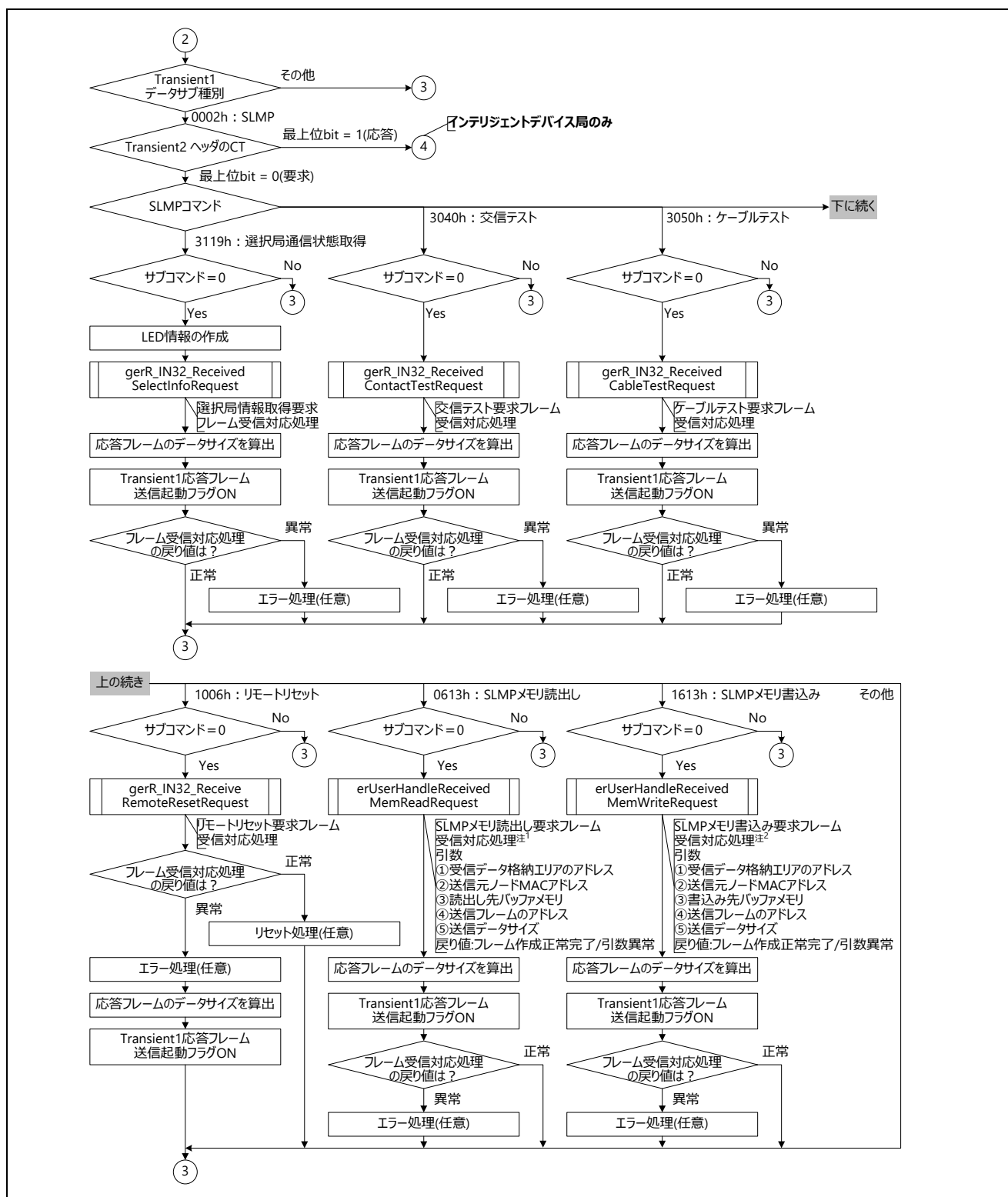


図 4.17 Transient 受信データ処理 フロー図 (3/4)

注 1. 詳細は、「4.3.6 SLMP メモリ読出し要求フレーム受信対応処理」を参照してください。

注 2. 詳細は、「4.3.7 SLMP メモリ書込み要求フレーム受信対応処理」を参照してください。

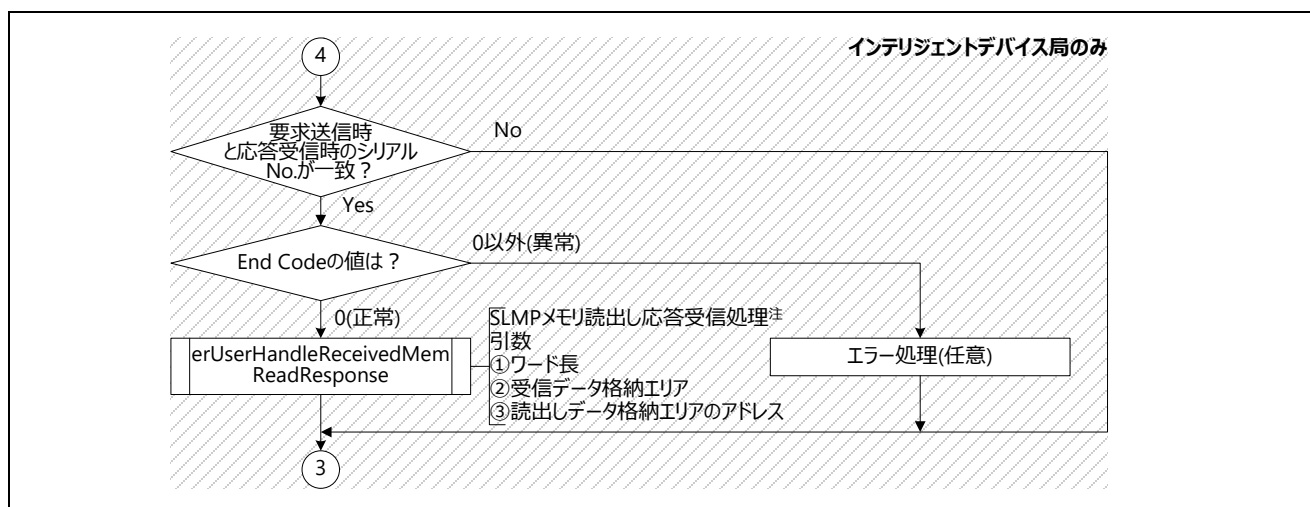


図 4.17 Transient 受信データ処理 フロー図 (4/4)

注. 詳細は、「4.3.18 SLMP メモリ読出し応答受信処理」を参照してください。

(1) ノード情報配信について

ノード情報は、他局の局番と MAC アドレスとの対応を示した情報です。

インテリジェントデバイス局が他局へトランジェント要求を送信する場合（クライアント時）、ノード情報を利用します。

- ノード情報を受信する場合（トランジェント要求を送信する場合）

iUserInitialization（「4.2.2 初期化处理」参照）にて、R_IN32_UNITINIT_T の blMACAddressTableRequest（ノード情報配信の要求初期値）と共に gblUserMACAddressTableRequest（ノード情報配信要求フラグ）に、R_IN32_TRUE をセットしてください。

なお、受信したノード情報配信フレームに対して、TransientAck およびトランジェント応答は不要です。

- ノード情報を受信しない場合（トランジェント要求を送信しない場合）

iUserInitialization（「4.2.2 初期化处理」参照）、R_IN32_UNITINIT_T の blMACAddressTableRequest（ノード情報配信の要求初期値）と共に gblUserMACAddressTableRequest（ノード情報配信要求フラグ）に、R_IN32_FALSE をセットしてください。

(2) マスタ局からの SLMP 要求受信について

エンジニアリング・ツールを使うことで、「CC-Link IE フィールド・ネットワーク診断」と「スレーブ局パラメータ処理・コマンド実行」を実行することができます。これらの機能は、マスタ局からの SLMP 要求フレームに自局が応答することで実現します。

下図は、選択局情報取得コマンドで示しています。送信テスト、ケーブルテスト、リモートリセットコマンド、並びに CSP+に記述したコマンドにおいても、送受信する処理は同様です。

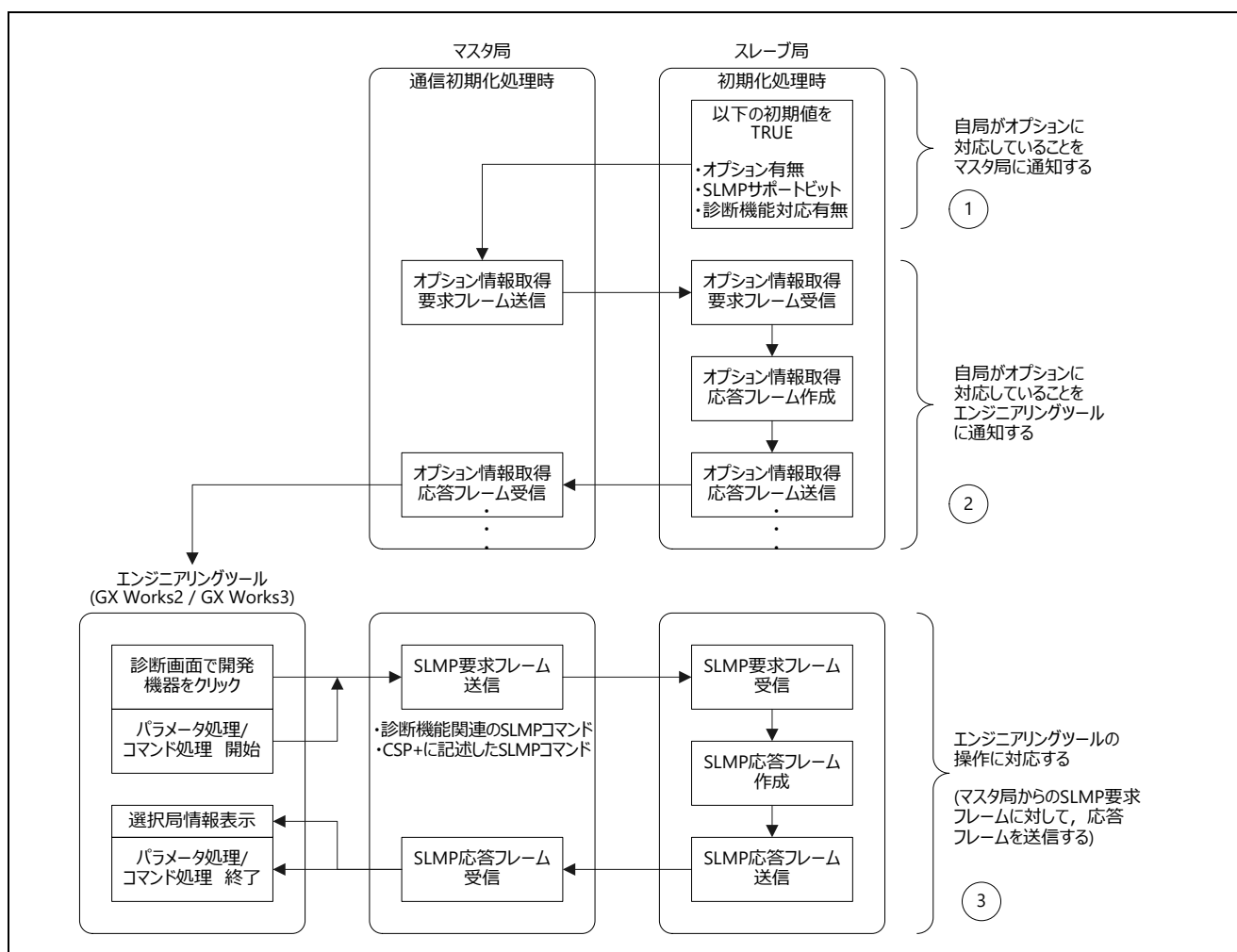


図 4.18 SLMP 要求の受信手順

1. R_IN32_UNITINIT_T の設定 (R-IN32M4-CL3 初期設定)

R_IN32_UNITINIT_T の以下のメンバに R_IN32_TRUE をセットしてください。

詳細は「表 5.7 R_IN32_UNITINIT_T 一覧」を参照してください。

- ulOptionSupport (オプション有無初期値)
- ulSlmpSupport (SLMP サポートビット初期値)
- ulSlmpDiagnosisSupport (診断機能対応有無初期値)

2. オプション情報取得要求フレームに応答

R-IN32M4-CL3 ドライバ・インタフェース関数 gerR_IN32_ReceivedOptionInfoRequest が、マスタ局からのオプション情報取得要求フレームに応答します。

3. 選択局情報取得要求フレームに応答

R-IN32M4-CL3 ドライバ・インタフェース関数 gerR_IN32_ReceivedSelectInfoRequest が、マスタ局からの選択局情報取得要求フレームに応答します。

(3) 選択局情報取得フレームの LED について

選択局通信状態モニタに自局の LED 状態を表示させるために、自局の状態に応じた LED 情報を、R_IN32_SELECTINFO_LED_INFO_T に格納してください。LED 情報は、R-IN32M4-CL3 適用回路に実装した LED の状態と一致させてください。

表 4.7 R_IN32_SELECTINFO_LED_INFO_T

No.	メンバ		内容
1	UCHAR	uchRow	LED 配列行数 (04h 固定)
2	UCHAR	uchColumn	LED 配列列数 (02h 固定)
3	R_IN32_LED_INFO_T	stLedInf[8]	LED 情報 1~8

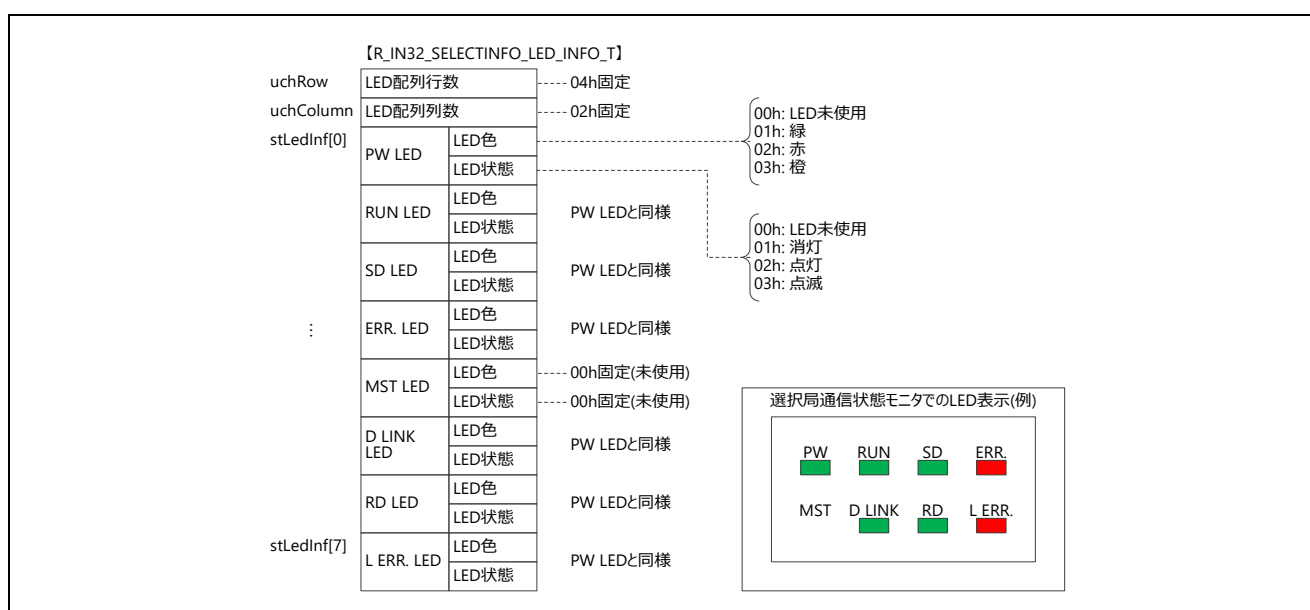


図 4.19 選択局情報取得フレームの LED

注意. 選択局情報取得の通信間隔よりも短い間隔で、R-IN32M4-CL3 上の LED 状態 (点灯/消灯/点滅) が変化すると、エンジニアリング・ツールへ LED 状態の変化が伝わりません。

(SD、RD など高速に点灯/消灯を繰り返す LED など)

GX Works2、GX Works3 を使用した場合、選択局情報取得の通信間隔は約 5 秒ですので、診断画面の LED 表示が実際の LED 状態と異なります。

4.3.4 Transient1 受信データ組立て開始処理

Transient1 フレームのデータ組立て処理を開始します。

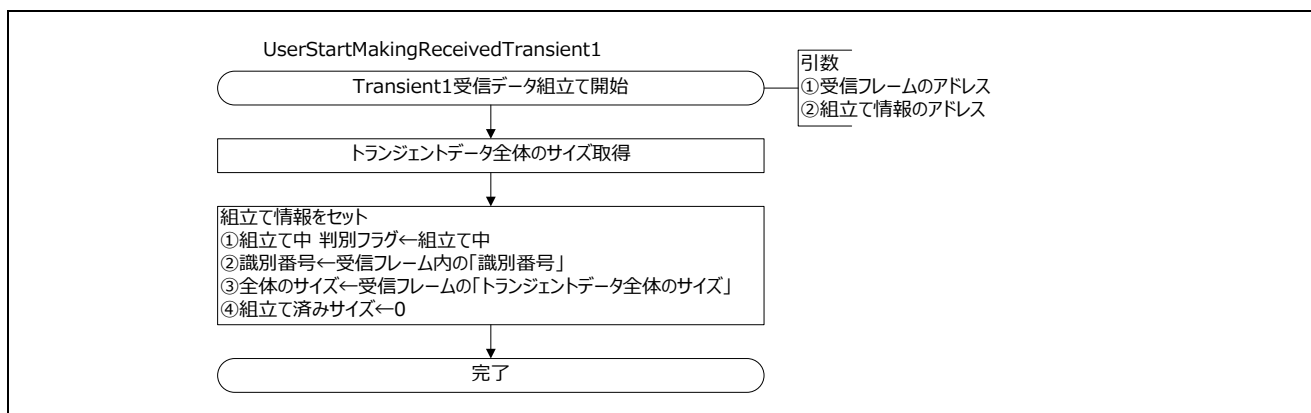


図 4.20 Transient1 受信データ組み立て開始 フロー図

4.3.5 Transient1 受信データ組立て処理

Transient1 フレームのデータ組立て処理を行います。

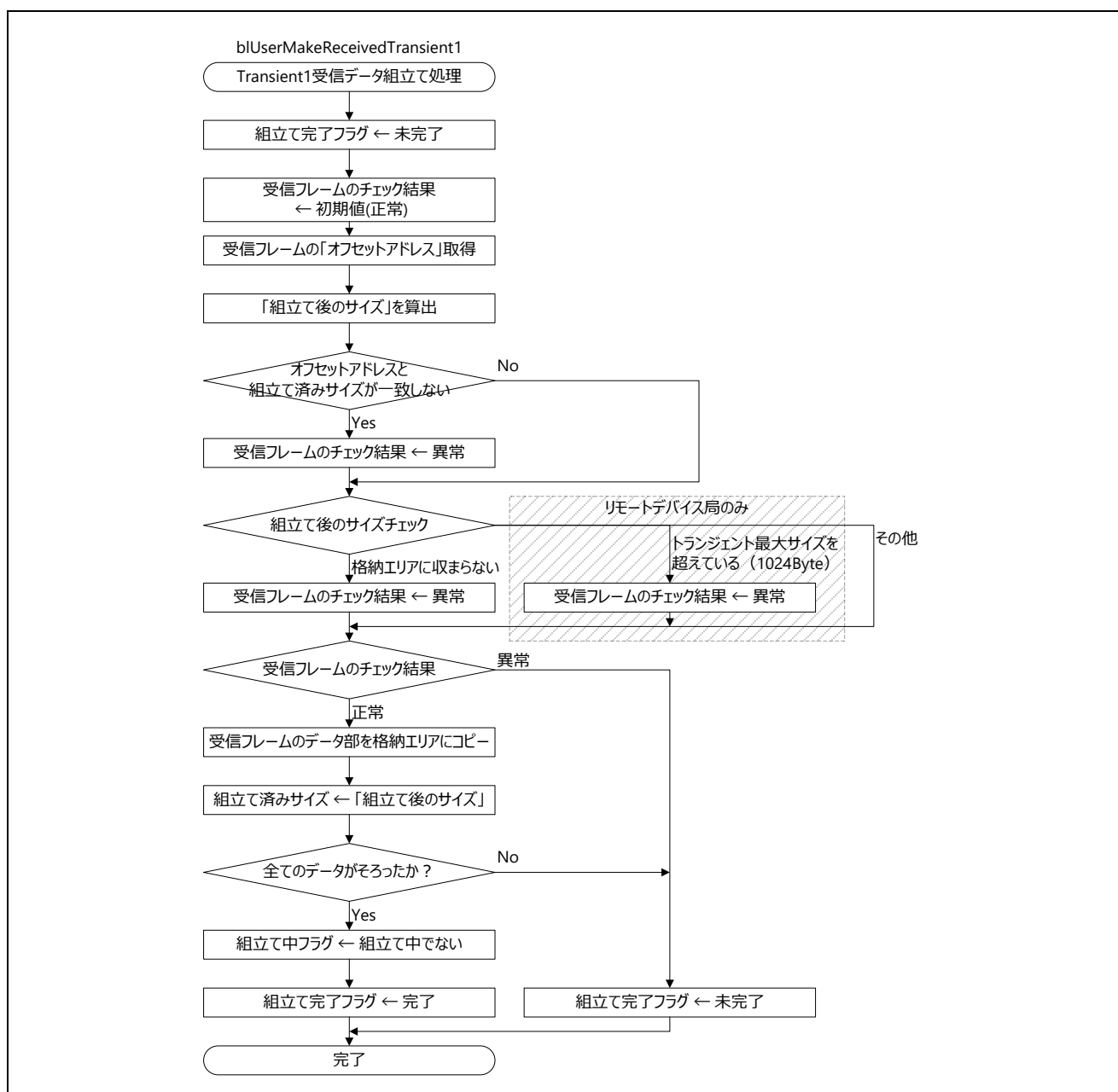


図 4.21 Transient1 受信データ処理 フロー図

4.3.6 SLMP メモリ読出し要求フレーム受信対応処理

他局から自局への SLMP メモリ読出し要求フレームを受信したときのフレーム受信処理を行います。

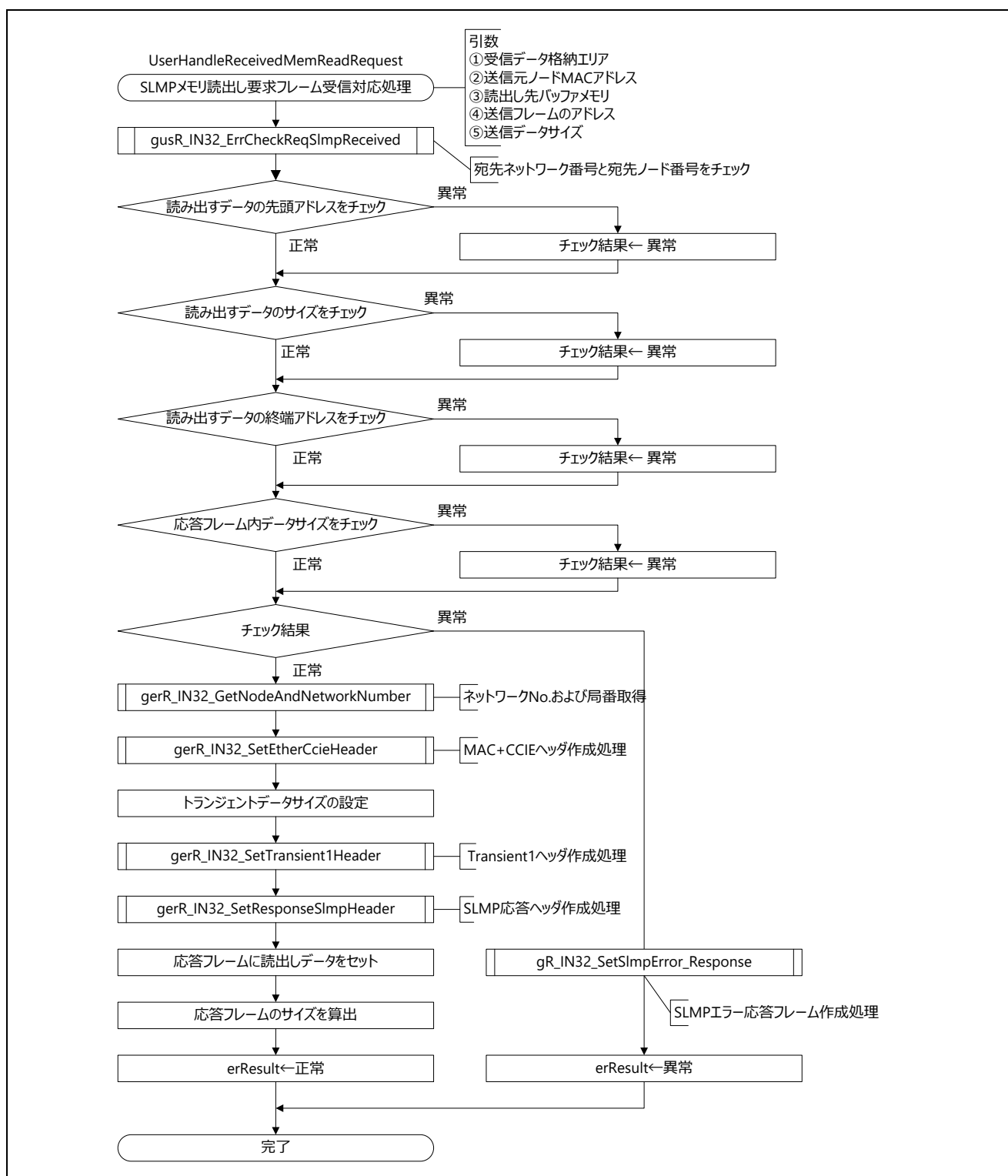


図 4.22 SLMP メモリ読出し要求フレーム受信対応処理 フロー図

4.3.7 SLMP メモリ書き込み要求フレーム受信対応処理

他局から自局への SLMP メモリ書き込み要求フレームを受信したときのフレーム受信処理を行います。

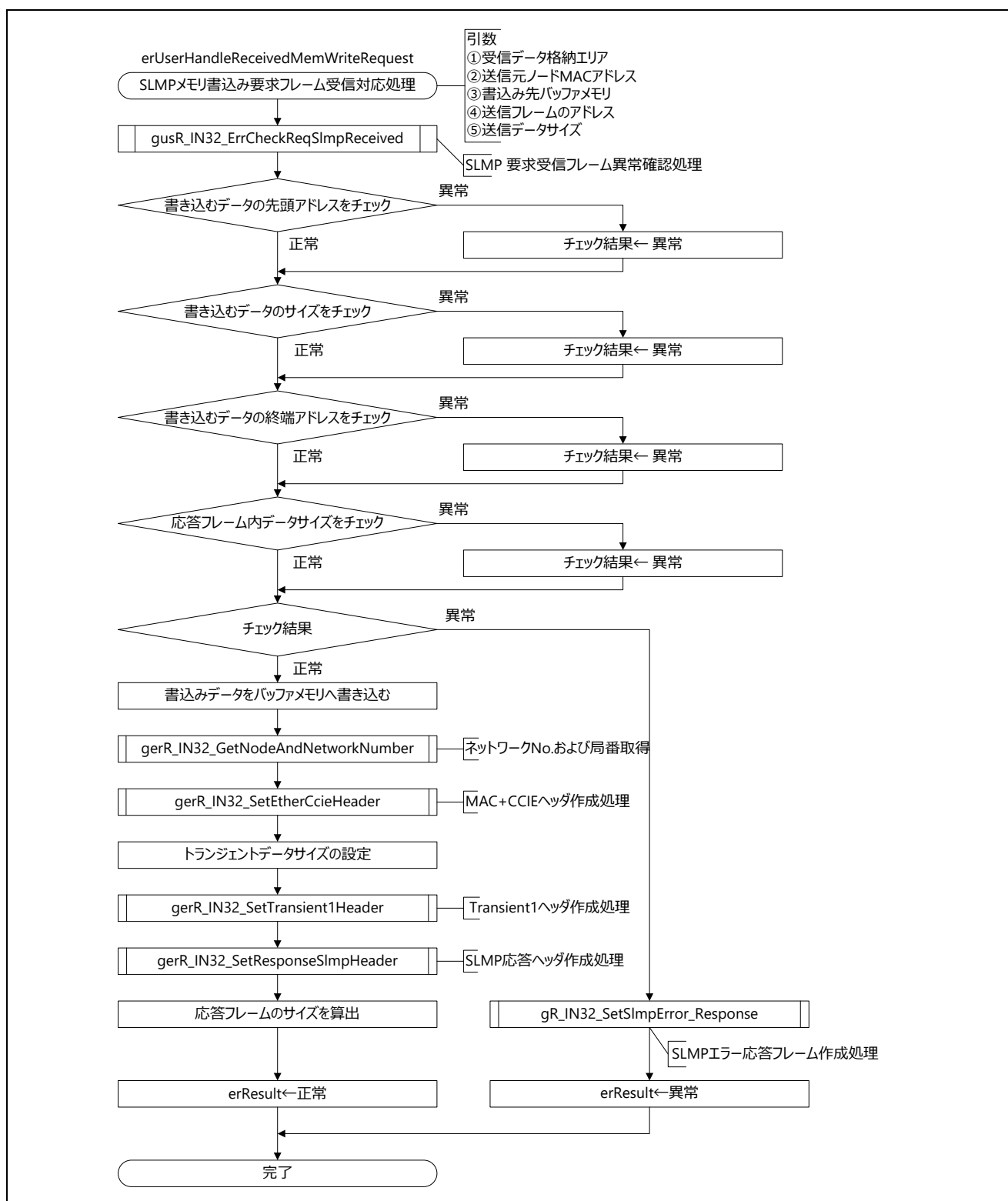
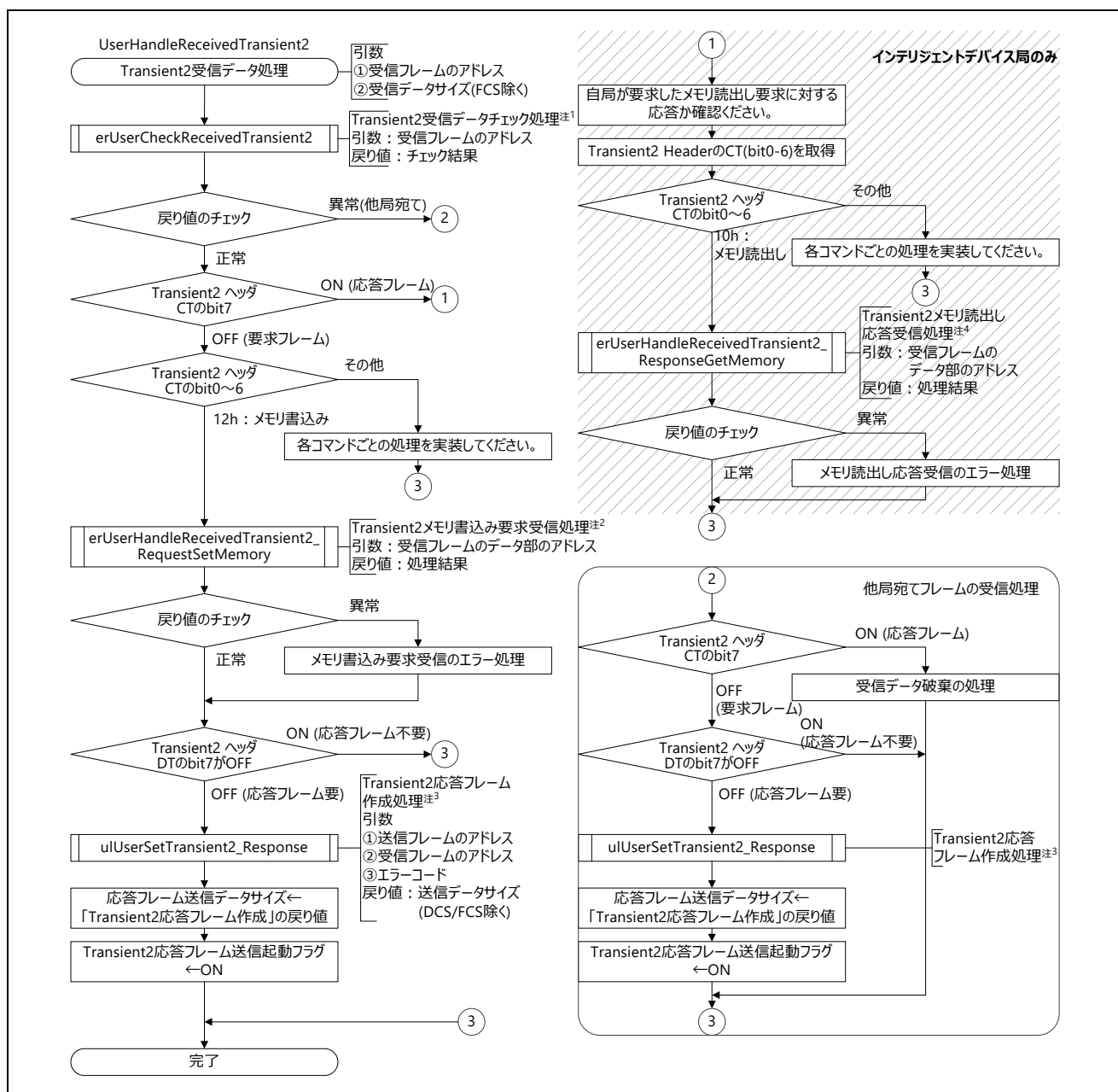


図 4.23 SLMP メモリ書き込み要求フレーム受信対応処理 フロー図

4.3.8 Transient2 受信データ処理

受信した Transient2 フレームを解析し、解析結果に応じて応答フレームの作成または応答フレームの受信を行います。



注 1. 詳細は、「4.3.9 Transient2 受信データチェック処理」を参照してください。

注 2. 詳細は、「4.3.10 Transient2 メモリ書き込み要求受信処理」を参照してください。

注 3. 詳細は、「4.3.11 Transient2 応答フレーム作成処理」を参照してください。

注 4. 詳細は、「4.3.21 Transient2 メモリ読み出し応答受信処理」を参照してください。

4.3.9 Transient2 受信データチェック処理

受信した Transient2 フレームが自局宛てか、宛先局番（DA・DS）、および宛先ネットワーク No.（DNA）をチェックします。

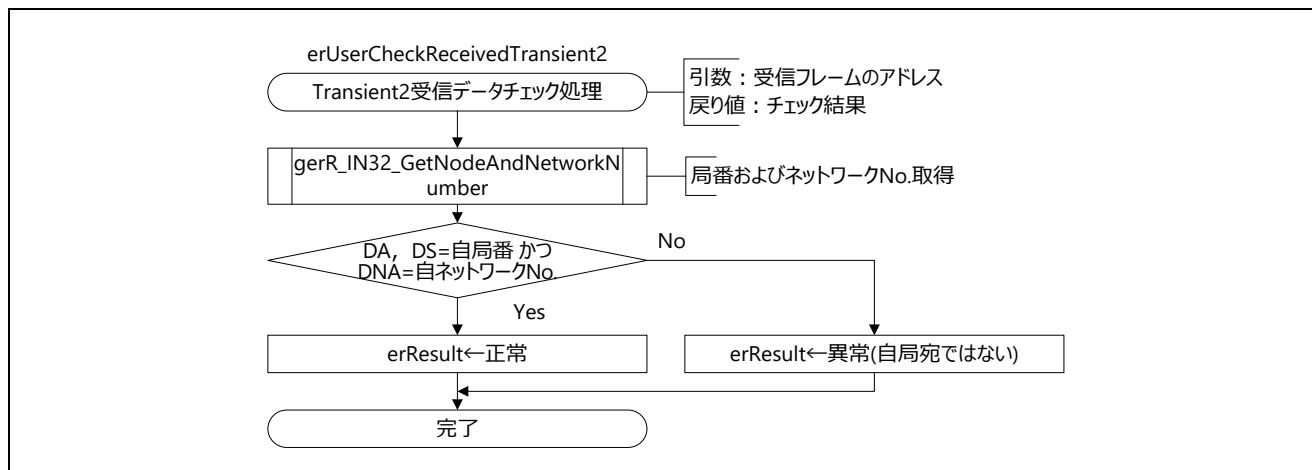


図 4.25 Transient2 受信データチェック処理 フロー図

4.3.10 Transient2 メモリ書き込み要求受信処理

他局から自局への Transient2 メモリ書き込み要求フレームを受信したときのフレーム受信処理を行います。

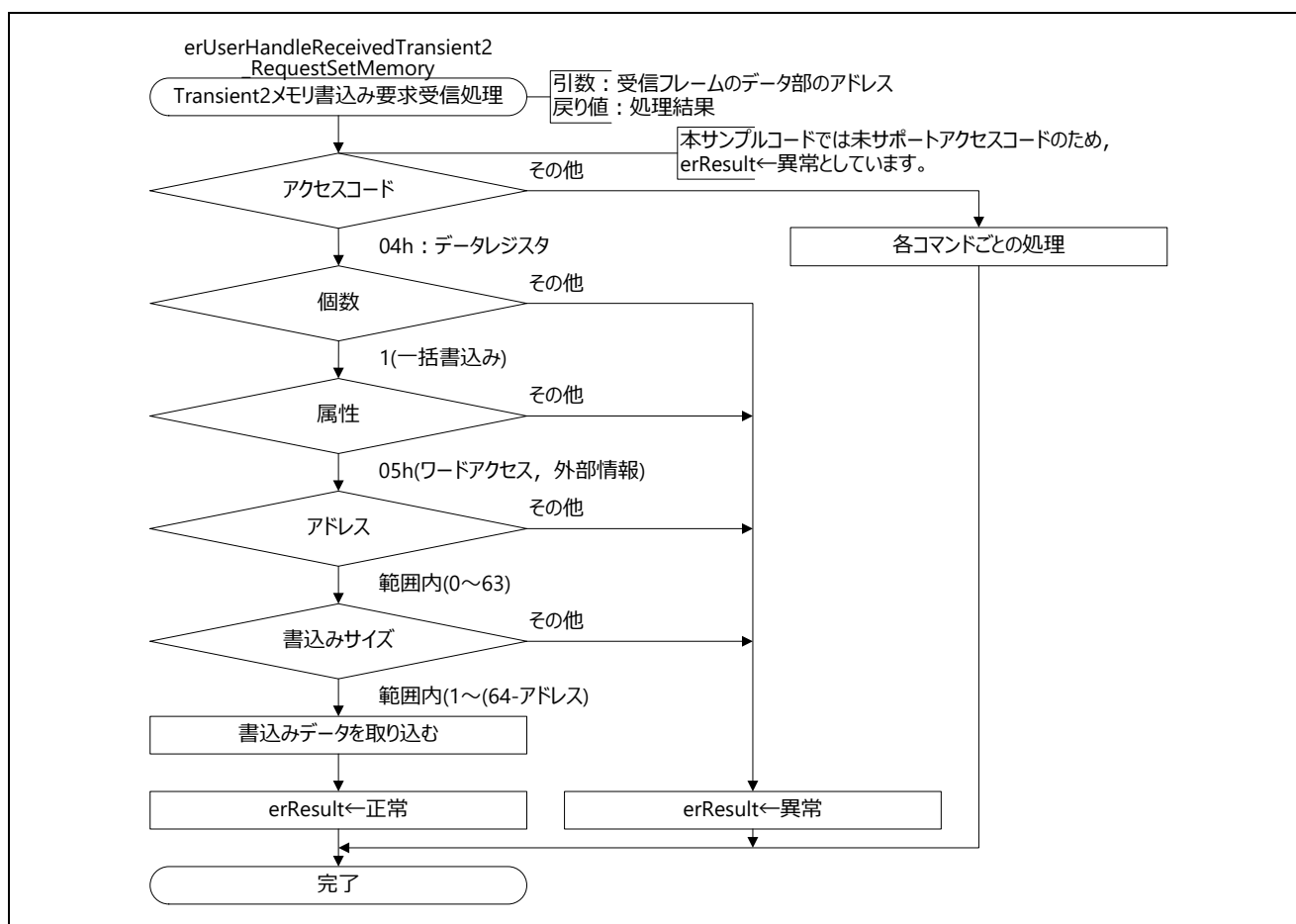


図 4.26 Transient2 メモリ書き込み要求受信処理 フロー図

本フローは、下記設定でのコマンド処理を例示しています。下記以外の場合、異常とします。

- (1) メモリ書き込み
- (2) アクセス・コード : 04h (データレジスタ)
- (3) 属性 : 05h (ワードアクセス、外部情報)
- (4) アドレス (書き込み先の先頭アドレス) : 0
- (5) 書き込みサイズ : 1~ (64-アドレス)

4.3.11 Transient2 応答フレーム作成処理

Transient2 応答フレームを作成します。

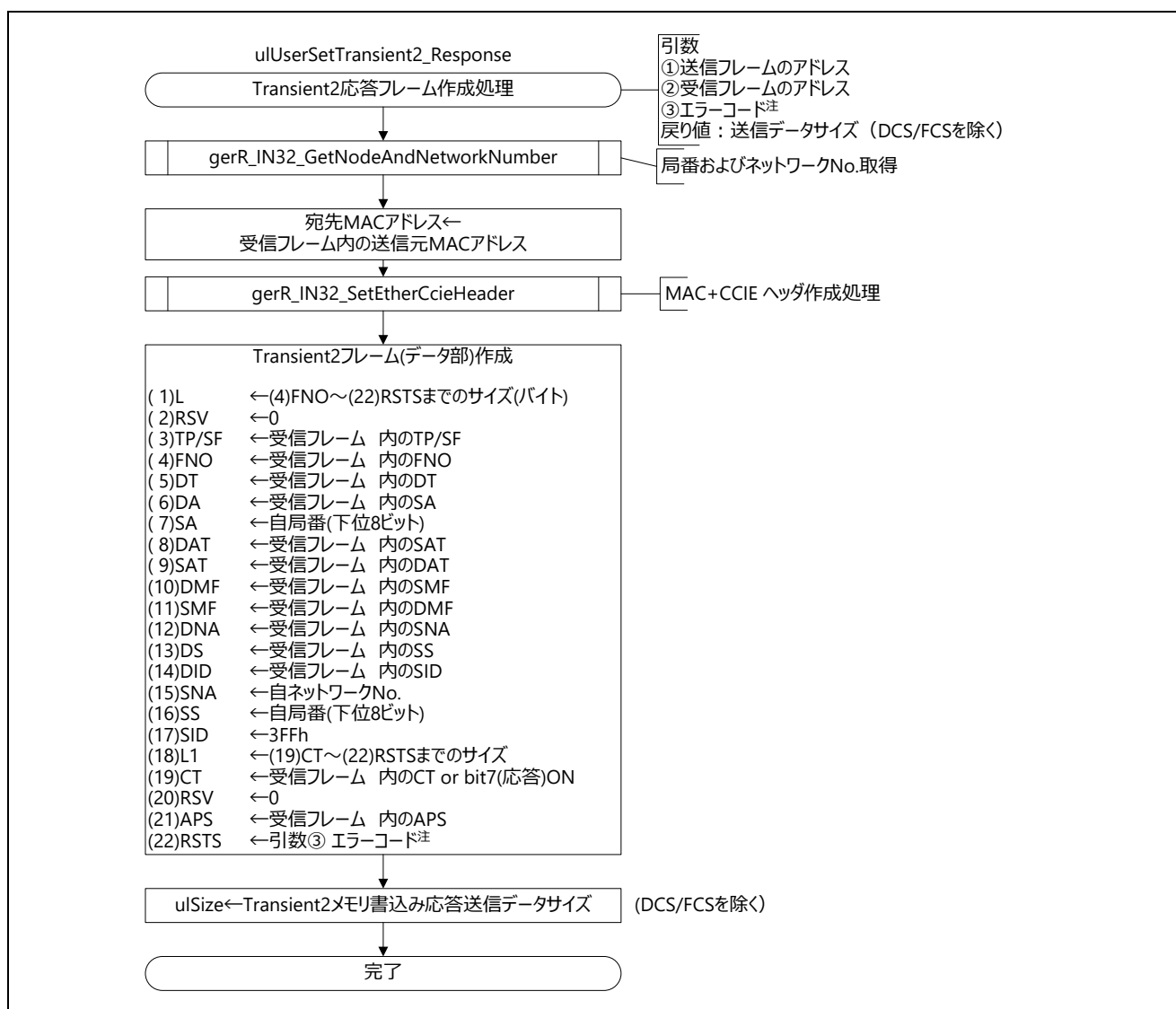


図 4.27 Transient2 応答フレーム作成処理 フロー図

注. ③エラー・コードの詳細は、「7.3.3 リターン・コード (RSTS) 詳細」を参照してください。

4.3.12 TransientAck 受信データ処理

受信した TransientAck フレームを解析、および解析結果に応じた処理を追加してください。

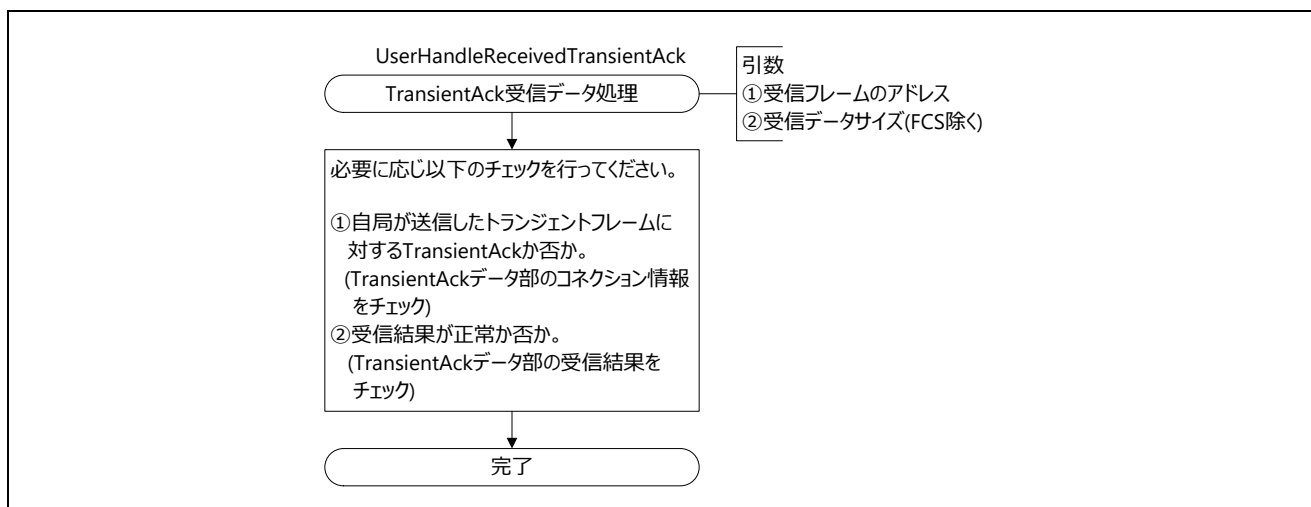


図 4.28 TransientAck 受信データ処理 フロー図

4.3.13 TransientAck フレーム作成処理

TransientAck フレームを作成します。

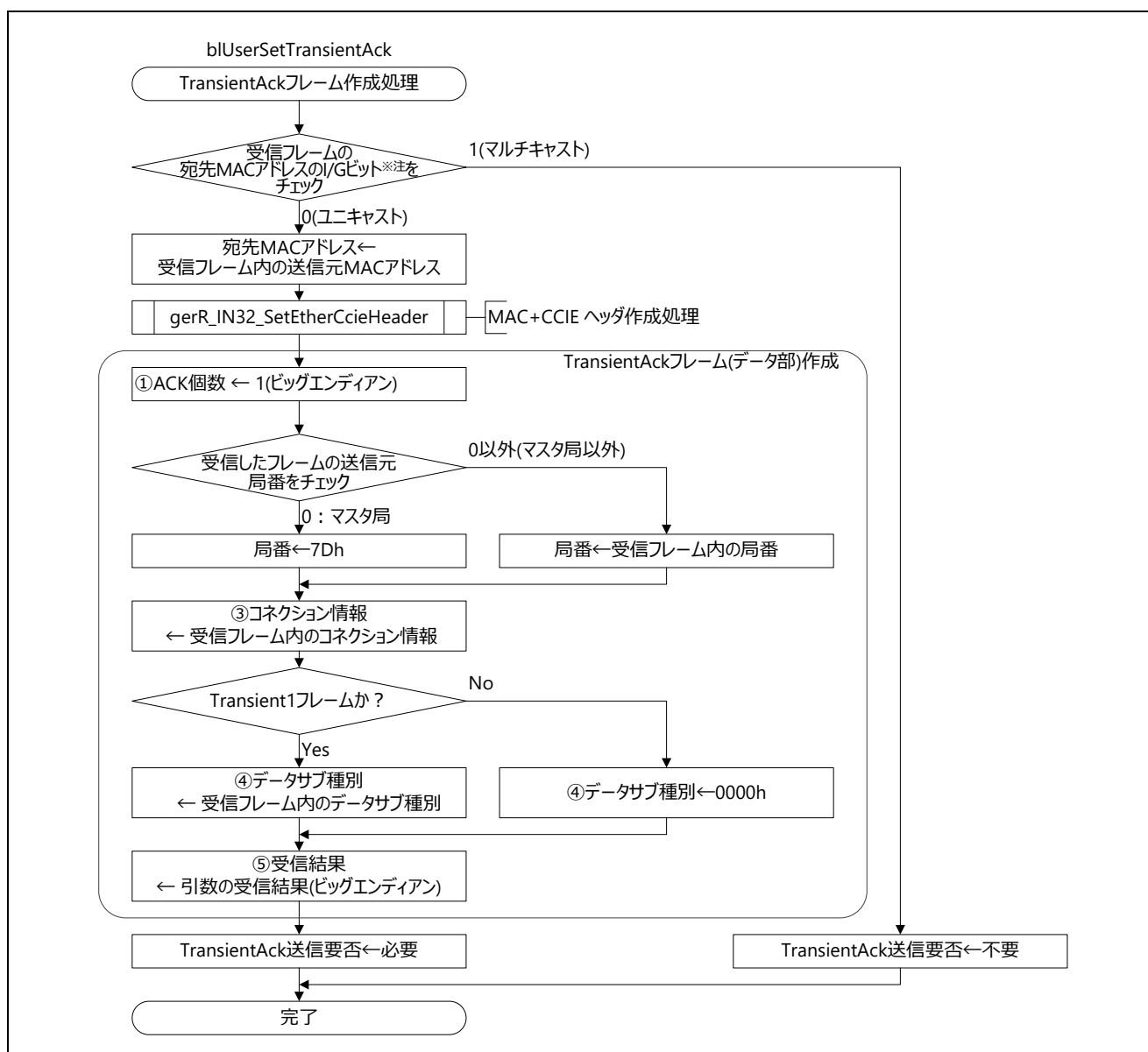
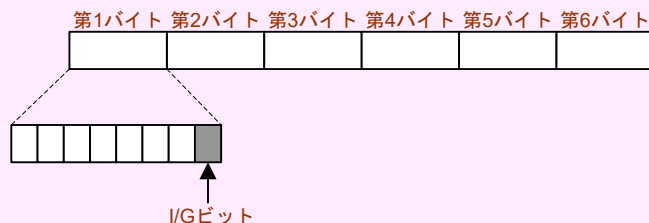


図 4.29 TransientAck フレーム作成処理 フロー図

注. I/G ビットとは、MAC アドレスの第 1 バイト（オクテット）の最下位ビットです。



4.3.14 Transient1、Transient2、TransientAck 送信処理

Transient1、Transient2、TransientAck フレームを送信します。

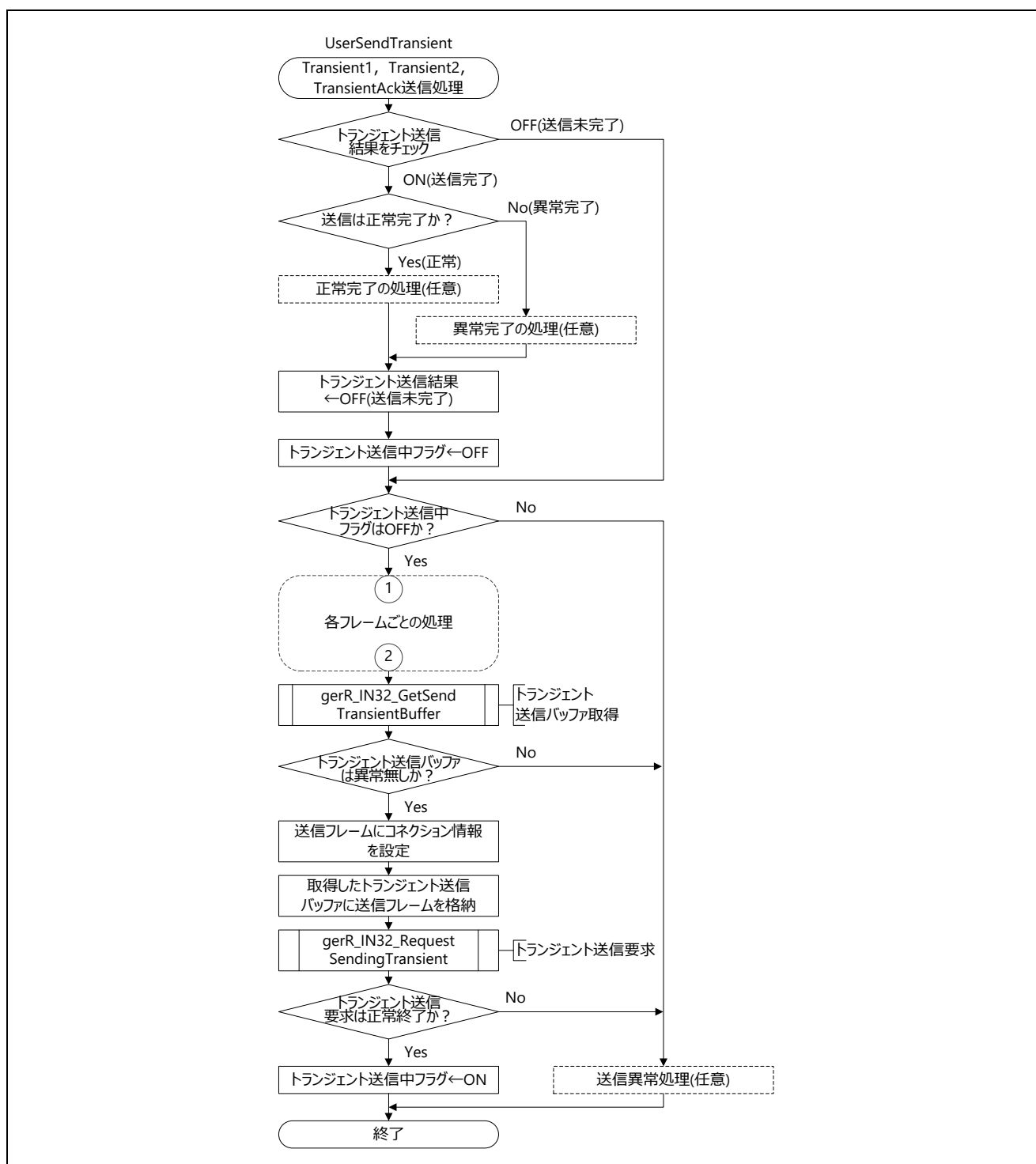


図 4.30 Transient1、Transient2、TransientAck 送信処理 フロー図 (1/2)

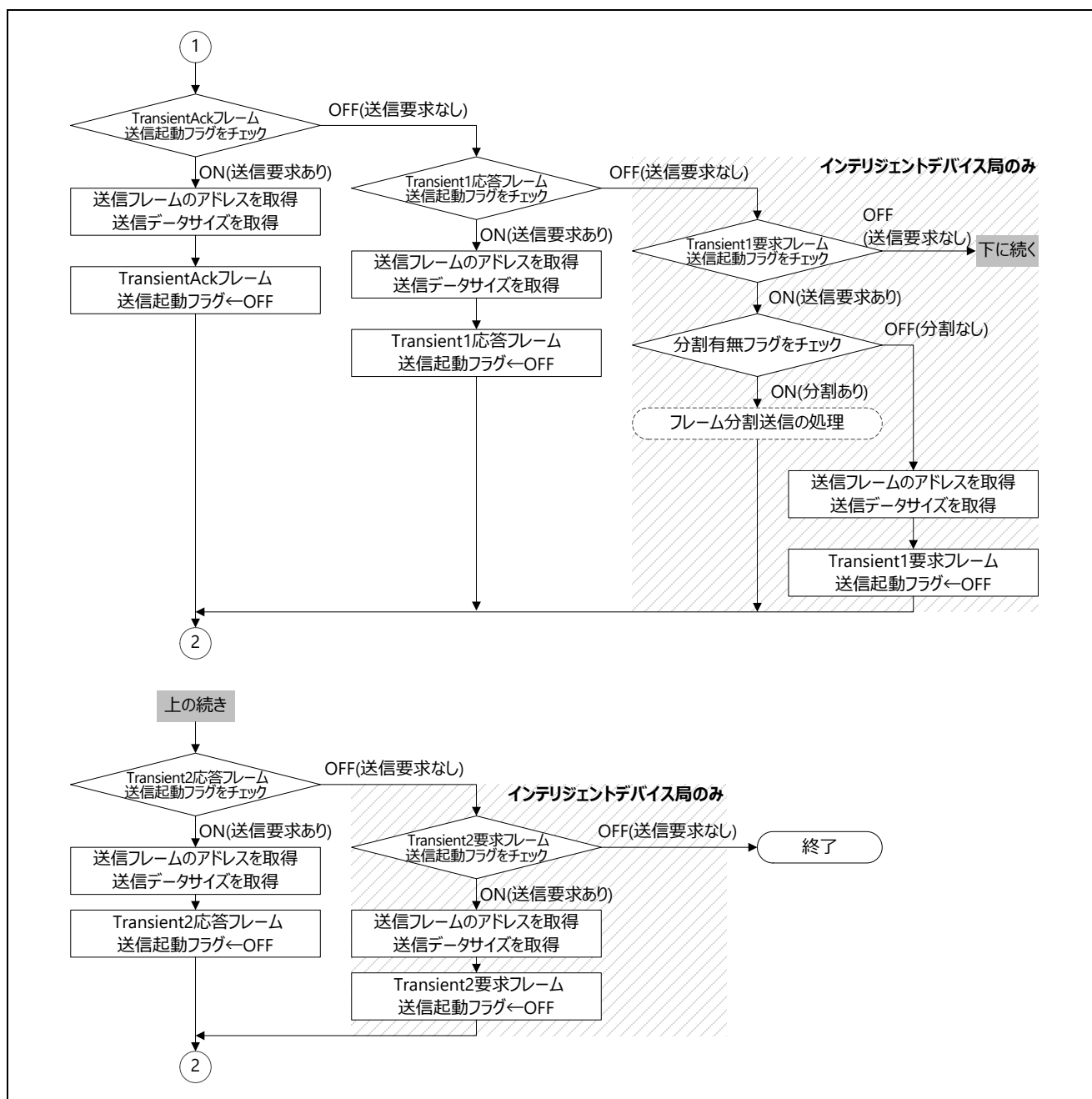


図 4.30 Transient1、Transient2、TransientAck 送信処理 フロー図 (2/2)

【分割送信について】

要求送信するトランジェントデータが 1466 バイト～2048 バイトある場合、トランジェントデータを分割して送信することができます。R-IN32M4-CL3 適用製品の仕様に応じて、本処理を実装してください。

以下に分割送信処理のイメージを示します。Transient1 フレームの詳細は「7.2 Transient1 フレーム」を参照してください。

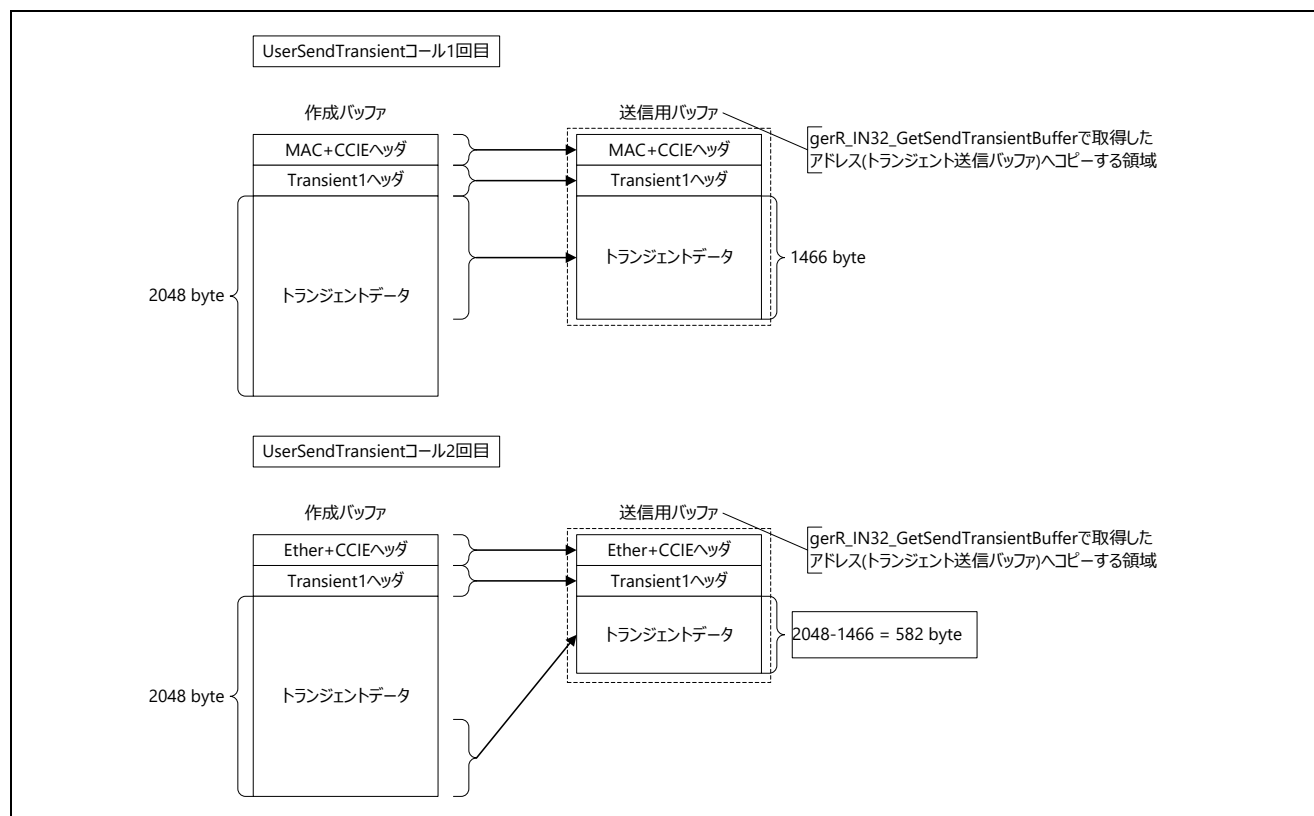


図 4.31 トランジェントデータのサイズが 2048byte の分割転送（分割フラグが ON のとき）

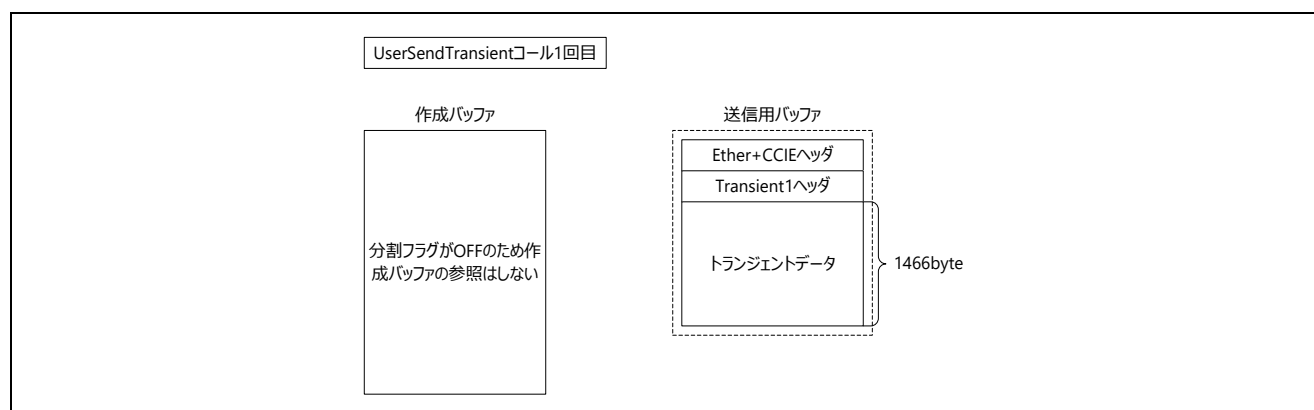


図 4.32 トランジェントデータのサイズが 1466byte 以下の転送（分割フラグが OFF のとき）

4.3.15 SLMP メモリ読出し要求フレーム作成処理

自局から他局へ SLMP 要求フレームをするときの、要求フレーム作成の処理を SLMP メモリ読出し要求の例で示します。

その他のコマンドは必要に応じて処理を追加してください。

本処理は、自局がインテリジェントデバイス局の場合のみ実装してください。

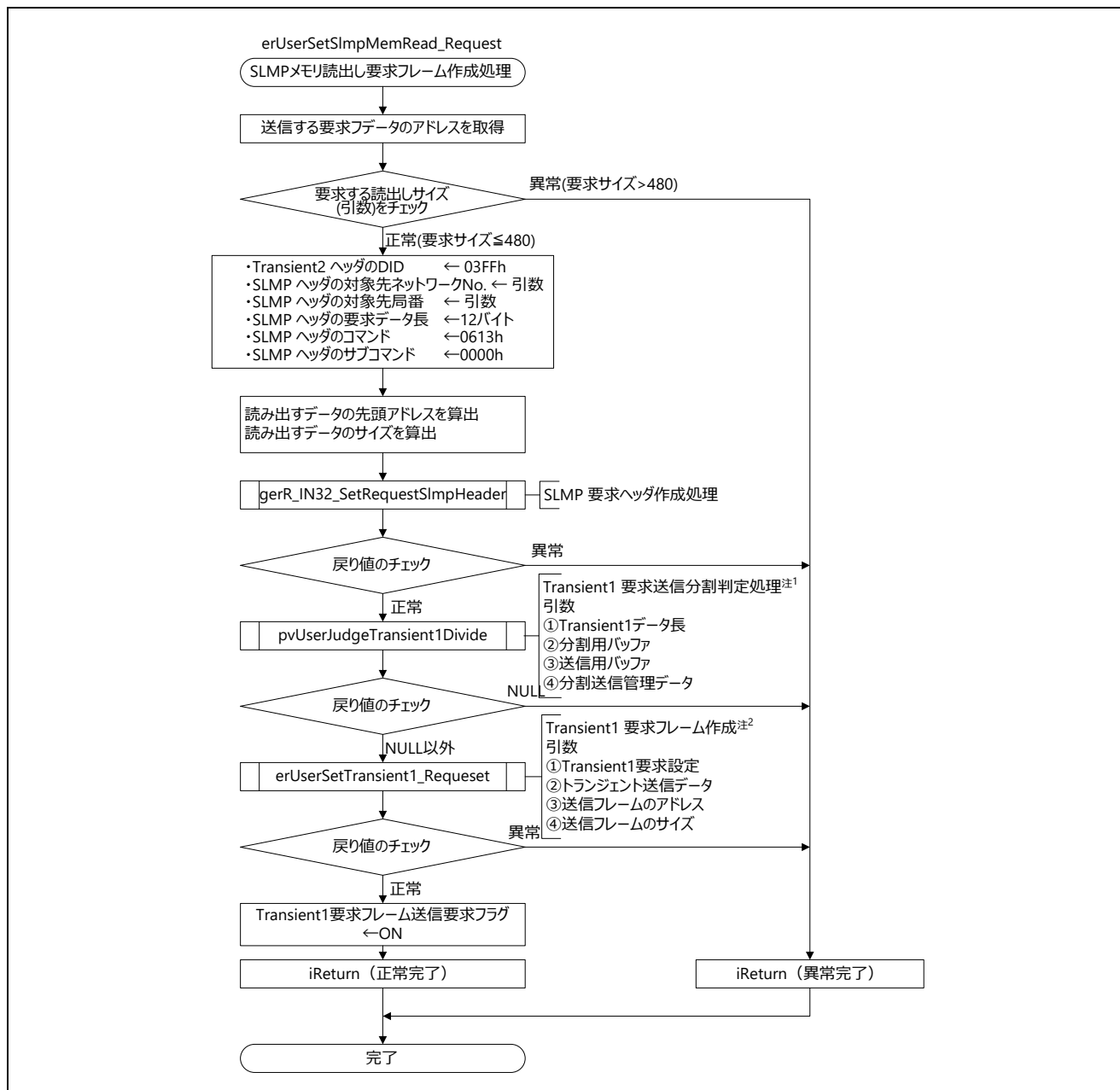


図 4.33 SLMP メモリ読出し要求フレーム作成処理 フロー図

注 1. 詳細は、「4.3.16 Transient1 要求送信分割判定処理」を参照してください。

2. 詳細は、「4.3.17 Transient1 要求フレーム作成処理」を参照してください。

4.3.16 Transient1 要求送信分割判定処理

Transient1 要求フレームの作成時、フレームを分割して送信するかを判定します。

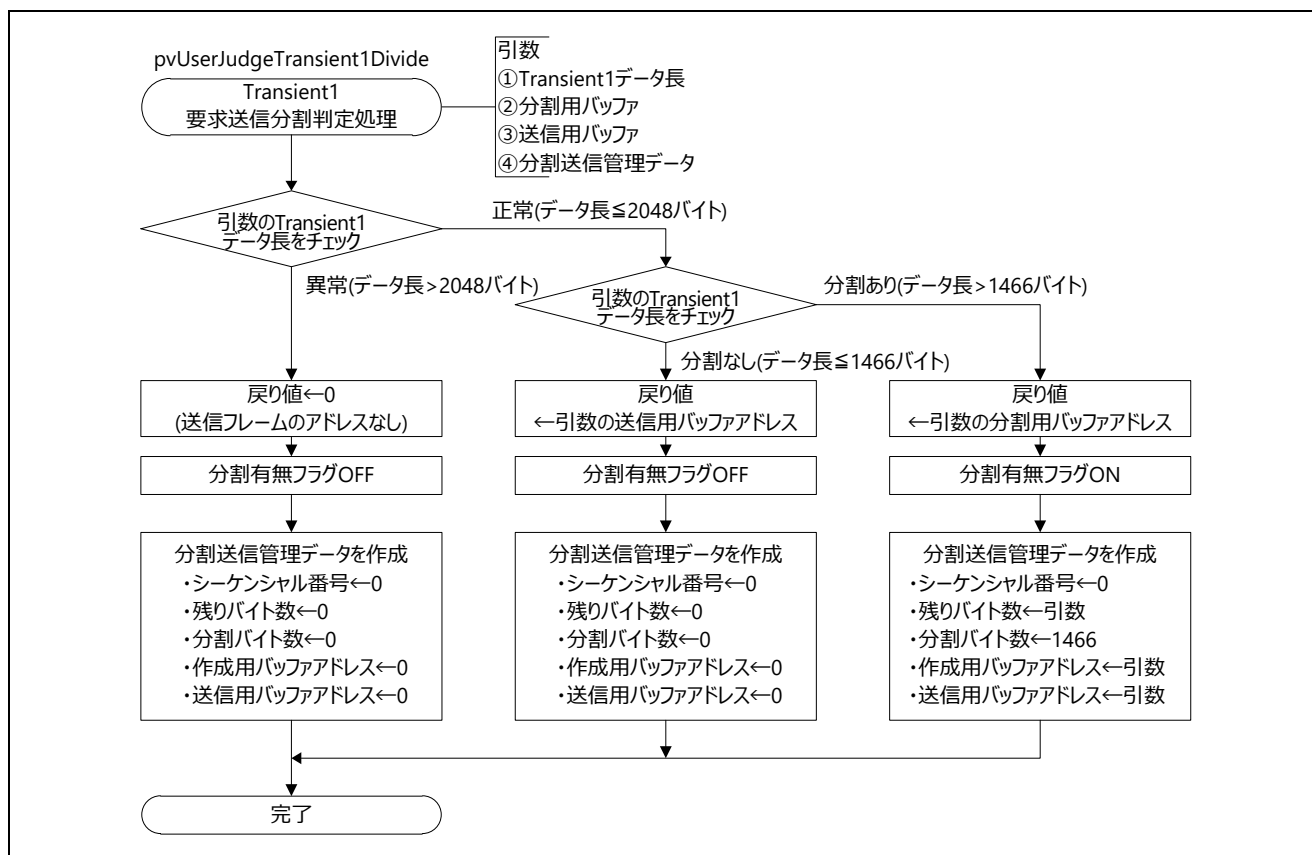


図 4.34 Transient1 要求送信分割判定処理 フロー図

4.3.17 Transient1 要求フレーム作成処理

自局から他局へ SLMP メモリ読出し要求を送信するときの、MAC ヘッダから Transient1 ヘッダまでの要求フレームを作成します。

本処理は、自局がインテリジェントデバイス局の場合のみ実装してください。

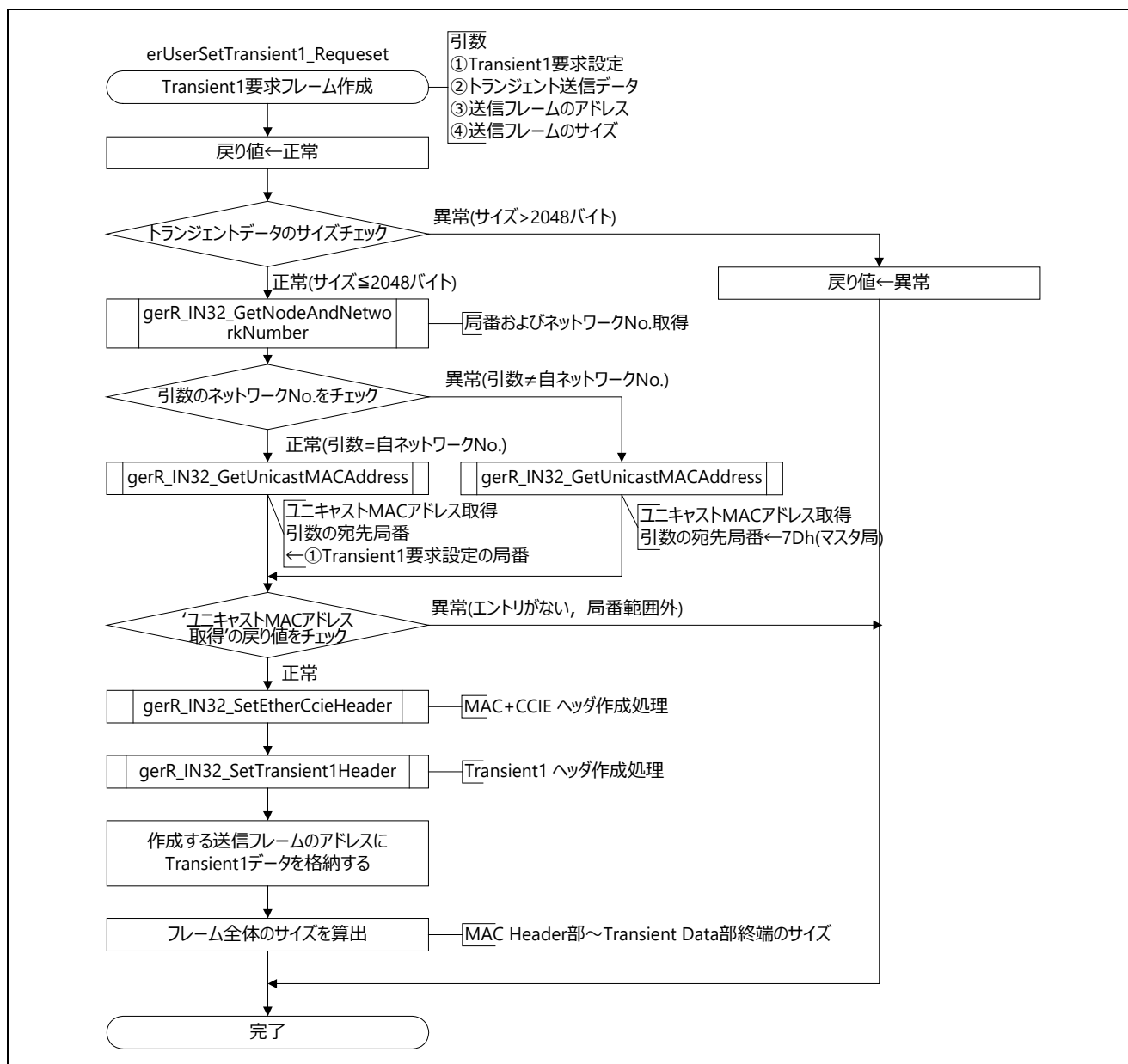


図 4.35 Transient1 要求フレーム作成処理 フロー図

4.3.18 SLMP メモリ読出し応答受信処理

自局が他局へ要求した SLMP メモリ読出しの応答フレーム受信処理を行います。本処理は、自局がインテリジェントデバイス局の場合のみ実装してください。

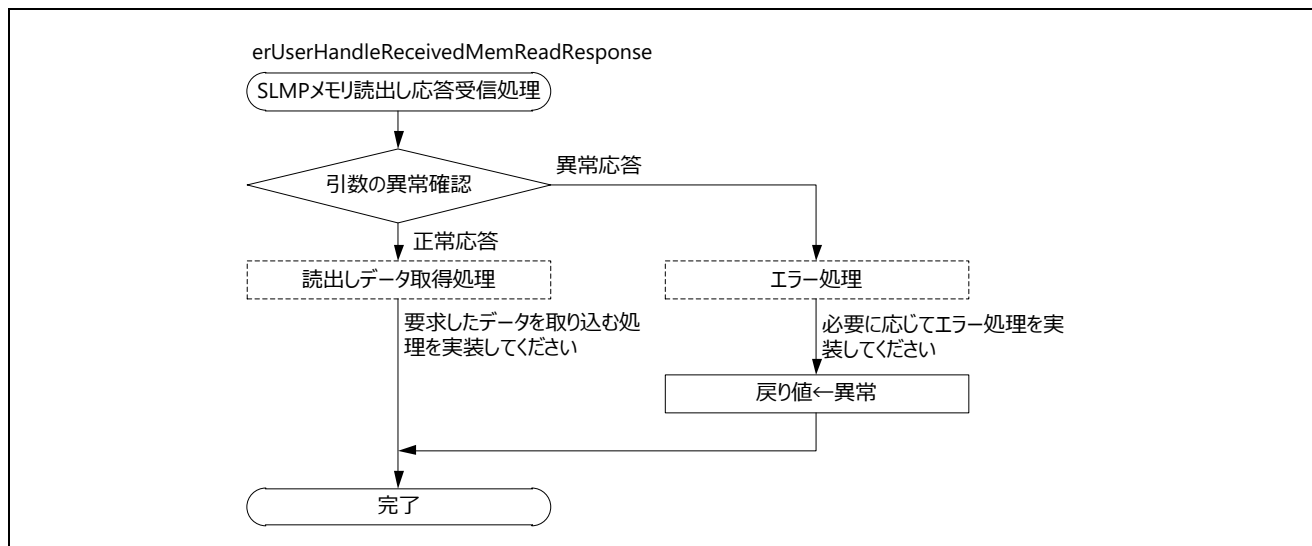


図 4.36 SLMP メモリ読出し応答受信処理 フロー図

4.3.19 Transient2 要求フレーム作成処理

自局から他局へ Transient2 要求フレームをするときの要求フレーム作成の処理を Transient2 メモリ読出し要求の例で示します。その他のコマンドは必要に応じて処理を追加してください。

本処理は、自局がインテリジェントデバイス局の場合のみ実装してください。

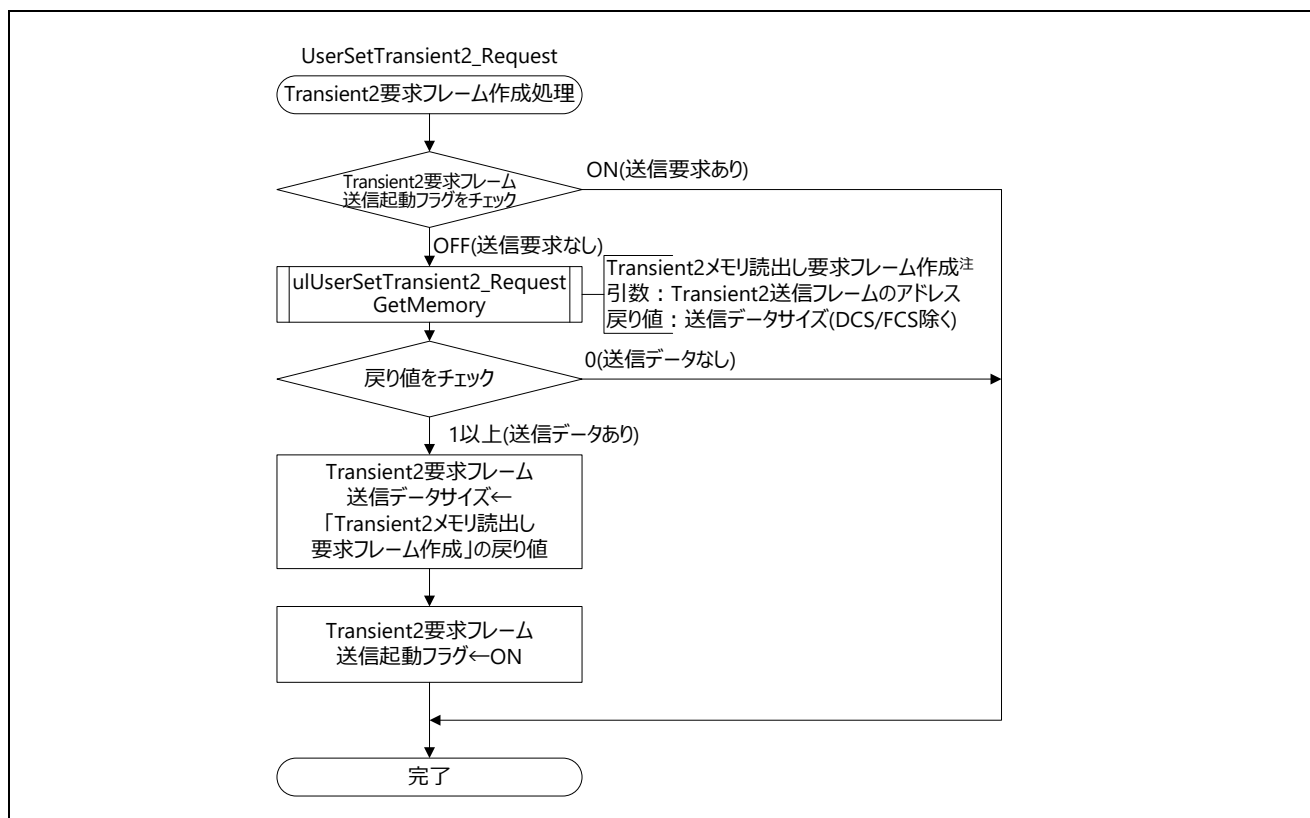


図 4.37 Transient2 要求フレーム作成処理 フロー図

注. 詳細は、「4.3.20 Transient2 メモリ読出し要求フレーム作成処理」を参照してください。

4.3.20 Transient2 メモリ読出し要求フレーム作成処理

自局から他局へ Transient2 メモリ読出し要求を送信するときの要求フレームを作成します。本処理は、自局がインテリジェントデバイス局の場合のみ実装してください。

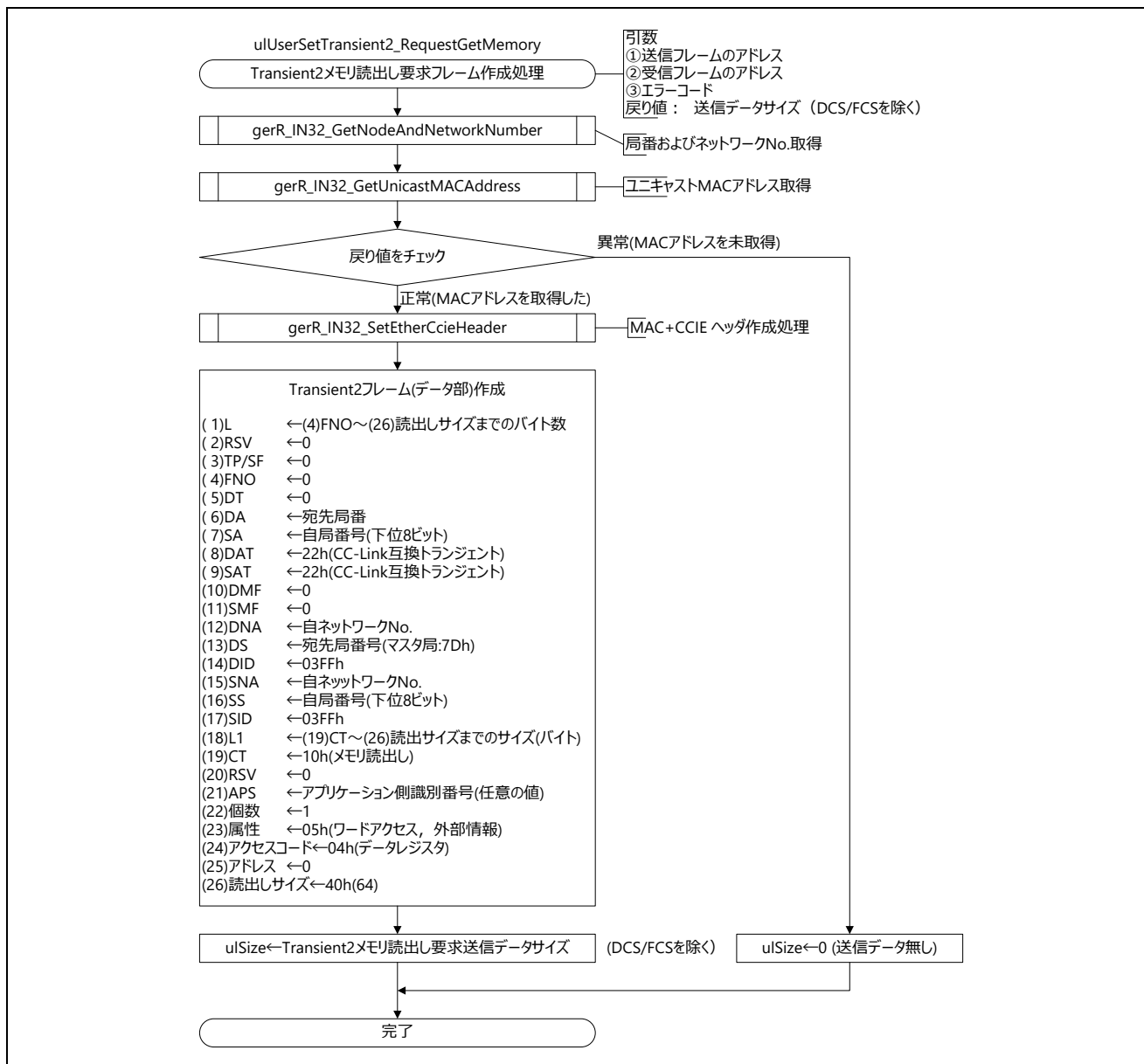


図 4.38 Transient2 メモリ読出し要求フレーム作成処理 フロー図

本フローは、下記設定でのコマンド処理を例示しています。下記以外の場合、異常とします。

- (1) 宛先局番：7Dh（マスタ局）
- (2) アクセス・コード：04h（データレジスタ）
- (3) 属性：05h（ワードアクセス、外部情報）
- (4) アドレス（読出し先の先頭アドレス）：0
- (5) 読出しサイズ：40h（64）

4.3.21 Transient2 メモリ読出し応答受信処理

自局が他局へ要求した Transient2 メモリ読出しの応答フレーム受信処理を行います。本処理は、自局がインテリジェントデバイス局の場合のみ実装してください。

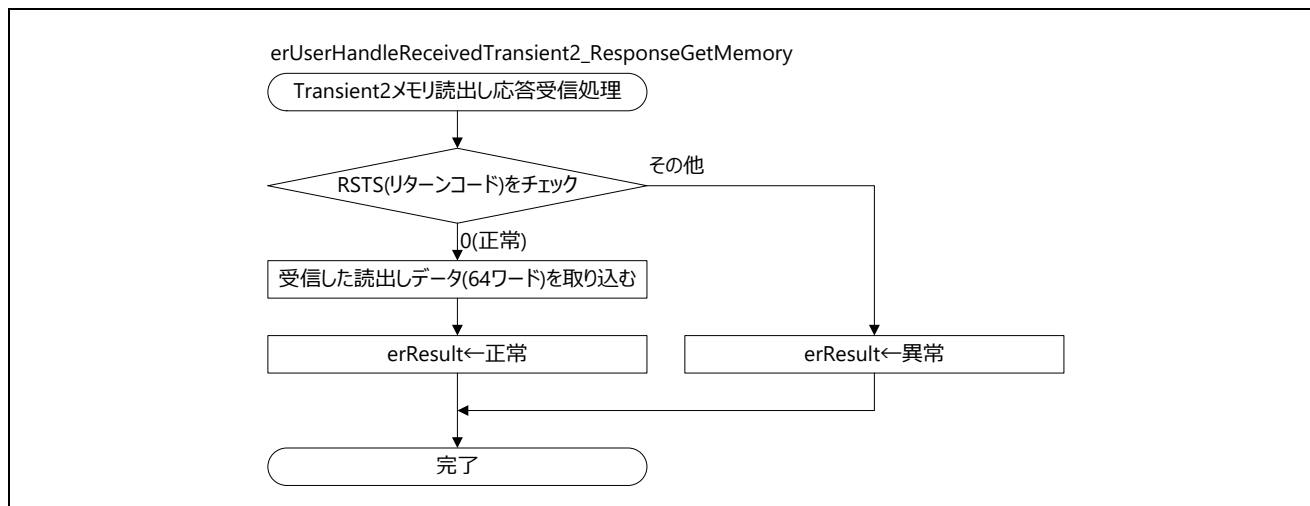


図 4.39 Transient2 メモリ読出し応答受信処理 フロー図

本フローは、下記設定でのコマンド処理を例示しています。下記以外の場合、異常とします。

- (1) 宛先局番：7Dh（マスタ局）
- (2) アクセス・コード：04h（データレジスタ）
- (3) 属性：05h（ワードアクセス、外部情報）
- (4) アドレス（読出し先の先頭アドレス）：0
- (5) 読出しサイズ：40h（64）

4.4 ユーザ・プログラム詳細 (H/W テスト関連)

H/W テストはメイン処理 (iUserMainRoutine) では実施せず、独立した処理として実装してください。

(実装例：R-IN32M4-CL3 適用製品の外部 (局番/ネットワーク No.スイッチ、またはエンジニアリング・ツールなど) から、通常運転時のモード (オンラインモード) 以外の「ハードウェア・テストモード (オフラインモード)」に切り替えられる機能を実装する。)

4.4.1 H/W テスト (IEEE802.3ab コンプライアンス・テスト)

コンフォーマンス・テストの「1000BASE-T コンプライアンス・テスト試験」を実施するために本機能を実装する必要があります。

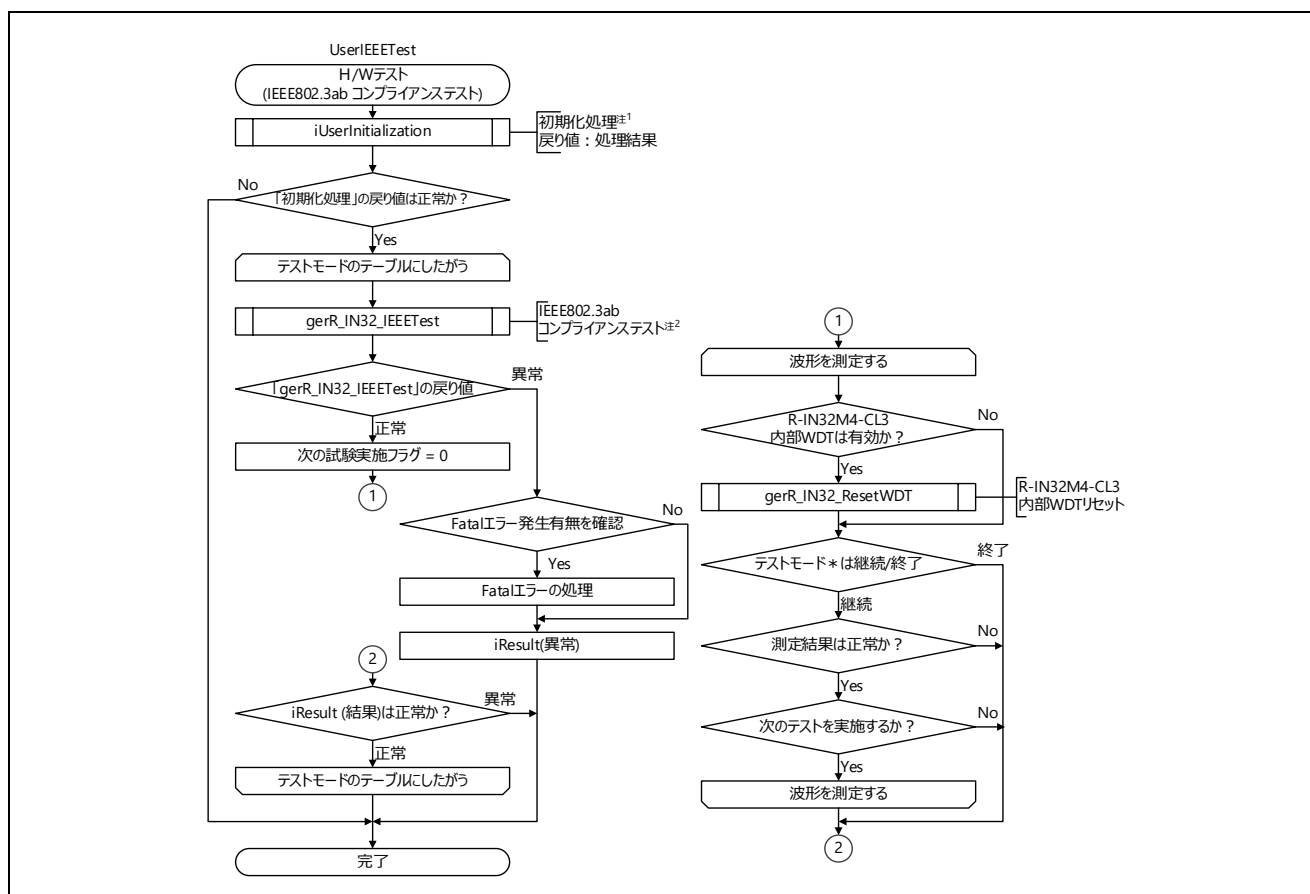


図 4.40 H/W テスト (IEEE802.3ab コンプライアンス・テスト) フロー図

注 1. 詳細は、「4.2.2 初期化処理」を参照してください。

2. R-IN32M4-CL3 に Fatal エラーが発生した場合、関数 gerR_IN32_IEEETest より、ユーザ作成の以下の関数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。

- ・ gR_IN32_CallbackFatalError

4.4.2 H/W テスト（折り返し通信テスト）

R-IN32M4-CL3 に H/W の異常がないかをチェックします。

本テストは、Ethernet PORT1 と 2 を Ethernet ケーブルで接続して実施してください。

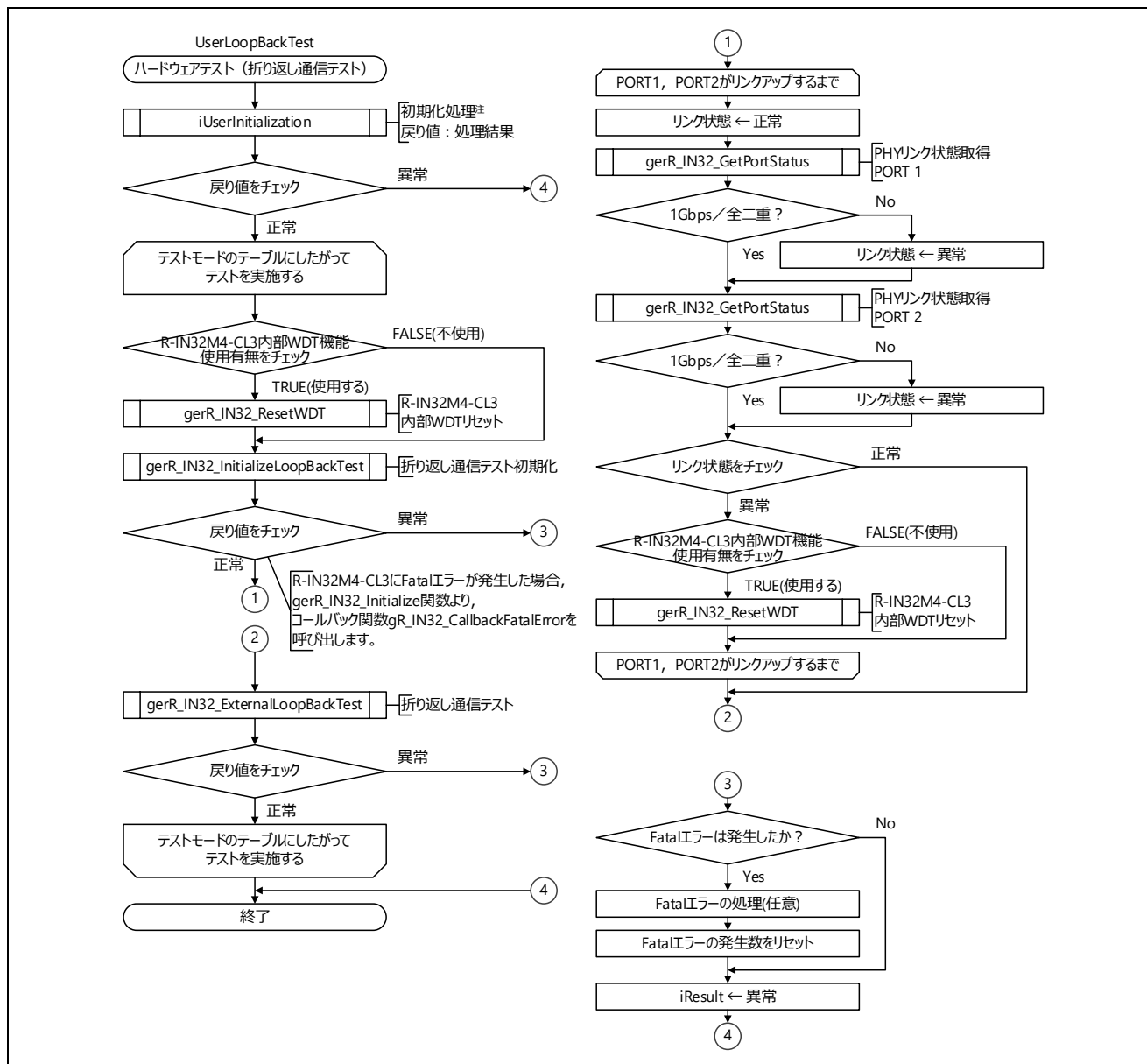


図 4.41 H/W テスト（折り返し通信テスト）フロー図

注. 詳細は、「4.2.2 初期化処理」を参照してください。

5. 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ドライバ・インタフェース関数、ドライバ・コールバック関数の仕様について示します。

5.1 各関数の概要

5.1.1 概要

各関数の概要と変更の要否を下表に示します。

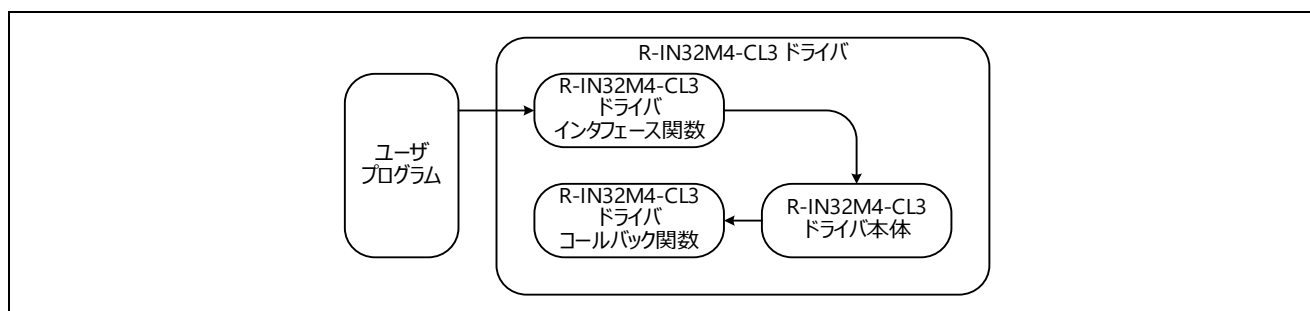


図 5.1 各関数の関連

表 5.1 各関数の概要

プログラム部品名	概要	変更要否
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コールする関数。 (ファイル: R_IN32_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・プログラム側の処理を記述する。 (ファイル: R_IN32M4_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_IN32_Interface.c を除いた driver フォルダ以下のファイル)	×

5.1.2 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 5.2 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 5.3 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_IN32_OK	0	正常
2	R_IN32_ERR	-1	異常終了（状態異常/不一致）
3	R_IN32_ERR_OTHER	-2	（ライブラリ内ドライバで異常発生）
4	R_IN32_ERR_OUTOFRANGE	-3	範囲外
5	R_IN32_ERR_EMPTY	-4	エンプティ
6	R_IN32_ERR_OVERFLOW	-5	オーバフロー
7	R_IN32_ERR_NOENTRY	-6	エントリがない
8	R_IN32_ERR_NOPERMIT	-7	許可されていない
9	R_IN32_ERR_NODATA	-8	データがない
10	R_IN32_ERR_NOMYSTATUS	-9	有効な MyStatus がない

5.1.3 記述仕様

各関数の記述仕様を示します。

表 5.4 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	—

5.2 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 5.5 R-IN32M4-CL3 ドライバ・インタフェース関数一覧

(1/3)

機能分類 (参照 項)	関数名	関数型	概要
初期設定 (5.3.1 初期設定)	gulR_IN32_GetResetStatus	ULONG	リセット状態取得
	gerR_IN32_Initialize	ERRCODE	R-IN32M4-CL3 の初期化
	gerR_IN32_SetNodeAndNetworkNumber	ERRCODE	局番およびネットワーク No. 設定
	gerR_IN32_Start	ERRCODE	R-IN32M4-CL3 通信開始
WDT (5.3.2 ウォッチ ドッグタイマ)	gerR_IN32_ResetWDT	ERRCODE	R-IN32M4-CL3 内部 WDT リセット
	gerR_IN32_DisableWDT	ERRCODE	R-IN32M4-CL3 内部 WDT 無効
	gerR_IN32_EnableWDT	ERRCODE	R-IN32M4-CL3 内部 WDT 有効
	gerR_IN32_SetWDT	ERRCODE	R-IN32M4-CL3 内部 WDT 時限設定
イベント (5.3.3 イベント)	gerR_IN32_GetEvent	ERRCODE	R-IN32M4-CL3 イベント検出
	gerR_IN32_Main	ERRCODE	R-IN32M4-CL3 イベント検出メイン処理
	gerR_IN32_RestartEvent	ERRCODE	R-IN32M4-CL3 イベント再開
	gerR_IN32_UpdateMIB	ERRCODE	MIB 情報更新
サイクリック伝送 (5.3.4 サイク リック伝送)	gerR_IN32_SetCyclicStop	ERRCODE	ユーザ・アプリケーション都合による サイクリック伝送停止
	gerR_IN32_ClearCyclicStop	ERRCODE	ユーザ・アプリケーション都合による サイクリック伝送停止を解除
	gerR_IN32_GetReceivedCyclicData	ERRCODE	サイクリック受信データ取得
	gerR_IN32_GetMasterNodeStatus	ERRCODE	マスタ局状態取得
	gerR_IN32_SetMyStatus	ERRCODE	MyStatus 送信データ設定
	gerR_IN32_SetSendCyclicData	ERRCODE	サイクリック送信データ設定
自局状態設定 (5.3.5 自局状態 設定)	gerR_IN32_SetNodeStatus	ERRCODE	自局状態設定
	gerR_IN32_ForceStop	ERRCODE	R-IN32M4-CL3 自局エラー設定
自局状態取得 (5.3.6 自局状態 取得)	gerR_IN32_GetNodeAndNetworkNumber	ERRCODE	局番およびネットワーク No. 取得
	gerR_IN32_GetCurrentCyclicSize	ERRCODE	マスタ局から指定されたサイクリック 伝送サイズ取得
	gerR_IN32_GetCommunicationStatus	ERRCODE	データリンク状態取得
	gerR_IN32_GetPortStatus	ERRCODE	PHY リンク状態取得
	gerR_IN32_GetCyclicStatus	ERRCODE	サイクリック伝送状態取得
	gerR_IN32_GetMIB	ERRCODE	MIB 情報取得
LED 制御 (5.3.7 LED 制御)	gerR_IN32_ClearMIB	ERRCODE	MIB 情報クリア
	gerR_IN32_SetERRLED	ERRCODE	LED 点灯制御 (ERR)
	gerR_IN32_SetUSER1LED	ERRCODE	LED 点灯制御 (User LED 1)
	gerR_IN32_SetUSER2LED	ERRCODE	LED 点灯制御 (User LED 2)
	gerR_IN32_SetRUNLED	ERRCODE	LED 点灯制御 (RUN)
	gerR_IN32_DisableLED	ERRCODE	LED 点灯機能無効
	gerR_IN32_EnableLED	ERRCODE	LED 点灯機能有効
	gerR_IN32_UpdateLedStatus	ERRCODE	通信状態表示 LED の更新

表 5.5 R-IN32M4-CL3 ドライバ・インタフェース関数一覧

(2/3)

機能分類 (参照項)	関数名	関数型	概要
ネットワーク時刻 (5.3.8 ネットワーク時刻)	gerR_IN32_GetNetworkTime	ERRCODE	ネットワーク時刻 (シリアル値) 取得
	gerR_IN32_SetNetworkTime	ERRCODE	ネットワーク時刻 (シリアル値) 設定
	gerR_IN32_NetworkTimeToDate	ERRCODE	ネットワーク時刻 (シリアル値) →時計情報変換
	gerR_IN32_DateToNetworkTime	ERRCODE	時計情報 →ネットワーク時刻 (シリアル値) 変換
MDIO アクセス (5.3.9 MDIO アクセス)	gerR_IN32_EnableMACIPAccess	ERRCODE	MAC IP アクセス許可
	gerR_IN32_DisableMACIPAccess	ERRCODE	MAC IP アクセス禁止
	gerR_IN32_WritePHY	ERRCODE	PHY 内部レジスタライト
	gerR_IN32_ReadPHY	ERRCODE	PHY 内部レジスタリード
	gerR_IN32_CheckPHY	ERRCODE	PHY チェック処理
トランジェント受信 (5.3.10 トランジェント受信処理)	gerR_IN32_MainReceiveTransient1	ERRCODE	トランジェント受信メイン処理 1
	gerR_IN32_MainReceiveTransient2	ERRCODE	トランジェント受信メイン処理 2
	gerR_IN32_EnableReceiveTransient	ERRCODE	ユーザ理由によるトランジェント 受信許可設定
	gblR_IN32_GetReceiveTransientStatus	BOOL	ユーザ理由によるトランジェント 受信許可設定状態取得
	gerR_IN32_SetMACAddressTableData	ERRCODE	ノード情報配信データ (MAC アドレス表) 設定
トランジェント 要求受信対応 (5.3.11 トランジェント要求受信対応)	gerR_IN32_ReceivedMACAddressData	ERRCODE	ノード情報配信フレーム受信対応処理
	gerR_IN32_ReceivedStatisticInfoRequest	ERRCODE	統計情報取得要求フレーム受信対応処理
	gerR_IN32_ReceivedUnitInfoRequest	ERRCODE	ノード詳細情報取得要求フレーム受信 対応処理
	gerR_IN32_ReceivedOptionInfoRequest	ERRCODE	オプション情報取得要求フレーム受信 対応処理
	gerR_IN32_ReceivedSelectInfoRequest	ERRCODE	選択局情報取得要求フレーム受信対応処理
	gerR_IN32_ReceivedContactTestRequest	ERRCODE	交信テスト要求フレーム受信対応処理
	gerR_IN32_ReceivedCableTestRequest	ERRCODE	ケーブルテスト要求フレーム受信対応処理
	gerR_IN32_ReceiveRemoteResetRequest	ERRCODE	SLMP リモートリセット要求フレーム 受信対応処理
	gR_IN32_SetSimpError_Response	ULONG	SLMP エラー応答フレーム作成処理
	gerR_IN32_ErrCheckReqFieldNetworkReceived	ERRCODE	CC-Link IE Field 固有要求受信フレーム 異常確認処理
	gusR_IN32_ErrCheckReqSimpReceived	USHORT	SLMP 要求受信フレーム異常確認処理

表 5.5 R-IN32M4-CL3 ドライバ・インタフェース関数一覧

(3/3)

機能分類 (参照項)	関数名	関数型	概要
トランジェント送信 フレームヘッダ作成 (5.3.12 トランジェ ント送信フレームヘッ ダ作成)	gerR_IN32_SetEtherCcieHeader	ERRCODE	MAC+CCIE ヘッダ作成処理
	gerR_IN32_SetTransient1Header	ERRCODE	Transient1 ヘッダ作成処理
	gerR_IN32_SetRequestSimpHeader	ERRCODE	SLMP 要求ヘッダ作成処理
	gerR_IN32_SetResponseSimpHeader	ERRCODE	SLMP 応答ヘッダ作成処理
トランジェント送信 (5.3.13 トランジェ ント送信処理)	gerR_IN32_GetUnitInformation	ERRCODE	ユニット情報取得
	gusR_IN32_GetNodeID	USHORT	ノード ID 取得
	gerR_IN32_GetMulticastMACAddress	ERRCODE	マルチキャスト MAC アドレス取得
	gerR_IN32_GetUnicastMACAddress	ERRCODE	ユニキャスト MAC アドレス取得
	gerR_IN32_GetSendTransientBuffer	ERRCODE	トランジェント送信バッファ取得
	gerR_IN32_RequestSendingTransient	ERRCODE	トランジェント送信要求
H/W テスト (5.3.14 H/W テスト)	gerR_IN32_MainSendTransient	ERRCODE	トランジェント送信メイン処理
	gerR_IN32_IEEE8023abTest	ERRCODE	IEEE802.3ab コンプライアンス・テスト
	gerR_IN32_InitializeLoopBackTest	ERRCODE	内部折り返し/外部折り返し通信 テスト初期化
	gerR_IN32_ExternalLoopBackTest	ERRCODE	外部折り返し通信テスト

5.3 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。

5.3.1 初期設定

(1) gulR_IN32_GetResetStatus

機能	リセット状態取得			
呼び出し形式	ULONG gulR_IN32_GetResetStatus (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_RESET_PWRON(1) : パワー・オン・リセット R_IN32_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。 本関数の呼び出しは、gerR_IN32_Initialize を呼び出す前に実施してください。			

(2) gerR_IN32_Initialize

機能	R-IN32M4-CL3 の初期化			
呼び出し形式	ERRCODE gerR_IN32_Initialize (const UCHAR* puchMACAddr, const R_IN32_UNITINFO_T *pstUnitInfo, const R_IN32_UNITINIT_T *pstUnitInit)			
引数	型名	変数名	内容	入出力
	const UCHAR	*puchMACAddr	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。 puchMACAddr[0] : 12h puchMACAddr[1] : 34h puchMACAddr[2] : 56h puchMACAddr[3] : 78h puchMACAddr[4] : 90h puchMACAddr[5] : ABh	入力
	const R_IN32_UNITINFO_T	*pstUnitInfo	R-IN32M4-CL3 ユニット情報 詳細は「表 5.6R_IN32_UNITINFO_T 一覧」を参照してください。	入力
	const R_IN32_UNITINIT_T	*pstUnitInit	R-IN32M4-CL3 初期設定 詳細は「表 5.7R_IN32_UNITINIT_T 一覧」を参照してください。	入力
戻り値	R_IN32_OK : 正常終了			
説明	R-IN32M4-CL3 の初期化および PHY のリセットを行います。 本関数を呼び出すと R-IN32M4-CL3 内部 WDT が無効になります。 R-IN32M4-CL3 内部 WDT を使用する場合は、gerR_IN32_EnableWDT を呼び出してください。			

注意. R-IN32M4-CL3にFatalエラーが発生した場合、本関数よりユーザ作成の
gR_IN32_CallbackFatalErrorを起動しますので、エラー・コードに対応した
エラー処理を実行してください。

gerR_IN32_Initialize の引数「R_IN32_UNITINFO_T」の構成を以下に示します。

表 5.6 R_IN32_UNITINFO_T 一覧

(1/3)

No.	メンバ		概要	設定内容
1	ULONG	ulMaxRySize	RY サイズ (バイト)	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。 インテリジェントデバイス局の場合、256 バイトが最大です。 リモートデバイス局の場合、16 バイトが最大です。
2	ULONG	ulMaxRWwSize	RWw サイズ (ワード)	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。 インテリジェントデバイス局の場合、1024 ワードが最大です。 リモートデバイス局の場合、64 ワードが最大です。
3	ULONG	ulMaxRxSize	RX サイズ (バイト)	自局が通信可能な RX のサイズ (ワード) を 1 バイト単位で設定します。 インテリジェントデバイス局の場合、256 バイトが最大です。 リモートデバイス局の場合、16 バイトが最大です。
4	ULONG	ulMaxRWrSize	RWr サイズ (ワード)	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。 インテリジェントデバイス局の場合、1024 ワードが最大です。 リモートデバイス局の場合、64 ワードが最大です。
5	ULONG	ulMyStationPortTotalNumber	自局ポート数	自局が保有する物理的な CC-Link IE フィールド・ネットワーク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。
6	ULONG	ulTokenHoldTime	トークン 保持時間	トークン巡回開始後に自局がトークンを保持する時間の最大値を us 単位で設定します。23us を設定してください。
7	ULONG	ulIOType	ノード情報 (I/O タイプ)	I/O タイプを設定します。 00b は裏表混在、01b は入力、10b は出力、11b は混在を表します。 裏表混在は、入力と出力が混在している場合で、かつ入力と出力が同じアドレスを使用している場合です。 混在は、入力と出力が混在している場合で、かつ入力と出力が異なるアドレスを使用している場合です。
8	ULONG	ulNetVersion	ネットワークの ファームウェア バージョン	ネットワークのファームウェアバージョンを設定します。
9	ULONG	ulNetModelType	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。 ^注
10	ULONG	ulNetUnitModelCode	ネットワークの 型名コード	ネットワークの型名コードを設定します。 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・コード内で一意になるように管理してください。

注: 「2.2 ベンダ・コードの取得と機種タイプの選択」を参照してください。

表 5.6 R_IN32_UNITINFO_T 一覧

(2/3)

No.	メンバ	概要	設定内容
11	ULONG	ulNetVendorCode	ネットワークのベンダ・コード CC-Link 協会入会時に取得したベンダ・コード (vendorCode) を BCD で設定します。 (ベンダ・コードが 5678 の場合、5678h を設定します。) 注1
12	UCHAR	auchNetUnitModelName[20]	ネットワークの型名名称 ネットワークの型名名称を設定します。 (20 バイトの文字列 (ASCII コード)) 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・コード内で一意になるように管理してください。
13	UCHAR	auchNetVendorName[32]	ネットワークのベンダ名称 ネットワークのベンダ名称を設定します。 (32 バイトの文字列 (ASCII コード)) ベンダ名称は、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。
14	USHORT	usHwVersion	ネットワークのハードウェアバージョン ネットワークのハードウェアバージョンを設定します。
15	USHORT	usDeviceVersion	機器バージョン 機器バージョン (Version) を設定します。 機器バージョン (Version) 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。 注2
16	BOOL	blInformationFlag	コントローラ情報有無フラグ 本表の No.14~No.21 の有効/無効を設定します。 R_IN32_FALSE は無効、R_IN32_TRUE は有効を表します。 通信機能しか持たない場合 (例: QJ71GF11-T2 など)、コントローラ情報有無フラグを無効にします。
17	ULONG	ulCtrlVersion	コントローラのファームウェアバージョン コントローラのファームウェアバージョンを設定します。
18	ULONG	ulCtrlModelType	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。 注1
19	ULONG	ulCtrlUnitModelCode	コントローラの型名コード コントローラの型名コードを設定します。 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・コード内で一意になるように管理してください。

注 1. 「2.2 ベンダ・コードの取得と機種タイプの選択」を参照してください。

2. CSP+における機器バージョンについて、以下に補足します。

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。

備考. 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、センサ部分、温調部分など) を指します。

表 5.6 R_IN32_UNITINFO_T 一覧

(3/3)

No.	メンバ		概要	設定内容
20	ULONG	ulCtrlVendorCode	コントローラのベンダ・コード	CC-Link 協会入会時に取得したベンダ・コード (vendorCode) を BCD で設定します。(ベンダ・コードが 5678 の場合、5678h を設定します。) 注
21	UCHAR	auchCtrlUnitModelName[20]	コントローラの型名名称	コントローラの型名名称を設定します。(20 バイトの文字列 (ASCII コード)) 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・コード内で一意になるように管理してください。
22	UCHAR	auchCtrlVendorName[32]	コントローラのベンダ名称	コントローラのベンダ名称を設定します。(32 バイトの文字列 (ASCII コード)) ベンダ名称は、ユーザが任意に定義してください。
23	ULONG	ulVendorInformation	コントローラのベンダ機器固有情報	コントローラのベンダ機器固有情報を設定します。 ベンダ機器固有情報は、ユーザが任意に定義してください。

注. 「2.2 ベンダ・コードの取得と機種タイプの選択」を参照してください。

備考 1. 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、センサ部分、温調部分など) を指します。

2. 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、スレーブ局の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。

【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。

gerR_IN32_Initialize の引数「R_IN32_UNITINIT_T」の構成を以下に示します。

表 5.7 R_IN32_UNITINIT_T 一覧

(1/3)

No.	メンバ		概要	設定内容
1	BOOL	bINMIUse	NMI 割込み使用	R-IN32M4-CL3 の内部 WDT 機能を使用する場合に R_IN32_TRUE、使用しない場合 R_IN32_FALSE を設定してください。 R_IN32_TRUE に設定すると、R-IN32M4-CL3 の内部 WDT がオーバフローしたときに CCI_NMIZ が“Low”になります。
2	BOOL	bInterruptUse	CPU 割込み機能使用	R-IN32M4-CL3 の CPU 割込み機能を使用する場合に R_IN32_TRUE、使用しない場合 R_IN32_FALSE を設定してください。 R_IN32_TRUE に設定すると、R-IN32M4-CL3 の割込みが発生したとき、CCI_INTZ が“Low”になります。
3	BOOL	bFailedProcess1	故障処理設定 1	R_IN32_TRUE を設定してください。 以下のいずれかの信号が真となったとき、R-IN32M4-CL3 はバイパスモードとなります。（フレームを送受信しません。受信したフレームは、他ポートへそのままフォワードします。） ①WDTIL 信号が真（Low） ②R-IN32M4-CL3 内部 WDT の時限がタイムアウトしたとき バイパスモードを解除する場合、パワー・オン・リセットまたはシステム・リセットを行う必要があります。
4	BOOL	bFailedProcess2	故障処理設定 2	R_IN32_TRUE を設定してください。 自局エラーを設定（gerR_IN32_ForceStop を起動）すると、R-IN32M4-CL3 はバイパスモードとなります。 （フレームを送受信しません。受信したフレームは、他ポートへそのままフォワードします。） 自局エラーを解除する場合は、パワー・オン・リセットまたはシステム・リセットを行う必要があります。 gerR_IN32_ForceStop については、「5.3.5 自局状態設定」を参照してください。
5	ULONG	ulNodeType	ノード種別	自局の局種別を設定します。 インテリジェントデバイス局の場合、(33h) を設定してください。 リモートデバイス局の場合、(34h) を設定してください。
6	BOOL	bTransientReceiveEnable	トランジェント受信機能	R_IN32_TRUE を設定してください。 トランジェント受信機能の有無を設定します。 R_IN32_TRUE は機能あり、R_IN32_FALSE は機能なしを表します。

表 5.7 R_IN32_UNITINIT_T 一覧

(2/3)

No.	メンバ		概要	設定内容
7	BOOL	blMACAddress TableRequest	ノード情報配信 の要求初期値	【インテリジェントデバイス局の場合】 トランジェント伝送のクライアント機能を実装する場合 R_IN32_TRUE、実装しない場合、R_IN32_FALSE を設定してください。 【リモートデバイス局の場合】 トランジェント伝送のクライアント機能がないため、R_IN32_FALSE を設定してください。 【ノード情報について】 ノード情報は、他局の局番と MAC アドレスの対応を示した情報です。R_IN32_TRUE に設定すると、マスタ局からノード情報がマルチキャスト配信されます。R_IN32_FALSE に設定した場合、受信したノード情報配信要求フレームをユーザ・プログラムにて破棄してください。 【トランジェント・フレームを送信するとき】 能動的に送信する場合（クライアント時）、ノード情報を利用します。送信元に応答を返す場合（サーバ時）、送信元 MAC アドレスを利用して返信できますので、ノード情報を利用しません
8	ULONG	ulRunStatus	アプリケーション 詳細動作状態 初期値	MyStatus フレームの nodeStatus 内のアプリケーション詳細動作状態の初期値を設定します。 R_IN32_RUNSTS_UNSUPPORTED (0000h): アプリケーション詳細動作状態通知未サポート R_IN32_RUNSTS_STOP (0001h): アプリケーションが停止中 R_IN32_RUNSTS_RUN (0002h): アプリケーションが動作中 R_IN32_RUNSTS_NOTEXIST (0003h): アプリケーション実体が存在しない
9	ULONG	ulErrorStatus	アプリケーション 詳細エラー状 態初期値	MyStatus フレーム nodeStatus フィールドのアプリケーション詳細エラー状態の初期値を設定します。 R_IN32_ERRSTS_NONE (0000h): エラーなし R_IN32_ERRSTS_WARNING (0001h): 軽度異常 R_IN32_ERRSTS_ERROR (0002h): 中度異常 R_IN32_ERRSTS_FATALERROR (0003h): 重度異常

表 5.7 R_IN32_UNITINIT_T 一覧

(3/3)

No.	メンバ		概要	設定内容
10	ULONG	ulUserInformation	ペンダ固有のノード情報初期値	MyStatus フレームの vendorSpfNodeInfo の初期値を設定します。
11	ULONG	ulOptionSupport	オプション有無初期値	オプションへ対応する場合、R_IN32_TRUE（推奨）、対応しない場合 R_IN32_FALSE を設定してください。 【オプションについて】 オプションは、SLMP フレームの送受信機能および CC-Link IE フィールド・ネットワーク診断機能を含む、CC-Link IE フィールド・ネットワークの拡張機能です。
12	ULONG	ulSlmpSupport	SLMP サポートビット初期値	SLMP フレームを送受信する場合 R_IN32_TRUE（推奨）、送受信しない場合 R_IN32_FALSE を設定してください。 ^{注1}
13	ULONG	ulSlmpDiagnosisSupport	診断機能対応有無初期値	CC-Link IE フィールド・ネットワーク診断機能に対応する場合 R_IN32_TRUE（推奨）、対応しない場合 R_IN32_FALSE を設定してください。 ^{注2}
14	R_IN32_PHY_SETTING_T	stPHYSetting[2]	PHY 設定初期値	ポートごとに、MDI/MDI-X およびマスタ/スレーブの設定を行います。 「表 5.8R_IN32_PHY_SETTING_T 一覧」を参照してください。

注 1. SLMP フレーム送受信するために、本設定と共に、「オプション有無初期値」を R_IN32_TRUE に設定してください。

2. CC-Link IE フィールド・ネットワーク診断機能に対応するために、本設定と共に、「オプション有無初期値」および「SLMP サポートビット初期値」を R_IN32_TRUE に設定してください。

表 5.8 R_IN32_PHY_SETTING_T 一覧

No.	メンバ		概要	設定内容
1	ULONG	uIMDI	MDI 設定	ポートごとに MDI/MDI-X を指定します。 ファストリンクアップ機能を使う場合、 PORT 1 は R_IN32_MDI_FORCED_MDI、 PORT 2 は R_IN32_MDI_FORCED_MDIX に設定してください。 使わない場合 R_IN32_MDI_AUTO を設定してください。 R_IN32_MDI_AUTO (0000h) : オート R_IN32_MDI_FORCED_MDI (0001h) : 強制 MDI R_IN32_MDI_FORCED_MDIX (0002h) : 強制 MDI-X
2	ULONG	uIClk	1000BASE-T Clock 設定	ポートごとに Master/Slave を指定します。 ファストリンクアップ機能を使う場合、 PORT 1 を R_IN32_CLOCK_MASTER に、 PORT 2 を R_IN32_CLOCK_SLAVE に設定してください。 使わない場合 R_IN32_CLOCK_AUTO を設定してください。 R_IN32_CLOCK_AUTO (0000h) : オート R_IN32_CLOCK_MASTER (0001h) : 強制マスタ R_IN32_CLOCK_SLAVE (0002h) : 強制スレーブ

(3) gerR_IN32_SetNodeAndNetworkNumber

機能	局番およびネットワーク No. 設定			
呼び出し形式	ERRCODE gerR_IN32_SetNodeAndNetworkNumber (UCHAR uchNetworkNumber, USHORT usNodeNumber)			
引数	型名	変数名	内容	入出力
	UCHAR	uchNetworkNumber	ネットワーク No. (値域 : 1~239)	入力
	USHORT	usNodeNumber	局番 (値域 : 1~120)	入力
戻り値	R_IN32_OK : 正常終了 R_IN32_ERR : 異常終了 (ライブラリ内での状態異常) R_IN32_ERR_OUTOFRANGE : 局番範囲外またはネットワーク No. 範囲外の場合			
説明	R-IN32M4-CL3 に局番およびネットワーク No. を設定します。 戻り値が R_IN32_ERR_OUTOFRANGE の場合、局番とネットワーク No. は設定されませんので、エラー処理を呼び出し元の関数に追加してください。			

注意. 本関数の呼び出しは、iUserInitialization (「4.2.2 初期化処理」参照) の後、iUserStart (「4.2.3 通信開始処理」参照) で gerR_IN32_Start を呼び出す前に行う必要があります。前述の処理を実行する前に本関数を呼び出すと R_IN32_ERR : 異常終了 (ライブラリ内での状態異常) となります。

(4) gerR_IN32_Start

機能	R-IN32M4-CL3 通信開始			
呼び出し形式	ERRCODE gerR_IN32_Start(VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
	R_IN32_ERR : 異常終了			
説明	R-IN32M4-CL3 に通信開始を指示します。			

注意. R-IN32M4-CL3 に Fatal エラーが発生した場合、本関数よりユーザ作成の
gR_IN32_CallbackFatalError を呼び出しますので、エラーコードに対応したエラー処理を
実行してください。

5.3.2 ウォッチドッグタイマ

(1) gerR_IN32_ResetWDT

機能	R-IN32M4-CL3 内部 WDT リセット			
呼び出し形式	ULONG gerR_IN32_ResetWDT (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	R-IN32M4-CL3 内部 WDT をリセットします。			

注意. 本関数呼び出し後、本項内の関数を呼び出す場合、1.032us 以上あけてください。

(2) gerR_IN32_DisableWDT

機能	R-IN32M4-CL3 内部 WDT 無効			
呼び出し形式	ULONG gerR_IN32_DisableWDT (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	R-IN32M4-CL3 内部 WDT を無効にします。			

注意. 本関数呼び出し後、本項内の関数を呼び出す場合、1.032us 以上あけてください。

R-IN32M4-CL3 は、リセット直後は、R-IN32M4-CL3 内部 WDT は有効になっています。

(R-IN32M4-CL3 内部 WDT 時限設定の初期値は、3.2 秒)

gerR_IN32_Initialize 起動にて R-IN32M4-CL3 内部 WDT が無効になりますが、

gerR_IN32_Initialize 処理開始まで時間がかかる場合は以下のいずれかを実施してください。

- ・ 本関数を呼び出し、R-IN32M4-CL3 内部 WDT を無効にします。
- ・ gerR_IN32_ResetWDT を呼び出し、R-IN32M4-CL3 内部 WDT をリセットします。
(R-IN32M4-CL3 内部 WDT がタイムアウトしないようにしてください。)

(3) gerR_IN32_EnableWDT

機能	R-IN32M4-CL3 内部 WDT 有効			
呼び出し形式	ULONG gerR_IN32_EnableWDT (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	R-IN32M4-CL3 内部 WDT を有効にします。			

注意. 本関数呼び出し後、本項内の関数を呼び出す場合、1.032us 以上あけてください。

gerR_IN32_Initialize にて、R-IN32M4-CL3 内部 WDT が無効となりますので R-IN32M4-CL3 内部 WDT を使用する場合は、本関数を呼び出してください。

(4) gerR_IN32_SetWDT

機能	R-IN32M4-CL3 内部 WDT 時限設定			
呼び出し形式	ULONG gerR_IN32_SetWDT (USHORT usWDTCOUNT)			
引数	型名	変数名	内容	入出力
	USHORT	usWDTCOUNT	R-IN32M4-CL3 内部 WDT 時限設定 0000h : 100ms 0001h : 200ms 0002h : 300ms ... 001Fh : 3.2 秒	入力
戻り値	R_IN32_OK : 正常終了			
説明	R-IN32M4-CL3 内部 WDT の時限を設定します。			

注意. 本関数呼び出し後、本項内の関数を呼び出す場合、1.032us 以上あけてください。

R-IN32M4-CL3 内部 WDT 実行中 (gerR_IN32_EnableWDT 呼び出し後) に本関数で R-IN32M4-CL3 内部 WDT 時限設定を変更した場合、gerR_IN32_ResetWDT を呼び出したときに変更した時限設定で R-IN32M4-CL3 内部 WDT が動作します。(gerR_IN32_ResetWDT を呼び出すまでは変更前の R-IN32M4-CL3 内部 WDT 時限設定で動作します。)

5.3.3 イベント

(1) gerR_IN32_GetEvent

機能	R-IN32M4-CL3 イベント検出			
呼び出し形式	ERRCODE gerR_IN32_GetEvent (R_IN32_EVTPRM_INTERRUPT_T *pstEvent)			
引数	型名	変数名	内容	入出力
	R_IN32_EVTPRM_INTERRUPT_T	*pstEvent	割込み要因 詳細は「表 5.9 R_IN32_EVTPRM_INTERRUPT_T 一覧」を参照してください。	出力
戻り値	R_IN32_OK : 正常終了			
説明	R-IN32M4-CL3 のイベントを検出します。			

R_IN32_EVTPRM_INTERRUPT_T の構成を以下に示します。

表 5.9 R_IN32_EVTPRM_INTERRUPT_T 一覧

No.	メンバ		概要
1	ULONG	b1ZCommConnect	通信参加
2	ULONG	b1ZCommDisconnect	通信解列
3	ULONG	b1ZCommConnectToDisconnect	通信参加→通信解列
4	ULONG	b1ZCommDisconnectToConnect	通信解列→通信参加
5	ULONG	b1ZChangeStNoNetNo	局番、ネットワーク No.変更
6	ULONG	b1ZChangeActCommand	動作コマンド変更
7	ULONG	b1ZPrmFrmRcv_OK	Parameter フレーム受信
8	ULONG	b1ZReserve1	予約
9	ULONG	b1ZPrmChkFrmRcv_OK	ParamCheck フレーム受信（パラメーター一致時）
10	ULONG	b3ZReserve2	予約
11	ULONG	b1ZRecvNonCyclic	トランジェント受信
12	ULONG	b1ZSendFinNonCyclic	トランジェント送信完了
13	ULONG	b7ZReserve3	予約
14	ULONG	b1ZMasterWatchTimeout	マスタウォッチタイマ タイムアウト発生
15	ULONG	bAZReserve4	予約

(2) gerR_IN32_Main

機能	R-IN32M4-CL3 イベント検出メイン処理			
呼び出し形式	ERRCODE gerR_IN32_Main (const R_IN32_EVTPRM_INTERRUPT_T *pstEvent)			
引数	型名	変数名	内容	入出力
	const R_IN32_EVTPRM_INTERRUPT_T	*pstEvent	割込み要因	入力
戻り値	R_IN32_OK : 正常終了 R_IN32_ERR : 異常終了（ライブラリ内での状態異常）			
説明	R-IN32M4-CL3 イベントに対する処理を行います。			

注意. 本関数は、iUserInitialization（「4.2.2 初期化処理」参照）の後、iUserStart（「4.2.3 通信開始処理」参照）の後で呼び出す必要があります。前述の処理を実行する前に本関数を呼び出すと R_IN32_ERR : 異常終了（ライブラリ内での状態異常）となります。

(3) gerR_IN32_RestartEvent

機能	R-IN32M4-CL3 イベント再開			
呼び出し形式	ERRCODE gerR_IN32_RestartEvent (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	gerR_IN32_GetEvent で停止したイベントを再開します。			

(4) gerR_IN32_UpdateMIB

機能	MIB 情報更新			
呼び出し形式	ERRCODE gerR_IN32_UpdateMIB (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了 R_IN32_ERR : 異常終了 (MIB 情報収集異常 (ライブラリ内での状態異常/不一致)) R_IN32_ERR_OTHER : 異常終了 (MIB 情報収集異常 (ライブラリ内ドライバで異常発生))			
説明	MIB 情報を更新します。			

注意. 本関数の戻り値が R_IN32_OK 以外の場合、ユーザ作成の gR_IN32_CallbackFatalError を呼び出しますので、エラー・コードに対応したエラー処理を実行してください。

5.3.4 サイクリック伝送

(1) gerR_IN32_SetCyclicStop

機能	ユーザ・アプリケーション都合によるサイクリック伝送停止			
呼び出し形式	ERRCODE gerR_IN32_SetCyclicStop (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	機器側都合によりサイクリック伝送を停止します。 解除する場合、gerR_IN32_ClearCyclicStop を呼び出してください。			

(2) gerR_IN32_ClearCyclicStop

機能	ユーザ・アプリケーション都合によるサイクリック伝送停止を解除			
呼び出し形式	ERRCODE gerR_IN32_ClearCyclicStop (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	gerR_IN32_SetCyclicStop の呼び出しによるサイクリック伝送停止を解除します。			

(3) gerR_IN32_GetReceivedCyclicData

機能	サイクリック受信データ取得			
呼び出し形式	ERRCODE gerR_IN32_GetReceivedCyclicData (VOID *pRyDst, VOID *pRWwDst, BOOL blEnable)			
引数	型名	変数名	内容	入出力
	VOID	*pRyDst	RY 領域	出力
	VOID	*pRWwDst	RWw 領域 ^{注1}	出力
	BOOL	blEnable	コピー有効/無効 R_IN32_TRUE : 有効 R_IN32_FALSE : 無効	入力
戻り値	R_IN32_OK : 正常終了 (受信データあり) R_IN32_ERR : 異常終了 (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を pRyDst および pRWwDst の示すアドレスに格納します。ただし、blEnable が R_IN32_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_IN32_ERR ^{注2} となります。			

注 1. RWw 領域の先頭アドレスは、4 バイト単位 (0 または 4 の倍数) としてください。

2. R_IN32_ERR : 異常終了 (受信データなし)

前回呼び出した gerR_IN32_GetReceivedCyclicData から、
今回呼び出す gerR_IN32_GetReceivedCyclicData までの間で
サイクリック受信していない場合に発生しますが異常ではありません。

(4) gerR_IN32_GetMasterNodeStatus

機能	マスタ局状態取得			
呼び出し形式	ERRCODE gerR_IN32_GetMasterNodeStatus (BOOL *pblRunSts, BOOL *pblErrSts, ULONG *pulErrCode)			
引数	型名	変数名	内容	入出力
	BOOL	*pblRunSts	マスタ局のアプリケーション動作状態 R_IN32_TRUE : 実行中 R_IN32_FALSE : 停止中	出力
	BOOL	*pblErrSts	マスタ局のアプリケーションエラー状態 R_IN32_TRUE : エラー発生中 R_IN32_FALSE : エラーなし	出力
	ULONG	*pulErrCode	マスタ局のエラー・コード	出力
戻り値	R_IN32_OK : 正常終了 (マスタ局からの MyStatus フレーム受信あり) R_IN32_ERR : 異常終了 (データリンク未実施 (解列中) のため、 マスタ局からの MyStatus フレームを受信していない)			
説明	マスタ局から受信した MyStatus フレームよりマスタ局の状態を取得します。 データリンク未実施 (解列中) のためマスタ局からの MyStatus 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_IN32_FALSE pblErrSts : R_IN32_FALSE pulErrCode : 0			

(5) gerR_IN32_SetMyStatus

機能	MyStatus 送信データ設定			
呼び出し形式	ERRCODE gerR_IN32_SetMyStatus (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	gerR_IN32_SetNodeStatus で設定した自局状態を R-IN32M4-CL3 に設定します。			

(6) gerR_IN32_SetSendCyclicData

機能	サイクリック送信データ設定			
呼び出し形式	ERRCODE gerR_IN32_SetSendCyclicData (const VOID *pRxSrc, const VOID *pRWwSrc, BOOL blEnable)			
引数	型名	変数名	内容	入出力
	const VOID	*pRxSrc	RX 領域	入力
	const VOID	*pRWwSrc	RWw 領域 ^注	入力
	BOOL	blEnable	更新 有効/無効 R_IN32_TRUE : 有効 R_IN32_FALSE : 無効	入力
戻り値	R_IN32_OK : 正常終了			
説明	pRxSrc および pRWwSrc 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。ただし、blEnable が R_IN32_FALSE の場合、サイクリック送信データの設定はしません。（戻り値は、R_IN32_ERR となります。）			

注. RWw 領域の先頭アドレスは、4 バイト単位（0 または 4 の倍数）としてください。

5.3.5 自局状態設定

(1) gerR_IN32_SetNodeStatus

機能	自局状態設定			
呼び出し形式	ERRCODE gerR_IN32_SetNodeStatus (ULONG ulRunSts, ULONG ulErrSts, ULONG ulErrCode, ULONG ulUserInformation)			
引数	型名	変数名	内容	入出力
	ULONG	ulRunSts	アプリケーション詳細動作状態 R_IN32_RUNSTS_UNSUPPORTED(0) : アプリケーション詳細動作状態通知未サポート R_IN32_RUNSTS_STOP(1) : アプリケーションが停止中 R_IN32_RUNSTS_RUN(2) : アプリケーションが動作中 R_IN32_RUNSTS_NOTEXIST(3) : アプリケーション実体が存在しない	入力
	ULONG	ulErrSts	アプリケーション詳細エラー状態 R_IN32_ERRSTS_NONE(0) : エラーなし R_IN32_ERRSTS_WARNING(1) : 軽度異常 R_IN32_ERRSTS_ERROR(2) : 中度異常 R_IN32_ERRSTS_FATALERROR(3) : 重度異常	入力
	ULONG	ulErrCode	エラー・コード	入力
	ULONG	ulUserInformation	ベンダ固有ノード情報	入力
戻り値	R_IN32_OK : 正常終了			
説明	MyStatus フレームにて送信する情報として、自局状態を設定します。			

(2) gerR_IN32_ForceStop

機能	自局エラー設定			
呼び出し形式	ERRCODE gerR_IN32_ForceStop (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	R-IN32M4-CL3 を自局エラーに設定します。自局エラーを解除する場合は、パワー・オン・リセットまたはシステム・リセットを行う必要があります。			

5.3.6 自局状態取得

(1) gerR_IN32_GetNodeAndNetworkNumber

機能	局番およびネットワーク No.取得			
呼び出し形式	ERRCODE gerR_IN32_GetNodeAndNetworkNumber (USHORT *pusNodeNumber, UCHAR *puchNetworkNumber)			
引数	型名	変数名	内容	入出力
	USHORT	*pusNodeNumber	局番	出力
	UCHAR	*puchNetworkNumber	ネットワーク No.	出力
戻り値	R_IN32_OK : 正常終了			
説明	局番およびネットワーク No.を取得します。			

(2) gerR_IN32_GetCurrentCyclicSize

機能	マスタ局から指定されたサイクリック伝送サイズ取得			
呼び出し形式	ERRCODE gerR_IN32_GetCurrentCyclicSize (R_IN32_CYCLIC_SIZE_T *pstCyclicSize)			
引数	型名	変数名	内容	入出力
	R_IN32_CYCLIC_SIZE_T	*pstCyclicSize	サイクリック伝送サイズ ulRySize : RY サイズ (バイト) ulRWwSize : RWw サイズ (バイト) ulRxSize : RX サイズ (バイト) ulRWrSize : RWr サイズ (バイト)	出力
戻り値	R_IN32_OK : 正常終了			
説明	Parameter フレームにてマスタ局から設定されたサイクリック伝送サイズを取得します。 gerR_IN32_GetReceivedCyclicData、および gerR_IN32_SetSendCyclicData 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。			

(3) gerR_IN32_GetCommumicationStatus

機能	データリンク状態取得			
呼び出し形式	ERRCODE gerR_IN32_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_IN32_COMMSTS_CYC_DLINK(2) : データリンク中（サイクリック伝送中） R_IN32_COMMSTS_TOKEN_PASS(1) : データリンク中（サイクリック伝送停止中） R_IN32_COMMSTS_DISCONNECT(0) : データリンク未実施（解列中）	出力
戻り値	R_IN32_OK : 正常終了			
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・点滅・消灯させてください。 R_IN32_COMMSTS_CYC_DLINK(2) : LED 点灯 R_IN32_COMMSTS_TOKEN_PASS(1) : LED 点滅 R_IN32_COMMSTS_DISCONNECT(0) : LED 消灯			

注意. D LINK LED の点灯/消灯の制御は、UserUpdateLed（「4.2.11 LED 更新処理」参照）を参照してください。

(4) gerR_IN32_GetPortStatus

機能	PHY リンク状態取得			
呼び出し形式	ERRCODE gerR_IN32_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_IN32_PORT1(0) : PORT1 R_IN32_PORT2(1) : PORT2	入力
	ULONG	*pulLinkStatus	リンク状態 R_IN32_LINKUP(1) : リンクアップ R_IN32_LINKDOWN(0) : リンクダウン	出力
	ULONG	*pulSpeed	速度 ^注 R_IN32_SPEED_1G(0) : 1Gbps R_IN32_SPEED_100M(1) : 100Mbps R_IN32_SPEED_10M(2) : 10Mbps	出力
	ULONG	*pulDuplex	全二重/半二重 ^注 R_IN32_DUPLEX_FULL(0) : 全二重 R_IN32_DUPLEX_HALF(1) : 半二重	出力
戻り値	R_IN32_OK : 正常終了			
説明	PHY リンク状態を取得します。（*pulLinkState が LinkDown のときは、don't care）			

注. 第 2 引数*pulLinkStatus が R_IN32_LINKUP(1)のときに有効です。R_IN32_LINKDOWN(0)のときは使用しないでください。

(5) gerR_IN32_GetCyclicStatus

機能	サイクリック伝送状態取得			
呼び出し形式	ULONG gerR_IN32_GetCyclicStatus (R_IN32_CYCLIC_STA_T *pstCyclicStatus)			
引数	型名	変数名	内容	入出力
	R_IN32_CYCLIC_STA_T	*pstCyclicStatus	サイクリック伝送状態 詳細は「表 5.10 R_IN32_CYCLIC_STA_T 一覧」を参照してください。	出力
戻り値	R_IN32_OK : 正常終了			
説明	サイクリック伝送状態を取得します。			

R_IN32_CYCLIC_STA_T の構成を以下に示します。

表 5.10 R_IN32_CYCLIC_STA_T 一覧

No.	メンバ		ビット	概要	内容
1	USHORT	b3ZComonParamkeep Cond	b2-0	サイクリック伝送 パラメータ保有状態	001b : パラメータ正常受信済み 010b : 未受信または ID 不一致 011b : 確認中 100b : パラメータ異常受信済み
2	USHORT	b1ZParamCheckCond	b3	サイクリック伝送 パラメータ確認状態	0b : 確認完了 1b : 確認中
3	USHORT	b1ZMyNodeNoRangeOut	b4	局番不正設定状態	0b : 範囲内 1b : 範囲外
4	USHORT	b1ZMyNodeReserve Setup	b5	予約局設定状態	0b : 予約局設定なし 1b : 予約局設定あり
5	USHORT	b1ZCyclicOpelInstruct Package	b6	サイクリック伝送実施 指示（一括）設定状態	0b : 動作設定 1b : 停止設定
6	USHORT	b1ZCyclicOpelInstruct Various	b7	サイクリック伝送実施 指示（個別）設定状態	0b : 動作設定 1b : 停止設定
7	USHORT	b1ZReserved1	b8	予約	—
8	USHORT	b1ZMyMpuAbnomal	b9	サイクリック伝送継続 不可異常状態	0b : 異常なし 1b : サイクリック伝送継続不可 異常有
9	USHORT	b1ZMyNodeNumber Duplicate	b10	局番重複状態	0b : 重複なし 1b : 重複あり
10	USHORT	b1ZReserved2	b11	予約	—
11	USHORT	b1ZNodeTypeWrong	b12	局種別不正・指定サイズ 不正状態	0b : 正常 1b : 不正
12	USHORT	b1ZReserved3	b13	予約	—
13	USHORT	b1ZDLinkState	b14	解列発生状態	0b : 解列なし (サイクリック伝送実施中 またはトークンパス実施中) 1b : 解列中
14	USHORT	b1ZCyclicState	b15	自己理由による停止状態	0b : 非停止、 1b : 上記以外の理由で サイクリック伝送停止中

(6) gerR_IN32_GetMIB

機能	MIB 情報取得			
呼び出し形式	ERRCODE gerR_IN32_GetMIB (R_IN32_MIB_T *pstMIB)			
引数	型名	変数名	内容	入出力
	R_IN32_MIB_T	*pstMIB	R-IN32M4-CL3 MIB 情報	出力
戻り値	R_IN32_OK : 正常終了			
説明	MIB 情報を取得します。			

(7) gerR_IN32_ClearMIB

機能	MIB 情報クリア			
呼び出し形式	ERRCODE gerR_IN32_ClearMIB (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	MIB 情報をクリアします。			

注意. MIB 情報は非公開情報ですので、エンド・ユーザに公開しないでください。

5.3.7 LED 制御

(1) gerR_IN32_SetERRLED

機能	LED 点灯制御 (ERR)			
呼び出し形式	ERRCODE gerR_IN32_SetERRLED (ULONG ulCtrl)			
引数	型名	変数名	内容	入出力
	ULONG	ulCtrl	LED 点灯制御 R_IN32_LED_OFF : LED 消灯 R_IN32_LED_ON : LED 点灯 R_IN32_LED_BLINK : LED 点滅	入力
戻り値	R_IN32_OK : 正常終了			
説明	ERR LED の点消灯を行います。			

注意. R-IN32M4-CL3 内部 WDT/外部 WDT/自局エラー発生中は、消灯または点滅にできません。

(2) gerR_IN32_SetUSER1LED

機能	LED 点灯制御 (User LED 1)			
呼び出し形式	ERRCODE gerR_IN32_SetUSER1LED (ULONG ulCtrl)			
引数	型名	変数名	内容	入出力
	ULONG	ulCtrl	LED 点灯制御 R_IN32_LED_OFF : LED 消灯 R_IN32_LED_ON : LED 点灯 R_IN32_LED_BLINK : LED 点滅	入力
戻り値	R_IN32_OK : 正常終了			
説明	User LED 1 の点消灯を行います。			

注意. R-IN32M4-CL3 内部 WDT/外部 WDT/自局エラー発生中は、消灯または点滅にできません。

(3) gerR_IN32_SetUSER2LED

機能	LED 点灯制御 (User LED 2)			
呼び出し形式	ERRCODE gerR_IN32_SetUSER2LED (ULONG ulCtrl)			
引数	型名	変数名	内容	入出力
	ULONG	ulCtrl	LED 点灯制御 R_IN32_LED_OFF : LED 消灯 R_IN32_LED_ON : LED 点灯 R_IN32_LED_BLINK : LED 点滅	入力
戻り値	R_IN32_OK : 正常終了			
説明	User LED 2 の点消灯を行います。			

注意. R-IN32M4-CL3 内部 WDT/外部 WDT/自局エラー発生中は、消灯または点滅にできません。

(4) gerR_IN32_SetRUNLED

機能	LED 点灯制御 (RUN)			
呼び出し形式	ERRCODE gerR_IN32_SetRUNLED (ULONG ulCtrl)			
引数	型名	変数名	内容	入出力
	ULONG	ulCtrl	LED 点灯制御 R_IN32_LED_OFF : LED 消灯 R_IN32_LED_ON : LED 点灯	入力
戻り値	R_IN32_OK : 正常終了			
説明	RUN LED の点消灯を行います。			

注意. R-IN32M4-CL3 内部 WDT/外部 WDT/自局エラー発生中は、消灯または点滅にできません。

(5) gerR_IN32_DisableLED

機能	LED の点灯機能無効			
呼び出し形式	ERRCODE gerR_IN32_DisableLED (USHORT usBitPattern)			
引数	型名	変数名	内容	入出力
	USHORT	usBitPattern	LED 点灯機能無効 (ON : 無効、OFF : 前置保持) Bit 0 : RUN LED 無効 Bit 2 : User LED 2 無効 Bit 4 : User LED 1 無効 Bit 6 : D LINK LED 無効 Bit 8 : ERR LED 無効 Bit10 : PORT1 L ER LED 無効 Bit11 : PORT2 L ER LED 無効 (Bit1、3、5、7、9、12~15 : 未使用)	入力
戻り値	R_IN32_OK : 正常終了			
説明	LED の点灯機能を無効にします。			

注意. R-IN32M4-CL3 内部 WDT/外部 WDT/自局エラー発生中は、消灯または点滅にできません。

(6) gerR_IN32_EnableLED

機能	LED の点灯機能有効			
呼び出し形式	ERRCODE gerR_IN32_EnableLED (USHORT usBitPattern)			
引数	型名	変数名	内容	入出力
	USHORT	usBitPattern	LED 点灯機能有効 (ON : 有効設定、OFF : 前置保持) Bit 0 : RUN LED 有効 Bit 2 : User LED 2 有効 Bit 4 : User LED 1 有効 Bit 6 : D LINK LED 有効 Bit 8 : ERR LED 有効 Bit10 : PORT1 L ER LED 有効 Bit11 : PORT2 L ER LED 有効 (Bit1、3、5、7、9、12～15 : 未使用)	入力
戻り値	R_IN32_OK : 正常終了			
説明	LED の点灯機能を有効にします。			

(7) gerR_IN32_UpdateLedStatus

機能	通信状態表示 LED の更新			
呼び出し形式	ERRCODE gerR_IN32UpdateLedStatus (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	データリンク状態取得 gerR_IN32_GetCommunicationStatus より取得する引数のデータリンク状態に応じて、自局の D LINK LED と ERR LED を制御します。			
	データリンク状態		D LINK LED	ERR LED
	データリンク中（サイクリック伝送中）		点灯	消灯
	データリンク中（サイクリック伝送停止中）		点滅	消灯
	また、リング接続時のトークン巡回の結果に応じて L ER1 LED、L ER2 LED を制御します。			
	トークン巡回	L ER1 LED	L ER2 LED	
	トークン未巡回（解列中）	消灯	消灯	
	トークン巡回中で両 PORT 有効	消灯	消灯	
	トークン巡回中で PORT 1 のみ有効	消灯	消灯	
	トークン巡回中で PORT 2 のみ有効	消灯	消灯	

5.3.8 ネットワーク時刻

(1) gerR_IN32_GetNetworkTime

機能	ネットワーク時刻（シリアル値）取得			
呼び出し形式	ERRCODE gerR_IN32_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 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_IN32_OK : 正常終了			
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。			

(2) gerR_IN32_SetNetworkTime

機能	ネットワーク時刻（シリアル値）設定			
呼び出し形式	ERRCODE gerR_IN32_SetNetworkTime (const USHORT *pusSerial)			
引数	型名	変数名	内容	入出力
	const USHORT	*pusSerial	ネットワーク時刻 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_IN32_OK : 正常終了			
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）を設定します。			

(3) gerR_IN32_NetworkTimeToDate

機能	ネットワーク時刻（シリアル値）→時計情報変換			
呼び出し形式	ERRCODE gerR_IN32_NetworkTimeToDate (R_IN32_TIMEINFO_T *pstTimeInfo, const USHORT *pusSerial)			
引数	型名	変数名	内容	入出力
	R_IN32_TIMEINFO_T	*pstTimeInfo	時計情報 詳細は「表 5.11 R_IN32_TIMEINFO_T 一覧」を参照してください。	出力
	const USHORT	*pusSerial	ネットワーク時刻 pusSerial[0] : ネットワーク時刻 (bit31-16) pusSerial[1] : ネットワーク時刻 (bit47-32)	入力
戻り値	R_IN32_OK : 正常終了			
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、秒単位のシリアル値）を時計情報（西暦・月・日・時・分・秒・ミリ秒（0 固定）・曜日）に変換します。			

R_IN32_TIMEINFO_T の構成を以下に示します。

表 5.11 R_IN32_TIMEINFO_T 一覧

No.	メンバ		概要	備考
1	USHORT	usYear	西暦 (2000-2136)	—
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))	—

(4) gerR_IN32_DateToNetworkTime

機能	時計情報→ネットワーク時刻 (シリアル値) 変換			
呼び出し形式	ERRCODE gerR_IN32_DateToNetworkTime (const R_IN32_TIMEINFO_T *pstTimeInfo, USHORT *pusSerial)			
引数	型名	変数名	内容	入出力
	const R_IN32_TIMEINFO_T	*pstTimeInfo	時計情報	入力
	USHORT	*pusSerial	ネットワーク時刻 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_IN32_OK : 正常終了 R_IN32_ERR : 異常終了			
説明	時計情報 (西暦・月・日・時・分・秒) をネットワーク時刻 (2000 年 1 月 1 日 0 時 0 分 0 秒を基点とした秒単位のシリアル値) に変換します。 (ausSerial[0] : ネットワーク時刻 (bit15-0) は、「0 固定」となります。)			

注意. 西暦が、2000 年～2136 年以外の場合、R_IN32_ERR 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は行っていないので、

うるう年・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。

5.3.9 MDIO アクセス

(1) gerR_IN32_EnableMACIPAccess

機能	MAC IP アクセス許可			
呼び出し形式	ERRCODE gerR_IN32_EnableMACIPAccess (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
	R_IN32_ERR : 異常終了 (MDIO コマンド終了待ち異常)			
説明	MAC IP のアクセスを許可します。			

注意. MAC IP アクセス許可～MAC IP アクセス禁止（「5.3.9(2) gerR_IN32_DisableMACIPAccess」参照）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。）
本関数の戻り値が R_IN32_OK 以外の場合、ユーザ作成の gR_IN32_CallbackFatalError を呼び出しますので、エラー・コードに対応したエラー処理を実行してください。

(2) gerR_IN32_DisableMACIPAccess

機能	MAC IP アクセス禁止			
呼び出し形式	ERRCODE gerR_IN32_DisableMACIPAccess (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	MAC IP のアクセスを禁止します。			

(3) gerR_IN32_WritePHY

機能	PHY 内部レジスタライト			
呼び出し形式	ERRCODE gerR_IN32_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_IN32_PORT1(0) : PORT1 R_IN32_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_IN32_OK : 正常終了 R_IN32_ERR : 異常終了 (MDIO コマンド終了待ち異常)			
説明	MDIO にて PHY 内部レジスタをライトします。			

注意. gerR_IN32_EnableMACIPAccess～gerR_IN32_DisableMACIPAccess の間で使用してください。
 本関数の戻り値が R_IN32_OK 以外の場合、ユーザ作成の gR_IN32_CallbackFatalError を
 呼び出しますので、エラー・コードに対応したエラー処理を実行してください。

(4) gerR_IN32_ReadPHY

機能	PHY 内部レジスタリード			
呼び出し形式	ERRCODE gerR_IN32_ReadPHY (ULONG ulPort, ULONG ulAddr, ULONG *ulData)			
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_IN32_PORT1(0) : PORT1 R_IN32_PORT2(1) : PORT2	入力
	ULONG	ulAddr	PHY レジスタアドレス	入力
	ULONG	*ulData	PHY からリードしたデータ	出力
戻り値	R_IN32_OK : 正常終了 R_IN32_ERR : 異常終了 (MDIO コマンド終了待ち異常)			
説明	MDIO にて PHY 内部レジスタをライトします。			

注意. gerR_IN32_EnableMACIPAccess～gerR_IN32_DisableMACIPAccess の間で使用してください。
 本関数の戻り値が R_IN32_OK 以外の場合、ユーザ作成の gR_IN32_CallbackFatalError を
 呼び出しますので、エラー・コードに対応したエラー処理を実行してください。

(5) gerR_IN32_CheckPHY

機能	PHY チェック処理			
呼び出し形式	ERRCODE gerR_IN32_CheckPHY(VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	PHY のリンク状態を取得します。			

5.3.10 トランジェント受信処理

(1) gerR_IN32_MainReceiveTransient1

機能	トランジェント受信メイン処理 1			
呼び出し形式	ERRCODE gerR_IN32_MainReceiveTransient1 (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	R-IN32M4-CL3 が受信したトランジェント受信フレームを取得します。			

(2) gerR_IN32_MainReceiveTransient2

機能	トランジェント受信メイン処理 2			
呼び出し形式	ERRCODE gerR_IN32_MainReceiveTransient2 (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	gerR_IN32_MainReceiveTransient1 で取得したトランジェント受信フレームを、gerR_IN32_CallbackReceivedTransient でユーザ・プログラムに渡します。			

(3) gerR_IN32_EnableReceiveTransient

機能	ユーザ理由によるトランジェント受信許可設定			
呼び出し形式	ULONG gerR_IN32_EnableReceiveTransient (BOOL blEnable)			
引数	型名	変数名	内容	入出力
	BOOL	blEnable	受信許可設定 R_IN32_TRUE : 受信許可 R_IN32_FALSE : 受信不許可	入力
戻り値	R_IN32_OK : 正常終了			
説明	ユーザ理由によるトランジェント受信許可/不許可を設定します。 ユーザ作成の gerR_IN32_CallbackReceivedTransient の戻り値が R_IN32_ERR の場合、「ユーザ理由によるトランジェント受信許可設定状態」を「受信不許可」とするので、受信可能な状態になりましたら本関数で「受信許可」としてください。			

(4) gblR_IN32_GetReceiveTransientStatus

機能	ユーザ理由によるトランジェント受信許可設定状態取得			
呼び出し形式	BOOL gblR_IN32_GetReceiveTransientStatus (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	受信許可設定状態 R_IN32_TRUE : 受信許可 R_IN32_FALSE : 受信不許可			
説明	ユーザ理由によるトランジェント受信許可設定状態を取得します。			

(5) gerR_IN32_SetMACAddressTableData

機能	ノード情報配信データ（MAC アドレス表）設定			
呼び出し形式	ERRCODE gerR_IN32_SetMACAddressTableData (UCHAR uchSeqNumber, R_IN32_MACADDRESSDATA_T *pstMacAddrDat)			
引数	型名	変数名	内容	入出力
	UCHAR	uchSeqNumber	配信シーケンシャル番号 (値域：1～7)	入力
	R_IN32_MACADDRESSDATA_T	*pstMacAddrDat	MAC アドレスなどの情報 (MAC アドレス表)	入力
戻り値	R_IN32_OK：正常終了 R_IN32_ERR_OUTOFRANGE：局番範囲外または配信シーケンシャル番号範囲外			
説明	ノード情報配信で取得した MAC アドレスなどの情報（MAC アドレス表）および配信シーケンシャル番号を設定します。			

注意. マスタ局の局番は、7Dh として登録してください。

「表 5.7 R_IN32_UNITINIT_T 一覧」の「No.7 ノード情報配信の要求」で R_IN32_FALSE と設定した場合、本関数の呼び出しは不要です。

R_IN32_MACADDRESSDATA_T の構成を以下に示します。

表 5.12 R_IN32_MACADDRESSDATA_T 一覧

No.	メンバ		概要	備考
1	USHORT	usNodeNumber	局番（1～120、マスタ局：7Dh）	—
2	UCHAR	uchTransientReceiveEnable	トランジェント受信機能 (R_IN32_ENABLE/R_IN32_DISABLE)	—
3	UCHAR	auchMacAddress[6]	MAC アドレス	—

5.3.11 トランジェント要求受信対応

(1) gerR_IN32_ReceivedMACAddressData

機能	ノード情報配信フレーム受信対応処理			
呼び出し形式	ERRCODE gerR_IN32_ReceivedMACAddressData (const VOID* pvReceivedData, ULONG ulDataSize)			
引数	型名	変数名	内容	入出力
	const VOID*	pvReceivedData	受信データ格納エリア	入力
	ULONG	ulDataSize	受信データサイズ	入力
戻り値	R_IN32_OK : 正常 R_IN32_ERR : 受信データの宛先局番または宛先ネットワーク No.が異常			
説明	gerR_IN32_SetMACAddressTableData による他局の MAC アドレスデータ作成を実施する。			

(2) gerR_IN32_ReceivedStatisticInfoRequest

機能	統計情報取得要求フレーム受信対応処理			
呼び出し形式	ERRCODE gerR_IN32_ReceivedStatisticInfoRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA)			
引数	型名	変数名	内容	入出力
	VOID*	pvSendFrame	送信フレーム	出力
	ULONG*	pulDataSize	送信データサイズ	出力
	const VOID*	pvReceivedData	受信データ格納エリア	入力
	const UCHAR*	puchSA	送信元ノード MAC アドレス	入力
戻り値	R_IN32_OK : 正常 R_IN32_ERR : 受信データの宛先局番または宛先ネットワーク No.が異常			
説明	引数で指定した受信データより統計情報取得応答フレームの作成を実施する。			

(3) gerR_IN32_ReceivedUnitInfoRequest

機能	ノード詳細情報取得要求フレーム受信対応処理			
呼び出し形式	ERRCODE gerR_IN32_ReceivedUnitInfoRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA)			
引数	型名	変数名	内容	入出力
	VOID*	pvSendFrame	送信フレーム	出力
	ULONG*	pulDataSize	送信データサイズ	出力
	const VOID*	pvReceivedData	受信データ格納エリア	入力
	const UCHAR*	puchSA	送信元ノード MAC アドレス	入力
戻り値	R_IN32_OK : 正常 R_IN32_ERR : 受信データの宛先局番または宛先ネットワーク No.が異常			
説明	引数で指定した受信データよりノード詳細情報取得応答フレームの作成を実施する。			

(4) gerR_IN32_ReceivedOptionInfoRequest

機能	オプション情報取得要求フレーム受信対応処理			
呼び出し形式	ERRCODE gerR_IN32_ReceivedOptionInfoRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA, const USHORT usSupportFunction)			
引数	型名	変数名	内容	入出力
	VOID*	pvSendFrame	送信フレーム	出力
	ULONG*	pulDataSize	送信データサイズ	出力
	const VOID*	pvReceivedData	受信データ格納エリア	入力
	const UCHAR*	puchSA	送信元ノード MAC アドレス	入力
	const USHORT	usSupportFunction	オプション機能対応	入力
戻り値	R_IN32_OK : 正常 R_IN32_ERR : 受信データの宛先局番または宛先ネットワーク No.が異常			
説明	引数で指定した受信データよりオプション情報取得応答フレームの作成を実施する。			

(5) gerR_IN32_ReceivedSelectInfoRequest

機能	選択局情報取得要求フレーム受信対応処理			
呼び出し形式	ERRCODE gerR_IN32_ReceivedSelectInfoRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA, const VOID* pvR_IN32TLedInfo)			
引数	型名	変数名	内容	入出力
	VOID*	pvSendFrame	送信フレームのアドレス	出力
	ULONG*	pulDataSize	送信データサイズ	出力
	const VOID*	pvReceivedData	受信データ格納エリア	入力
	const UCHAR*	puchSA	送信元ノード MAC アドレス	入力
	const VOID*	pvR_IN32TLedInfo	自局 LED 情報	入力
戻り値	R_IN32_OK : 正常 R_IN32_ERR : 受信データの宛先局番または宛先ネットワーク No.が異常			
説明	引数で指定した受信データより選択局情報取得応答フレームの作成を実施する。			

(6) gerR_IN32_ReceivedContactTestRequest

機能	交信テスト要求フレーム受信対応処理			
呼び出し形式	ERRCODE gerR_IN32_ReceivedContactTestRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA)			
引数	型名	変数名	内容	入出力
	VOID*	pvSendFrame	送信フレーム	出力
	ULONG*	pulDataSize	送信データサイズ	出力
	const VOID*	pvReceivedData	受信データ格納エリア	入力
	const UCHAR*	puchSA	送信元ノード MAC アドレス	入力
戻り値	R_IN32_OK : 正常 R_IN32_ERR : 受信データの宛先局番または宛先ネットワーク No.が異常			
説明	引数で指定した受信データより交信テスト応答フレームの作成を実施する。			

(7) gerR_IN32_ReceivedCableTestRequest

機能	ケーブルテスト要求フレーム受信対応処理			
呼び出し形式	ERRCODE gerR_IN32_ReceivedCableTestRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA)			
引数	型名	変数名	内容	入出力
	VOID*	pvSendFrame	送信フレーム	出力
	ULONG*	pulDataSize	送信データサイズ	出力
	const VOID*	pvReceivedData	受信データ格納エリア	入力
	const UCHAR*	puchSA	送信元ノード MAC アドレス	入力
戻り値	R_IN32_OK : 正常 R_IN32_ERR : 受信データの宛先局番または宛先ネットワーク No.が異常			
説明	引数で指定した受信データよりケーブルテスト応答フレームの作成を実施する。			

(8) gerR_IN32_ReceiveRemoteResetRequest

機能	SLMP リモートリセット要求フレーム受信対応処理			
呼び出し形式	ERRCODE gerR_IN32_ReceiveRemoteResetRequest (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA)			
引数	型名	変数名	内容	入出力
	VOID*	pvSendFrame	送信フレーム	出力
	ULONG*	pulDataSize	送信データサイズ	出力
	const VOID*	pvReceivedData	受信データ格納エリア	入力
	const UCHAR*	puchSA	送信元ノード MAC アドレス	入力
戻り値	R_IN32_OK : 正常 R_IN32_ERR : 受信データの宛先局番または宛先ネットワーク No.が異常			
説明	引数で指定した SLMP リモートリセット要求フレームの局番・ネットワーク No.の範囲外チェックを実施する。SLMP リモートリセット要求を正常受信した場合、応答フレームは送信しない。			

(9) gR_IN32_SetSimpError_Response

機能	SLMP エラー応答フレーム作成処理			
呼び出し形式	VOID gR_IN32_SetSimpError_Response (VOID* pvSendFrame, ULONG* pulDataSize, const VOID* pvReceivedData, const UCHAR* puchSA, USHORT usFinCode)			
引数	型名	変数名	内容	入出力
	VOID*	pvSendFrame	送信フレーム	出力
	ULONG*	pulDataSize	送信データサイズ	出力
	const VOID*	pvReceivedData	受信データ格納エリア	入力
	const UCHAR*	puchSA	送信元ノード MAC アドレス	入力
	USHORT	usFinCode	終了コード	
戻り値	なし			
説明	SLMP エラー応答フレームを作成する。作成する応答フレームの終了コードには、引数 (usFinCode) を格納する。			

(10) gerR_IN32_ErrCheckReqFieldNetworkReceived

機能	CC-Link IE フィールド固有要求受信フレーム異常確認処理			
呼び出し形式	ERRCODE gerR_IN32_ErrCheckReqFieldNetworkReceived (const VOID* pvReceivedData)			
引数	型名	変数名	内容	入出力
	const VOID*	pvReceivedData	受信データ格納エリア	入力
戻り値	R_IN32_OK : 正常 R_IN32_ERR : 受信データの宛先局番または宛先ネットワーク No.が異常			
説明	受信した CC-Link IE フィールド固有要求フレームの宛先局番とネットワーク No.が設定範囲内であることを確認する。			

(11) gerR_IN32_ErrCheckReqSlmpReceived

機能	SLMP 要求受信フレーム異常確認処理			
呼び出し形式	USHORT gusR_IN32_ErrCheckReqSlmpReceived (const VOID* pvReceivedData)			
引数	型名	変数名	内容	入出力
	const VOID*	pvReceivedData	受信データ格納エリア	入力
戻り値	R_IN32_SLMP_FINCODE_OK : 正常 (0000h) 上記以外 : 受信データの宛先局番または宛先ネットワーク No.が異常			
説明	受信した SLMP 要求フレームの宛先局番とネットワーク No.が設定範囲内であることを確認する。			

5.3.12 トランジェント送信フレームヘッダ作成

(1) gerR_IN32_SetEtherCcieHeader

機能	MAC+CCIE ヘッダ作成処理			
呼び出し形式	ERRCODE gerR_IN32_SetEtherCcieHeader (const UCHAR* puchSndMac, const UCHAR* puchMyMac, UCHAR uchFrameClassification, UCHAR uchDataClassification, R_IN32_NONCICLIC_FRAME_T* pstCOMMON)			
引数	型名	変数名	内容	入出力
	CONST UCHAR*	puchSndMac	宛先 MAC アドレス	入力
	const UCHAR*	puchMyMac	自 MAC アドレス	入力
	UCHAR	uchFrameClassification	フレーム種別	入力
	UCHAR	uchDataClassification	データ種別	入力
	R_IN32_NONCICLIC_FRAME_T*	pstCOMMON	作成する MAC+CCIE ヘッダ のアドレス	出力
戻り値	erR_IN32T_TxFrame_CreateEtherCcieHeader の戻り値 (R_IN32_OK : 正常終了のみ)			
説明	erR_IN32T_TxFrame_CreateEtherCcieHeader の呼び出しにより pstCOMMON に対して、MAC Header と CCIE ヘッダを作成します。			

(2) gerR_IN32_SetTransient1Header

機能	Transient1 ヘッダ作成処理			
呼び出し形式	ERRCODE gerR_IN32_SetTransient1Header (USHORT usDataSubClassification, USHORT usTransientDataSize, R_IN32_TRAN1_HEAD_T* pstHEAD)			
引数	型名	変数名	内容	入出力
	USHORT	usDataSubClassification	データサブ種別	入力
	USHORT	usTransientDataSize	トランジェントデータのサイズ	入力
	R_IN32_TRAN1_HEAD_T*	pstHEAD	作成する Transient1 ヘッダの アドレス	出力
戻り値	erR_IN32T_TxFrame_CreateTransient1Header の戻り値 (R_IN32_OK : 正常終了 / R_IN32_ERR : 引数の内容が異常)			
説明	erR_IN32T_TxFrame_CreateTransient1Header の呼び出しにより pstHEAD に対して Transient1 ヘッ ダを作成します。			

(3) gerR_IN32_SetRequestSimpHeader

機能	SLMP 要求ヘッダ作成処理			
呼び出し形式	ERRCODE gerR_IN32_SetRequestSimpHeader (R_IN32_SLMP_REQUSET_SETTING_T* pstSimpReqSetting, R_IN32_SLMP_REQUEST_FRAME_T* pstSimpExHead, ULONG* pulAllDataSize, USHORT* pusReqSerialNo)			
引数	型名	変数名	内容	入出力
	R_IN32_SLMP_REQUSET_SETTING_T*	pstSimpReqSetting	対象局設定	入力
	R_IN32_SLMP_REQ_FRAME_T*	pstSimpExHead	作成する Transient2+SLMP ヘッダのアドレス	出力
	ULONG*	pulAllDataSize	SLMP データ部のサイズ	出力
	USHORT*	pusReqSerialNo	シリアル番号	出力
戻り値	gerR_IN32T_TxFrame_CreateRequestSimpHeader の戻り値 (R_IN32_OK : 正常終了 / R_IN32_ERR : 引数の内容が異常)			
説明	erR_IN32T_TxFrame_CreateRequestSimpHeader の呼び出しにより pstSimpExHead に対して Transient2 ヘッダと SLMP ヘッダ（要求用）を作成します。			

(4) gerR_IN32_SetResponseSimpHeader

機能	SLMP 応答ヘッダ作成処理			
呼び出し形式	ERRCODE gerR_IN32_SetResponseSimpHeader (const R_IN32_SLMP_REQUEST_FRAME_T* pstReqSimpExHead, USHORT usSimpDataSize, R_IN32_SLMP_RESPONSE_FRAME_T* pstSimpExHead)			
引数	型名	変数名	内容	入出力
	R_IN32_SLMP_REQ_FRAME_T*	pstReqSimpExHead	受信した SLMP 要求フレーム	入力
	ULONG	usSimpDataSize	SLMP データ部のサイズ	入力
	R_IN32_SLMP_RES_FRAME_T*	pstSimpExHead	作成する Transient2+SLMP ヘッダ	出力
戻り値	erR_IN32T_TxFrame_CreateResponseSimpHeader の戻り値 (R_IN32_OK : 正常終了 / R_IN32_ERR : 引数の内容が異常)			
説明	erR_IN32T_TxFrame_CreateResponseSimpHeader の呼び出しにより pstSimpExHead に対して Transient2 ヘッダと SLMP ヘッダ（応答用）を作成します。			

5.3.13 トランジェント送信処理

(1) gerR_IN32_GetUnitInformation

機能	ユニット情報取得			
呼び出し形式	ERRCODE gerR_IN32_GetUnitInformation (R_IN32_UNITINFO_T *pstUnitInfo, R_IN32_UNITNETWORKSETTING_T *pstUnitNetworkSetting)			
引数	型名	変数名	内容	入出力
	R_IN32_UNITINFO_T	*pstUnitInfo	ユニット情報	出力
	R_IN32_UNITNETWORKSETTING_T	*pstUnitNetworkSetting	ネットワーク動作設定	出力
戻り値	R_IN32_OK : 正常終了			
説明	自局の設定情報を取得します。 取得した設定情報は、ノード詳細情報取得応答フレーム作成時に使用します。			

R_IN32_UNITNETWORKSETTING_T の構成を以下に示します。

表 5.13 R_IN32_UNITNETWORKSETTING_T 一覧

No.	メンバ		概要	備考
1	ULONG	ulFrameSendCount	トークン保持時の送信回数設定	—
2	ULONG	ulFrameSendInterval	フレーム送信間隔設定	—
3	ULONG	ulTokenSendCount	トークン送信回数設定	—

(2) gusR_IN32_GetNodeID

機能	ノード ID 取得			
呼び出し形式	USHORT gusR_IN32_GetNodeID (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	ノード ID			
説明	ノード ID を取得します。 取得したノード ID は、トランジェント送信する場合に使用します。			

(3) gerR_IN32_GetMulticastMACAddress

機能	マルチキャスト MAC アドレス取得			
呼び出し形式	ERRCODE gerR_IN32_GetMulticastMACAddress (UCHAR *puchMACAddr)			
引数	型名	変数名	内容	入出力
	UCHAR	*puchMACAddr	マルチキャストアドレス 13-34-56-78-90-AB が設定されている場合、以下のアドレスが返ります。 puchMACAddr[0] : 13h puchMACAddr[1] : 34h puchMACAddr[2] : 56h puchMACAddr[3] : 78h puchMACAddr[4] : 90h puchMACAddr[5] : ABh	出力
戻り値	R_IN32_OK : 正常終了 R_IN32_ERR : 異常終了 (データリンク未実施 (解列中) のため、マルチキャスト MAC アドレスを取得できなかった。)			
説明	マルチキャスト MAC アドレスを取得します。 取得したマルチキャスト MAC アドレスは、ネットワークに接続するすべての局にトランジェント送信する場合の宛先アドレスとして使用します。			

(4) gerR_IN32_GetUnicastMACAddress

機能	ユニキャスト MAC アドレス取得			
呼び出し形式	ERRCODE gerR_IN32_GetUnicastMACAddress (USHORT usNodeNumber,UCHAR *puchMACAddr)			
引数	型名	変数名	内容	入出力
	USHORT	usNodeNumber	局番 (値域 : 1~120、マスタ局 : 7Dh)	入力
	UCHAR	*puchMACAddr	ユニキャストアドレス 12-34-56-78-90-AB が設定されている場合、以下のアドレスが返ります。 puchMACAddr[0] : 12h puchMACAddr[1] : 34h puchMACAddr[2] : 56h puchMACAddr[3] : 78h puchMACAddr[4] : 90h puchMACAddr[5] : ABh	出力
戻り値	R_IN32_OK : 正常終了 R_IN32_ERR_NOENTRY : エントリがない R_IN32_ERR_OUTOFRANGE : 局番範囲外			
説明	マスタ局から受信したノード情報配信より局番に対応したユニキャスト MAC アドレスを取得します。 マスタ局の局番は、7Dh としてください。			

注意. データリンク未実施 (解列中) の場合、ユニキャスト MAC アドレスの取得はできません。
(R_IN32_ERR_NOENTRY となります。)

(5) gerR_IN32_GetSendTransientBuffer

機能	トランジェント送信バッファ取得			
呼び出し形式	ERRCODE gerR_IN32_GetSendTransientBuffer (USHORT usSize, VOID** ppvSendBuffAddr, UCHAR *puchSendBuffNo, UCHAR *puchConnectionInfo)			
引数	型名	変数名	内容	入出力
	USHORT	usSize	DCS/FCS を除く送信データサイズ	入力
	VOID	**ppvSendBuffAddr	トランジェント送信バッファアドレス	出力
	UCHAR	*puchSendBuffNo	トランジェント送信バッファ No.	出力
	UCHAR	*puchConnectionInfo	トランジェントコネクション情報	出力
戻り値	R_IN32_OK : 正常終了 (トランジェント送信バッファ取得) R_IN32_ERR : 異常終了 (トランジェント送信バッファ取得異常)			
説明	トランジェント送信エリアに「送信データサイズ」分の空きがあるか問い合わせし、空きがあれば以下の情報を返します。 ・ トランジェント送信バッファアドレス ・ トランジェント送信バッファ No. ・ トランジェントコネクション情報			

注意. 以下の場合、トランジェント送信ができないので、R_IN32_ERR : 異常終了で終了します。

- ・ データリンク未実施（解列中）の場合
- ・ 送信データサイズが 1510 バイトより大きい場合

トランジェント送信する場合、以下の処理を行ってください。

- ・ 本関数にてトランジェント送信バッファ No.を取得
- ・ 取得したトランジェント送信バッファに送信データを格納
- ・ gerR_IN32_RequestSendingTransient にてトランジェント送信要求

(6) gerR_IN32_RequestSendingTransient

機能	トランジェント送信要求			
呼び出し形式	ERRCODE gerR_IN32_RequestSendingTransient (UCHAR uchSendBuffNo, USHORT usSize)			
引数	型名	変数名	内容	入出力
	UCHAR	uchSendBuffNo	トランジェント送信バッファ No.	入力
	USHORT	usSize	DCS/FCS を除く送信データサイズ	入力
戻り値	R_IN32_OK : 正常終了 R_IN32_ERR : 異常終了 (トランジェント送信要求異常)			
説明	gerR_IN32_GetSendTransientBuffer で取得したトランジェント送信バッファ No.に対して送信設定を行います。本関数を実行する前に、以下の処理を行ってください。 ・ gerR_IN32_GetSendTransientBuffer で、トランジェント送信バッファを取得 ・ 取得したトランジェント送信バッファに送信データを格納 本関数にて送信要求後に発生した異常は、gerR_IN32_MainSendTransient の戻り値で通知します。送信データサイズは、gerR_IN32_GetSendTransientBuffer にて指定した値と同じにしてください。			

注意. 以下の場合、トランジェント送信ができないので、R_IN32_ERR : 異常終了で終了します。

- ・ データリンク未実施（解列中）の場合

(7) gerR_IN32_MainSendTransient

機能	トランジェント送信メイン処理			
呼び出し形式	ULONG gerR_IN32_MainSendTransient (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了			
説明	トランジェント送信完了結果を取得します。 本関数から、gerR_IN32_CallbackTransientSendingComplete を呼び出して、対象の送信ディスクリプタの状態（送信結果）を通知します。			

5.3.14 H/W テスト

(1) gerR_IN32_IEEEtest

機能	IEEE802.3ab コンプライアンス・テスト			
呼び出し形式	ERRCODE gerR_IN32_IEEEtest (USHORT usMode)			
引数	型名	変数名	内容	入出力
	USHORT	usMode	IEEE802.3ab コンプライアンス・テストモード R_IN32_IEEE_MODE1(1) : MODE1 R_IN32_IEEE_MODE2(2) : MODE2 R_IN32_IEEE_MODE3(3) : MODE3 R_IN32_IEEE_MODE4(4) : MODE4 R_IN32_IEEE_END(5) : テスト終了	入力
戻り値	R_IN32_OK : 正常終了 R_IN32_ERR : 異常終了			
説明	引数の IEEE802.3ab コンプライアンス・テストモードにしたがい、テストモード用の波形出力を PHY に設定します。			

注意. 本関数の戻り値が R_IN32_OK 以外の場合、ユーザ作成の gR_IN32_CallbackFatalError を呼び出しますので、エラー・コードに対応したエラー処理を実行してください。

(2) gerR_IN32_InitializeLoopBackTest

機能	折り返し通信テスト初期化			
呼び出し形式	ERRCODE gerR_IN32_InitializeLoopBackTest (VOID)			
引数	型名	変数名	内容	入出力
	—	—	—	—
戻り値	R_IN32_OK : 正常終了 R_IN32_ERR : 異常終了			
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。			

注意. 本関数の戻り値が R_IN32_OK 以外の場合、ユーザ作成の gR_IN32_CallbackFatalError を呼び出しますので、エラー・コードに対応したエラー処理を実行してください。

(3) gerR_IN32_ExternalLoopBackTest

機能	折り返し通信テスト			
呼び出し形式	ERRCODE gerR_IN32_ExternalLoopBackTest (ULONG ulPort)			
引数	型名	変数名	内容	入出力
	ULONG	ulPort	送信元ポート R_IN32_PORT1(0) : PORT1 R_IN32_PORT2(1) : PORT2	入力
戻り値	R_IN32_OK : 正常終了 R_IN32_ERR : 異常終了			
説明	引数で指定されたテスト対象ポートからフレームを送信し、他方のポートで受信した結果を確認します。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。			

注意. 本関数の戻り値が R_IN32_OK 以外の場合、ユーザ作成の gR_IN32_CallbackFatalError を呼び出しますので、エラー・コードに対応したエラー処理を実行してください。

5.4 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 5.14 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	エラー処理	gR_IN32_CallbackFatalError	VOID	R-IN32M4-CL3 の Fatal エラー取得
2	自局状態取得	gerR_IN32_CallbackCommandFromMaster	ERRCODE	マスタ局からのコマンド取得
3	トランジェント	gerR_IN32_CallbackReceivedTransient	ERRCODE	受信トランジェント・フレーム取得
4	送受信	gerR_IN32_CallbackTransientSendingComplete	ERRCODE	トランジェント送信完了状態取得

5.5 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、ユーザでカスタマイズする必要があります。
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_IN32_CallbackFatalError

機能	R-IN32M4-CL3 の Fatal エラー取得			
呼び出し形式	VOID gR_IN32_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	なし			
説明	R-IN32M4-CL3 の Fatal エラーを取得します。 R-IN32M4-CL3 ドライバは、R-IN32M4-CL3 の Fatal エラーを検出したとき本関数を呼び出します。 関数内部はベンダが任意に実装します。			

表 5.15 gR_IN32_CallbackFatalError 関数の Fatal エラー・コード一覧

Fatal エラー・コード (ulErrorCode)	Fatal エラー情報 (ulErrorInfo)	Fatal エラー内容	処置
D529	ドライバ内呼び出し元関数 gerR_IN32D_ClearTxRxRAM のアドレス	通信 LSI 異常	ノイズなどによる誤動作が考えられます。電線やケーブルの距離、各機器の接地などを確認してノイズ対策を行ってください。 ハードウェアテストを実行してください。再度異常になったときはハードウェアに異常が考えられます。
D52A	ドライバ内呼び出し元関数 erR_IN32D_MDIO_WaitCommandComplete のアドレス	通信 LSI 異常	
D52B	ドライバ内呼び出し元関数 erR_IN32D_ResetMAC のアドレス	通信 LSI 異常	
D52C	ドライバ内呼び出し元関数 gerR_IN32D_StartRing のアドレス	通信 LSI 異常	

(2) gerR_IN32_CallbackCommandFromMaster

機能	マスタ局からのコマンド取得			
呼び出し形式	ERRCODE gerR_IN32_CallbackCommandFromMaster (ULONG pulCommand)			
引数	型名	変数名	内容	入出力
	ULONG	pulCommand	マスタ局からのコマンド状態。 詳細は「表 5.16 ulCommand 一覧」を参照してください。	入力
戻り値	R_IN32_OK : 正常終了			
説明	マスタ局からの Parameter フレーム受信によるコマンドを取得します。 R-IN32M4-CL3 ドライバは、マスタ局から Parameter フレームを受信したときに本関数を呼び出します。関数内部はユーザが任意に実装します。			

ulCommand の構成を以下に示します。

表 5.16 ulCommand 一覧

No.	ビット	概要	内容
1	b0	サイクリック伝送停止指示（自局の局番範囲外）	1b : 停止指示あり 0b : 停止指示なし
2	b1	サイクリック伝送停止指示（予約局設定）	1b : 停止指示あり 0b : 停止指示なし
3	b2	サイクリック伝送停止指示（マスタ局指示）	1b : 停止指示あり 0b : 停止指示なし
4	b3	サイクリック伝送停止指示（局番重複）	1b : 停止指示あり 0b : 停止指示なし
5	b15-4	予約	—
6	b16	局種別不正（マスタ局が指定した局種別と自局の局種別が不一致）	1b : 局種別不正あり 0b : 局種別不正なし
7	b17	指定サイズ不正 ^注	1b : 指定サイズ不正あり 0b : 指定サイズ不正なし
8	b31-18	予約	—

注. マスタ局が指定したサイクリック伝送サイズが自局のサイクリック伝送可能な最大サイズ（gerR_IN32_Initialize にて設定したサイズ）より大きかった。

(3) gerR_IN32_CallbackReceivedTransient

機能	受信トランジェント・フレーム取得			
呼び出し形式	ERRCODE gerR_IN32_CallbackReceivedTransient (VOID *pvRcv, USHORT usFrameSize)			
引数	型名	変数名	内容	入出力
	VOID	*pvRcv	受信バッファ	入力
	USHORT	usFrameSize	FCSを除くフレームサイズ	入力
戻り値	R_IN32_OK : 正常終了 R_IN32_ERR : 異常終了			
説明	受信したトランジェント・フレームを取得します。 R-IN32M4-CL3 ドライバは、トランジェント・フレームを受信したときに本関数を呼び出します。 関数内部はユーザが任意に実装します。			

注意. 受信バッファの先頭アドレスは、4 バイト単位（0 または 4 の倍数）としてください。

戻り値が R_IN32_OK 以外の場合、「ユーザ理由によるトランジェント受信許可設定」を「受信不許可」とするので、受信可能な状態になりましたら gerR_IN32_EnableReceiveTransient にて「受信許可」としてください。

(4) gerR_IN32_CallbackTransientSendingComplete

機能	トランジェント送信完了状態取得			
呼び出し形式	ERRCODE gerR_IN32_CallbackTransientSendingComplete (UCHAR uchSendBuffNo, ERRCODE erSendStatus)			
引数	型名	変数名	内容	入出力
	UCHAR	uchSendBuffNo	トランジェント送信バッファ No.	入力
	ERRCODE	erSendStatus	対象のトランジェント送信バッファの状態（送信結果） R_IN32_OK : トランジェント送信正常完了 R_IN32_ERR : トランジェント送信異常完了	入力
戻り値	R_IN32_OK : 正常終了			
説明	トランジェント送信バッファの送信状態（送信結果）を取得します。 R-IN32M4-CL3 ドライバは、トランジェント・フレームの送信が完了したときに本関数を呼び出します。 関数内部はユーザが任意に実装します。			

6. リンクデバイスのシステム領域

CC-Link IE フィールド・ネットワークに接続するスレーブ局は、保有するリンクデバイスの一部をシステム領域として定義することができます。システム領域は、自局状態を他局に通知するため、およびマスタ局から自局に対して指示するために使用します。

リンクデバイスの一部をシステム領域として定義することは任意です。システム領域を定義する場合は、リンクデバイスのビットを「表 6.1 システム領域のビット割り付け（例）」のように割り付けてください。

リモート入力（RX）、リモート出力（RY）にシステム領域を定義する例を下表に示します。リモートレジスタ（RW_r、RW_w）に定義する場合は、RX を RW_w、RY を RW_w に置き換えてください。

表 6.1 システム領域のビット割り付け（例）

ビット	名称	ビット	名称
RX (S+0)	リザーブ	RY (S+0)	リザーブ
RX (S+1)		RY (S+1)	
RX (S+2)		RY (S+2)	
RX (S+3)		RY (S+3)	
RX (S+4)		RY (S+4)	
RX (S+5)		RY (S+5)	
RX (S+6)		RY (S+6)	
RX (S+7)	ワーニング状態フラグ	RY (S+7)	
RX (S+8)	イニシャルデータ処理要求フラグ	RY (S+8)	イニシャルデータ処理完了フラグ
RX (S+9)	イニシャルデータ設定完了フラグ	RY (S+9)	イニシャルデータ設定要求フラグ
RX (S+A)	エラー状態フラグ	RY (S+A)	エラーリセット要求フラグ
RX (S+B)	リモート Ready	RY (S+B)	リザーブ
RX (S+C)	リザーブ	RY (S+C)	
RX (S+D)		RY (S+D)	
RX (S+E)		RY (S+E)	
RX (S+F)		RY (S+F)	

備考. S : システム領域の先頭番号

6.1 システム領域詳細

システム領域の各ビットの詳細を、リモート入力 (RX)、リモート出力 (RY) の例で示します。

(1) リモート Ready : RX (S+B)

マスタ局と自局間でのデータ送受信が可能なことを表します。

電源投入後またはハードウェアリセット後に本ビットを ON してください。

エラー状態フラグにより、マスタ局と自局とのデータの送受信ができないときは OFF してください。

ただし、ワーニング状態フラグでは、ON のままにしてください。

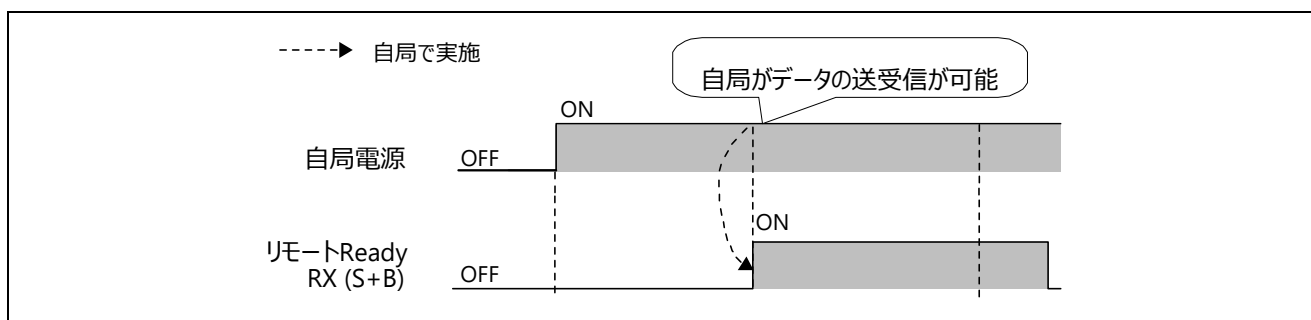


図 6.1 リモート Ready タイミング図

(2) イニシャルデータ処理要求フラグ : RX (S+8) / イニシャルデータ処理完了フラグ : RY (S+8)

自局の電源投入後またはハードウェアリセット後に、自局がマスタ局に対して、イニシャルデータ処理を要求するときに使用します。イニシャルデータ処理完了後に、リモート Ready を ON してください。

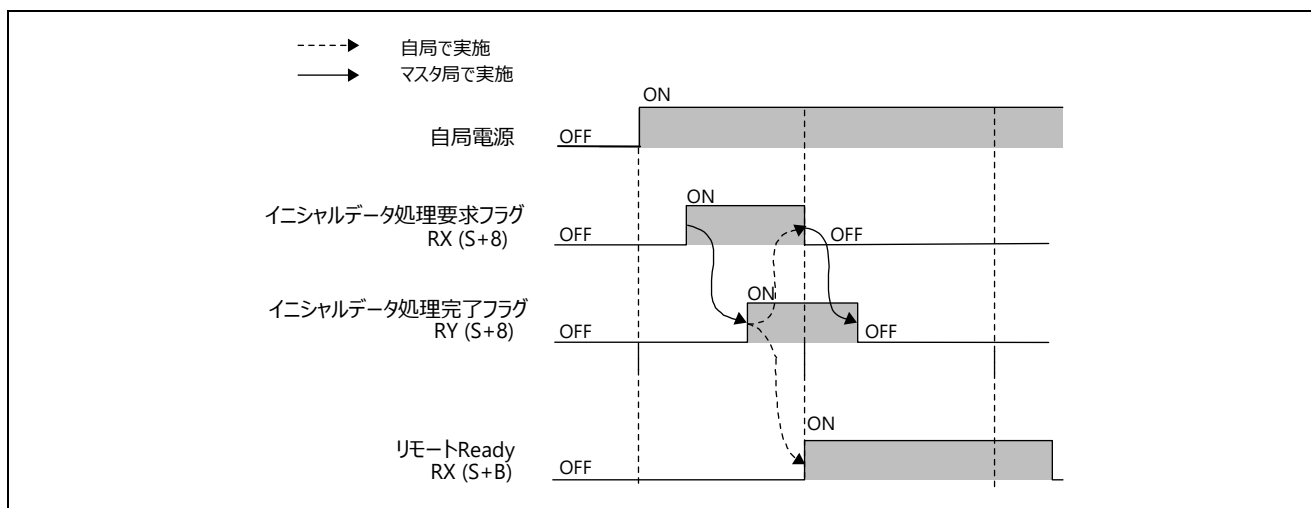


図 6.2 イニシャルデータ処理要求/完了フラグ タイミング図

(3) イニシャルデータ設定完了 : RX (S+9) / イニシャル設定要求フラグ : RY (S+9)

マスタ局から自局に対して、イニシャルデータ設定を要求するときに使用します。

イニシャルデータ設定完了後に、リモート Ready を ON してください。

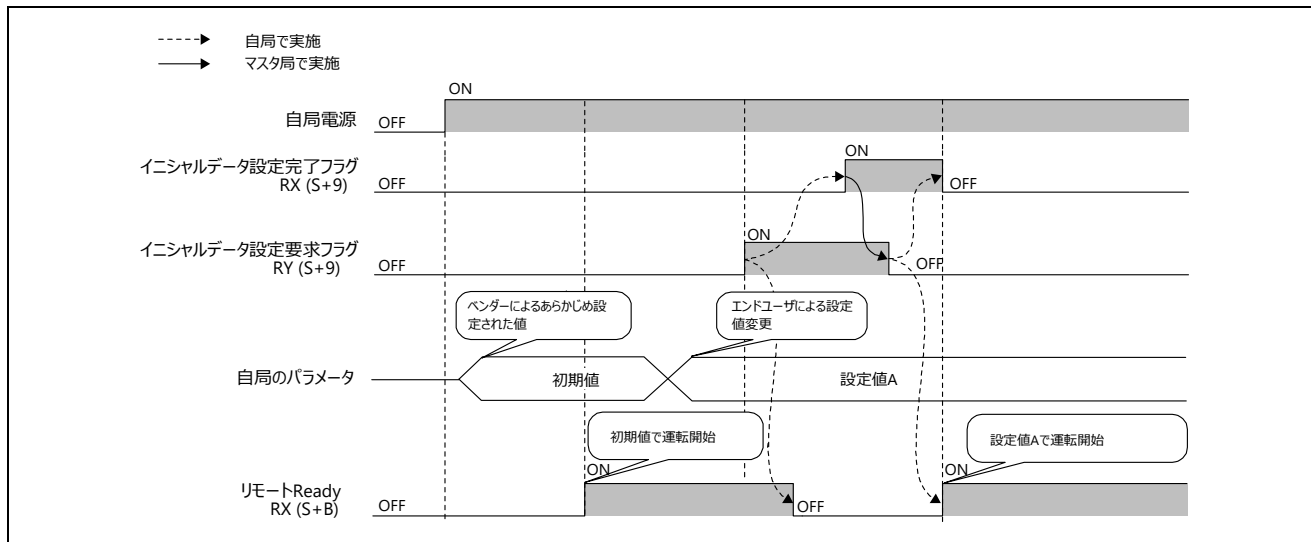


図 6.3 イニシャルデータ設定完了/要求フラグ タイミング図

(4) イニシャルデータ処理要求/完了フラグと、イニシャルデータ設定完了/要求フラグの両方を実装する場合

両方を実装する場合は、イニシャルデータ処理およびイニシャルデータ設定の両処理の完了後にリモート Ready を ON してください。

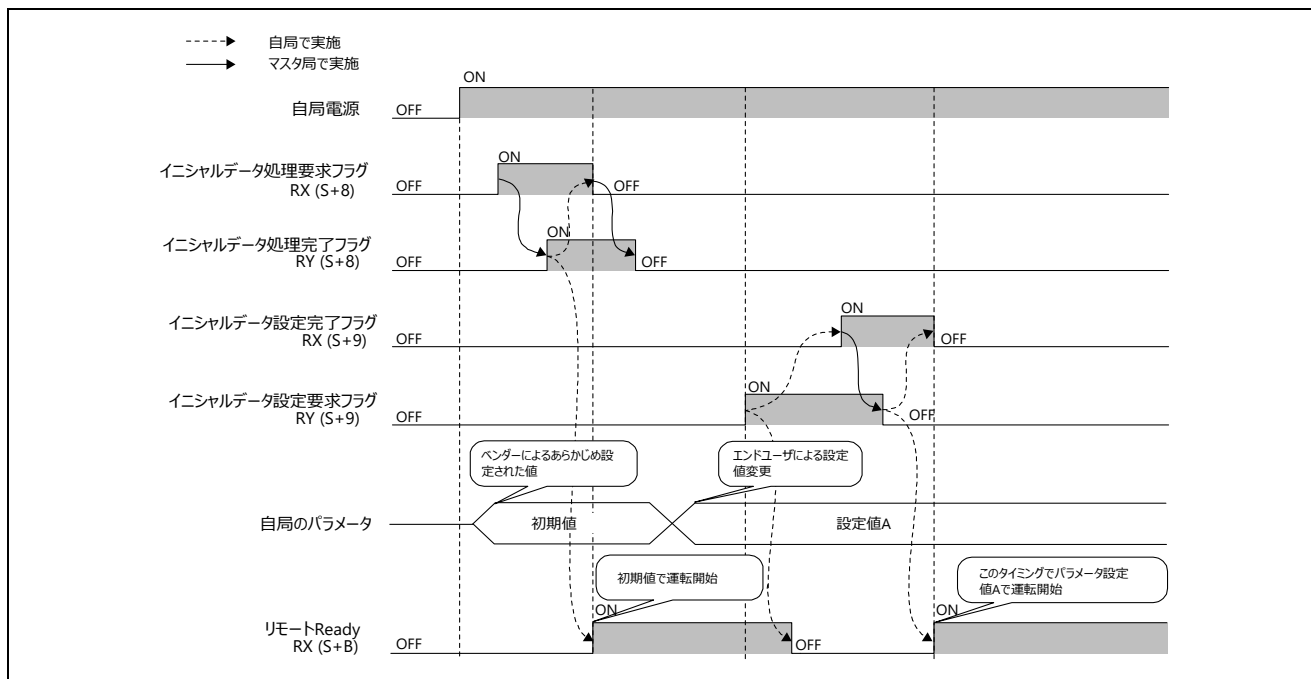


図 6.4 イニシャルデータ処理および設定 タイミング図

(5) エラー状態フラグ : RX (S+A) / エラーリセット要求フラグ : RY (S+A)

自局が中度/重度エラー（自局の動作が継続不可能な異常状態）の発生を通知/解除するために使用します。

自局に中度/重度エラーが発生したときに、エラー状態フラグを ON してください。

マスタ局は、エラー状態を解除しエラーリセット要求フラグを ON します。

自局は、エラー状態フラグを OFF してから、エラー・コード格納エリアをクリアしてください。

エラー発生からエラーリセットまで、リモート Ready を OFF してください。

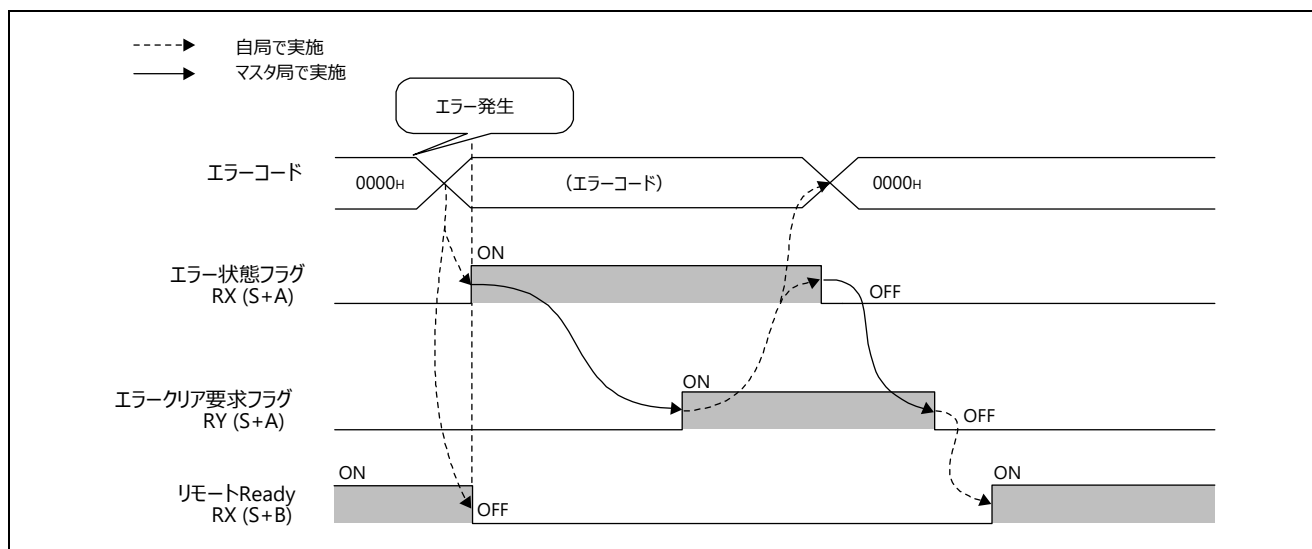


図 6.5 エラー状態/リセット要求フラグ タイミング図

(6) ワーニング状態フラグ : RX (S+7)

自局が軽度エラー（自局の動作が継続可能な異常状態）の発生を通知するために使用します。

自局に軽度エラーが発生したとき、ワーニング状態フラグを ON してください。

マスタ局が発生した軽度エラーの要因を取り除いたとき、自局はワーニングコードをクリアし、ワーニング状態フラグを OFF してください。ワーニング発生からワーニング解除まで、リモート Ready は ON のままにしてください。

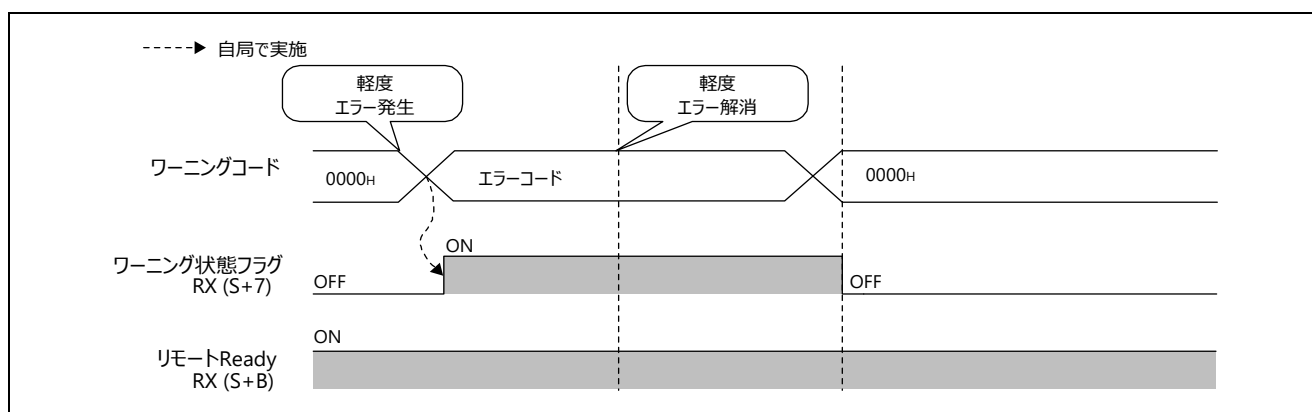


図 6.6 ワーニング状態フラグ タイミング図

7. フレーム・フォーマット

下表のフレームは、ユーザ・プログラムが送信時の作成または受信時の解析を行うフレームです。本章は、フレーム作成または解析の参考として、表 7.1 にフレーム・フォーマットを示します。

表 7.1 ユーザ・プログラムが作成/解析するフレーム一覧

No.	フレーム名称	フレーム種別 (FType)		データ種別 (DataType)		データサブ種別	
1	SLMP	22h	Transient1	05h	ネットワーク共通	0002h	SLMP
2	CC-Link 互換トランジェント	25h	Transient2	04h	CC-Link 互換	—	—
3	TransientAck	23h	TransientAck	注	注	注	注

注. TransientAck は受信したフレームのデータ種別、データサブ種別を使用して確認応答します。

7.1 共通フォーマット

CC-Link IE フィールド・ネットワークのフレームは、IEEE802.3 の Ethernet フレームに準拠しています。
CC-Link IE フィールド・ネットワークのフレームに共通するフォーマットを以下に示します。

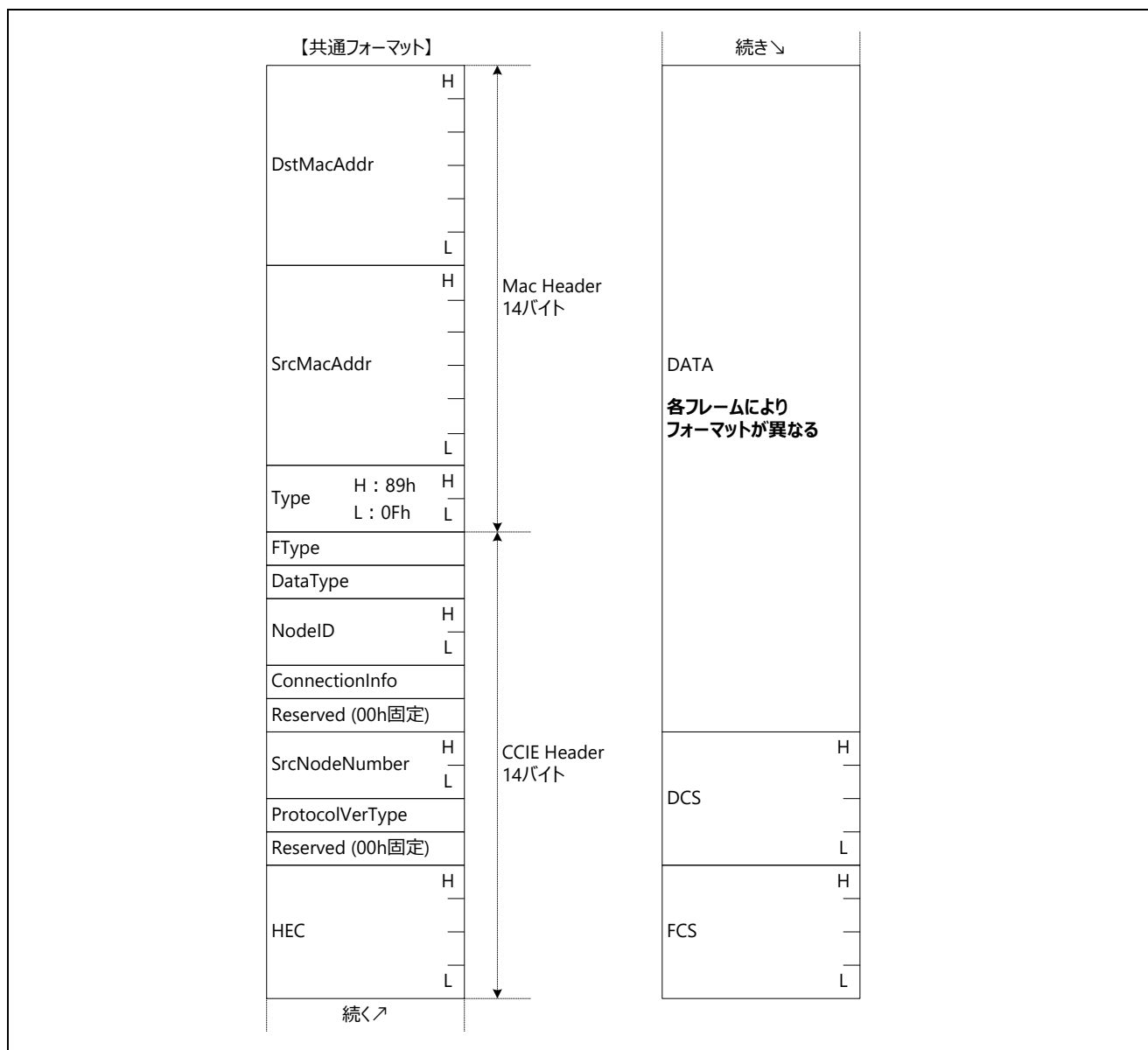


図 7.1 フレーム共通フォーマットの概略

フレーム共通フォーマットで定義されている各項目の詳細を以下に示します。なお、本表の全項目は R-IN32M4-CL3 ドライバが送信時にセットします。

表 7.2 フレーム共通フォーマットの詳細

項目	内容	値	備考
Dst/SrcMacAddr (第 1 オクテット)	宛先/送信元の MAC アドレス	IEEE が管理する値 ^{注1}	MAC アドレスが、01-23-45-67-89-AB のとき、01h
Dst/SrcMacAddr (第 2 オクテット)	宛先/送信元の MAC アドレス		MAC アドレスが、01-23-45-67-89-AB のとき、23h
Dst/SrcMacAddr (第 3 オクテット)	宛先/送信元の MAC アドレス		MAC アドレスが、01-23-45-67-89-AB のとき、45h
Dst/SrcMacAddr (第 4 オクテット)	宛先/送信元の MAC アドレス	ユーザが管理する値 ^{注1}	MAC アドレスが、01-23-45-67-89-AB のとき、67h
Dst/SrcMacAddr (第 5 オクテット)	宛先/送信元の MAC アドレス		MAC アドレスが、01-23-45-67-89-AB のとき、89h
Dst/SrcMacAddr (第 6 オクテット)	宛先/送信元の MAC アドレス		MAC アドレスが、01-23-45-67-89-AB のとき、ABh
Type	フレームタイプ	890Fh 固定 ^{注1}	CC-Link IE フィールド・ネットワークのフレームであることを示します。
FType	フレームの種別	注2	—
DateType	データの種別		—
NodeID	ノード識別子	00h~F0h (0~240) ^{注1}	マスタ局の各スレーブの管理情報 (局番と異なる)
ConnectionInfo	トランジェントの識別情報	01h~FFh (1~255)	「1 回のトークン保持時に送信するトランジェント・フレーム」を識別するための情報です。
SrcNodeNumber	自局番号	0001h~0078h (1~120) ^{注1}	—
ProtocolVerType	bit7-4 プロトコルバージョン	0h 固定	—
	bit3-0 プロトコルの種別	1h 固定	—
HEC	Header Error Control	—	—
DCS	Data Check Sequence	—	—
FCS	Frame Check Sequence	—	—

注 1. ビッグエンディアンで設定してください。

2. 「表 7.1 ユーザ・プログラムが作成/解析するフレーム一覧」を参照してください。

7.2 Transient1 フレーム

Transient1 フレームの基本フォーマットを示します。

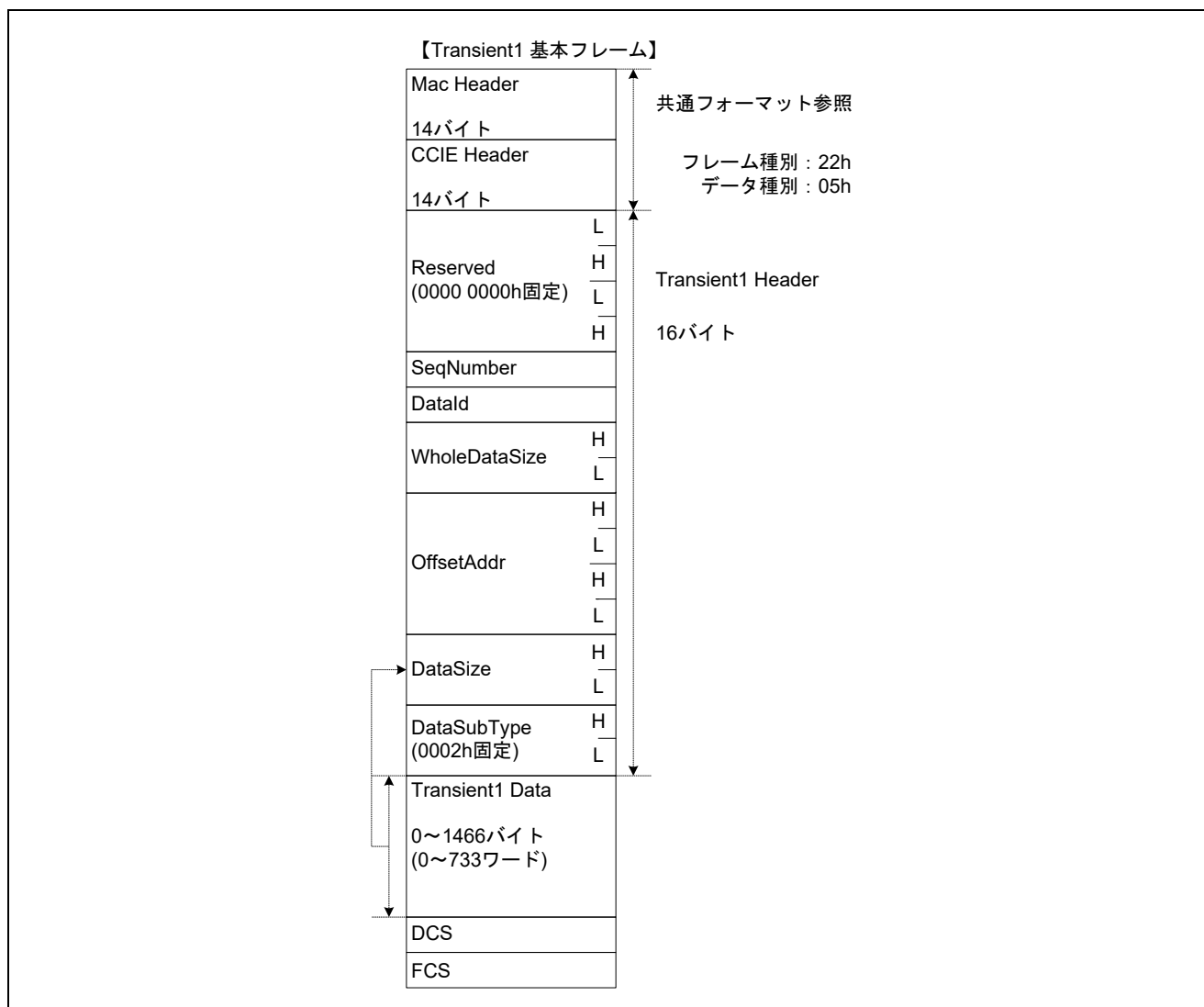


図 7.2 Transient1 フレーム基本フォーマットの概略

Transient1 フレームの基本フォーマットで定義されている各項目の詳細を以下に示します。

表 7.3 Transient1 フレーム基本フォーマットの詳細

項目	内容		値	備考
SeqNumber	bit7	最終フレーム判別	0b : 分割途中フレーム 1b : 分割最終フレーム	トランジェントデータを分割する場合の番号です。
	bit0-6	Transient1 フレームのシーケンシャル番号	00h~7Fh (0~127)	
DataId	トランジェントデータの識別番号		00h~FFh (0~255)	分割後のフレームは同じ識別番号を設定してください。
WholeDataSize	トランジェントデータの全体のサイズ	インテリジェントデバイス局	0000h~0800h (0~2048) 注	分割前のトランジェントデータ全体のサイズ (バイト単位)
		リモートデバイス局	0000h~0400h (0~1024) 注	
OffsetAddr	トランジェントデータ全体の先頭からのオフセットアドレス		0000 0000h~7FFF FFFFh 注	分割しない場合 : 0 固定 分割する場合の先頭フレーム : 0 固定 2 フレーム目からは本フレーム内のトランジェントデータが、全体のトランジェントデータのどこに格納されるかを先頭からのオフセットアドレスで示します。
DataSize	本フレーム内のトランジェントデータのサイズ		0000h~05BAh (0~1466) 注	分割後のトランジェントデータのサイズ (バイト単位)
DataSubType	データサブ種別		0002h : SLMP 注	本書は SLMP のみ記載します。

注. ビッグエンディアンで設定してください。

Transient1 フレームを分割した場合のフォーマットを示します。

なお下図は、トランジェントデータを SLMP^注の場合で、トランジェントデータエリアのアドレスを 1000h ~1800h の場合で例示します。

注. SLMP の詳細は「7.4 SLMP フレーム」を参照してください。

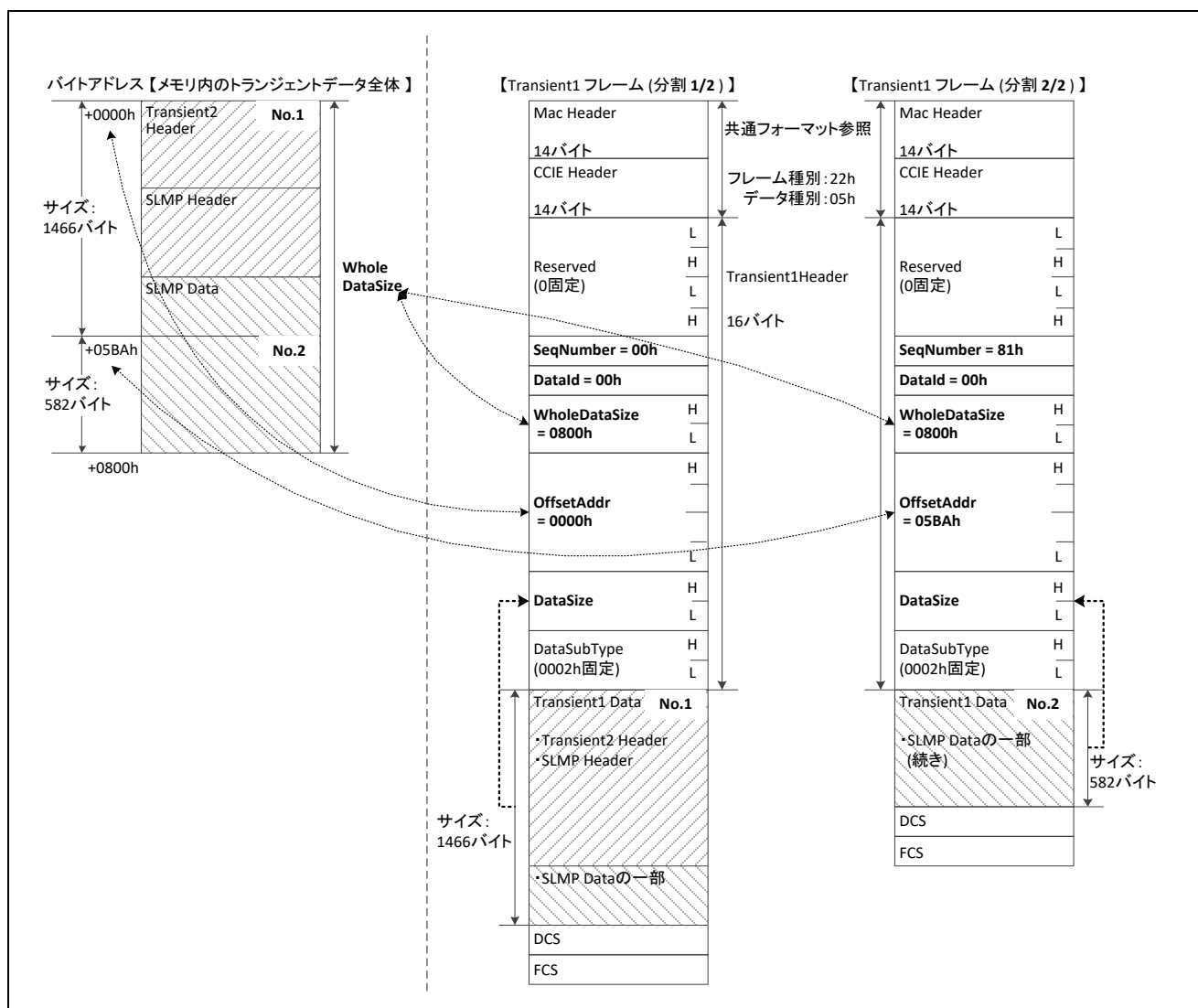


図 7.3 Transient1 フレームの分割例

7.3 CC-Link 互換トランジェント・フレーム

CC-Link 互換トランジェント・フレームの基本フォーマットを示します。

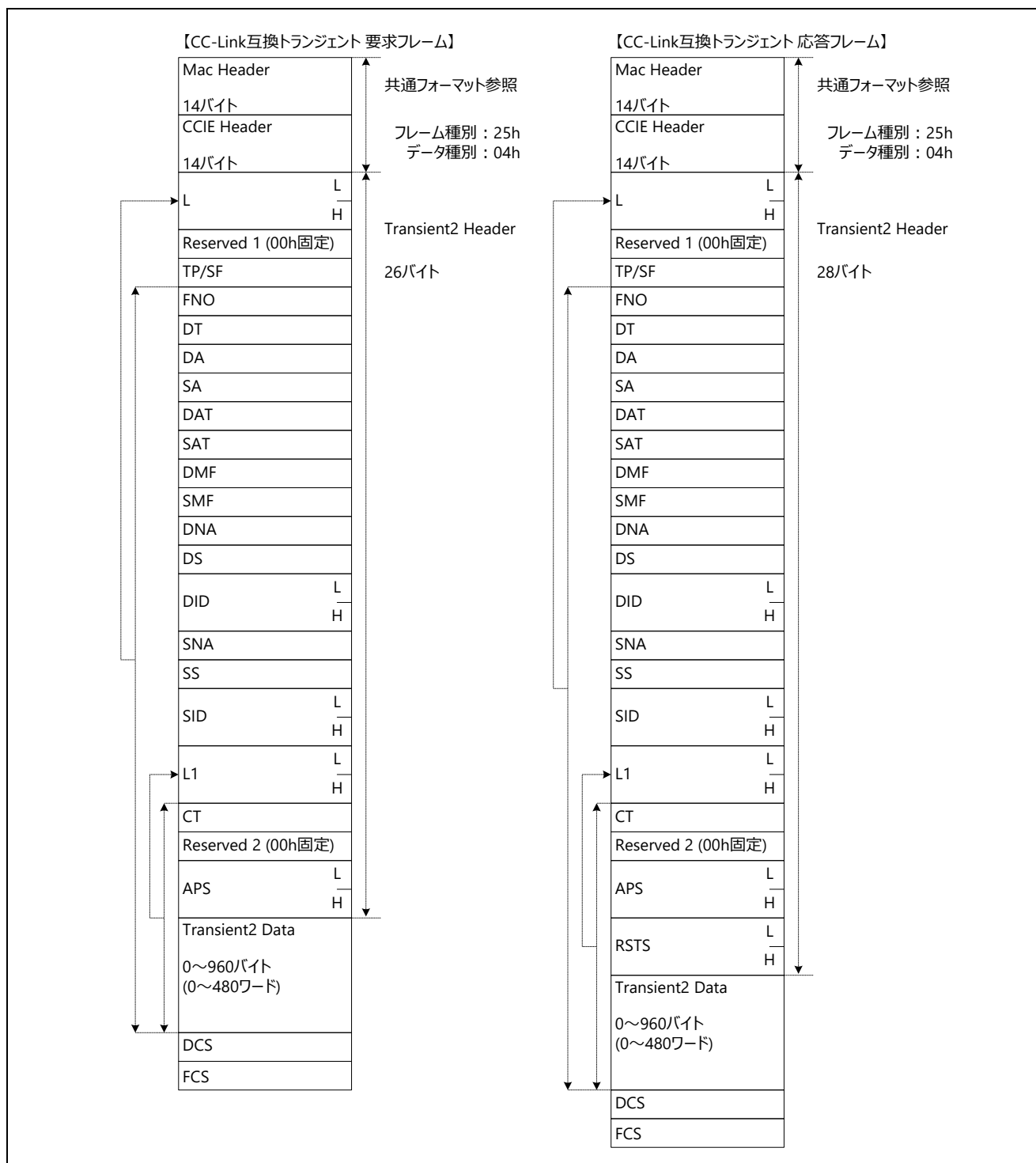


図 7.4 CC-Link 互換トランジェント・フレーム基本フォーマットの概略

7.3.1 CC-Link 互換トランジェント・フレームの詳細

CC-Link 互換トランジェント・フレームの基本フォーマットで定義されている各項目の詳細を以下に示します。

表 7.4 CC-Link 互換トランジェント・フレーム基本フォーマットの詳細

(1/2)

項目	内容	値	備考
L	フレーム長	—	FNO 以降のデータ長をバイト単位で設定してください。
TP/SF	タイプ/シーケンス番号	00h 固定	—
FNO	先頭フレーム識別/分割フレーム番号	00h 固定	—
DT	プライオリティ/応答フレーム要否	00h 固定	—
DA	対象先局番	01h~78h (1~120) : 局番 7Dh : 指定管理局/マスタ局 7Eh : 現在管理局/マスタ局 FFh : グローバル要求	対象局の局番を設定してください。 (DS と同じ値)
SA	起動元局番	01h~78h (1~120) : 局番	実行局の局番を設定してください。 (SS と同じ値)
DAT	対象先アプリケーションタイプ	22h 固定	—
SAT	起動元アプリケーションタイプ	22h 固定	—
DMF	実行モジュール先フラグ	00h : Transient2 03h : SLMP	—
SMF	実行モジュール元フラグ	00h : Transient2 03h : SLMP	—
DNA	対象先ネットワーク No.	01h~EFh (1~239)	対象局の所属するネットワーク No.を設定してください。
DS	対象先局番 02	01h~78h (1~120) : 局番 7Dh : 指定管理局/マスタ局 7Eh : 現在管理局/マスタ局 FFh : グローバル要求	対象局の局番を設定してください。 (DA と同じ値)
DID	対象先識別番号	03FFh 固定	—
SNA	起動元ネットワーク No.	01h~EFh (1~239)	実行局の所属するネットワーク No.を設定してください。
SS	起動元局番 02	01h~78h (1~120) : 局番	実行局の局番を設定してください。 (SA と同じ値)
SID	起動元識別番号	03FFh 固定	—
L1	データ長	—	CT 以降のデータ長をバイト単位で設定してください。
CT	コマンドタイプ	04h~1Fh : Transient2 30h : SLMP の要求 B0h : SLMP の応答	Transient2 のコマンドタイプの詳細は「7.3.2 コマンドタイプ」を参照してください。

表 7.4 CC-Link 互換トランジェント・フレーム基本フォーマットの詳細

(2/2)

項目	内容		値	備考
APS	アプリケーション No.	bit 15-8	00h 固定	起動元が要求送信するときに、何番目のフレームかを識別するための番号を設定してください。
		bit 7-0	00h~FFh	
RSTS	リターン・コード		0000h : 正常 0000h 以外 : エラー・コード	応答時のみリターン・コードの詳細は「7.3.3 リターン・コード (RSTS) 詳細」を参照してください。

7.3.2 コマンドタイプ (CT) 詳細

コマンドタイプ (CT) のデータ構造を、以下に示します。

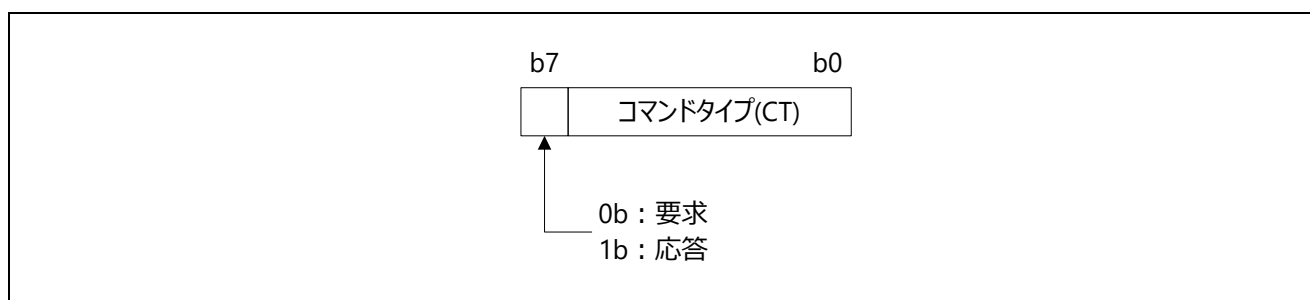


図 7.5 CC-Link 互換トランジェントコマンドタイプのデータ構造

本書に記載する CC-Link 互換トランジェントのコマンドを以下に示します。各コマンドのフレーム・フォーマットは、「7.3.4 メモリアクセス情報取得」以降を参照してください。

表 7.5 CC-Link 互換トランジェントコマンド一覧

コマンドタイプ	内容
04h	メモリアクセス情報取得要求
84h	メモリアクセス情報取得応答
08h	リモート RUN 要求
88h	リモート RUN 応答
09h	リモート STOP 要求
89h	リモート STOP 応答
10h	メモリ読出し要求
90h	メモリ読出し応答
12h	メモリ書込み要求
92h	メモリ書込み応答

7.3.3 リターン・コード (RSTS) 詳細

リターンコード (RSTS) は、クライアントの要求フレームにエラーがあるときに、サーバが応答フレームにエラー・コードを格納する領域です。

【自局がクライアントのとき】

応答フレーム受信処理において、自局が送信した要求フレームに対するエラーのエラー・コードが格納されます。要求先機器 (MELSEC またはパートナーメーカ製品) のユーザーズ・マニュアルを参照して、要求フレームの作成処理または要求送信処理を見直してください。

【自局がサーバのとき】

応答フレーム送信処理において、クライアントの要求フレームに対するエラーのエラー・コードを格納してください。格納するエラー・コードはユーザの任意です。参考として、格納するエラー・コードの例を「表 7.6 リターン・コードに格納するエラー・コード例一覧」に示します。(表中の No.2~7 は、サンプルコードに記述しています。)

表 7.6 リターン・コードに格納するエラー・コード例一覧

No.	エラーコード	内容 (R_IN32M4_Transient.h ファイル内の定義)	処置方法
1	0000h	正常	—
2	D203h	トランジェントデータの読出し、書込みアドレス指定異常 (USER_ERR_WRREQ_ADDRESS_OUTOFRANGE)	トランジェント要求元で読出しアドレス、および書込みアドレス指定を修正後、再度実行してください。
3	D213h	トランジェントデータのコマンド異常 (USER_ERR_WRREQ_COMMAND_OUTOFRANGE)	トランジェント要求元で要求コマンドを修正後、再度実行してください。
4	D218h	トランジェントデータの読出し、書込みデータ数異常 (USER_ERR_WRREQ_SIZE_OUTOFRANGE)	トランジェント要求元で読出し/書込みデータ数を修正後、再度実行してください。
5	D219h	トランジェントデータの属性コード異常 (USER_ERR_WRREQ_ATTRIBUTE_OUTOFRANGE)	トランジェント要求元で属性コードを修正後、再度実行してください。
6	D21Ah	トランジェントデータのアクセス・コード異常 (USER_ERR_WRREQ_ACCESSCODE_OUTOFRANGE)	トランジェント要求元でアクセス・コードを修正後、再度実行してください。
7	D2AEh	トランジェントデータの対象局番異常 (USER_ERR_NOTTOONESELF)	異なるネットワーク/局番宛てのトランジェントデータを受信しました。 ネットワーク No./対象局番号を確認後、再度実行してください。
8	D2A0h	受信バッファフル異常	・ CC-Link IE フィールド・ネットワーク診断にてネットワーク状態を確認してください。 ・ 対象局のトランジェントデータ受信が過負荷の場合は、送信元が任意の時間経過後に送信してください。
9	D2A1h	送信バッファフル異常	・ トランジェント伝送の使用頻度を下げた後に再度実行してください。 ・ 要求元のケーブルおよびスイッチングハブの接続に異常がないか確認してください。
10	D2A3h	トランジェントデータのフレーム長 (L) 異常	Transient2 ヘッダの該当箇所を修正後、再度実行してください。
11	D2A4h	トランジェントデータのリザーブ (RSV) 異常	
12	D2A5h	トランジェントデータの対象局番号 (DA) 異常	
13	D2A6h	トランジェントデータの要求元番号 (SA) 異常	
14	D2A7h	トランジェントデータの対象先アプリケーションタイプ (DAT) 異常	
15	D2A8h	トランジェントデータの要求元アプリケーションタイプ (SAT) 異常	
16	D2A9h	トランジェントデータの対象ネットワーク No. (DNA) 異常	
17	D2AAh	トランジェントデータの対象局番 (DS) 異常	
18	D2ABh	トランジェントデータの要求元ネットワーク No. (SNA) 異常	
19	D2ACh	トランジェントデータの要求元番号 (SS) 異常	
20	D2ADh	トランジェントデータのデータ長 (L1) 異常	

7.3.4 メモリアクセス情報取得

メモリアクセス情報取得コマンド (CT=04h) を実行することで、対象先コントローラなどの使用可能なデバイスとアクセス・コードを知ることができます。メモリアクセス情報取得フレーム・フォーマットを以下に示します。

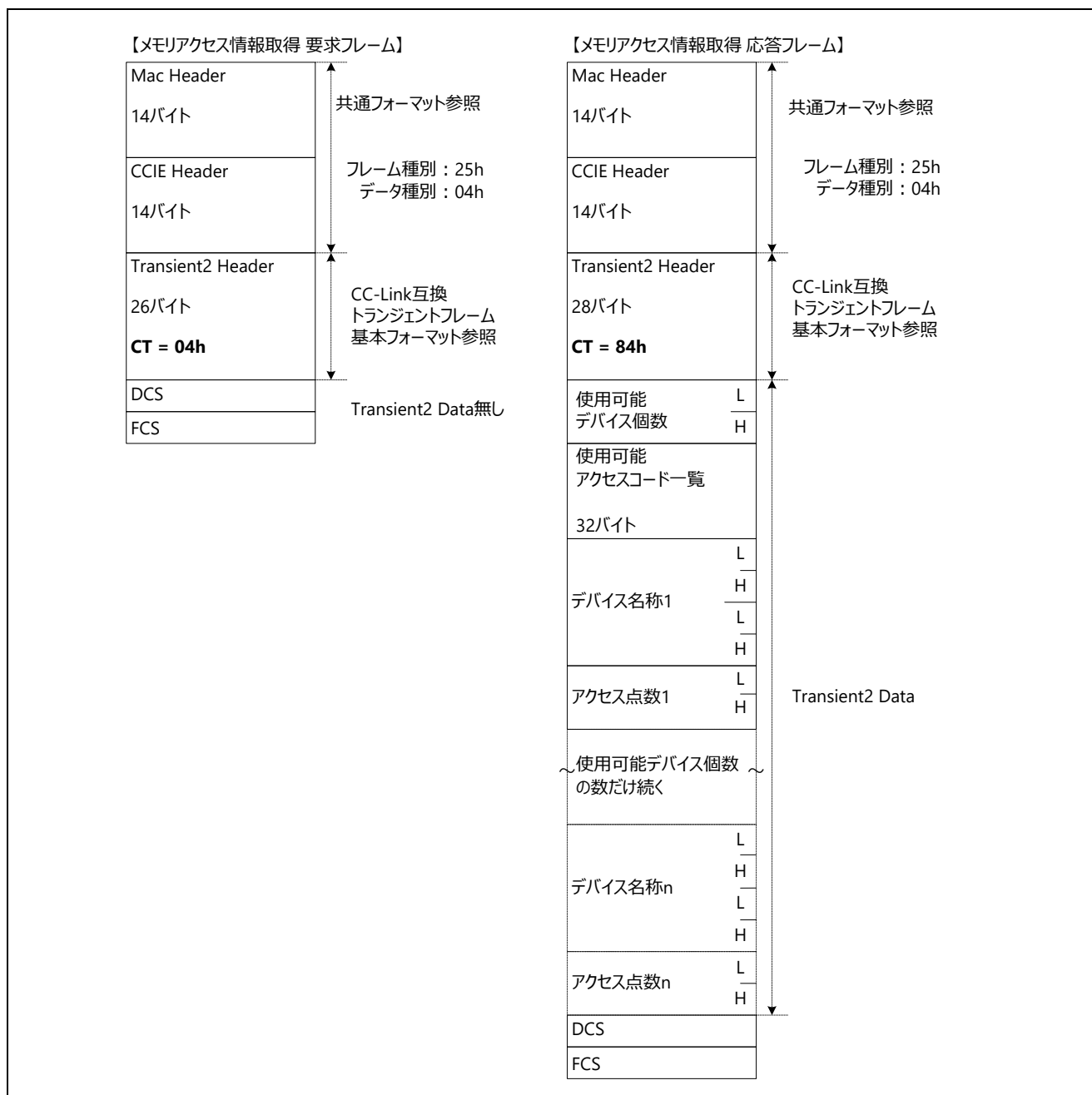


図 7.6 メモリアクセス情報取得フレーム・フォーマットの概略

メモリアクセス情報取得フレーム・フォーマットで定義されている各項目の詳細を以下に示します。

表 7.7 メモリアクセス情報取得フレーム・フォーマットの詳細

項目	内容	値	備考																																																																																																																																																																																																																																																																
使用可能デバイス個数	メモリ読出しコマンド、メモリ書込みコマンドで使用可能なデバイスの個数を格納します。	—	—																																																																																																																																																																																																																																																																
使用可能 アクセス・コード一覧	メモリ読出し/書込みコマンドの対象局コントローラのデバイスが、アクセス可能かどうかを以下の bit パターンに格納します。 デバイスは、00h～FFh のアクセス・コードで表しています。 b15b14b13b12b11b10b9b8b7b6b5b3b3b2b1b0 <table><tr><td>0Fh</td><td>0Eh</td><td>0Dh</td><td>0Ch</td><td>0Bh</td><td>0Ah</td><td>09h</td><td>08h</td><td>07h</td><td>06h</td><td>05h</td><td>04h</td><td>03h</td><td>02h</td><td>01h</td><td>00h</td></tr><tr><td>1Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>10h</td></tr><tr><td>2Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>20h</td></tr><tr><td>3Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>30h</td></tr><tr><td>4Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>40h</td></tr><tr><td>5Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>50h</td></tr><tr><td>6Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>60h</td></tr><tr><td>7Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>70h</td></tr><tr><td>8Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>80h</td></tr><tr><td>9Fh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>90h</td></tr><tr><td>AFh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>A0h</td></tr><tr><td>BFh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>B0h</td></tr><tr><td>CFh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>C0h</td></tr><tr><td>DFh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>D0h</td></tr><tr><td>EFh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>E0h</td></tr><tr><td>FFh</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>F0h</td></tr></table> 0b：使用不可アクセスコード 1b：使用可アクセスコード	0Fh	0Eh	0Dh	0Ch	0Bh	0Ah	09h	08h	07h	06h	05h	04h	03h	02h	01h	00h	1Fh															10h	2Fh															20h	3Fh															30h	4Fh															40h	5Fh															50h	6Fh															60h	7Fh															70h	8Fh															80h	9Fh															90h	AFh															A0h	BFh															B0h	CFh															C0h	DFh															D0h	EFh															E0h	FFh															F0h	—	—
0Fh	0Eh	0Dh	0Ch	0Bh	0Ah	09h	08h	07h	06h	05h	04h	03h	02h	01h	00h																																																																																																																																																																																																																																																				
1Fh															10h																																																																																																																																																																																																																																																				
2Fh															20h																																																																																																																																																																																																																																																				
3Fh															30h																																																																																																																																																																																																																																																				
4Fh															40h																																																																																																																																																																																																																																																				
5Fh															50h																																																																																																																																																																																																																																																				
6Fh															60h																																																																																																																																																																																																																																																				
7Fh															70h																																																																																																																																																																																																																																																				
8Fh															80h																																																																																																																																																																																																																																																				
9Fh															90h																																																																																																																																																																																																																																																				
AFh															A0h																																																																																																																																																																																																																																																				
BFh															B0h																																																																																																																																																																																																																																																				
CFh															C0h																																																																																																																																																																																																																																																				
DFh															D0h																																																																																																																																																																																																																																																				
EFh															E0h																																																																																																																																																																																																																																																				
FFh															F0h																																																																																																																																																																																																																																																				
デバイス名称 1	デバイス名称を格納します。デバイス名称は最大 4 文字のアスキーコードを格納します。	—	—																																																																																																																																																																																																																																																																
アクセス点数 1	対象先コントローラなどが提供するデバイスのアクセスサイズをワード単位で格納します。	—	—																																																																																																																																																																																																																																																																
...	コントローラが使用可能なデバイス個数分繰り返します。	—	—																																																																																																																																																																																																																																																																
デバイス名称 n	デバイス名称を格納します。デバイス名称は最大 4 文字のアスキーコードを格納します。	—	注																																																																																																																																																																																																																																																																
アクセス点数 n	対象先コントローラなどが提供するデバイスのアクセスサイズをワード単位で格納します。	—	注																																																																																																																																																																																																																																																																

注. デバイス名称 n とアクセス点数 n の先頭アドレスは、以下の計算式で求めることができます。

デバイス名称 n : + 3Eh + (06h × n)

アクセス点数 n : + 3Eh + (06h × n) + 04h

アクセス・コードの定義を以下に示します。

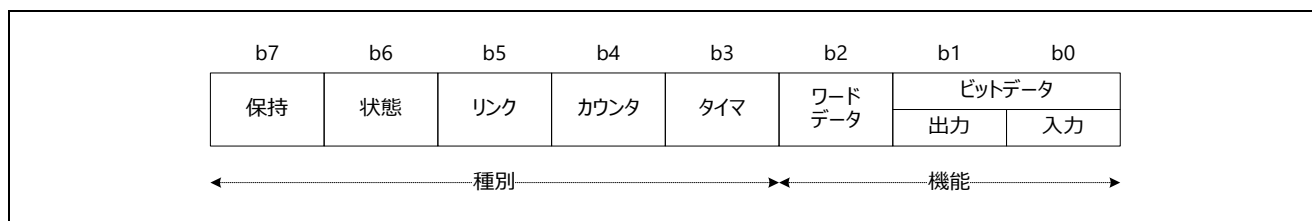


図 7.7 アクセス・コード定義

7.3.5 リモート RUN

リモート RUN コマンド (CT=08h) を実行することで、他局コントローラの状態を RUN にすることができます。リモート RUN フレーム・フォーマットを以下に示します。

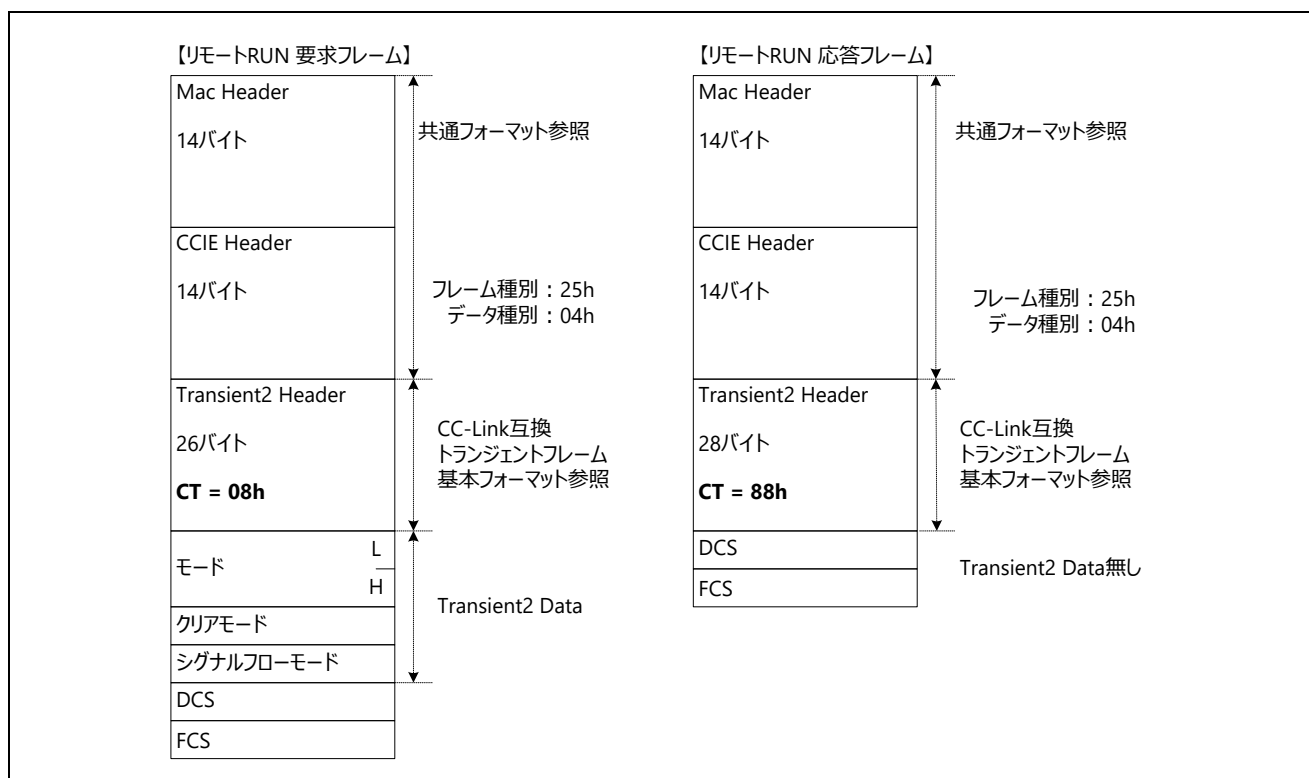


図 7.8 リモート RUN フレーム・フォーマットの概略

リモート RUN フレーム・フォーマットで定義されている各項目の詳細を以下に示します。

表 7.8 リモート RUN フレーム・フォーマットの詳細

項目	内容	値	備考
モード	STOP 状態の他局コントローラを強制的に RUN 状態にするか設定してください。	1: 強制 RUN 3: 通常 RUN	—
クリアモード	リモート RUN 実行時に、コントローラが持つデバイスを初期化するか設定してください。	0: 初期化せず 1: ラッチデバイス以外を初期化 2: すべてのデバイスを初期化	—
シグナルフローモード	クリアモードで設定した初期化の方法を指定してください。	0: コマンド実行前の状態保持 1: すべて 0 (オフ) 2: すべて 1 (オン)	—

7.3.6 リモート STOP

リモート STOP コマンド (CT=09h) を実行することで、他局コントローラの状態を STOP にすることができます。リモート STOP フレーム・フォーマットを以下に示します。

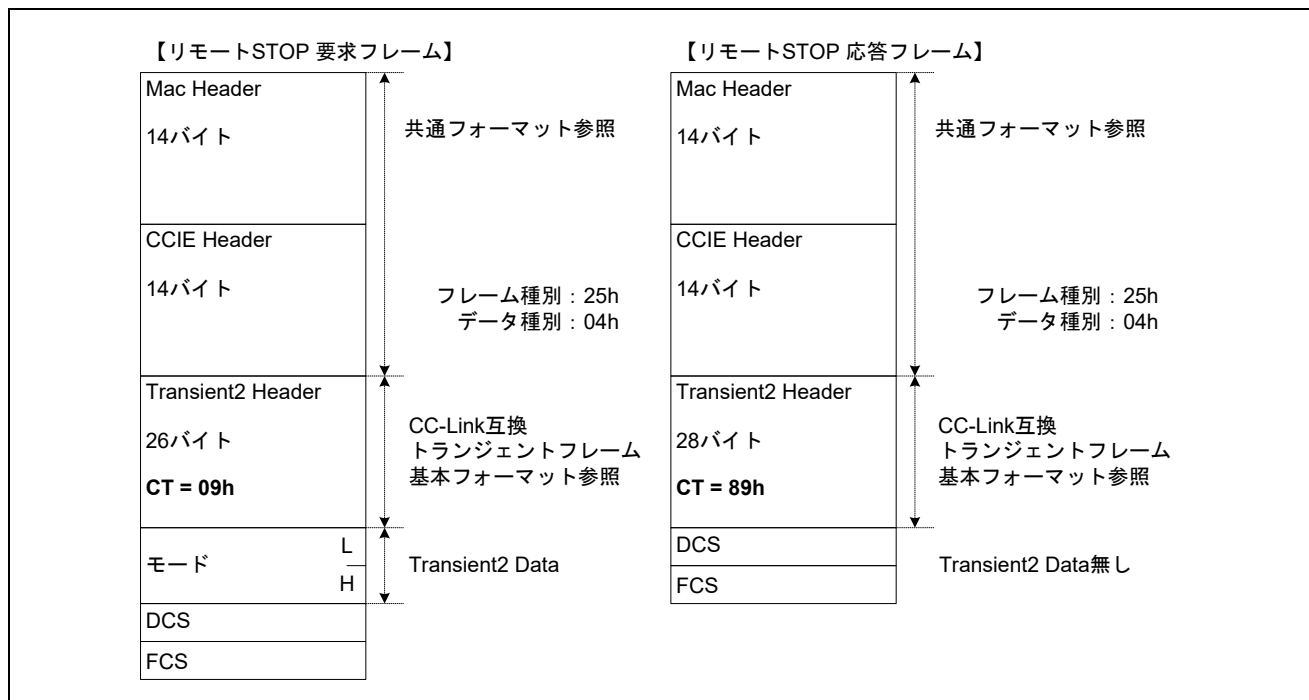


図 7.9 リモート STOP フレーム・フォーマットの概略

リモート STOP フレーム・フォーマットで定義されている各項目の詳細を以下に示します。

表 7.9 リモート STOP フレーム・フォーマットの詳細

項目	内容	値	備考
モード	RUN 状態の他局コントローラを強制的に STOP 状態にするか設定してください。	1: 強制 STOP 3: 通常 STOP	—

7.3.7 メモリ読出し

メモリ読出しコマンド (CT=10h) を実行することで、他局コントローラなどのデバイスから値を読み出すことができます。メモリ読出しフレーム・フォーマットを以下に示します。なお、三菱電機のシーケンサで専用命令 **RIRD** を実行したとき、本フレームが送信されます。

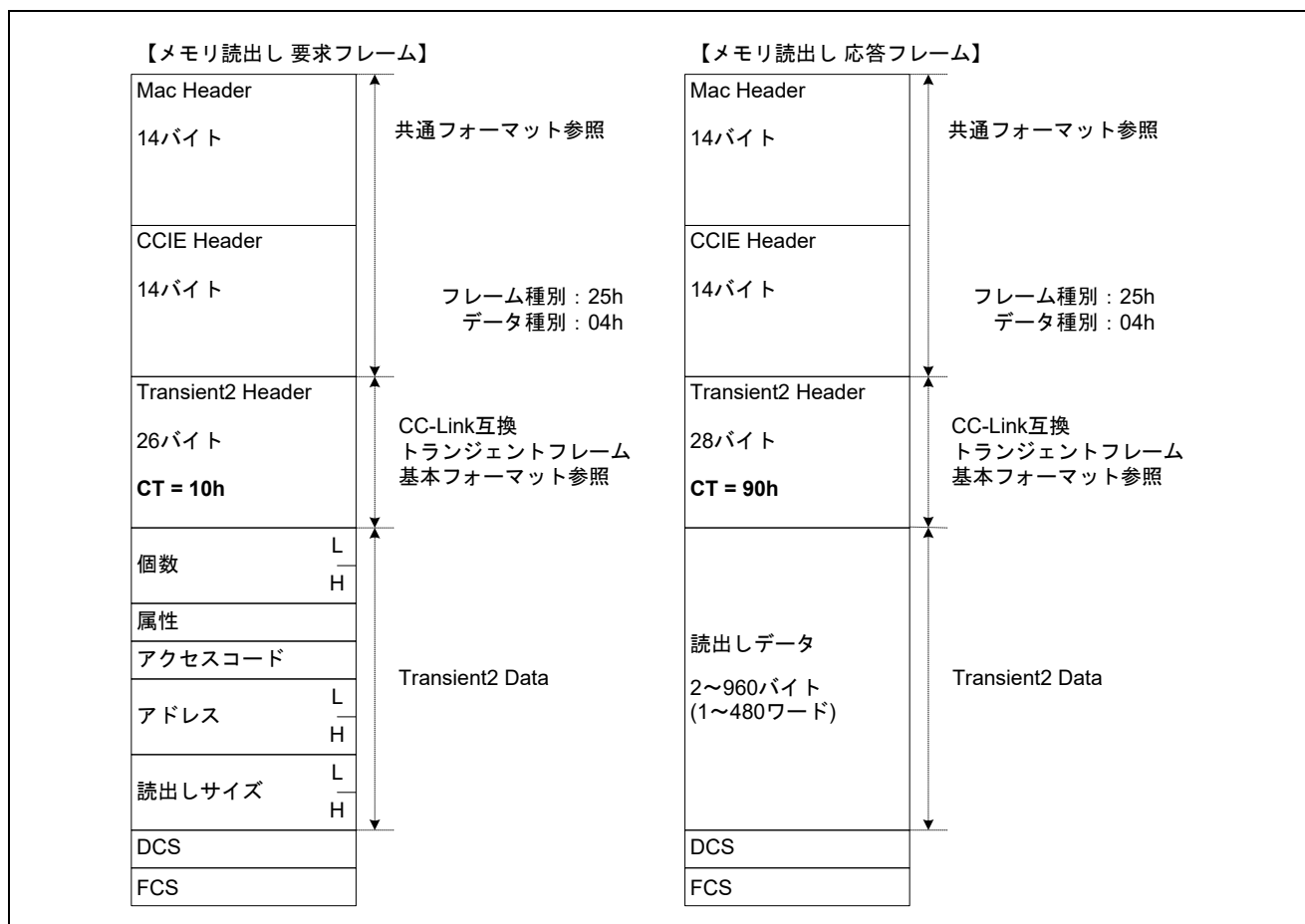


図 7.10 メモリ読出しフレーム・フォーマットの概略

メモリ読出しフレーム・フォーマットで定義されている各項目の詳細を以下に示します。

表 7.10 メモリ読出しフレーム・フォーマットの詳細

項目	内容	値	備考
個数	読み出すデバイスの個数を設定してください。	0001h 固定	—
属性	要求先の対象デバイスの属性を設定してください。	—	詳細は「7.3.9 アクセス・コードと属性」を参照してください。
アクセス・コード	要求先の対象デバイスのアクセス・コードを設定してください。	—	詳細は「7.3.9 アクセス・コードと属性」を参照してください。
アドレス	読み出すデバイスの先頭アドレスを設定してください。（アクセス範囲内の値を設定してください。）	任意のオフセット値	デバイスタイプがビットの場合は 0 または 16 の倍数で指定し、デバイスタイプがバイトの場合は 0 または 2 の倍数で指定してください。
読出しサイズ	読み出すデータのサイズを設定してください。（ワード単位）	1～480 ワード	—

7.3.8 メモリ書込み

メモリ書込みコマンド (CT=12h) を実行することで、他局コントローラなどのデバイスへ値を書込むことができます。メモリ書込みフレーム・フォーマットを以下に示します。三菱電機のシーケンサで専用命令 RIWT を実行したとき、本フレームが送信されます。

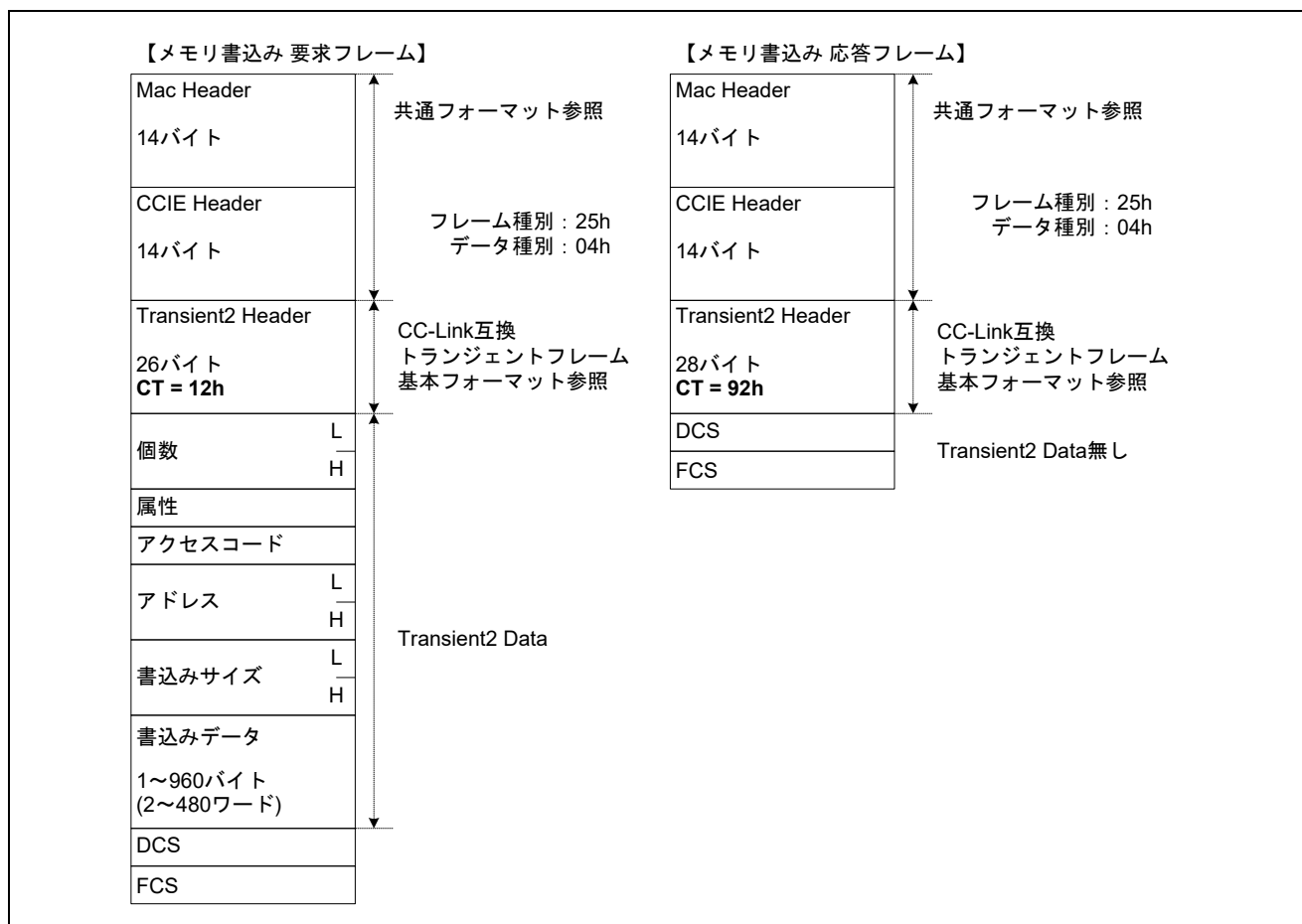


図 7.11 メモリ書込みフレーム・フォーマットの概略

メモリ書込みフレーム・フォーマットで定義されている各項目の詳細を以下に示します。

表 7.11 メモリ書込みフレーム・フォーマットの詳細

項目	内容	値	備考
個数	書き込むデバイスの個数を設定してください。	0001h 固定	—
属性	要求先の対象デバイスの属性を設定してください。	—	詳細は「7.3.9 アクセス・コードと属性」を参照してください。
アクセス・コード	要求先の対象デバイスのアクセス・コードを設定してください。	—	詳細は「7.3.9 アクセス・コードと属性」を参照してください。
アドレス	書き込むデバイスの先頭アドレスを設定してください。(アクセス範囲内の値を設定してください。)	任意のオフセット値	デバイスタイプがビットの場合は 0 または 16 の倍数で指定し、 デバイスタイプがバイトの場合は 0 または 2 の倍数で指定してください。
書込みサイズ	書き込むデータのサイズを設定してください。(ワード単位)	1~480 ワード	—
書込みデータ	書き込むデータを設定してください。	書込みを行う 任意の値	—

7.3.9 アクセス・コードと属性

アクセス・コードと属性の定義を以下に示します。メモリ読出し/書込みコマンドで、他局から自局へアクセスできるように、自局のデバイス/バッファメモリを定義してください。

他局：MELSEC 製品およびユーザが開発した CC-Link ファミリ接続対応製品

自局：R-IN32M4-CL3 適用製品



図 7.12 アクセス・コード定義

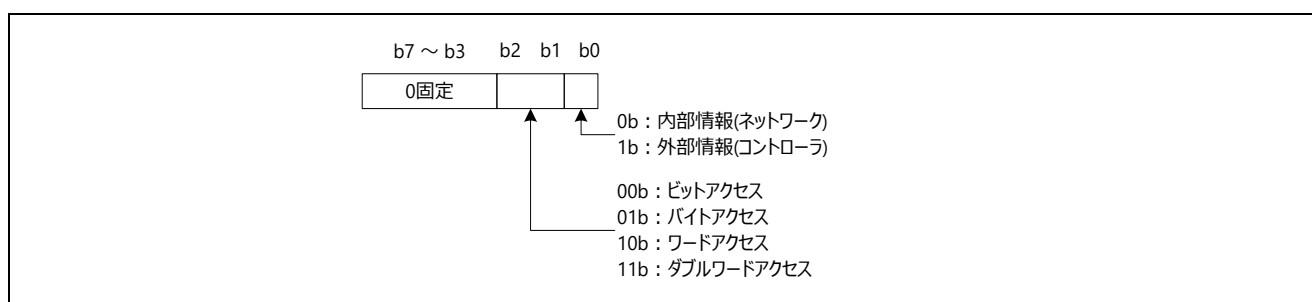


図 7.13 属性定義

MELSEC 製品のアクセス・コードと属性を以下に示します。メモリ読出し/書込みコマンドで、自局から他局（MELSEC 製品）へアクセスする場合、下表を参照してください。デバイスの点数（サイズ）は個々のシーケンサによって異なります。そのため、アクセスできる範囲は、各シーケンサのユーザーズ・マニュアルを参照してください。

表 7.12 MELSEC 製品のアクセス・コード一覧

デバイス内容	名称	デバイスタイプ		単位	アクセス・コード ^{注1}	属性コード ^{注1}
		ビット	ワード			
入力リレー	X	○	—	16 進数	01h	05h
出力リレー	Y	○	—	16 進数	02h	
特殊リレー	SM	○	—	10 進数	43h	
特殊レジスタ	SD	—	○	10 進数	44h	
内部リレー	M	○	—	10 進数	03h	
ラッチリレー	L	○	—	10 進数	83h	
タイマ（接点）	T	○	—	10 進数	09h	
タイマ（コイル）	T	○	—	10 進数	0Ah	
タイマ（現在値）	T	—	○	10 進数	0Ch	
積算タイマ（接点）	ST	○	—	10 進数	89h	
積算タイマ（コイル）	ST	○	—	10 進数	8Ah	
積算タイマ（現在値）	ST	—	○	10 進数	8Ch	
カウンタ（接点）	C	○	—	10 進数	11h	
カウンタ（コイル）	C	○	—	10 進数	12h	
カウンタ（現在値）	C	—	○	10 進数	14h	
データレジスタ	D ^{注2}	—	○	10 進数	04h	
ファイルレジスタ	R	—	○	10 進数	84h	
リンクリレー	B	○	—	16 進数	23h	
リンクレジスタ	W ^{注2}	—	○	16 進数	24h	
リンク特殊リレー	SB	○	—	16 進数	63h	
リンク特殊レジスタ	SW	—	○	16 進数	64h	

注 1. 対象局がマスタ・ローカルユニット以外の場合、アクセス・コード/属性コードは対象局のユーザーズ・マニュアルを参照してください。

2. D65536 以降の拡張データレジスタ、および W10000 以降の拡張リンクレジスタは、指定できません。

7.4 SLMP フレーム

SLMP フレームの基本フォーマットを示します。

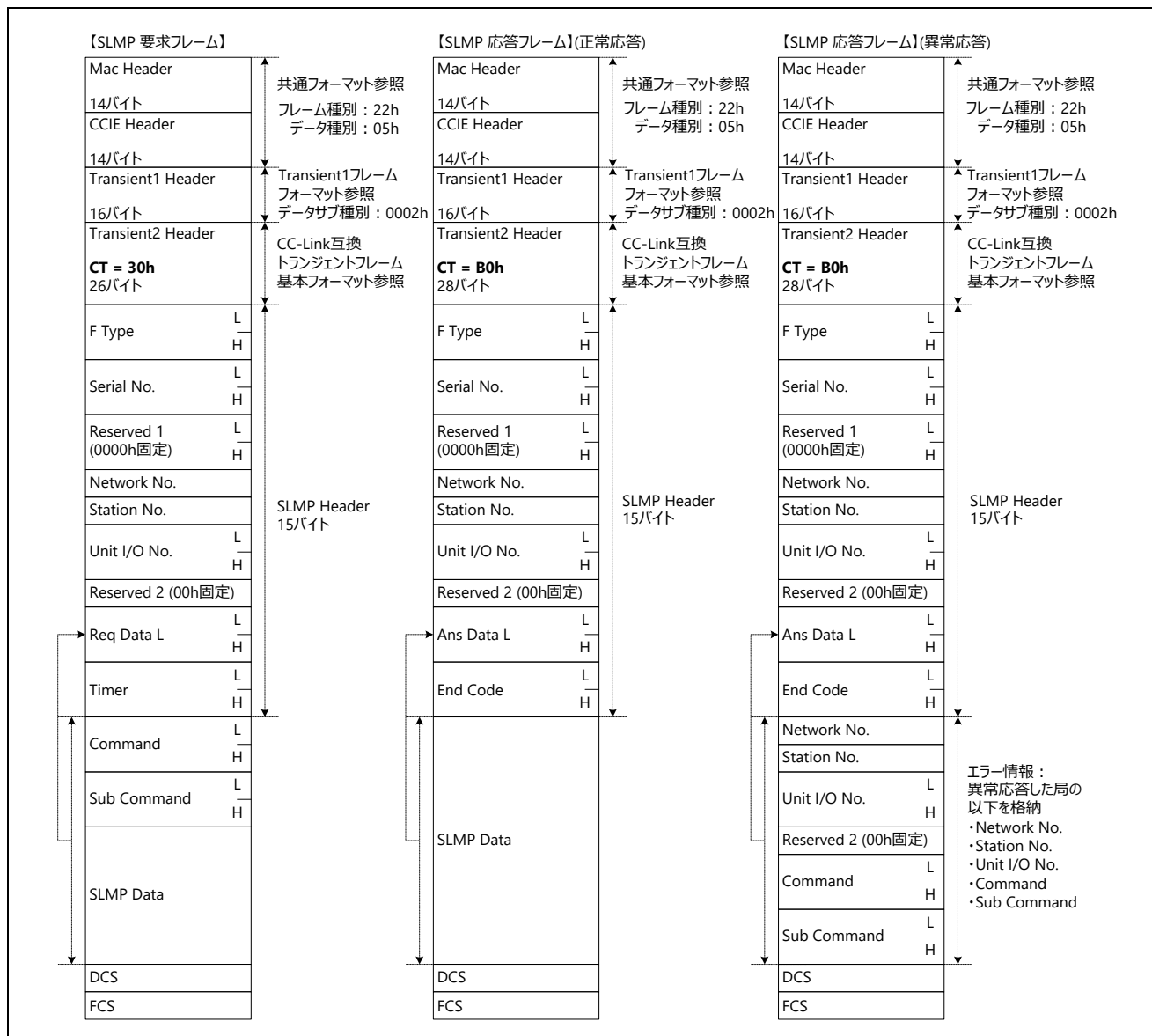


図 7.14 SLMP フレーム基本フォーマットの概略

7.4.1 SLMP フレームの基本フォーマットの詳細

SLMP フレームの基本フォーマットで定義されている各項目の詳細を以下に示します。

表 7.13 SLMP フレーム基本フォーマットの詳細

項目	内容	値	備考
F Type	フレームの種別	0054h : 要求時 00D4h : 応答時	—
Serial No.	シリアル番号	0000h~FFFFh	フレームを区別するための任意の番号を設定してください。要求時と応答時で同値に設定してください。
Network No.	対象先ネットワーク No.	00h : 自局 01h~EFh (1~239) : 他局	対象となる局が所属するネットワーク No.を設定してください。
Station No.	対象先局番	01h~78h (1~120) : 局番 7Dh : 指定管理局/マスタ局 7Eh : 現在管理局/マスタ局 FFh : 自局 ^注	対象となる局番を設定してください。 ^注
Unit I/O No.	対象先ユニット I/O 番号	03FFh : 固定	アクセス先の CPU ユニットを設定してください。
Req Data L	要求データ長	—	要求時の Timer からデータ部の最後までサイズのバイト単位で設定してください。
Ans Data L	応答データ長	—	応答時の End Code からデータ部の最後までサイズのバイト単位で設定してください。
Timer	監視タイマ	0001h~FFFFh 0000h : 無限	要求時のみ。 サーバが応答を返すまでのクライアントの待ち時間を設定してください。(単位 250ms) 推奨値 自局 : 0001h~0028h (0.25~10 秒) 他局 : 0002h~00F0h (0.5~60 秒)
End Code	終了コード	0000h : 正常終了 0000h 以外 : エラー・コード	応答時のみ。 終了コードの詳細は、「7.4.2 終了コード (End Code) 詳細」を参照してください。

注. Network No.が 00h の場合のみ有効

7.4.2 終了コード（End Code）詳細

終了コード（End Code）は、クライアントの要求フレームにエラーがあるときに、サーバが応答フレームにエラー・コードを格納する領域です。

【自局がクライアントのとき】

応答フレーム受信処理において、自局が送信した要求フレームに対するエラーのエラー・コードが格納されます。要求先機器（MELSEC または パートナ・メーカ製品）のユーザーズ・マニュアルを参照して、要求フレームの作成処理または要求送信処理を見直してください。

【自局がサーバのとき】

応答フレーム送信処理において、クライアントの要求フレームに対するエラーのエラー・コードを格納してください。格納するエラー・コードはユーザの任意です。参考として、格納するエラー・コードの例を表 7.16 に示します。

表 7.14 終了コードに格納するエラー・コード例一覧

No.	カテゴリ	エラーコード	内容	処置
1	処理成功	0000h	要求を正常に処理した。	—
2	一般エラー	C059h	・ コマンド・サブコマンドの指定に誤りがある。 ・ 規定されたシーケンス以外のコマンドを受信した。	コマンド・サブコマンドを見直し、再度送信する。
3		C05Ch	要求伝文に誤りがある。	要求内容を見直し、再度送信する。
4		C061h	要求データ長が、データ数と一致しない。	要求データの内容、または要求データ長を見直し、再度送信する。
5		CEE0h	他の要求が実行中のため、要求を処理できない。	しばらく待った後、再度要求を送信する。
6		CEE1h	要求伝文サイズが処理可能な範囲を超えた。	要求内容を見直し、再度送信する。
7		CEE2h	応答伝文サイズが処理可能な範囲を超えた。	要求内容を見直し、再度送信する。
8	サーバ情報	CF10H	指定されたサーバ情報 No.が存在しない。	サーバ情報 No.を見直し、再度送信する。
9	通信設定	CF20H	設定できない項目が要求伝文に含まれている。	設定項目（CSP+）を見直し、再度送信する。
10	パラメータ設定	CF30H	指定されたパラメータ ID は存在しない。	パラメータ、パラメータ ID（CSP+）を見直し、再度送信する。
11		CF31H	書込み排他開始処理が行われていないため、要求を処理できない。	書込み排他処理を実行し、再度送信する。
12	通信状態	CF70H	中継先の通信経路に異常が発生したため、要求は処理できない。	通信経路を確認し、再度送信する。
13		CF71H	タイムアウトが発生したため、処理を中断した。	対象機器の状態を確認し、再度送信する。

7.4.3 SLMP メモリ読出しフレーム

SLMP メモリ読出しコマンド (Command=0613h、Sub Command=0000h) を実行することで、他局 (SLMP 対応機器) のバッファメモリからデータを読み出すことができます。SLMP メモリ読出しフレーム・フォーマットを以下に示します。

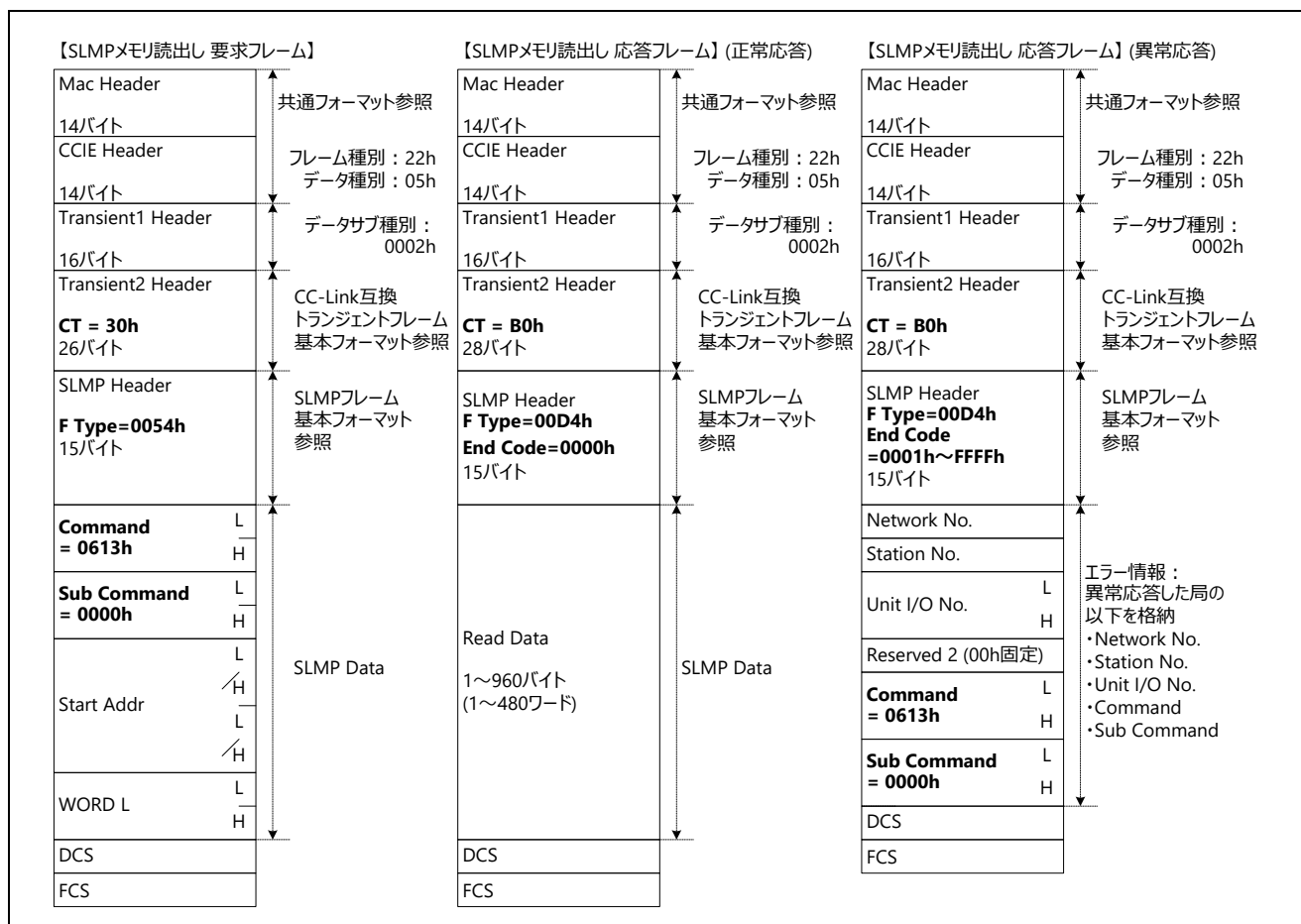


図 7.15 SLMP メモリ読出しフレーム・フォーマットの概略

SLMP メモリ読出しフレーム・フォーマットで定義されている各項目の詳細を以下に示します。

表 7.15 SLMP メモリ読出しフレーム・フォーマットの詳細

項目	内容	値	備考
Command	コマンド	0613h 固定	—
Sub Command	サブコマンド	0000h 固定	—
Start Addr	先頭アドレス	—	読み出すバッファメモリの先頭アドレスを指定してください。
Word L	ワード長	1h~1E0h (1~480)	読み出すバッファメモリのワード長を指定してください。

7.4.4 SLMP メモリ書き込みフレーム

SLMP メモリ書き込みコマンド (Command=1613h、Sub Command=0000h) を実行することで、他局 (SLMP 対応機器) のバッファメモリにデータを書き込むことができます。SLMP メモリ書き込みフレーム・フォーマットを以下に示します。

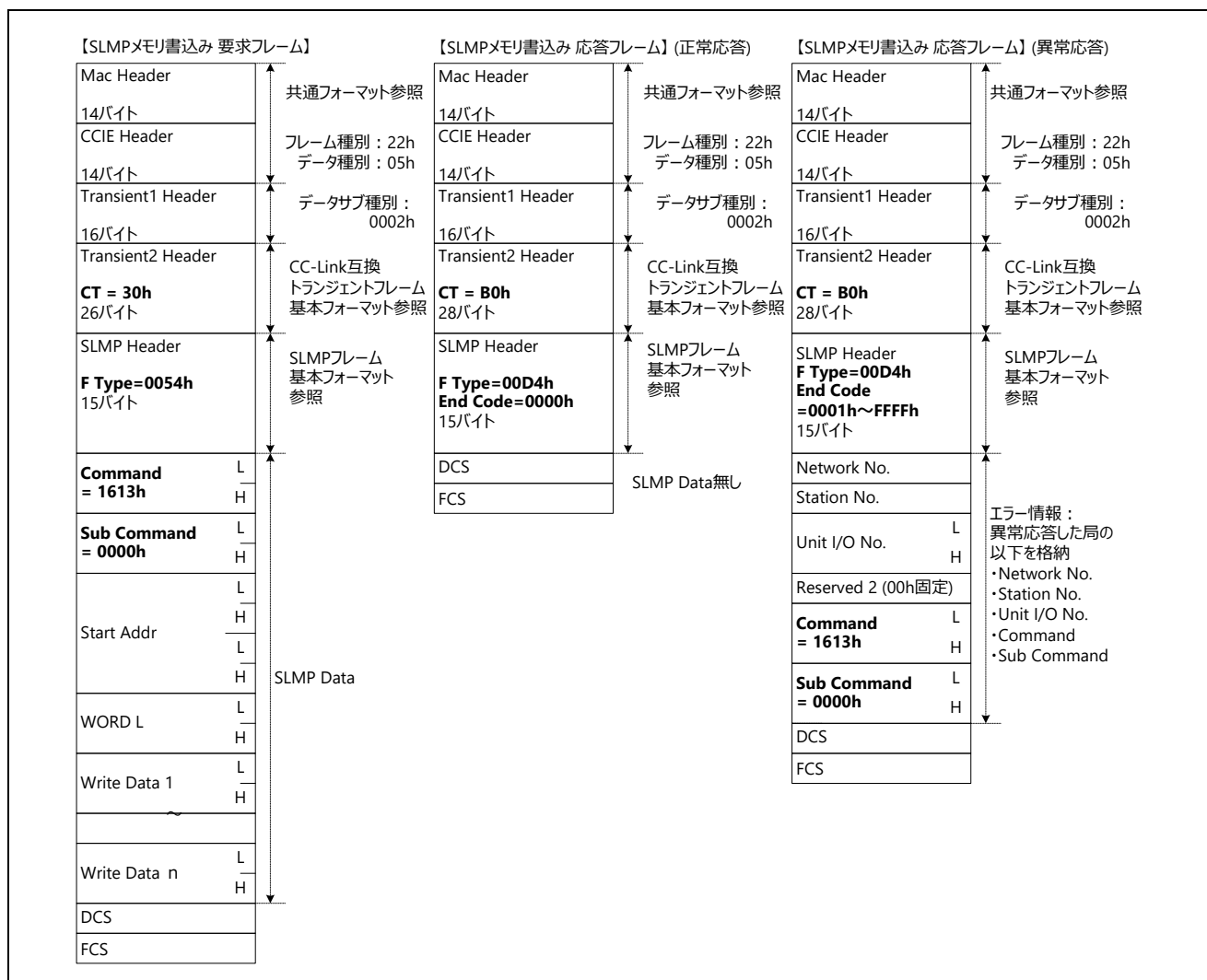


図 7.16 SLMP メモリ書き込みフレーム・フォーマットの概略

SLMP メモリ書き込みフレーム・フォーマットで定義されている各項目の詳細を以下に示します。

表 7.16 SLMP メモリ書き込みフレーム・フォーマットの詳細

項目	内容	値	備考
Command	コマンド	1613h 固定	—
Sub Command	サブコマンド	0000h 固定	—
Start Addr	先頭アドレス	—	書き込むバッファメモリの先頭アドレスを指定してください。
Word L	ワード長	1h~1E0h (1~480)	書き込むバッファメモリのワード長を指定してください。

7.4.5 TransientAck フレーム

TransientAck フレームのフォーマットを示します。

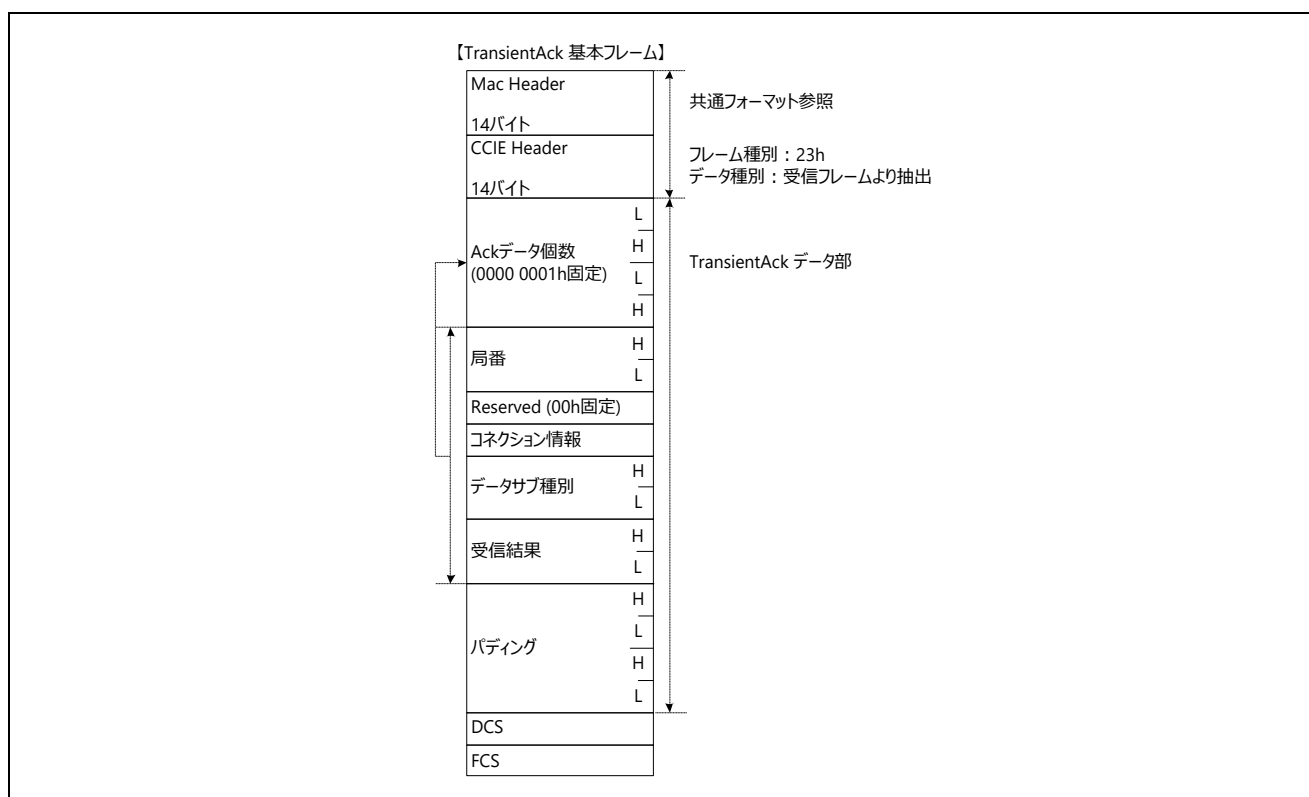


図 7.17 TransientAck フレーム・フォーマットの概略

TransientAck フレーム・フォーマットで定義されている各項目の詳細を以下に示します。

表 7.17 TransientAck フレーム・フォーマットの詳細

項目	内容	値	備考
Ack データ個数	局番～受信結果までの個数	固定値：0000 0001h	—
局番	TransientAck フレーム送信先局番	受信したトランジェント・フレームの送信元局番	注 1
コネクション情報	Ack 送信対象フレームのコネクション情報折り返し値 (ConnectionInformation)	受信したトランジェント・フレームのコネクション情報	—
データサブ種別	受信したトランジェント・フレームのデータサブ種別	Transient1：0002h Transient2：0000h	注 2
受信結果	トランジェント・フレームの受信結果 (RET)	正常：0000h 異常：0000h 以外	—
パディング	パディング 16 バイト	—	注 3

注 1. マスタ局（送信元局番が 0000h）からトランジェント・フレームを受信した場合は、送信先局番を 007Dh に変換して設定してください。

2. Transient2 フレームはデータサブ種別がないため、Transient2 フレームに対する確認応答時は 0000h を設定します。

3. Ethernet フレーム最小サイズ 64 バイトを満たすため、R-IN32M4-CL3 が自動的にパディングします。

改訂記録	R-IN32M4-CL3 ユーザーズ・マニュアル CC-Link IE Field 編
------	---

Rev.	発行日	改訂内容	
		ページ	ポイント
1.00	2019.11.1	－	初版発行

[メ モ]

R-IN32M4-CL3 ユーザーズ・マニュアル
CC-Link IE Field編

発行年月日 2019年11月 1日 Rev.1.00

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 Field 編