

# R-IN32M3 シリーズ

ユーザーズマニュアル

(CC-Link IE Field インテリジェントデバイス局編)

- ・ R-IN32M3-CL

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 : R18UZ0014JJ0500

発行年月 : 2018.1.31

## ご注意書き

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

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

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

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

6. 当社製品をご使用の際は、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他の保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
8. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。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, AMBA, ARM Cortex, ThumbおよびARM Cortex-M3はARM LimitedのEUおよびその他の国における商標および登録商標です。

○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。

○EtherCATは、Beckhoff Automation GmbH, Germanyの登録商標です。

○CC-Link及びCC-Link IE Fieldは、CC-Link協会（CC-Link Partner Association : CLPA）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。

○リアルタイムOSアクセラレータ及びハードウェア・リアルタイムOSはカーネロンシリコン社製IP「ARTESSO技術」で使用されているハードウェア・リアルタイムOSを採用しています。

# 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-IN32M3-CL」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、論理回路、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-IN32M3に関する資料

| 資料名   | 資料番号            |
|---|-----------------|
| R-IN32M3 シリーズ データシート                          | R18DS0007JJ**** |
| R-IN32M3-CL ユーザーズ・マニュアル                       | R18UZ0004JJ**** |
| R-IN32M3 シリーズ周辺機能編                            | R18UZ0006JJ**** |
| R-IN32M3 シリーズ プログラミング・マニュアル ドライバ編             | R18UZ0008JJ**** |
| R-IN32M3 シリーズ プログラミング・マニュアル OS 編              | R18UZ0010JJ**** |
| R-IN32M3 シリーズ CC-Link IE Field インテリジェントデバイス局編 | このマニュアル         |

## 2. 数や記号の表記

データ表記の重み：左が上位桁、右が下位桁

アクティブ・ローの表記：

xxxZ (端子、信号名称のあとにZ)

またはxxx\_N (端子、信号名称のあとに\_N)

またはxxnx (端子、信号名称にnを含む)

注：

本文中につけた注の説明

注意：

気をつけて読んでいただきたい内容

備考：

本文の補足説明

数の表記：

2 進数 … xxxx, xxxxB または n'bxxxx(nビット)

10 進数 … xxxx

16 進数 … xxxxH または n'hxxxx(nビット)

2のべき数を示す接頭語 (アドレス空間、メモリ容量)：

K (キロ) …  $2^{10} = 1024$

M (メガ) …  $2^{20} = 1024^2$

G (ギガ) …  $2^{30} = 1024^3$

データ・タイプ：

ワード … 32 ビット

ハーフワード … 16 ビット

バイト … 8 ビット

### 【備考】

- 通信速度を表す場合は  $10^6$  を示します。

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

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

# 目次

|  |    |
|--|----|
| 1. 概要.....                               | 1  |
| 1.1 R-IN32M3-CLの性能仕様 .....               | 1  |
| 1.2 開発の準備 .....                          | 2  |
| 1.2.1 MACアドレスの取得 .....                   | 3  |
| 1.2.2 ベンダーコードの取得と機種タイプの選択.....           | 3  |
| 1.2.3 ノード番号・ネットワーク番号の書込み方法.....          | 3  |
| 1.2.4 1000BASE-Tコンプライアンステストモードの実装検討..... | 5  |
| 1.2.5 エンジニアリングツール各種機能への対応準備.....         | 6  |
| 1.2.6 エラー状態、エラーコードの仕様検討.....             | 7  |
| 1.2.7 リンクデバイスの属性の割り当て検討.....             | 8  |
| 1.2.8 Hold/Clear処理の実装検討.....             | 9  |
| 1.2.9 CSP+の作成準備.....                     | 10 |
| 1.2.10 コンフォーマンステスト .....                 | 12 |
| 2. R-IN32M3-CLの機能.....                   | 13 |
| 2.1 通信機能 .....                           | 14 |
| 2.2 状態表示機能 .....                         | 14 |
| 2.3 割込み .....                            | 15 |
| 2.4 バイパスモード .....                        | 16 |
| 2.5 MIB情報.....                           | 16 |
| 2.6 CC-Link IEフィールドネットワーク診断機能.....       | 17 |
| 2.6.1 選択局通信状態モニタのLED.....                | 18 |
| 3. 基板設計上の注意事項.....                       | 19 |
| 3.1 部品選定 .....                           | 19 |
| 3.2 回路設計 .....                           | 20 |
| 3.3 パターン設計 .....                         | 20 |
| 4. 状態表示機能 .....                          | 21 |
| 4.1 LEDによる状態表示.....                      | 21 |
| 4.1.1 User LED 1とUser LED 2について .....    | 22 |
| 4.2 LEDの制御.....                          | 22 |
| 4.2.1 LEDの制御概要.....                      | 22 |
| 4.2.2 User LED 1とUser LED 2の制御.....      | 23 |
| 4.2.3 L ERR. LEDの制御 .....                | 24 |

|        |   |     |
|--------|---|-----|
| 4.3    | LEDの有効/無効.....                              | 24  |
| 5.     | CC-Link IE フィールドネットワークのデータ通信方式.....         | 25  |
| 5.1    | サイクリック伝送概要.....                             | 25  |
| 5.2    | トランジェント伝送概要.....                            | 27  |
| 5.2.1  | Transient1要求の受信手順.....                      | 30  |
| 5.2.2  | Transient1要求の送信手順.....                      | 31  |
| 5.2.3  | Transient2要求の受信手順.....                      | 32  |
| 5.2.4  | Transient2要求の送信手順.....                      | 33  |
| 5.3    | トランジェント伝送のフレームフォーマット概要.....                 | 34  |
| 5.3.1  | トランジェントフレーム共通フォーマット.....                    | 34  |
| 5.3.2  | CC-Link IEフィールド固有トランジェントフレームフォーマット.....     | 38  |
| 5.3.3  | TransientAckフレームフォーマット.....                 | 54  |
| 5.3.4  | CC-Link互換トランジェントフレームフォーマット.....             | 56  |
| 5.3.5  | SLMPフレームフォーマット.....                         | 69  |
| 5.4    | MyStatus概要.....                             | 75  |
| 5.4.1  | MyStatus送信.....                             | 75  |
| 5.4.2  | MyStatus受信.....                             | 76  |
| 6.     | ファームウェアの開発.....                             | 77  |
| 6.1    | 開発の手順.....                                  | 77  |
| 6.1.1  | サンプルコードのファイルリスト.....                        | 79  |
| 6.2    | サンプルフローチャート.....                            | 81  |
| 6.2.1  | メイン処理.....                                  | 83  |
| 6.2.2  | 初期化処理.....                                  | 85  |
| 6.2.3  | 通信開始処理.....                                 | 87  |
| 6.2.4  | PHYチェック処理.....                              | 88  |
| 6.2.5  | PHY設定変更処理.....                              | 89  |
| 6.2.6  | 自局エラー処理.....                                | 90  |
| 6.2.7  | サイクリック伝送停止処理.....                           | 90  |
| 6.2.8  | イベント処理.....                                 | 91  |
| 6.2.9  | マスタ局からのMyStatusおよびサイクリック受信処理.....           | 92  |
| 6.2.10 | MyStatus送信処理.....                           | 94  |
| 6.2.11 | サイクリック送信処理.....                             | 94  |
| 6.2.12 | 通信状態更新処理.....                               | 95  |
| 6.2.13 | サイクリック伝送状態更新処理.....                         | 97  |
| 6.2.14 | MIB情報取得処理.....                              | 98  |
| 6.2.15 | Transient1、Transient2、TransientAck受信処理..... | 100 |
| 6.2.16 | Transient2要求フレーム作成処理.....                   | 102 |

|        |   |     |
|--------|---|-----|
| 6.2.17 | Transient1、Transient2、TransientAck送信処理..... | 103 |
| 6.2.18 | Transient1受信データ処理.....                      | 106 |
| 6.2.19 | Transient1受信データ組立て開始処理.....                 | 111 |
| 6.2.20 | Transient1受信データ組立て処理.....                   | 112 |
| 6.2.21 | ノード情報配信フレーム受信対応処理.....                      | 113 |
| 6.2.22 | ノード情報配信フレームチェック処理.....                      | 114 |
| 6.2.23 | 統計情報取得要求フレーム受信対応処理.....                     | 115 |
| 6.2.24 | 統計情報取得応答フレーム作成処理.....                       | 116 |
| 6.2.25 | ノード詳細情報取得要求フレーム受信対応処理.....                  | 117 |
| 6.2.26 | ノード詳細情報取得応答フレーム作成処理.....                    | 118 |
| 6.2.27 | オプション情報取得要求フレーム受信対応処理.....                  | 119 |
| 6.2.28 | 選択局情報取得要求フレーム受信対応処理.....                    | 120 |
| 6.2.29 | 交信テスト要求フレーム受信対応処理.....                      | 121 |
| 6.2.30 | ケーブルテスト要求フレーム受信対応処理.....                    | 122 |
| 6.2.31 | Transient2受信データ処理.....                      | 123 |
| 6.2.32 | Transient2受信データチェック処理.....                  | 125 |
| 6.2.33 | TransientAck受信データ処理.....                    | 125 |
| 6.2.34 | TransientAckフレーム作成処理.....                   | 126 |
| 6.2.35 | Transient2応答フレーム作成処理.....                   | 127 |
| 6.2.36 | Transient2メモリ読出し要求フレーム作成処理.....             | 128 |
| 6.2.37 | Transient2メモリ書込み要求受信処理.....                 | 130 |
| 6.2.38 | Transient2メモリ読出し応答受信処理.....                 | 131 |
| 6.2.39 | SLMPメモリ読出し要求フレーム受信対応処理.....                 | 132 |
| 6.2.40 | SLMPメモリ書込み要求フレーム受信対応処理.....                 | 133 |
| 6.2.41 | SLMPメモリ読出し要求フレーム作成処理.....                   | 134 |
| 6.2.42 | Transient1要求送信分割判定処理.....                   | 135 |
| 6.2.43 | Transient1要求フレーム作成処理.....                   | 136 |
| 6.2.44 | SLMPメモリ読出し応答受信処理.....                       | 137 |
| 6.2.45 | ハードウェアテスト（IEEE802.3abコンプライアンステスト）.....      | 138 |
| 6.2.46 | ハードウェアテスト（折り返し通信テスト）.....                   | 139 |
| 6.3    | R-IN32M3-CLドライバインタフェース関数一覧.....             | 141 |
| 6.4    | R-IN32M3-CLドライバインタフェース関数詳細.....             | 143 |
| 6.4.1  | 初期設定.....                                   | 144 |
| 6.4.2  | ウォッチドグタイマ.....                              | 154 |
| 6.4.3  | イベント.....                                   | 156 |
| 6.4.4  | サイクリック伝送.....                               | 159 |
| 6.4.5  | 自局状態設定.....                                 | 162 |
| 6.4.6  | 自局状態取得.....                                 | 163 |
| 6.4.7  | LED制御.....                                  | 169 |

|        |                                      |     |
|--------|--------------------------------------|-----|
| 6.4.8  | ネットワーク時刻 .....                       | 173 |
| 6.4.9  | MDIOアクセス.....                        | 175 |
| 6.4.10 | トランジェント受信処理 .....                    | 177 |
| 6.4.11 | トランジェント送信処理 .....                    | 179 |
| 6.4.12 | 割込み .....                            | 188 |
| 6.4.13 | ハードウェアテスト .....                      | 189 |
| 6.5    | R-IN32M3-CLドライバターゲット依存関数のカスタマイズ..... | 191 |
| 6.5.1  | ヘッダファイルの変更 .....                     | 191 |
| 6.5.2  | R-IN32M3-CLドライバターゲット依存関数の作成.....     | 192 |
| 6.6    | R-IN32M3-CLドライバコールバック関数のカスタマイズ.....  | 195 |
| 7.     | リンクデバイスのシステム領域 .....                 | 198 |
| 7.1    | システム領域詳細 .....                       | 199 |

## 図の目次

|       |  |    |
|-------|--|----|
| 図1.1  | 開発の工程例 .....                                     | 2  |
| 図1.2  | ノード番号およびネットワーク番号の書込みイメージ .....                   | 4  |
| 図1.3  | CSP+ファイルのセクション構成 .....                           | 10 |
| 図1.4  | 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 .....        | 11 |
| 図2.1  | 診断画面の表示/操作箇所 (GX Works2の場合) .....                | 17 |
| 図2.2  | 選択局通信状態モニタ表示例 .....                              | 18 |
| 図4.1  | L ERR.点灯用外部AND論理 .....                           | 24 |
| 図5.1  | サイクリックデータの流れ .....                               | 26 |
| 図5.2  | トランジェントデータの流れ .....                              | 27 |
| 図5.3  | トランジェント クライアント・サーバ機能 .....                       | 27 |
| 図5.4  | Transient1要求の受信手順 .....                          | 30 |
| 図5.5  | Transient1要求の送信手順 .....                          | 31 |
| 図5.6  | Transient2要求の受信手順 .....                          | 32 |
| 図5.7  | Transient2要求の送信手順 .....                          | 33 |
| 図5.8  | トランジェントフレーム共通フォーマット .....                        | 35 |
| 図5.9  | MACアドレス I/Gビット .....                             | 36 |
| 図5.10 | CC-Link IEフィールド固有トランジェントフレームフォーマット概略 .....       | 39 |
| 図5.11 | Transient1ヘッダ トランジェントデータのシーケンシャル番号と識別番号の関係 ..... | 40 |
| 図5.12 | Transient1データ部 ノード情報配信要求の分割時のフレーム .....          | 42 |
| 図5.13 | Transient1データ部 ノード情報配信要求 .....                   | 44 |
| 図5.14 | Transient1データ部 ノード情報配信要求1フレーム目 .....             | 45 |
| 図5.15 | Transient1データ部 ノード情報配信要求2フレーム目 .....             | 46 |
| 図5.16 | Transient1データ部 統計情報取得要求 .....                    | 48 |
| 図5.17 | Transient1データ部 統計情報取得応答 .....                    | 49 |
| 図5.18 | Transient1データ部 ノード詳細情報取得要求 .....                 | 51 |
| 図5.19 | Transient1データ部 ノード詳細情報取得応答 .....                 | 52 |
| 図5.20 | TransientAckフレームフォーマット概略 .....                   | 54 |
| 図5.21 | CC-Link互換トランジェントフレームフォーマット概略 .....               | 57 |
| 図5.22 | コマンドタイプ (CT) のデータ構成 .....                        | 61 |
| 図5.23 | メモリアクセス情報取得フレームフォーマット概略 .....                    | 62 |
| 図5.24 | RUNフレームフォーマット概略 .....                            | 63 |
| 図5.25 | STOPフレームフォーマット概略 .....                           | 64 |
| 図5.26 | メモリ読出しフレームフォーマット概略 .....                         | 65 |
| 図5.27 | メモリ書込みフレームフォーマット概略 .....                         | 66 |
| 図5.28 | アクセスコード定義 .....                                  | 67 |
| 図5.29 | 属性定義 .....                                       | 67 |
| 図5.30 | SLMPフレームフォーマット概略 .....                           | 70 |
| 図5.31 | SLMPメモリ読出しフレーム .....                             | 73 |
| 図5.32 | SLMPメモリ書込みフレーム .....                             | 74 |
| 図6.1  | サンプルコードの構成 .....                                 | 77 |

|       |   |     |
|-------|---|-----|
| 図6.2  | ファームウェア開発手順 .....                                       | 78  |
| 図6.3  | メイン処理 フロー図 .....  | 83  |
| 図6.4  | 初期化处理 フロー図 .....  | 85  |
| 図6.5  | 通信開始処理 フロー図 .....                                       | 87  |
| 図6.6  | PHYチェック処理 フロー図.....                                     | 88  |
| 図6.7  | PHY設定変更処理 フロー図.....                                     | 89  |
| 図6.8  | 自局エラー処理 フロー図.....                                       | 90  |
| 図6.9  | サイクリック伝送停止処理 フロー図.....                                  | 90  |
| 図6.10 | イベント処理 フロー図 .....                                       | 91  |
| 図6.11 | 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.....                  | 92  |
| 図6.12 | MyStatus送信処理 フロー図 .....                                 | 94  |
| 図6.13 | サイクリック送信処理 フロー図.....                                    | 94  |
| 図6.14 | 通信状態更新処理 フロー図.....                                      | 95  |
| 図6.15 | サイクリック伝送状態更新処理 フロー図.....                                | 97  |
| 図6.16 | MIB情報取得処理 フロー図.....                                     | 98  |
| 図6.17 | Transient1、Transient2、TransientAck受信処理 フロー図 (1/2) ..... | 100 |
| 図6.18 | Transient2要求フレーム作成処理 フロー図.....                          | 102 |
| 図6.19 | Transient1、Transient2、TransientAck送信処理 フロー図 (1/2) ..... | 103 |
| 図6.20 | トランジェントフレーム分割送信処理のイメージ.....                             | 105 |
| 図6.21 | Transient1受信データ処理 フロー図 (1/3) .....                      | 106 |
| 図6.22 | マスタ局からのSLMP要求処理手順 (例) .....                             | 110 |
| 図6.23 | Transient1受信データ組立て開始処理 フロー図.....                        | 111 |
| 図6.24 | Transient1受信データ組立て処理 フロー図.....                          | 112 |
| 図6.25 | ノード情報配信フレーム受信対応処理 フロー図.....                             | 113 |
| 図6.26 | ノード情報配信フレームチェック処理 フロー図.....                             | 114 |
| 図6.27 | 統計情報取得要求フレーム受信対応処理 フロー図.....                            | 115 |
| 図6.28 | 統計情報取得応答フレーム作成処理 フロー図.....                              | 116 |
| 図6.29 | ノード詳細情報取得要求フレーム受信対応処理 フロー図.....                         | 117 |
| 図6.30 | ノード詳細情報取得応答フレーム作成処理 フロー図.....                           | 118 |
| 図6.31 | オプション情報取得要求フレーム受信対応処理 フロー図.....                         | 119 |
| 図6.32 | 選択局情報取得要求フレーム受信対応処理 フロー図.....                           | 120 |
| 図6.33 | 交信テスト要求フレーム受信対応処理 フロー図.....                             | 121 |
| 図6.34 | ケーブルテスト要求フレーム受信対応処理 フロー図.....                           | 122 |
| 図6.35 | Transient2受信データ処理 フロー図 (1/2) .....                      | 123 |
| 図6.36 | Transient2受信データチェック処理 フロー図.....                         | 125 |
| 図6.37 | TransientAck受信データ処理 フロー図.....                           | 125 |
| 図6.38 | TransientAckフレーム作成処理 フロー図.....                          | 126 |
| 図6.39 | Transient2応答フレーム作成処理 フロー図.....                          | 127 |
| 図6.40 | Transient2メモリ読み出し要求フレーム作成処理 フロー図.....                   | 128 |
| 図6.41 | Transient2メモリ書き込み要求受信処理 フロー図.....                       | 130 |
| 図6.42 | Transient2メモリ読み出し応答受信処理 フロー図.....                       | 131 |
| 図6.43 | SLMPメモリ読み出し要求フレーム受信対応処理 フロー図 .....                      | 132 |
| 図6.44 | SLMPメモリ書き込み要求フレーム受信対応処理 フロー図 .....                      | 133 |

|       |  |     |
|-------|--|-----|
| 図6.45 | SLMPメモリ読出し要求フレーム作成処理 フロー図 .....              | 134 |
| 図6.46 | Transient1要求送信分割判定処理 フロー図.....               | 135 |
| 図6.47 | Transient1要求フレーム作成処理 フロー図.....               | 136 |
| 図6.48 | SLMPメモリ読出し応答受信処理 フロー図 .....                  | 137 |
| 図6.49 | ハードウェアテスト (IEEE802.3abコンプライアンステスト) フロー図..... | 138 |
| 図6.50 | ポート概略図 .....                                 | 139 |
| 図6.51 | ハードウェアテスト (折り返し通信テスト) フロー図.....              | 140 |
| 図6.52 | ネットワークとコントローラの例.....                         | 148 |
| 図6.53 | 自局LED情報.....                                 | 184 |
| 図7.1  | リモートReady タイミング図 .....                       | 199 |
| 図7.2  | イニシャルデータ処理要求/完了フラグ タイミング図 .....              | 200 |
| 図7.3  | イニシャルデータ設定完了/要求フラグ タイミング図 .....              | 201 |
| 図7.4  | イニシャルデータ処理および設定 タイミング図.....                  | 202 |
| 図7.5  | エラー状態/リセット要求フラグ タイミング図.....                  | 203 |
| 図7.6  | ワーニング状態フラグ タイミング図.....                       | 204 |

## 表の目次

|       |  |    |
|-------|--|----|
| 表1.1  | 性能仕様（概要） .....                             | 1  |
| 表1.2  | ベンダーコードと機種タイプ .....                        | 3  |
| 表1.3  | H/Wスイッチを使用する方法（例1） .....                   | 4  |
| 表1.4  | H/Wスイッチの範囲.....                            | 4  |
| 表1.5  | エンジニアリングツールを使用する方法（例2） .....               | 4  |
| 表1.6  | 1000BASE-Tコンプライアンステストモードの切替え方法（例） .....    | 5  |
| 表1.7  | エンジニアリングツールの機能.....                        | 6  |
| 表1.8  | リンクデバイスの属性定義例（システム領域） .....                | 8  |
| 表1.9  | リンクデバイスの属性定義例（システム領域） .....                | 8  |
| 表1.10 | CSP+ファイルのセクション構成 .....                     | 10 |
| 表2.1  | R-IN32M3-CL機能一覧 .....                      | 13 |
| 表2.2  | 通信機能一覧 .....                               | 14 |
| 表2.3  | R-IN32M3-CL割込み一覧 .....                     | 15 |
| 表2.4  | 診断画面の表示／操作箇所とSLMP要求との対応 .....              | 17 |
| 表3.1  | 部品選定チェックシート .....                          | 19 |
| 表3.2  | 回路設計チェックシート .....                          | 20 |
| 表3.3  | <b>パターン設計チェックシート</b> .....                 | 20 |
| 表4.1  | LED状態表示一覧.....                             | 21 |
| 表4.2  | LED制御一覧.....                               | 23 |
| 表4.3  | LEDの有効／無効の切替えが可能なLED一覧 .....               | 24 |
| 表5.1  | トランジェントフレームの一覧と実装要否.....                   | 28 |
| 表5.2  | トランジェント伝送のコマンドのクライアント・サーバ機能実装要否 .....      | 29 |
| 表5.3  | Transient1要求の受信手順 .....                    | 30 |
| 表5.4  | Transient1要求の送信手順 .....                    | 31 |
| 表5.5  | Transient2要求の受信手順 .....                    | 32 |
| 表5.6  | Transient2要求の送信手順 .....                    | 33 |
| 表5.7  | トランジェントフレーム共通フォーマット概要.....                 | 34 |
| 表5.8  | MACヘッダ項目一覧 .....                           | 36 |
| 表5.9  | CC-Link IEヘッダ項目一覧.....                     | 37 |
| 表5.10 | CC-Link IEフィールド固有トランジェントフレームフォーマット概要 ..... | 38 |
| 表5.11 | Transient1ヘッダ項目一覧 .....                    | 40 |
| 表5.12 | 拡張ヘッダ項目一覧 .....                            | 41 |
| 表5.13 | CC-Link IEフィールド固有トランジェント伝送コマンド一覧.....      | 41 |
| 表5.14 | ノード情報配信のフレームフォーマット一覧.....                  | 43 |
| 表5.15 | ノード情報配信ヘッダ項目一覧.....                        | 47 |
| 表5.16 | ノード情報データ部項目一覧.....                         | 47 |
| 表5.17 | ノード種別一覧 .....                              | 47 |
| 表5.18 | 統計情報取得応答データ項目一覧.....                       | 50 |
| 表5.19 | ノード詳細情報取得応答データ部項目一覧.....                   | 53 |
| 表5.20 | TransientAckフレームフォーマット概要.....              | 54 |
| 表5.21 | TransientAckデータ部項目一覧.....                  | 55 |

|       |   |     |
|-------|---|-----|
| 表5.22 | CC-Link互換トランジェントフレームフォーマット概要.....                           | 56  |
| 表5.23 | Transient2ヘッダ項目 .....                                       | 58  |
| 表5.24 | リターンコードに格納するエラーコード例一覧.....                                  | 60  |
| 表5.25 | CC-Link互換トランジェントコマンドタイプ一覧.....                              | 61  |
| 表5.26 | RUN要求設定一覧.....  | 63  |
| 表5.27 | STOP要求設定一覧.....   | 64  |
| 表5.28 | メモリ読出し設定一覧 .....  | 65  |
| 表5.29 | メモリ書込み要求設定一覧.....   | 66  |
| 表5.30 | 三菱電機製品のアクセスコード一覧.....                                       | 68  |
| 表5.31 | SLMPフレームフォーマット概要 .....                                      | 69  |
| 表5.32 | SLMPヘッダ項目 .....   | 71  |
| 表5.33 | 終了コードに格納するエラーコード例一覧.....                                    | 72  |
| 表5.34 | SLMPメモリ読出しフレームフォーマットの詳細 .....                               | 73  |
| 表5.35 | SLMPメモリ書込みフレームフォーマットの詳細 .....                               | 74  |
| 表5.36 | MyStatus送信に関する情報 .....                                      | 75  |
| 表5.37 | MyStatus受信に関する情報 .....                                      | 76  |
| 表6.1  | サンプルコードに含まれるプログラム部品一覧.....                                  | 77  |
| 表6.2  | サンプルコードのファイルリスト (1/2) .....                                 | 79  |
| 表6.3  | イニシャル・サイクリック伝送関連サンプルフローチャート一覧 (R_IN32M3_sample.c) ....      | 81  |
| 表6.4  | トランジェント伝送関連サンプルフローチャート一覧 (R_IN32M3_Transient.c) (1/2) ..... | 81  |
| 表6.5  | ハードウェアテスト関連サンプルフローチャート一覧 (R_IN32M3_HWTest.c) .....          | 82  |
| 表6.6  | リング制御部 MIB情報一覧表 .....                                       | 98  |
| 表6.7  | MAC IP部 MIB情報一覧表 .....                                      | 99  |
| 表6.8  | その他 MIB情報一覧表.....   | 99  |
| 表6.9  | 折り返し通信テストによるトラブルシュート.....                                   | 139 |
| 表6.10 | テスト項目の注意事項 .....  | 139 |
| 表6.11 | R-IN32M3-CLドライバインタフェース関数一覧表 (1/2) .....                     | 141 |
| 表6.12 | アプリケーション詳細動作状態の初期値一覧.....                                   | 151 |
| 表6.13 | アプリケーション詳細エラー状態の初期値一覧.....                                  | 151 |
| 表6.14 | R-IN32M3-CLドライバターゲット依存関数一覧表.....                            | 192 |
| 表6.15 | R-IN32M3-CLドライバが使用するコールバック関数一覧表.....                        | 195 |
| 表6.16 | gR_IN32_CallbackFatalError関数のFatalエラーコード一覧 .....            | 195 |
| 表7.1  | システム領域のビット割付け (例) .....                                     | 198 |

## 1. 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用通信 LSI R-IN32M3-CL を使用して、インテリジェン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設計
- ・R-IN32M3-CL ドライバの仕様

### 1.1 R-IN32M3-CL の性能仕様

R-IN32M3-CL の性能仕様を以下に示します。

表1.1 性能仕様（概要）

| 項目                 | 仕様   |                              |
|--------------------|--|------------------------------|
| 局種別                | インテリジェントデバイス局  |                              |
| 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<br>（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）<br>スター接続時：システム構成による<br>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  |                              |
| 1 局あたりの<br>最大リンク点数 | RX   | 最大 2048 点（2048 ビット）、256 バイト  |
|                    | RY   | 最大 2048 点（2048 ビット）、256 バイト  |
|                    | RWr  | 最大 1024 点（1024 ワード）、2048 バイト |
|                    | RWw  | 最大 1024 点（1024 ワード）、2048 バイト |

## 1.2 開発の準備

本節は、開発するにあたり、あらかじめ準備や検討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。各工程であらかじめ準備や検討が必要なものを、次項以降に示します。

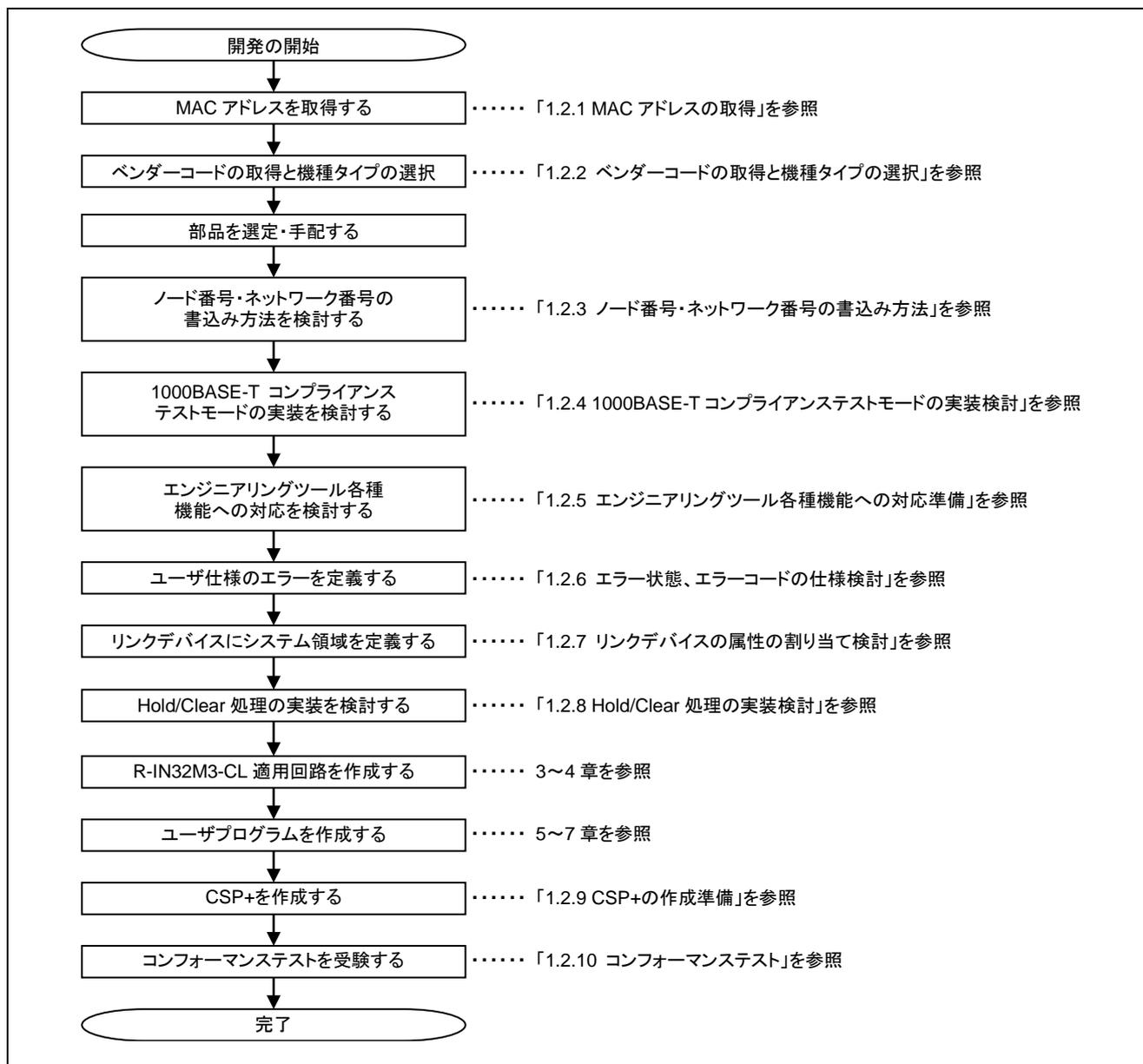


図1.1 開発の工程例

### 1.2.1 MAC アドレスの取得

CC-Link IE フィールドネットワーク機器は、Ethernet (IEEE 802.3ab) に準拠しているため、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 : <http://standards.ieee.org/develop/regauth/oui/>

### 1.2.2 ベンダーコードの取得と機種タイプの選択

CC-Link IE フィールドネットワーク機器は、ベンダーコードと機種タイプの登録が必要です。ベンダーコードと機種タイプの割り当てや管理は CC-Link 協会が行っています。ご不明な点があれば CC-Link 協会までお問い合わせください。

表1.2 ベンダーコードと機種タイプ

| 項目                   | 内容   |
|----------------------|--|
| ベンダーコード (vendorCode) | CC-Link 協会入会時に発行される ID 番号 (5 桁目~8 桁目) <sup>注</sup>   |
| 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、ベンダーコードは 5678 となります。**

### 1.2.3 ノード番号・ネットワーク番号の書き込み方法

自局をデータリンクさせるためには、ノード番号およびネットワーク番号を R-IN32M3-CL へ書き込む必要があります。そのため開発機器の仕様に応じた、ノード番号およびネットワーク番号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 (「6.2.2 初期化処理」参照) において R-IN32M3-CL ドライバインタフェース関数の gerR\_IN32\_SetNodeAndNetworkNumber (「6.4.1(3) gerR\_IN32\_SetNodeAndNetworkNumber」参照) を使用します。

ノード番号およびネットワーク番号の書き込み方法とイメージを以下に例示します。

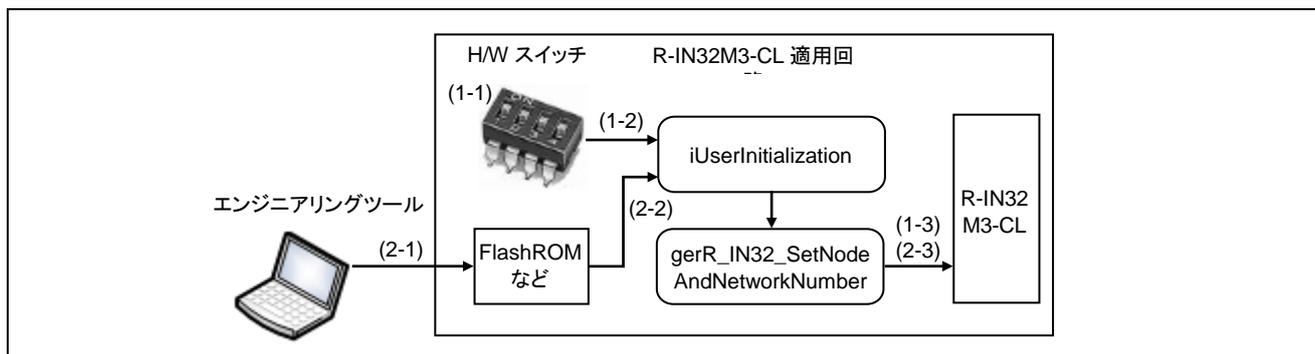


図1.2 ノード番号およびネットワーク番号の書き込みイメージ

表1.3 H/W スイッチを使用する方法（例 1）

| 手順  | 内容   |
|-----|--|
| 1-1 | H/W スイッチを使用してノード番号およびネットワーク番号を設定する。  |
| 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 R-IN32M3-CL ドライバインタフェース関数 gerR_IN32_SetNodeAndNetworkNumber の引数へセットする。<br>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し処理を追加してください。 |
| 1-3 | R-IN32M3-CL ドライバインタフェース関数の gerR_IN32_SetNodeAndNetworkNumber が、引数の値を R-IN32M3-CL に書き込む。  |

なお、H/W スイッチは、ノード番号およびネットワーク番号の値域に対応したスイッチを選定してください。

表1.4 H/W スイッチの範囲

| 項目       | 値域             |
|----------|----------------|
| ノード番号    | 01h~78h(1~120) |
| ネットワーク番号 | 01h~EFh(1~239) |

表1.5 エンジニアリングツールを使用する方法（例 2）

| 手順  | 内容   |
|-----|--|
| 2-1 | エンジニアリングツールを使用して、ノード番号およびネットワーク番号データを FlashROM などへ書き込む。  |
| 2-2 | ユーザプログラムの iUserInitialization が、FlashROM に書き込まれたデータを読み出して、R-IN32M3-CL ドライバインタフェース関数 gerR_IN32_SetNodeAndNetworkNumber の引数へセットする。<br>ユーザプログラムの iUserInitialization には、FlashROM に書き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 | R-IN32M3-CL ドライバインタフェース関数の gerR_IN32_SetNodeAndNetworkNumber が、引数の値を R-IN32M3-CL に書き込む。  |

### 1.2.4 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。

**注意. 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 ポートから 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、開発機器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、または開発機器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（「6.2.45 ハードウェアテスト（IEEE802.3abコンプライアンステスト」参照）から、R-IN32M3-CL ドライバインタフェース関数の gerR\_IN32\_IEEEETest（「6.4.13(1) gerR\_IN32\_IEEEETest」参照）を呼び出します。

表1.6 1000BASE-T コンプライアンステストモードの切替え方法（例）

| 手順 | 内容   |
|----|--|
| 1  | 開発機器の外部（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  | 開発機器の外部（H/W スイッチなど）から、R-IN32M3-CL ドライバインタフェース関数 gerR_IN32_IEEEETest の引数“MODE1~4”を指定できる処理を実装する。 |

### 1.2.5 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。開発機器（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。

#### 【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「2.6 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。

#### 【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、開発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「1.2.9(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、開発機器（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。

マスタ局からの SLMP フレームの要求受信・応答送信処理（「6.2.18 Transient1受信データ処理」参照）を実装するかをあらかじめ検討してください。

表1.7 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 | <ul style="list-style-type: none"> <li>・ SLMP フレームの要求受信・応答送信処理</li> </ul>   |
|     | 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      | <ul style="list-style-type: none"> <li>・ CSP+を「図1.3 CSP+ファイルのセクション構成」の範囲③まで記述する</li> <li>・ SLMP フレームの要求受信・応答送信処理</li> </ul> |

## 1.2.6 エラー状態、エラーコードの仕様検討

開発機器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。開発機器の仕様に応じて、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、本項の「(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、本項の「(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-IN32M3-CL 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<sup>注</sup>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-IN32M3-CL の H/W 異常のエラーコードは、R-IN32M3-CL 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。（「6.6(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。

### (1) エラー状態の仕様検討

R-IN32M3-CL 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。（「5.4 MyStatus概要」参照）

開発機器の仕様に応じた、軽度/中度/重度異常の状態をあらかじめ定義してください。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
- ・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  
(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) 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

開発機器が 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。

エンドユーザが開発機器のユーザ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、開発機器の仕様に応じたエラーコードをリターンコード (RSTS) に格納してください。詳細は、「5.3.4 CC-Link互換トランジェントフレームフォーマット」を参照してください。

開発機器が 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、開発機器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、「5.3.5 SLMPフレームフォーマット」を参照してください。

### 1.2.7 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。開発機器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。）

#### (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。（たとえば、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) 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「7 リンクデバイスのシステム領域」参照してください。）

表1.8 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      | —       | —               |

#### (3) ベンダー入出力グループ

ベンダー入出力グループは、リンクデバイスを任意に定義して使用します。

表1.9 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         |
|---------|------------|---------|------------|
| 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 |
| RWr00   | 出力周波数状態    | RWw00   | 出力周波数設定値   |

### 1.2.8 Hold/Clear 処理の実装検討

Hold/Clear 処理は、開発機器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) する処理です。

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、「6.2.9 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、「5.4.2 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、「6.2.12 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-IN32M3-CL 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、「6.2.9 マスタ局からのMyStatusおよびサイクリック受信処理」と、「6.2.12 通信状態更新処理」を参照してください。

### 1.2.9 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+を開発機器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としてCSP+ファイルを作成する範囲を示します。コンFORMANCE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。開発機器の仕様として、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。

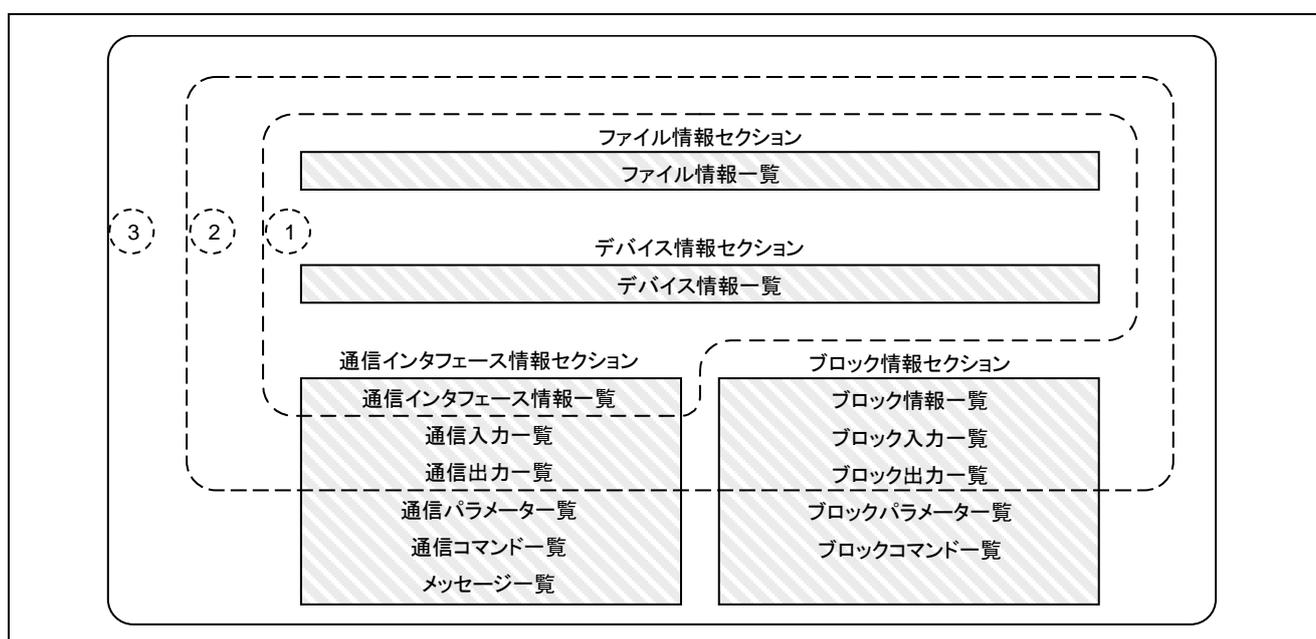


図1.3 CSP+ファイルのセクション構成

表1.10 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FORMANCEテストにおいて、必須項目確認のために必要な情報<br>【GX Works2、GX Works3 の場合】<br>CC IE Field 構成ウィンドウに開発機器が表示され、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  | スレーブ局パラメータ処理・コマンド実行をするために必要な情報 <sup>注</sup><br>【GX Works2、GX Works3 の場合】<br>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、本項「(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。

## (1) スレーブ局パラメータ処理・コマンド実行

開発機器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、開発機器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。

開発機器の仕様として、エンジニアリングツールのスレーブ局パラメータ処理・コマンド実行に対応するかを検討してください。

### 【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     |
|--|------------|------------|------------|----------|--------|
| <b>局単位パラメータ</b>  |            |            |            |          |        |
| <input checked="" type="checkbox"/> 入力応答時間設定                               | 5:10ms     | 5:10ms     | 5:10ms     |          |        |
| <input checked="" type="checkbox"/> 出力HOLD/CLEAR設定                         | 0: CLEAR   | 0: CLEAR   | 0: CLEAR   |          |        |
| <input checked="" type="checkbox"/> サイクリックデータ更新監視時間...                     | 0          | 0          | 0          | 0~20     | x100ms |
| <input checked="" type="checkbox"/> モード切替                                  | 9:自動判定...  | 9:自動判定...  | 9:自動判定...  |          |        |
| <input checked="" type="checkbox"/> 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... |          |        |
| <b>ユニット単位パラメータ</b>   |            |            |            |          |        |
| <input checked="" type="checkbox"/> <input type="checkbox"/> 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       |          |        |
| <input checked="" type="checkbox"/> <input type="checkbox"/> 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1.4 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「図1.3 CSP+ファイルのセクション構成」の範囲③まで記述する。
- ・CSP+で記述した SLMP フレームの送受信処理を開発機器に実装する。

### 1.2.10 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、認定製品として「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。

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

#### (1) コンフォーマンステストに最低限必要な事項

本書に記載する機能や処理のうち、コンフォーマンステストに最低限必要な事項を以下示します。

##### 【サイクリック伝送機能】

コンフォーマンステスト全体をとおしてサイクリック伝送機能が必要です。

「表6.3 イニシャル・サイクリック伝送関連サンプルフローチャート一覧 (R\_IN32M3\_sample.c)」に記載する、実装要否が必須である処理を実装してください。

##### 【トランジェント伝送機能】

Transient1 ノード詳細情報取得への応答が必要です。

「表6.4 トランジェント伝送関連サンプルフローチャート一覧 (R\_IN32M3\_Transient.c)」に記載する、実装要否が必須である処理を実装してください。

##### 【1000BASE-T コンプライアンステスト】

IEEE802.3 の仕様に基づき、伝送路の波形確認が必要です。

「表6.5 ハードウェアテスト関連サンプルフローチャート一覧 (R\_IN32M3\_HWTest.c)」に記載する処理を実装してください。

##### 【CSP+】

CSP+を「図1.3 CSP+ファイルのセクション構成」の範囲①を作成してください。

## 2. R-IN32M3-CL の機能

R-IN32M3-CL がサポートする機能について示します。

表2.1 R-IN32M3-CL 機能一覧

| 機能                         | 概要   |
|----------------------------|--|
| バスアクセス                     | 外部 16bit バスによる 16/32bit レジスタアクセス   |
| LED 状態表示                   | <ul style="list-style-type: none"> <li>・状態表示 (RUN、RD、SD、ERR、D LINK)</li> <li>・ポート状態表示 (ポート 1 L ER、ポート 2 L ER)</li> <li>・ユーザ LED×2</li> </ul> |
| 割込み                        | <ul style="list-style-type: none"> <li>・MPU 割込出力</li> <li>・外部 WDT 入力</li> <li>・内部 WDT 出力</li> <li>・マスタウォッチタイマ</li> </ul>                     |
| リセット                       | <ul style="list-style-type: none"> <li>・パワーオンリセット入力</li> <li>・システムリセット入力</li> <li>・PHY リセット出力</li> </ul>                                     |
| WDT                        | <ul style="list-style-type: none"> <li>・内部 WDT</li> <li>・外部 WDT</li> </ul>   |
| バイパスモード                    | <p>自局で通信に影響を及ぼす異常が発生しても、リンクアップを継続する。<br/>ポート 1 (ポート 2) で受信したフレームをポート 2 (ポート 1) で送信することができる。</p>  |
| MyStatus/サイクリック送信          | <p>送信データを格納しているアドレスを R-IN32M3-CL に設定することで、R-IN32M3-CL ドライバが自動的に MyStatus/サイクリック送信フレームを作成して送信する。</p>  |
| MyStatus/サイクリック受信          | <p>他局から受信した MyStatus/サイクリックフレームのデータを、R-IN32M3-CL ドライバが自動的に指定した格納先へ書き込む。</p>  |
| MDIO                       | <p>PHY の初期化と状態監視するための IF を持つ。<br/>MAC アクセスが許可の場合のみアクセス可能。</p>  |
| MIB (統計) 情報                | <p>2つのポートの HEC エラーフレーム受信や DCS/FCS エラーフレーム受信などの統計情報を取得できる。</p>  |
| トランジェント送信                  | <p>送信データを格納しているアドレスを R-IN32M3-CL に設定することで、R-IN32M3-CL ドライバがトランジェントフレームを送信する。</p>   |
| トランジェント受信                  | <p>他局から受信したトランジェントフレームのデータを、R-IN32M3-CL ドライバが指定した格納先へ書き込む。</p>   |
| CC-Link IE フィールドネットワーク診断機能 | <p>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が確認できる。<br/>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できる。</p>   |

## 2.1 通信機能

R-IN32M3-CL は、サイクリック伝送、トランジェント伝送、および MyStatus の通信機能をサポートします。

表2.2 通信機能一覧

| 名称           | 内容  |
|--------------|---|
| サイクリック伝送     | <p>マスタ局と周期的にデータを送受信します。サイクリック伝送のデータ送受信は R-IN32M3-CL が自動的に行います。</p> <p>データ交信には、リンクデバイス (RX、RY、RWw、RWr) を使用します。</p> <p>インテリジェントデバイス局が扱えるデータサイズを以下に示します。</p> <p>RX : 最大2048ビット (2048点)、256バイト</p> <p>RY : 最大2048ビット (2048点)、256バイト</p> <p>RWw : 最大1024ワード (1024点)、2048バイト</p> <p>RWr : 最大1024ワード (1024点)、2048バイト</p> |
| トランジェント伝送    | <p>ユーザプログラムからの要求時や他局からの交信要求時にデータを送受信します。</p> <p>インテリジェントデバイス局が扱える機能と、データサイズを以下に示します。</p> <p>クライアント機能 : あり</p> <p>サーバ機能 : あり</p> <p>データサイズ : 2048バイト (トランジェントフレームのデータ部サイズ)</p>   |
| MyStatus 送受信 | <p>R-IN32M3-CL が自局の情報を MyStatus フレームにセットしてマスタ局に通知します。</p> <p>また、マスタ局からの MyStatus フレームを受信して、マスタ局の状態を監視します。</p>   |

## 2.2 状態表示機能

R-IN32M3-CL は、自局の状態およびポートの状態を、LED を使用して表示することができます。各 LED の詳細については、「4 状態表示機能」を参照してください。

## 2.3 割込み

R-IN32M3-CL は、4つの割込み機能をサポートします。

割込み機能には、「MPU 割込み機能」、「マスタウォッチタイマ機能」、「内部 WDT 機能」および「外部 WDT 機能」があります。内部 WDT 機能と外部 WDT 機能は、両機能を同時に使用できませんので、排他的に使用してください。

表2.3 R-IN32M3-CL 割込み一覧

| 名称           | 信号名   | 割込種別 | 内容  |
|--------------|-------|------|---|
| MPU 割込み機能    | INTL  | 出力   | MPU 割込み機能は、R-IN32M3-CL ドライバインタフェース関数 <code>gerR_IN32_Initialize</code> で設定する「MPU 割込み機能使用」 <sup>注</sup> が“R_IN32_TRUE”の場合、R-IN32M3-CL はイベントが発生すると割込み信号 INTL を“Low”で出力します。<br>R-IN32M3-CL ドライバは、 <code>gerR_IN32_GetEvent</code> 関数を使用して R-IN32M3-CL のイベントを取得しますので、ベンダーが割込み信号 (INTL) を意識する必要がありません。 |
| マスタウォッチタイマ機能 | —     | 内部   | マスタウォッチタイマ機能は、マスタ局が異常動作したときに割込みを発生します。<br>R-IN32M3-CL は、マスタ局が送信する MyStatus フレームの受信間隔がタイムアウト時間内であるかを監視してマスタ局の異常を検知します。<br>R-IN32M3-CL はマスタ局の異常動作を検知すると割込みを発生させます。タイムアウト時間は、R-IN32M3-CL が自動的にマスタ局から受信し設定します。マスタウォッチタイマ割込み発生時の処理は、「6.2.8 イベント処理」を参照してください。   |
| 内部 WDT 機能    | NMIL  | 出力   | 内部 WDT 機能は、ユーザプログラムが異常動作したときに割込みを発生 (割込み信号 NMIL を“Low”で出力) します。また、R-IN32M3-CL はバイパスモードとなります。<br>R-IN32M3-CL は、ユーザプログラムから WDT リセット間隔がイニシャル処理で設定する WDT 監視時間内であるかを監視してユーザプログラムの異常を検知します。ユーザプログラムは、WDT 監視時間内に WDT をリセットする処理を実装します。<br>内部 WDT 機能の設定については、「6.2.2 初期化処理」を参照してください。                       |
| 外部 WDT 機能    | WDTIL | 入力   | 外部 WDT 機能は、外部 WDT 検出回路でユーザプログラムが正常に動作しているかを監視するものです。外部 WDT 機能を使用する場合、ユーザプログラムの異常を検出する WDT 検出回路を実装し、R-IN32M3-CL の WDTIL 端子に接続してください。外部 WDT 検出回路がエラーを検出した後は、割込み信号 WDTIL が“Low”を維持するように設計してください。<br>割込み信号 WDTIL に“Low”信号が入力されると、R-IN32M3-CL はユーザプログラムが異常と認識し、バイパスモードとなります。                           |

注. 「MPU 割込み機能使用」の詳細は「6.4.1(2) `gerR_IN32_Initialize`」を参照してください。

## 2.4 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、WDT エラー、自局エラーなどの通信に影響を与える異常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。

## 2.5 MIB 情報

R-IN32M3-CL は、通信状態を管理する情報として、ポートごとにフレーム受信回数やエラーフレーム受信回数などをカウントし MIB に格納します。

ベンダーは、MIB 情報を利用し、自局のポート 1、ポート 2 の通信異常状態を把握できます。

MIB 情報の詳細については、「6.2.14 MIB情報取得処理」(1)(2)(3)を参照してください。

### 2.6 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ตอบสนองすることで、CC-Link IE フィールドネットワーク診断画面に開発機器が表示されます。また、各種のテスト・操作を実行することができます。



図2.1 診断画面の表示／操作箇所 (GX Works2 の場合)

表2.4 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 (0x3119)  |
| 2   | 送信テスト      | 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。 | 送信テスト要求 (0x3040)      |
| 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 (0x3050)    |

**【SLMP要求フレームの応答】**

ユーザプログラムの UserHandleReceivedTransient1 (「6.2.18 Transient1受信データ処理」参照) にて、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上記 No.1~3 の処理を記述していますので、ご活用ください。(実装を推奨します。)

## 2.6.1 選択局通信状態モニタのLED

UserHandleReceivedSelectInfoRequest（「6.2.28 選択局情報取得要求フレーム受信対応処理」参照）にてLED情報を作成し、選択局通信状態取得要求に回答することで、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。

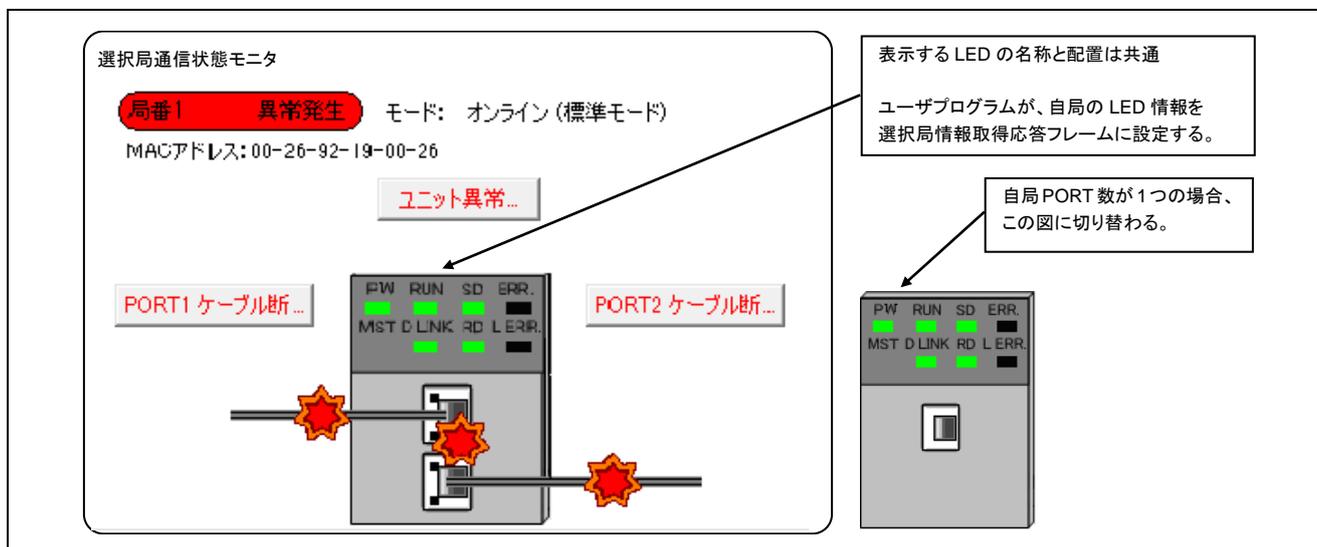


図2.2 選択局通信状態モニタ表示例

### 【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、開発機器のLED状態が目視できないなどの場合に、CC-LinkIE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<sup>注</sup>とLED配置は、上図のとおりです。

**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、「6.4.11(9) gulR\_IN32\_SetSelectInfo\_Response」を参照してください。

### 3. 基板設計上の注意事項

#### 3.1 部品選定

下表の内容に注意して部品を選定してください。

表3.1 部品選定チェックシート

| No. | 項目                       | 内容   | チェック |
|-----|--------------------------|--|------|
| 1   | MPU の選定について              | 以下の仕様を満たす MPU を選定しているか？<br>(1) データ幅 16 ビット以上<br>(2) アドレス幅 17 ビット以上<br>(3) エンディアンはリトルエンディアンであること。<br>(4) 4 章に示す各タイミングを満たすこと。  |      |
| 2   | RJ-45 コネクタの選定について        | シールド付きで ANSI/TIA/EIA-568-B の 8 ピンコネクタか？  |      |
| 3   | パルストランスの選定について           | IEEE802.3 1000BASE-T 対応の部品を選定しているか？  |      |
| 4   | PHY の選定について              | 以下の仕様を満たす部品を選定しているか？<br>(1) IEEE802.3 1000BASE-T 全二重対応品であること。<br>(2) オートネゴシエーション機能を持った部品であること。<br>(3) GMII インタフェースを持った部品であること。<br>(4) オート MDI/MDIX ネゴシエーション機能を持った部品であること。<br>(5) MDC クロック周波数が 7. 812MHz で動作可能な部品であること。 |      |
| 5   | 125MHz 水晶発振器の選定について      | 周波数偏差±50ppm 以内となる部品を選定しているか？   |      |
| 6   | 2.097152MHz 水晶発振器の選定について | 周波数偏差±50ppm 以内となる部品を選定しているか？   |      |
| 7   | PHY クロック用水晶発振器の選定について    | PHY クロック用水晶発振器は、使用する PHY の要求仕様に合わせて選定しているか？<br>(例)<br>・水晶発振器の周波数<br>・水晶発振器の総ジッター   |      |

### 3.2 回路設計

下表の内容に注意して、R-IN32M3-CL の周辺回路を設計してください。

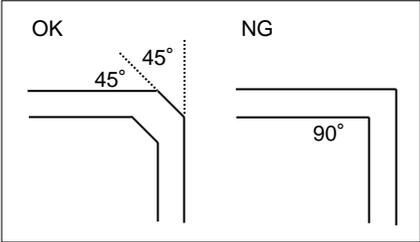
表3.2 回路設計チェックシート

| No. | 項目                      | 内容  | チェック |
|-----|-------------------------|---|------|
| 1   | GMII の配線について            | GMII 信号にオーバーシュート／アンダーシュート対策のダンピング抵抗を入れているか？   |      |
| 2   | PHY と RJ45 コネクタとの接続について | PHY (88E1111-B2-BAB1C000) と RJ45 コネクタ (JF-01J-0-H22-020-2) 間の信号線は、各端子の + 側と + 側、- 側と - 側同士を接続するようにしてください。<br>+ 側と - 側が繋がれている場合、1000BASE-T コンプライアンステストの判定が NG となります。 |      |
| 3   | データ信号について               | データ信号 D15～D00 に、プルアップ抵抗を付けているか？<br>(回路図例では、10kΩ のプルアップ抵抗を使用)  |      |
| 4   | PHY アドレスについて            | PHY アドレスは R-IN32M3-CL のポート番号と同じアドレスに設定しているか？<br>アドレス 1 に設定した PHY は、MAC ポート 1 に接続してください。<br>アドレス 2 に設定した PHY は、MAC ポート 2 に接続してください。                                  |      |

### 3.3 パターン設計

下表の内容に注意して、R-IN32M3-CL 周辺のパターン配線を設計してください。

表3.3 パターン設計チェックシート

| No. | 項目                                      | 内容  | チェック |
|-----|---|---|------|
| 1   | R-IN32M3-CL に接続する 2.097152MHz 水晶発振器について | R-IN32M3-CL に接続する 2.097152MHz 水晶発振器は、R-IN32M3-CL の近くに配置してください。また、CCI_CLK2_097M ピンまでのパターンは極力最短となるように、SG パターンでシールドしているか？            |      |
| 2   | GMII の配線について                            | R-IN32M3-CL と PHY を接続する信号 (GMII) は、最短でパターン配線し、インピーダンス 50Ω のパターンとなるように配線層、信号線の太さを決定しているか？  |      |
| 3   | 信号パターンの曲げ方について                          | パターンを曲げるとき、すべての信号パターンは、下図のように 45 度で曲げているか？<br> |      |
| 4   | 電源/GND パターンについて                         | 電源/GND パターンは、極力太いパターンで配線しているか？  |      |

## 4. 状態表示機能

### 4.1 LEDによる状態表示

R-IN32M3-CL 適用回路は、「表4.1 LED状態表示一覧」の自局状態表示用 LED およびポート 1、2 状態表示用 LED を実装できます。なお、エンドユーザの使い易さの観点から、すべての LED を実装することを推奨します。

LED の制御については、「4.2 LEDの制御」を参照してください。

表4.1 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  |                         |
|             | 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  |                         |
|             | 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-IN32M3-CL のエラー状態を表示します。  |                         |
|             | 点灯                      | 自局でエラーが発生  |                         |
|             | 消灯                      | 正常動作中  |                         |
|             | L ERR.                  | 受信データおよび回線のエラー状態を表示します。<br>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 | ベンダーが定義した状態を表示します。      |  |                         |
| ポート 1 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 |
| ポート 2 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 |

### 4.1.1 User LED 1 と User LED 2 について

User LED 1 と User LED 2 の名称は、仮名です。ベンダーで任意の LED 名を付けてください。ベンダーが定義した状態を表示できます。

例として、以下の状態を表示する場合があります。開発の参考にしてください。

- ・インテリジェントデバイス局の、オンライン/オフライン状態を表示する。
- ・インテリジェントデバイス局にハードウェアテストや回線テストなどを実装した場合、そのテストを実施中/通常運転中を表示する。

User LED 1 と User LED 2 の制御については、「4.2.2 User LED 1とUser LED 2の制御」を参照してください。

## 4.2 LED の制御

### 4.2.1 LED の制御概要

LED の制御は、ハードウェアで制御する LED とユーザプログラムから呼び出す R-IN32M3-CL ドライバインタフェース関数で制御する LED の 2 種類あります。

ハードウェアで制御する LED は、R-IN32M3-CL、PHY および電源チェック回路のいずれかで制御を行います。R-IN32M3-CL が制御する LED は、R-IN32M3-CL が自局の状態に応じて自動的に点灯制御します。PHY が制御する LED は、リンクアップしたとき自動で点灯制御します。

電源チェック回路が制御する LED は、実装する電源回路の状態に応じた点灯制御をしてください。

R-IN32M3-CL ドライバインタフェース関数で制御する LED は、LED 点灯制御用の関数で制御します。「6.4.7 LED制御」を参照してください。

表4.2 LED 制御一覧

| LED 名称          | R-IN32M3-CL<br>出力信号名 | 制御元   | リセット、エラー時の出力  |              |                                     |
|-----------------|----------------------|---|---------------|--------------|-------------------------------------|
|                 |                      |   | パワーオン<br>リセット | システム<br>リセット | 内部 WDT 注1/<br>外部 WDT 注1/<br>自局エラー注2 |
| 自局状態表示用 LED     |                      |   |               |              |                                     |
| PW              | —                    | 電源チェック回路  | —             | —            | —                                   |
| RUN             | RUNLEDL              | R-IN32M3-CL ドライバインタフェース関数、<br>R-IN32M3-CL                     | 消灯            | 消灯           | 消灯                                  |
| RD              | RDLEDL               | R-IN32M3-CL   | 消灯            | —            | —                                   |
| SD              | SDLEDL               | R-IN32M3-CL   | 消灯            | —            | —                                   |
| ERR.            | ERRLEDL              | R-IN32M3-CL ドライバインタフェース関数、<br>R-IN32M3-CL                     | 消灯            | 消灯           | 点灯                                  |
| D LINK          | DLINKLEDL            | R-IN32M3-CL ドライバインタフェース関数、<br>R-IN32M3-CL                     | 消灯            | 消灯           | 消灯                                  |
| User LED 1      | USER1LEDL            | R-IN32M3-CL ドライバインタフェース関数、<br>R-IN32M3-CL                     | 消灯            | 消灯           | 消灯                                  |
| User LED 2      | USER2LEDL            | R-IN32M3-CL ドライバインタフェース関数、<br>R-IN32M3-CL                     | 消灯            | 消灯           | 消灯                                  |
| L ERR.          | —                    | LERR1LEDL と LERR2LEDL の論理積により<br>点灯注3<br>(各ポートの L ER 信号を元にオン) | —             | —            | —                                   |
| ポート 1 状態表示用 LED |                      |   |               |              |                                     |
| LINK            | —                    | PHY<br>(リンクアップしたときに、点灯するように<br>配線してください。)                     | —             | —            | —                                   |
| L ER            | LERR1LEDL            | R-IN32M3-CL ドライバインタフェース関数、<br>R-IN32M3-CL                     | 消灯            | 消灯           | 消灯                                  |
| ポート 2 状態表示用 LED |                      |   |               |              |                                     |
| LINK            | —                    | PHY<br>(リンクアップしたときに、点灯するように<br>配線してください。)                     | —             | —            | —                                   |
| L ER            | LERR2LEDL            | R-IN32M3-CL ドライバインタフェース関数、<br>R-IN32M3-CL                     | 消灯            | 消灯           | 消灯                                  |

注 1. 内部 WDT/外部 WDT については、「2.3 割込み」を参照してください。

2. ユーザプログラムの都合によるエラーです。詳細は「6.2.6 自局エラー処理」および「6.4.5(2) gerR\_IN32\_ForceStop」を参照してください。

3. L ERR. LED の制御については、「4.2.3 L ERR. LEDの制御」を参照してください。

#### 4.2.2 User LED 1 と User LED 2 の制御

R-IN32M3-CL は、任意に機能を定義できる User LED 1 と User LED 2 の 2 点を用意しています。

User LED 1 および User LED 2 は、gerR\_IN32\_SetUSER1LED 関数および gerR\_IN32\_SetUSER2LED 関数を起動して、点灯制御ができます。

### 4.2.3 L ERR. LED の制御

L ERR. LED 信号は、下図に従って、LERR1LEDL および LERR2LEDL 信号の外部 AND 論理を設定してください。

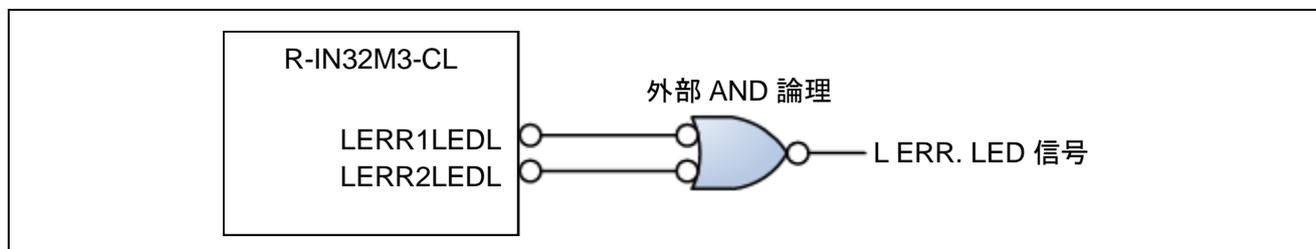


図4.1 L ERR.点灯用外部 AND 論理

## 4.3 LED の有効／無効

下表に示す LED は、LED の有効／無効を切り替えることができます。

LED の有効／無効は、以下に示す例のように、必要に応じてベンダーで仕様を決定してください。

(例) : ポート 1、ポート 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 の表示を無効にするには、gerR\_IN32\_DisableLED 関数を使用します。

LED の表示を有効にするには、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、「6.4.7 LED制御」を参照してください。

表4.3 LED の有効／無効の切替えが可能な LED 一覧

| LED 名称          | 機能                 |
|-----------------|--------------------|
| 自局状態表示用 LED     |                    |
| 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      | ベンダーが定義した状態表示      |
| User LED 2      | ベンダーが定義した状態表示      |
| ポート 1 状態表示用 LED |                    |
| L ER            | ポート 1 受信データエラー状態表示 |
| ポート 2 状態表示用 LED |                    |
| L ER            | ポート 2 受信データエラー状態表示 |

## 5. CC-Link IE フィールドネットワークのデータ通信方式

本章は、サイクリック伝送、トランジェント伝送、および MyStatus の概要について示します。

### 5.1 サイクリック伝送概要

サイクリック伝送は、リンクデバイスを使用して定期的にデータ交信する通信方式です。

マスタ局のリンクデバイス (RY、RWw) の状態がスレーブ局に出力され、スレーブ局からの入力のマスタ局のリンクデバイス (RX、RWr) に格納されます。

リンクデバイスの読書きは、R-IN32M3-CL ドライバインタフェース関数を起動するだけで、R-IN32M3-CL が自動で行います。（「6.2.9 マスタ局からのMyStatusおよびサイクリック受信処理」、「6.2.11 サイクリック送信処理」を参照）

- ・マスタ局にリンクデバイス (RX、RWr) を送信するときは、ユーザプログラムが送信データを格納しているアドレスを R-IN32M3-CL に設定してください。R-IN32M3-CL ドライバが自動的にサイクリック送信フレームを作成して送信します。
- ・マスタ局からリンクデバイス (RX、RWr) を受信したときは、サイクリック受信フレームのデータを、R-IN32M3-CL ドライバが自動的に指定した格納先に書き込みます。ユーザプログラムは格納先からデータを読み出してください。

サイクリックデータの流れを以下に例示します。

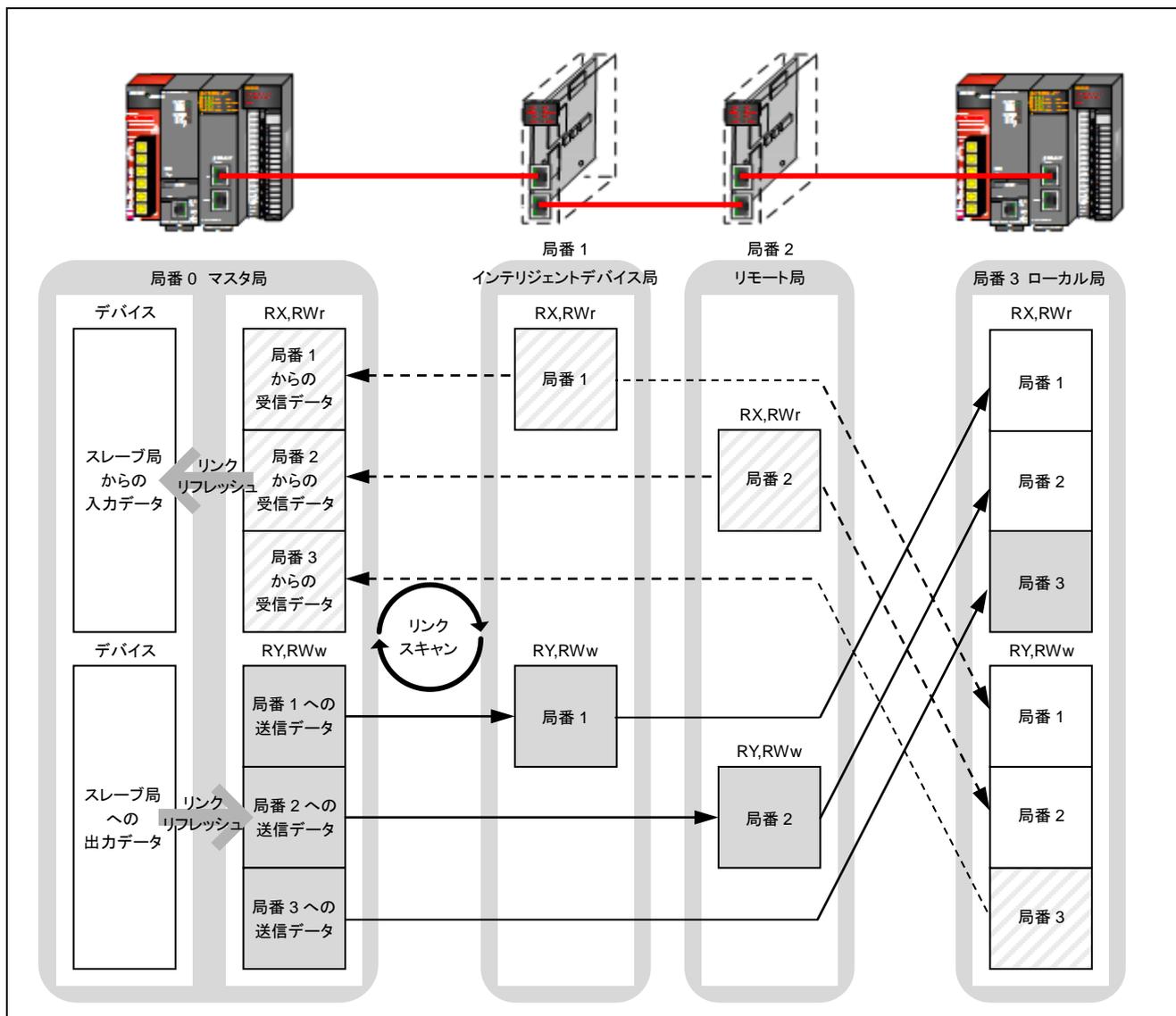


図5.1 サイクリックデータの流れ

## 5.2 トランジェント伝送概要

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

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。

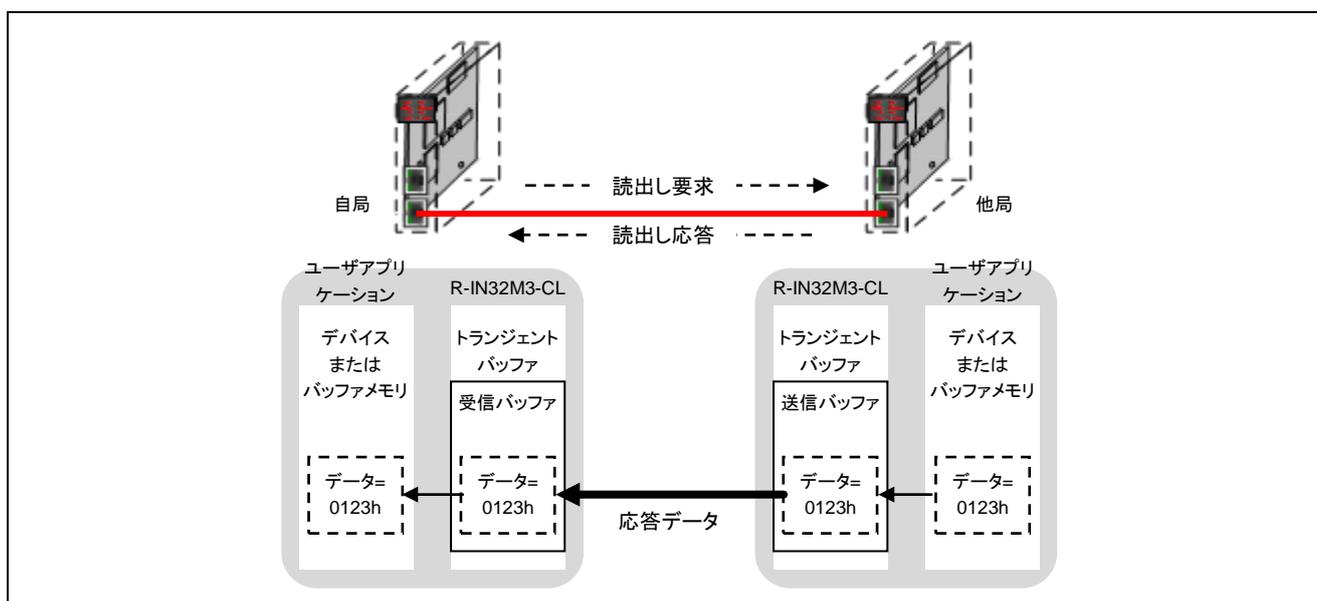


図5.2 トランジェントデータの流れ

### (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、サーバ機能を持つノード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ノード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。

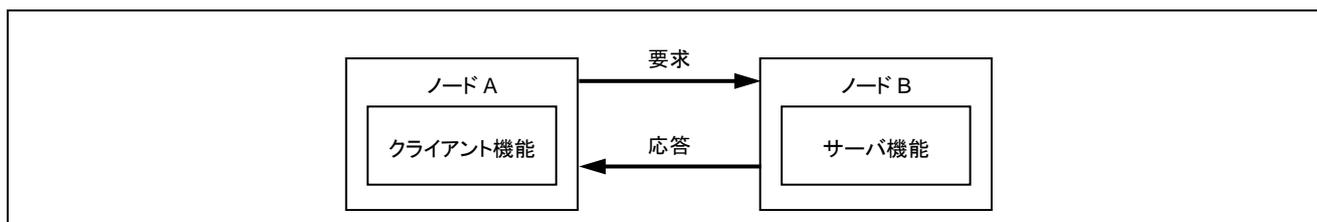


図5.3 トランジェント クライアント・サーバ機能

## (2) トランジェント伝送のトランジェントフレームについて

開発機器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下表に示します。

表5.1 トランジェントフレームの一覧と実装要否

| No. | フレーム名称 <sup>注1</sup>               | フレーム種別<br>(FType) |              | データ種別<br>(DataType) |                                    | データサブ種別 |        | 実装<br>要否 |
|-----|------------------------------------|-------------------|--------------|---------------------|------------------------------------|---------|--------|----------|
| 1   | CC-Link IE<br>フィールド固有<br>トランジェント伝送 | 0x22              | Transient1   | 0x07                | CC-Link IE<br>フィールド固有<br>トランジェント伝送 | 0x0002  | システム固有 | 必須       |
| 2   | SLMP                               | 0x22              | Transient1   | 0x05                | ネットワーク共通                           | 0x0002  | SLMP   | 任意       |
| 3   | CC-Link 互換<br>トランジェント伝送            | 0x25              | Transient2   | 0x04                | CC-Link 互換<br>トランジェント伝送            | —       | —      | 任意       |
| 4   | TransientAck                       | 0x23              | TransientAck | 注2                  | 注2                                 | 注2、3    | 注2、3   | 必須       |

**注 1. 本書に限り、各フレームを上記名称で記載します。**

**2. 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。**

**3. CC-Link 互換トランジェント伝送に対する TransientAck は、データサブ種別に固定値 0x0000 を格納します。**

1. CC-Link IE フィールド固有トランジェント伝送は、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伝送は、主にベンダー製品間の交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) トランジェント伝送のコマンドについて

クライアント・サーバ機能実装が必要なトランジェント伝送のコマンドは、ノード種別により異なります。本書に記載するトランジェント伝送のコマンドの実装要否を下表に示します。

表5.2 トランジェント伝送のコマンドのクライアント・サーバ機能実装要否

| 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    |                 | 【参考】<br>リモート<br>デバイス局 <sup>注1</sup> | 備考   |
|------------------------------------|-------------|------------------|-----------------|-------------------------------------|--|
|                                    |             | クライアント<br>機能（要求） | サーバ機能<br>（応答）   | サーバ機能<br>（応答）                       |  |
| CC-Link IE<br>フィールド固有<br>トランジェント伝送 | ノード情報配信     | ×                | △ <sup>注2</sup> | ×                                   | —  |
|                                    | 統計情報取得      | ×                | △               | △                                   | —  |
|                                    | ノード詳細情報取得   | ×                | ◎               | ◎                                   | —  |
|                                    | オプション情報取得   | ×                | ○               | ○                                   | 注3   |
| CC-Link 互換<br>トランジェント伝送            | メモリアクセス情報取得 | △                | △               | △                                   | アクセスコードを使用する場合に必要                                  |
|                                    | RUN         | △                | △               | △                                   | —  |
|                                    | STOP        | △                | △               | △                                   | —  |
|                                    | メモリ読出し      | △                | △               | △                                   | マスタ・ローカルユニットの専用命令 RIRD に相当                         |
|                                    | メモリ書込み      | △                | △               | △                                   | マスタ・ローカルユニットの専用命令 RIWT に相当                         |
| SLMP                               | 選択局情報取得     | ×                | ○               | ○                                   | 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必要 <sup>注4</sup> |
|                                    | 交信テスト       | ×                | ○               | ○                                   |  |
|                                    | ケーブルテスト     | ×                | ○               | ○                                   |  |
|                                    | メモリ読出し      | △                | △               | △                                   | —  |
|                                    | メモリ書込み      | △                | △               | △                                   | —  |

備考. ◎：必須 ○：推奨 △：任意 ×：不要

注 1. リモートデバイス局は上記コマンドのクライアント機能は不要です。

2. TransientAck および応答は不要です。配信される他局の MAC アドレスデータを受信する処理のみ必要です。

3. 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。

4. 「2.6 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。

### 5.2.1 Transient1 要求の受信手順

クライアントからの Transient1 要求フレームに対して、サーバが Transient1 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。ここでは、統計情報取得、ノード詳細情報取得、SLMP メモリ読出し、および SLMP メモリ書込みを例に示します。

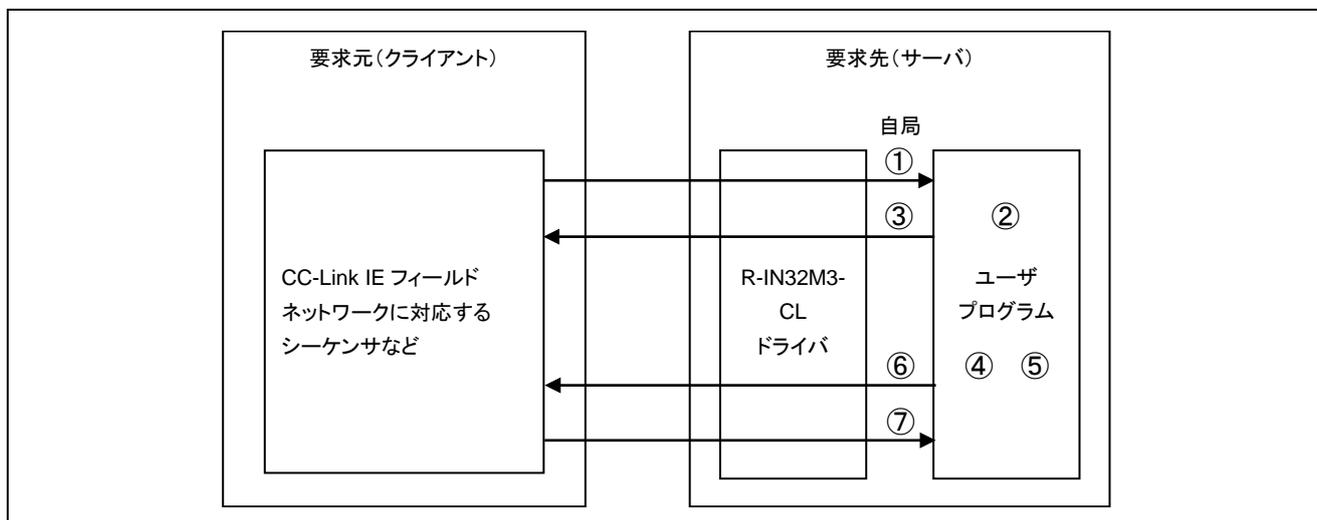


図5.4 Transient1 要求の受信手順

表5.3 Transient1 要求の受信手順

| No. | 処理                               | 参照項  |
|-----|----------------------------------|--|
| ①   | Transient1 要求フレームを受信します          | 「6.2.15 Transient1、Transient2、TransientAck受信処理」参照  |
| ②   | TransientAck フレームを作成します          | 「6.2.34 TransientAckフレーム作成」参照  |
| ③   | TransientAck フレームを送信します          | 「6.2.17 Transient1、Transient2、TransientAck送信処理」参照  |
| ④   | Transient1 要求フレームのコマンドを解析します     | 「6.2.18 Transient1受信データ処理」参照   |
| ⑤   | コマンドに応じて Transient1 応答フレームを作成します | CC-Link IE フィールド固有の場合：<br>「6.2.24 統計情報取得応答フレーム作成」、<br>「6.2.26 ノード詳細情報取得応答フレーム作成」参照<br>SLMP の場合：<br>「6.2.39 SLMPメモリ読出し要求フレーム受信対応処理」、<br>「6.2.40 SLMPメモリ書込み要求フレーム受信対応処理」参照 |
| ⑥   | Transient1 応答フレームを送信します          | 「6.2.17 Transient1、Transient2、TransientAck送信処理」参照  |
| ⑦   | TransientAck フレームを受信します          | 「6.2.15 Transient1、Transient2、TransientAck受信処理」参照  |

## 5.2.2 Transient1 要求の送信手順

クライアントが Transient1 要求フレームを送信して、サーバからの Transient1 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。ここでは、SLMP メモリ読出しを例に示します。

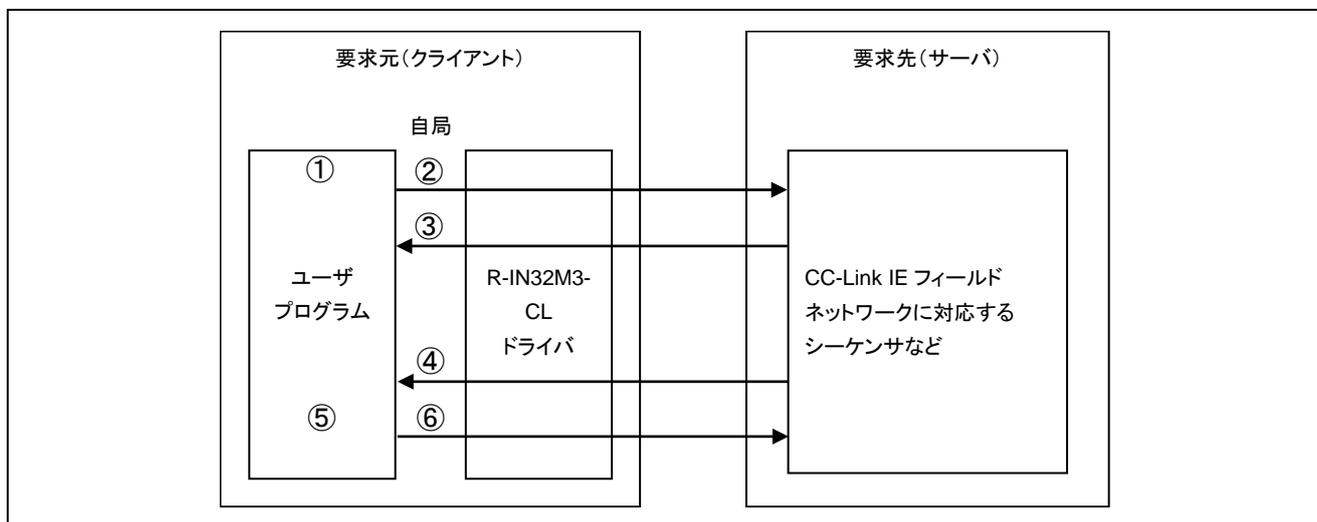


図5.5 Transient1 要求の送信手順

表5.4 Transient1 要求の送信手順

| No. | 処理                      | 参照項   |
|-----|-------------------------|---|
| ①   | Transient1 要求フレームを作成します | 「6.2.41 SLMPメモリ読出し要求フレーム作成処理」参照                   |
| ②   | TransientAck フレームを送信します | 「6.2.17 Transient1、Transient2、TransientAck送信処理」参照 |
| ③   | TransientAck フレームを受信します | 「6.2.15 Transient1、Transient2、TransientAck受信処理」参照 |
| ④   | Transient1 応答フレームを受信します | 「6.2.15 Transient1、Transient2、TransientAck受信処理」参照 |
| ⑤   | TransientAck フレームを作成します | 「6.2.34 TransientAckフレーム作成」参照                     |
| ⑥   | TransientAck フレームを送信します | 「6.2.17 Transient1、Transient2、TransientAck送信処理」参照 |

### 5.2.3 Transient2 要求の受信手順

クライアントからの Transient2 要求フレームに対して、サーバが Transient2 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。

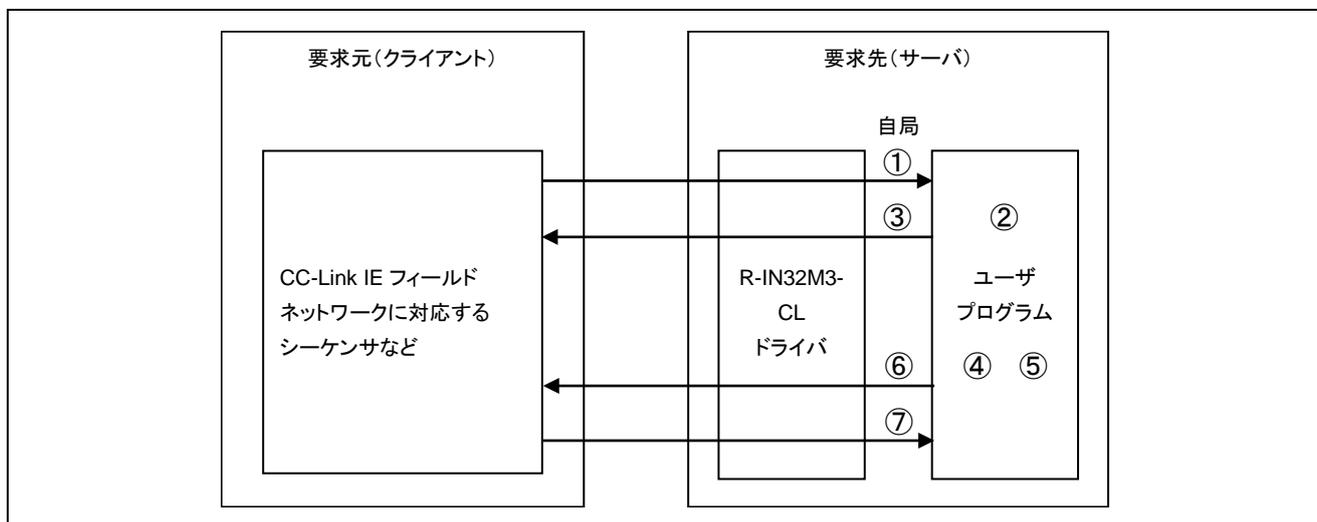


図5.6 Transient2 要求の受信手順

表5.5 Transient2 要求の受信手順

| No. | 処理                               | 参照項   |
|-----|----------------------------------|---|
| ①   | Transient2 要求フレームを受信します          | 「6.2.15 Transient1、Transient2、TransientAck受信処理」参照 |
| ②   | TransientAck フレームを作成します          | 「6.2.34 TransientAckフレーム作成」参照                     |
| ③   | TransientAck フレームを送信します          | 「6.2.17 Transient1、Transient2、TransientAck送信処理」参照 |
| ④   | Transient2 要求フレームのコマンドを解析します     | 「6.2.31 Transient2受信データ処理」参照                      |
| ⑤   | コマンドに応じて Transient2 応答フレームを作成します | 「6.2.35 Transient2応答フレーム作成」参照                     |
| ⑥   | Transient2 応答フレームを送信します          | 「6.2.17 Transient1、Transient2、TransientAck送信処理」参照 |
| ⑦   | TransientAck フレームを受信します          | 「6.2.15 Transient1、Transient2、TransientAck受信処理」参照 |

## 5.2.4 Transient2 要求の送信手順

クライアントが Transient2 要求フレームを送信して、サーバからの Transient2 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。

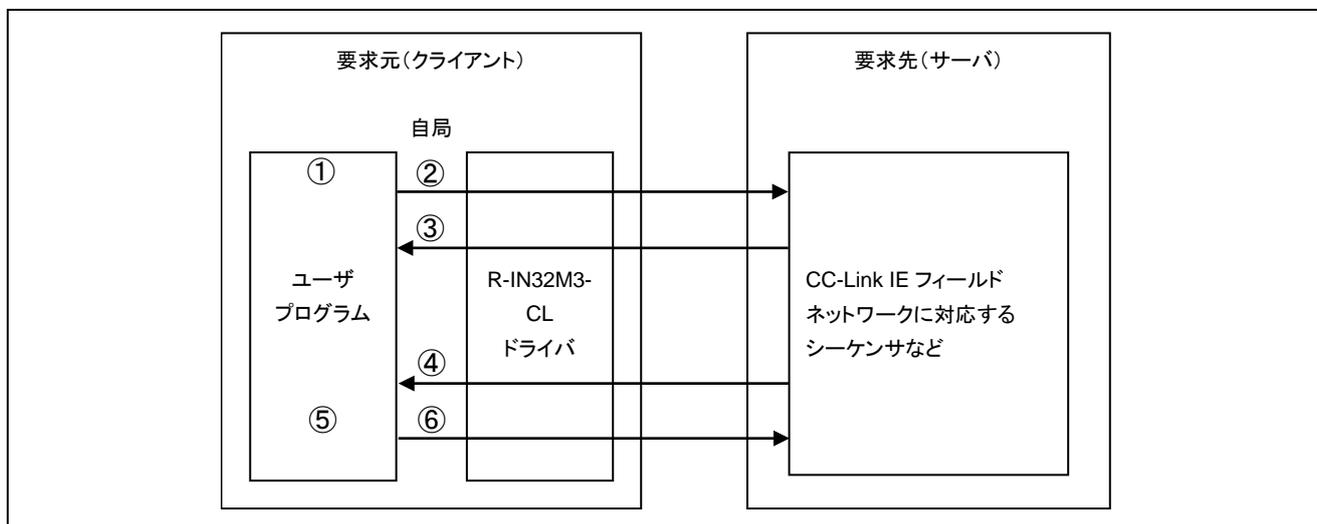


図5.7 Transient2 要求の送信手順

表5.6 Transient2 要求の送信手順

| No. | 処理                      | 参照項   |
|-----|-------------------------|---|
| ①   | Transient2 要求フレームを作成します | 「6.2.16 Transient2要求フレーム作成処理」参照                   |
| ②   | TransientAck フレームを送信します | 「6.2.17 Transient1、Transient2、TransientAck送信処理」参照 |
| ③   | TransientAck フレームを受信します | 「6.2.15 Transient1、Transient2、TransientAck受信処理」参照 |
| ④   | Transient2 応答フレームを受信します | 「6.2.15 Transient1、Transient2、TransientAck受信処理」参照 |
| ⑤   | TransientAck フレームを作成します | 「6.2.34 TransientAckフレーム作成」参照                     |
| ⑥   | TransientAck フレームを送信します | 「6.2.17 Transient1、Transient2、TransientAck送信処理」参照 |

### 5.3 トランジェント伝送のフレームフォーマット概要

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。Ethernet フレームのサイズは、MAC ヘッダから FCS までの 64～1518 バイトです。

本節では、ベンダーがユーザプログラム内で設定する必要がある以下のトランジェントフレームについて説明します。

- Transient1 フレーム (CC-Link IE フィールド固有トランジェント伝送および SLMP)
- Transient2 フレーム (CC-Link 互換トランジェント伝送)
- TransientAck フレーム

#### 5.3.1 トランジェントフレーム共通フォーマット

トランジェントフレーム共通フォーマットは、トランジェントフレームで共通に使用するフォーマットです。

表5.7 トランジェントフレーム共通フォーマット概要

| No. | 項目             | サイズ (バイト) | 備考                                |
|-----|----------------|-----------|-----------------------------------|
| 1   | MAC ヘッダ        | 14        | —                                 |
|     | CC-Link IE ヘッダ | 14        |                                   |
| 2   | トランジェントデータ     | 1482      | —                                 |
| 3   | DCS            | 4         | Data Check Sequence <sup>注</sup>  |
| 4   | FCS            | 4         | Frame Check Sequence <sup>注</sup> |

**注. R-IN32M3-CL が自動的に計算し付加します。**

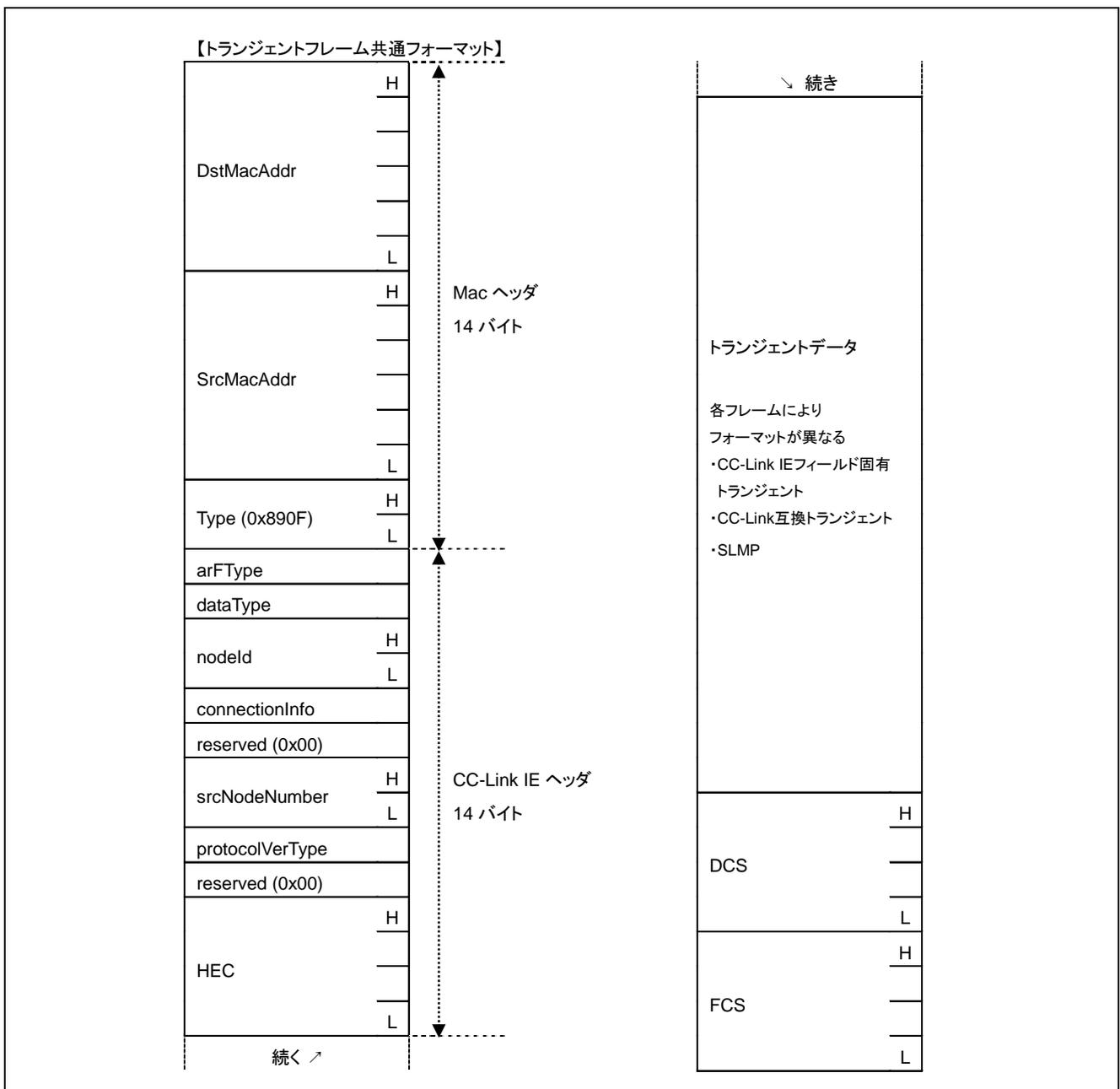


図5.8 トランジェントフレーム共通フォーマット

表5.8 MAC ヘッダ項目一覧

| 項目                          | 内容                  | 値           | 備考   |
|-----------------------------|---------------------|-------------|--|
| Dst/SrcMacAddr<br>(第1オクテット) | 宛先/送信元の<br>MAC アドレス | IEEE が管理する値 | MAC アドレスが、01-23-45-67-89-AB のとき、0x01<br>I/G ビットは、0b に設定してください。<br>1b に設定するとマルチキャストアドレスとなり、マスタ局と正常に通信できません。 |
| Dst/SrcMacAddr<br>(第2オクテット) | 宛先/送信元の<br>MAC アドレス |             | MAC アドレスが、01-23-45-67-89-AB のとき、0x23   |
| Dst/SrcMacAddr<br>(第3オクテット) | 宛先/送信元の<br>MAC アドレス |             | MAC アドレスが、01-23-45-67-89-AB のとき、0x45   |
| Dst/SrcMacAddr<br>(第4オクテット) | 宛先/送信元の<br>MAC アドレス | ベンダーで管理する値  | MAC アドレスが、01-23-45-67-89-AB のとき、0x67   |
| Dst/SrcMacAddr<br>(第5オクテット) | 宛先/送信元の<br>MAC アドレス |             | MAC アドレスが、01-23-45-67-89-AB のとき、0x89   |
| Dst/SrcMacAddr<br>(第6オクテット) | 宛先/送信元の<br>MAC アドレス |             | MAC アドレスが、01-23-45-67-89-AB のとき、0xAB   |
| Type                        | フレームタイプ             | 0x890F 固定   | CC-Link IE フィールドネットワークの伝送フレームであることを示す。   |

注. 本表の全項目はビッグエンディアンで設定してください。

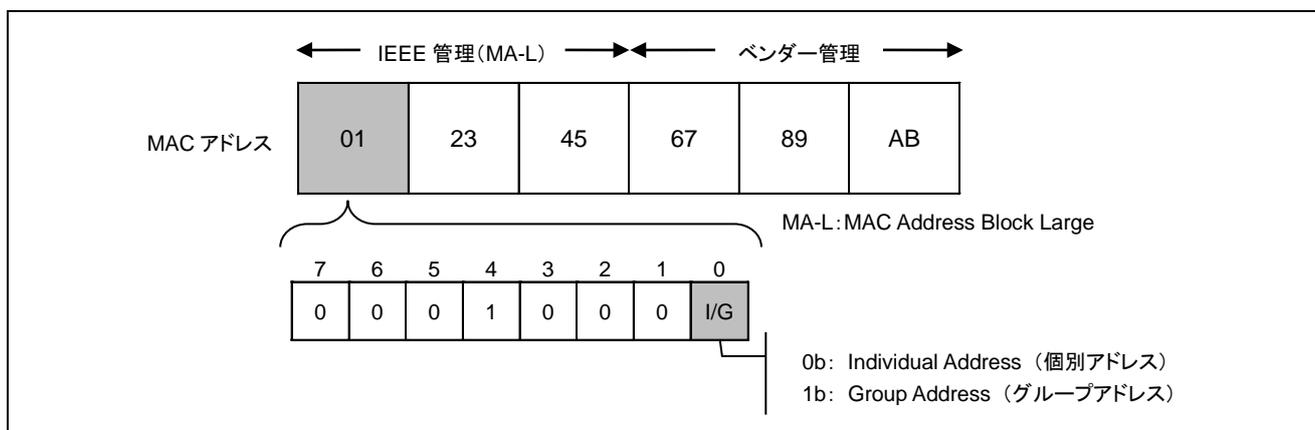


図5.9 MAC アドレス I/G ビット

表5.9 CC-Link IE ヘッダ項目一覧

| 項目              | 内容                   |            | 値                                   | 備考  |
|-----------------|----------------------|------------|-------------------------------------|---|
| arFType         | フレームの種別              |            | 「表5.1 トランジェントフレームの一覧と実装要否」参照してください。 | —   |
| dataType        | データの種別               |            |                                     | —   |
| nodeId          | ノード識別子               |            | 0x0000~0x00F0<br>(0~240)            | マスタ局の各スレーブの管理情報です。<br>(局番と異なる)<br>gusR_IN32_GetNodeID 関数で取得します。 <sup>注1</sup><br>ビッグエンディアンで設定してください。  |
| connectionInfo  | トランジェントの識別情報         |            | 0x01~0xFF<br>(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。<br>gerR_IN32_GetSendTransientBuffer 関数で取得します。 <sup>注2</sup> |
| reserved        | 予約                   |            | 0x00 固定                             | —   |
| srcNodeNumber   | 自ノード番号               |            | 0x0001~0x0078<br>(1~120)            | ビッグエンディアンで設定してください。   |
| protocolVerType | Bit7-4               | プロトコルバージョン | 0x0 固定                              | —   |
|                 | Bit3-0               | プロトコル種別    | 0x1 固定                              | 0x01 : CC-Link IE フィールドネットワーク   |
| HEC             | Header Error Control |            | R-IN32M3-CL が自動的に計算します。             | —   |

注 1. 「6.4.11(2) gusR\_IN32\_GetNodeID」を参照してください。

2. 「6.4.11(5) gerR\_IN32\_GetSendTransientBuffer」を参照してください。

### 5.3.2 CC-Link IE フィールド固有トランジェントフレームフォーマット

CC-Link IE フィールド固有トランジェントフレームフォーマットの概要を下表に示します。

表5.10 CC-Link IE フィールド固有トランジェントフレームフォーマット概要

| No. | 項目                               | サイズ (バイト)    | 備考                                    |   |
|-----|----------------------------------|--------------|---------------------------------------|---|
| 1   | MAC ヘッダ                          | 14           | 「5.3.1 トランジェントフレーム共通フォーマット」を参照してください。 |   |
|     | CC-Link IE ヘッダ                   | 14           |                                       |   |
| 2   | Transient1 ヘッダ                   | 16           | —                                     |   |
| 3   | Transient1<br>データ部 <sup>注2</sup> | 拡張ヘッダ<br>データ | 20<br>0~1446                          | — |
|     |                                  |              |                                       |   |
| 4   | DCS                              | 4            | Data Check Sequence <sup>注1</sup>     |   |
| 5   | FCS                              | 4            | Frame Check Sequence <sup>注1</sup>    |   |

注 1. R-IN32M3-CL が自動的に計算し付加します。

2. Transient1 データ部を“CC-Link IE フィールド固有トランジェント”として使う場合を示します。  
Transient1 データ部を“SLMP”として使う場合、「5.3.5 SLMPフレームフォーマット」を参照してください。

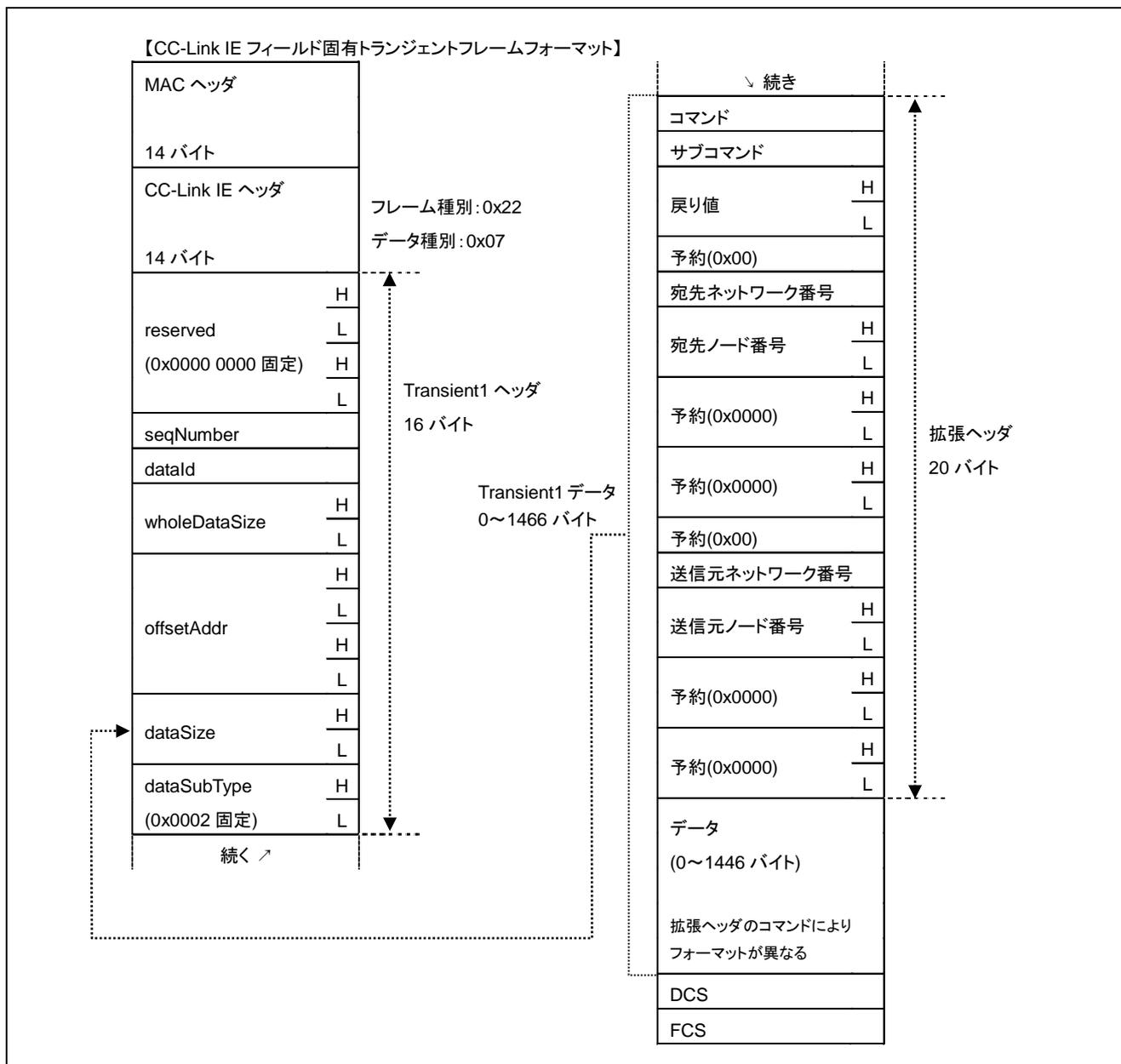


図5.10 CC-Link IE フィールド固有トランジェントフレームフォーマット概略

(1) MAC ヘッダ、CC-Link IE ヘッダ

「5.3.1 トランジェントフレーム共通フォーマット」を参照してください。

(2) Transient1 ヘッダ

Transient1 ヘッダは、Transient1 フレーム（CC-Link IE フィールド固有トランジェントと SLMP）に付加するヘッダです。

表5.11 Transient1 ヘッダ項目一覧

| 項目            | 内容                          |                           | 値                                | 備考   |
|---------------|-----------------------------|---------------------------|----------------------------------|--|
| reserved      | 予約                          |                           | 0x00000000 固定                    | —  |
| seqNumber     | Bit7                        | 最終フレーム判別                  | 0b : 分割途中フレーム<br>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。  |
|               | Bit6-0                      | Transient1 フレームのシーケンシャル番号 | 0x00~0x7F                        |  |
| dataId        | トランジェントデータの識別番号             |                           | 0x00~0xFF                        | 番号は任意ですが、分割したフレームは同じ識別番号を設定してください。   |
| wholeDataSize | Transient1 データ部全体のサイズ       |                           | 0x0000~0x0800 (0~2048)           | 分割前のトランジェントデータ全体のサイズ   |
| 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 |                           | 0x0000 0000~0x7FFF FFFF          | 分割しない場合は0 固定<br>分割する場合の先頭フレーム : 0 固定<br>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。 <sup>注</sup> |
| dataSize      | 本フレーム内のトランジェントデータのサイズ       |                           | 0x0000~0x05BA (0~1466)           | 分割後のトランジェントデータのサイズ <sup>注</sup>  |
| dataSubType   | データサブ種別                     |                           | 0x0002 : システム固有<br>0x0002 : SLMP | 注  |

**注. ビッグエンディアンで設定してください。**

Transient1 ヘッダのシーケンシャル番号と識別番号の関係を以下の例で説明します。

トランジェントデータ 1 個目は分割なし、  
トランジェントデータ 2 個目は 3 分割、  
トランジェントデータ 3 個目は 2 分割

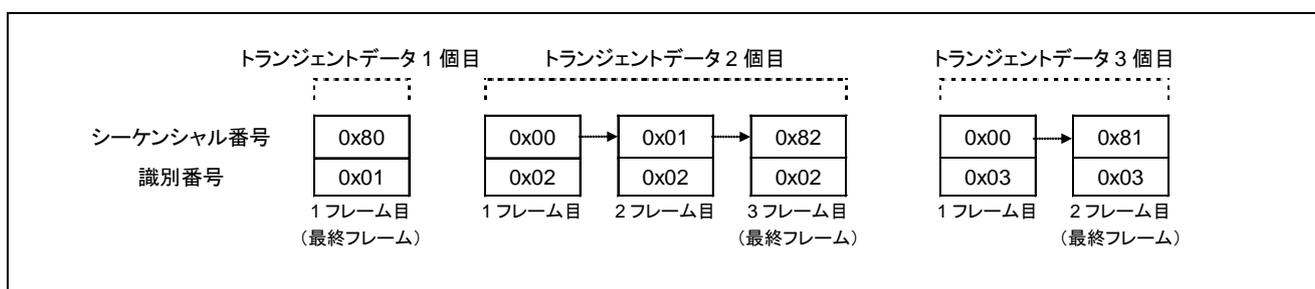


図5.11 Transient1 ヘッダ トランジェントデータのシーケンシャル番号と識別番号の関係

## (3) Transient1 データ部

CC-Link IE フィールド固有トランジェント伝送として使用する場合は、拡張ヘッダとデータで構成します。

表5.12 拡張ヘッダ項目一覧

| 項目          | 内容           | 値   |                                   | 備考 |
|-------------|--------------|---|-----------------------------------|----|
| コマンド        | コマンド         | 「表5.13 CC-Link IEフィールド固有トランジェント伝送コマンド一覧」を参照してください。  |                                   | —  |
| サブコマンド      | サブコマンド       |   |                                   |    |
| 戻り値         | 要求に対する戻り値    | 要求時   | 0x0000 固定                         | 注  |
|             |              | 応答時   | 0x0000 : 正常<br>0x0001~0xFFFF : 異常 | —  |
| 予約          | 予約           | 0 固定  |                                   | —  |
| 宛先ネットワーク番号  | 宛先ネットワーク番号   | 0 : ブロードキャスト<br>1~239 : 宛先ネットワーク                    |                                   | —  |
| 宛先ノード番号     | 宛先ノード番号      | 1~120 : スレーブ局<br>0x007D : マスタ局<br>0xFFFF : ブロードキャスト |                                   | 注  |
| 予約          | 予約           | 0 固定  |                                   | —  |
| 予約          | 予約           | 0 固定  |                                   | —  |
| 予約          | 予約           | 0 固定  |                                   | —  |
| 送信元ネットワーク番号 | 送信元のネットワーク番号 | 1~239   |                                   | —  |
| 送信元ノード番号    | 送信元のノード番号    | 1~120   |                                   | 注  |
| 予約          | 予約           | 0 固定  |                                   | —  |
| 予約          | 予約           | 0 固定  |                                   | —  |

注. ビッグエンディアンで設定してください。

表5.13 CC-Link IE フィールド固有トランジェント伝送コマンド一覧

| コマンド | サブコマンド | コマンド種別                   | 送信方向       | 備考   |
|------|--------|--------------------------|------------|------|
| 0x01 | 0x00   | ノード情報配信要求                | マスタ局→スレーブ局 | 応答不要 |
| 0x03 | 0x00   | 統計情報取得要求                 | マスタ局→スレーブ局 | —    |
| 0x03 | 0x80   | 統計情報取得応答                 | マスタ局←スレーブ局 | —    |
| 0x04 | 0x00   | ノード詳細情報取得要求              | マスタ局→スレーブ局 | —    |
| 0x04 | 0x80   | ノード詳細情報取得応答              | マスタ局←スレーブ局 | —    |
| 0x0A | 0x00   | オプション情報取得要求              | マスタ局→スレーブ局 | —    |
| 0x0A | 0x80   | オプション情報取得応答 <sup>注</sup> | マスタ局←スレーブ局 | —    |

注. オプション情報取得応答はインタフェース関数を使用して作成しますので、フレームフォーマットを意識する必要はありません。

(a) ノード情報配信

ノード情報配信は、スレーブ局がクライアント機能を使用するときに必要な宛先 MAC アドレスを、マスター局からスレーブ局へ配信するフレームです。

TransientAck および Transient1 応答フレーム（ノード情報配信要求の応答）の送信は不要です。

ノード情報の配信個数が 60 個以上の場合、フレームサイズが Ethernet フレームの最大値 1518 バイトを超えるため、2つのフレームに分割されます。この場合、ユーザプログラムで Transient1 受信データ組立て処理が必要となります。

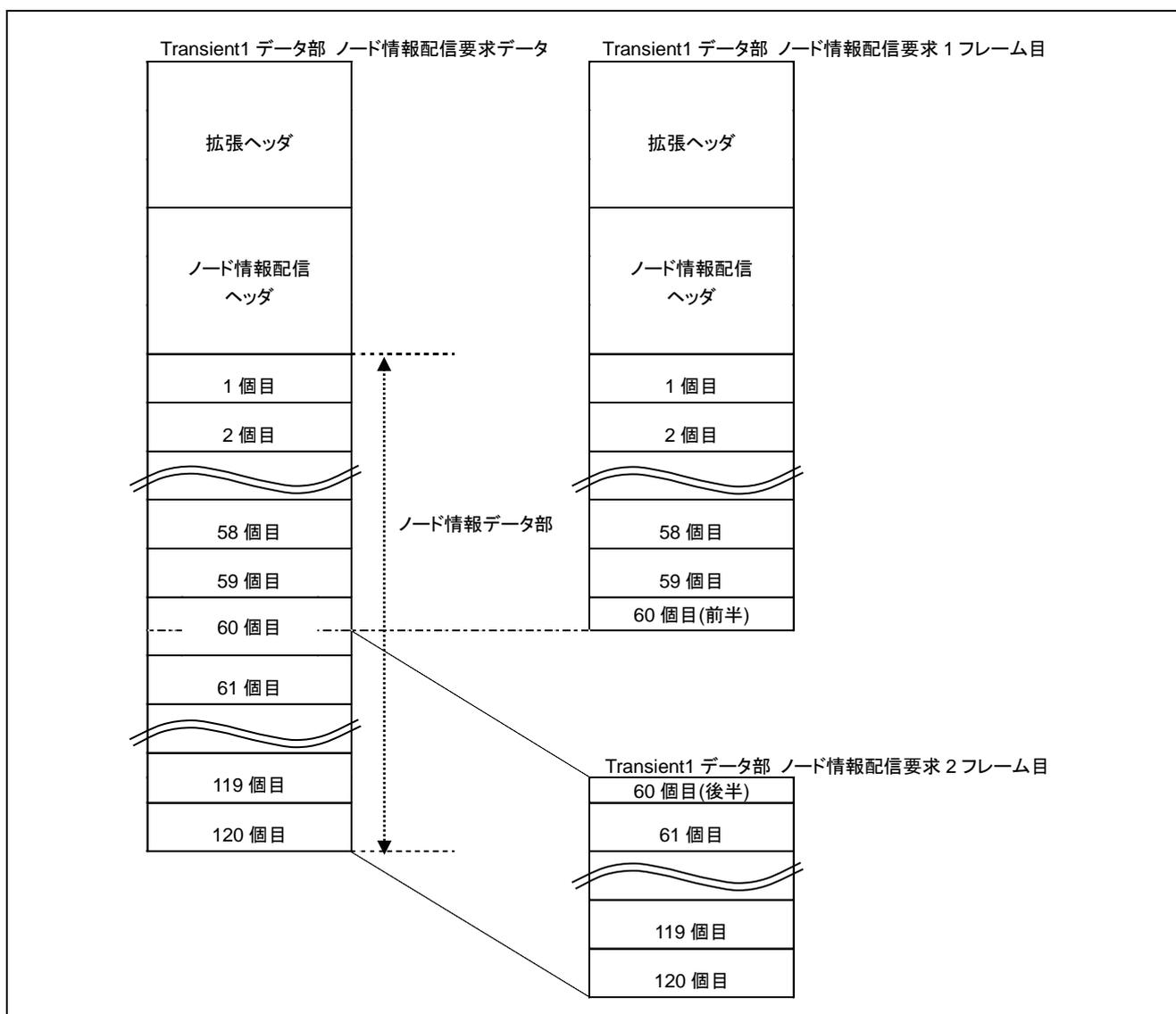


図5.12 Transient1 データ部 ノード情報配信要求の分割時のフレーム

フレームフォーマットは下表にしたがって、図5.13、図5.14および図5.15を参照してください。

表5.14 ノード情報配信のフレームフォーマット一覧

| 配信個数  | 参照先図番号   |
|-------|--|
| 60個未満 | 図5.13 Transient1データ部 ノード情報配信要求   |
| 60個以上 | 図5.14 Transient1データ部 ノード情報配信要求1フレーム目<br>図5.15 Transient1データ部 ノード情報配信要求2フレーム目 |

Transient1 受信データ組立て処理の詳細は「6.2.19 Transient1受信データ組立て開始」および「6.2.20 Transient1受信データ組立て」を参照してください。

配信個数が 60 個未満のノード情報配信要求フレームフォーマットを以下に示します。

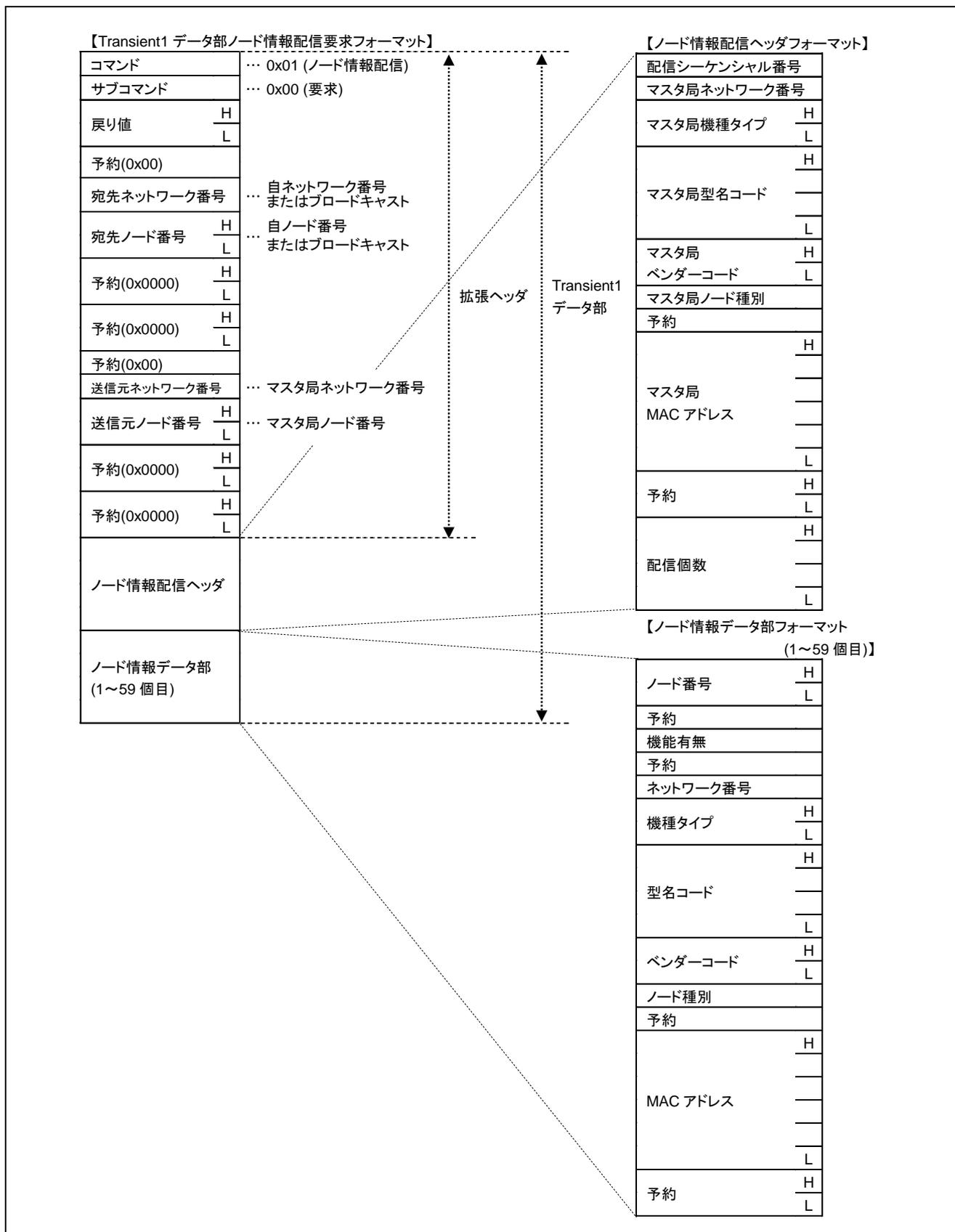


図5.13 Transient1 データ部 ノード情報配信要求

配信個数が 60 個以上のノード情報配信要求フレームフォーマットを以下に示します。

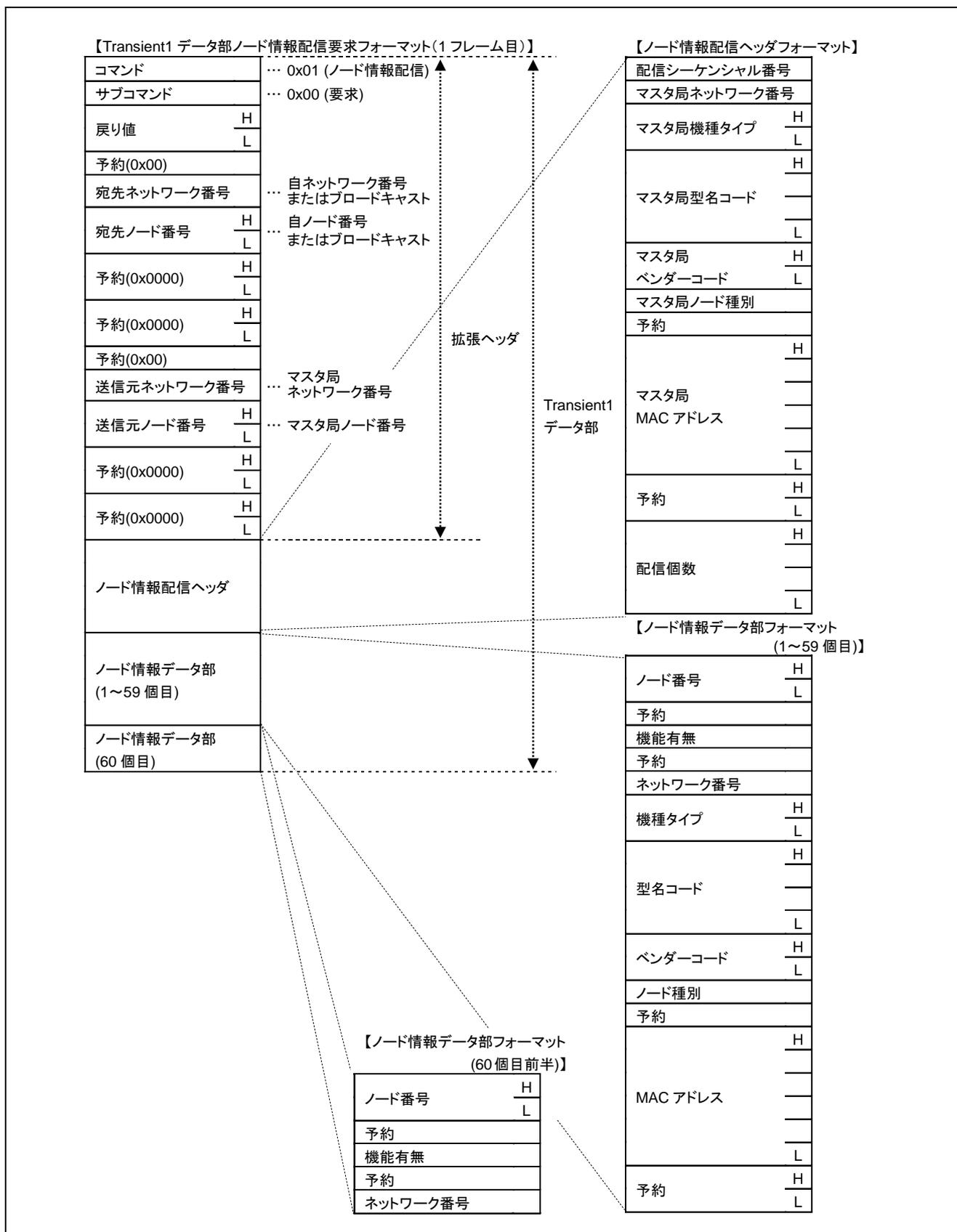


図5.14 Transient1 データ部 ノード情報配信要求 1 フレーム目

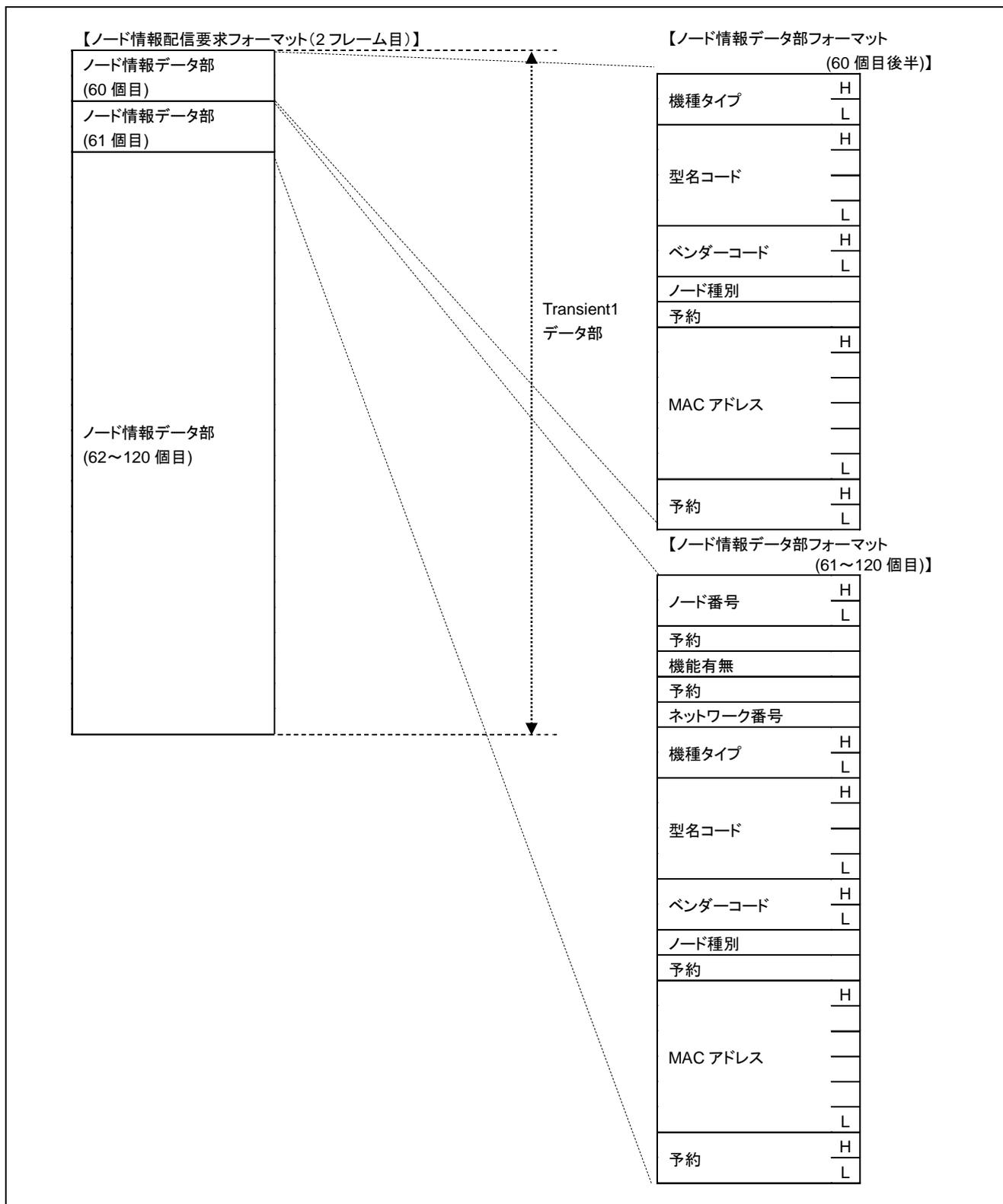


図5.15 Transient1 データ部 ノード情報配信要求 2 フレーム目

表5.15 ノード情報配信ヘッダ項目一覧

| 項目            | 内容             | 値                         | 備考                                     |
|---------------|----------------|---------------------------|--|
| 配信シーケンシャル番号   | 配信シーケンシャル番号    | 1~7                       | 配信シーケンシャル番号が同じときはノード情報が同じですので破棄してください。 |
| マスタ局ネットワーク番号  | マスタ局のネットワーク番号  | 1~239                     | —                                      |
| マスタ局機種タイプ     | マスタ局の機種タイプ     | 0x0001~0xFFFF             | CC-Link 協会が管理する機種タイプ                   |
| マスタ局型名コード     | マスタ局の型名コード     | 0x00000000~<br>0xFFFFFFFF | 同一ベンダーコード内で一意となるネットワークの型名コード           |
| マスタ局ベンダーコード   | マスタ局のベンダーコード   | 0x0000~0xFFFF             | CC-Link 協会が管理するベンダーコード                 |
| マスタ局ノード種別     | マスタ局のノード種別     | 固定値：0x30                  | —                                      |
| 予約            | 予約             | 固定値：0x00                  | —                                      |
| マスタ局 MAC アドレス | マスタ局の MAC アドレス | 6 バイトの<br>MAC アドレス        | —                                      |
| 予約            | 予約             | 固定値：0x00                  | —                                      |
| 配信個数          | ノード情報の配信個数     | 1~120                     | —                                      |

表5.16 ノード情報データ部項目一覧

| 項目       | 内容                  | 値                         | 備考                           |
|----------|---------------------|---------------------------|------------------------------|
| ノード番号    | スレーブ局のノード番号         | 1~120                     | —                            |
| 予約       | 予約                  | 固定値：0x00                  | —                            |
| 機能有無     | スレーブ局のトランジェント受信機能有無 | あり：0x01<br>なし：0x00        | —                            |
| 予約       | 予約                  | 固定値：0x00                  | —                            |
| ネットワーク番号 | スレーブ局のネットワーク番号      | 1~239                     | —                            |
| 機種タイプ    | スレーブ局の機種タイプ         | 0x0001~0xFFFF             | CC-Link 協会が管理する機種タイプ         |
| 型名コード    | スレーブ局の型名コード         | 0x00000000~<br>0xFFFFFFFF | 同一ベンダーコード内で一意となるネットワークの型名コード |
| ベンダーコード  | スレーブ局のベンダーコード       | 0x0000~0xFFFF             | CC-Link 協会が管理するベンダーコード       |
| ノード種別    | スレーブ局のノード種別         | 「表5.17 ノード種別一覧」を参照してください。 | —                            |
| 予約       | 予約                  | 固定値：0x00                  | —                            |
| MAC アドレス | スレーブ局の MAC アドレス     | 6 バイトの<br>MAC アドレス        | —                            |
| 予約       | 予約                  | 固定値：0x0000                | —                            |

表5.17 ノード種別一覧

| ノード種別 | 内容            | 備考 |
|-------|---------------|----|
| 0x30  | マスタ局          | —  |
| 0x31  | 予約            | —  |
| 0x32  | ローカル局         | —  |
| 0x33  | インテリジェントデバイス局 | —  |
| 0x34  | リモートデバイス局     | —  |
| 0x35  | リモート I/O 局    | —  |

(b) 統計情報取得

統計情報取得は、マスタ局がスレーブ局のポート 1 およびポート 2 に関するエラー情報を収集するために使用します。

統計情報取得要求フレームフォーマットを以下に示します。

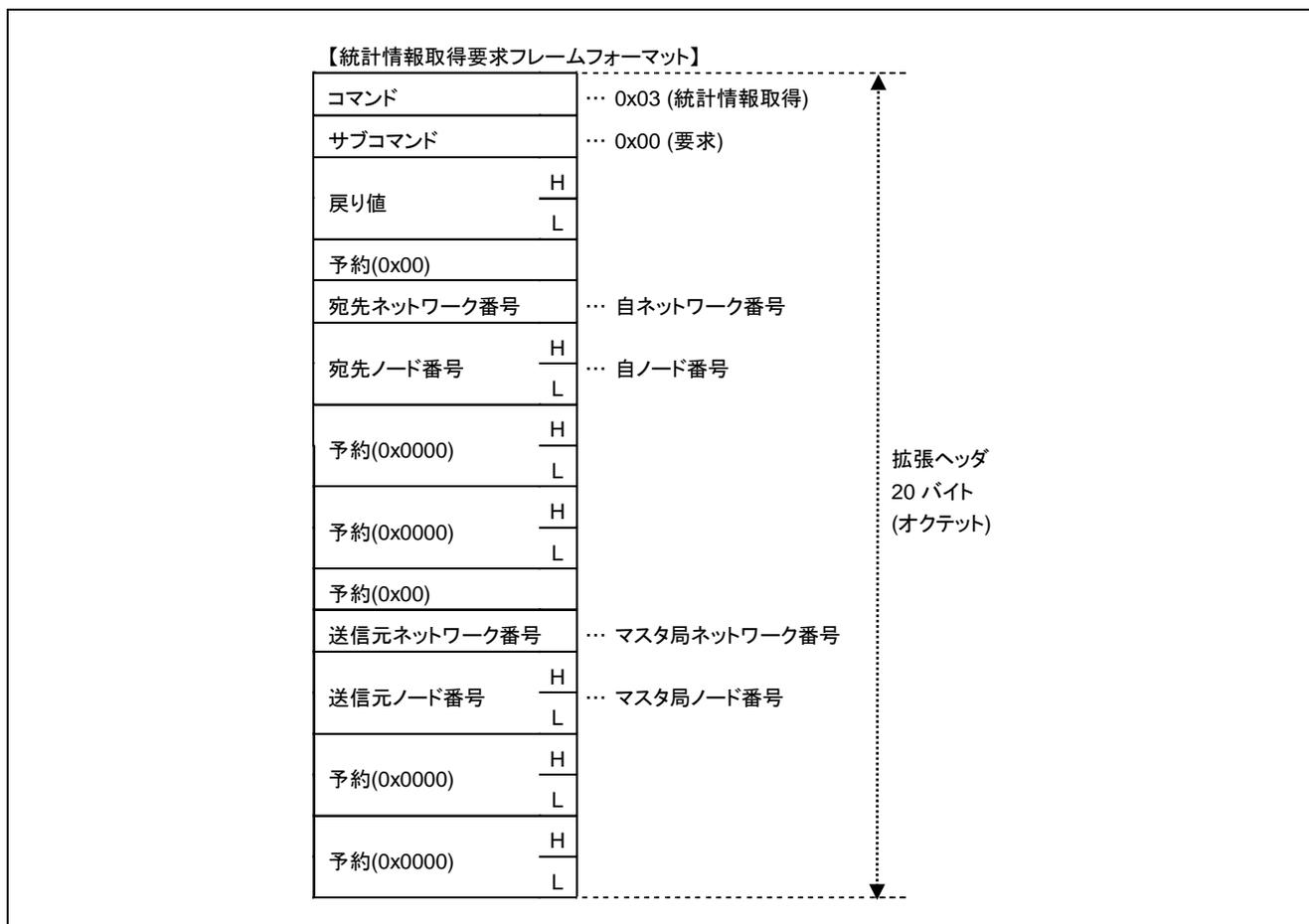


図5.16 Transient1 データ部 統計情報取得要求

上図の各項目については、「表5.12 拡張ヘッダ項目一覧」を参照してください。

統計情報取得応答フレームフォーマットを以下に示します。

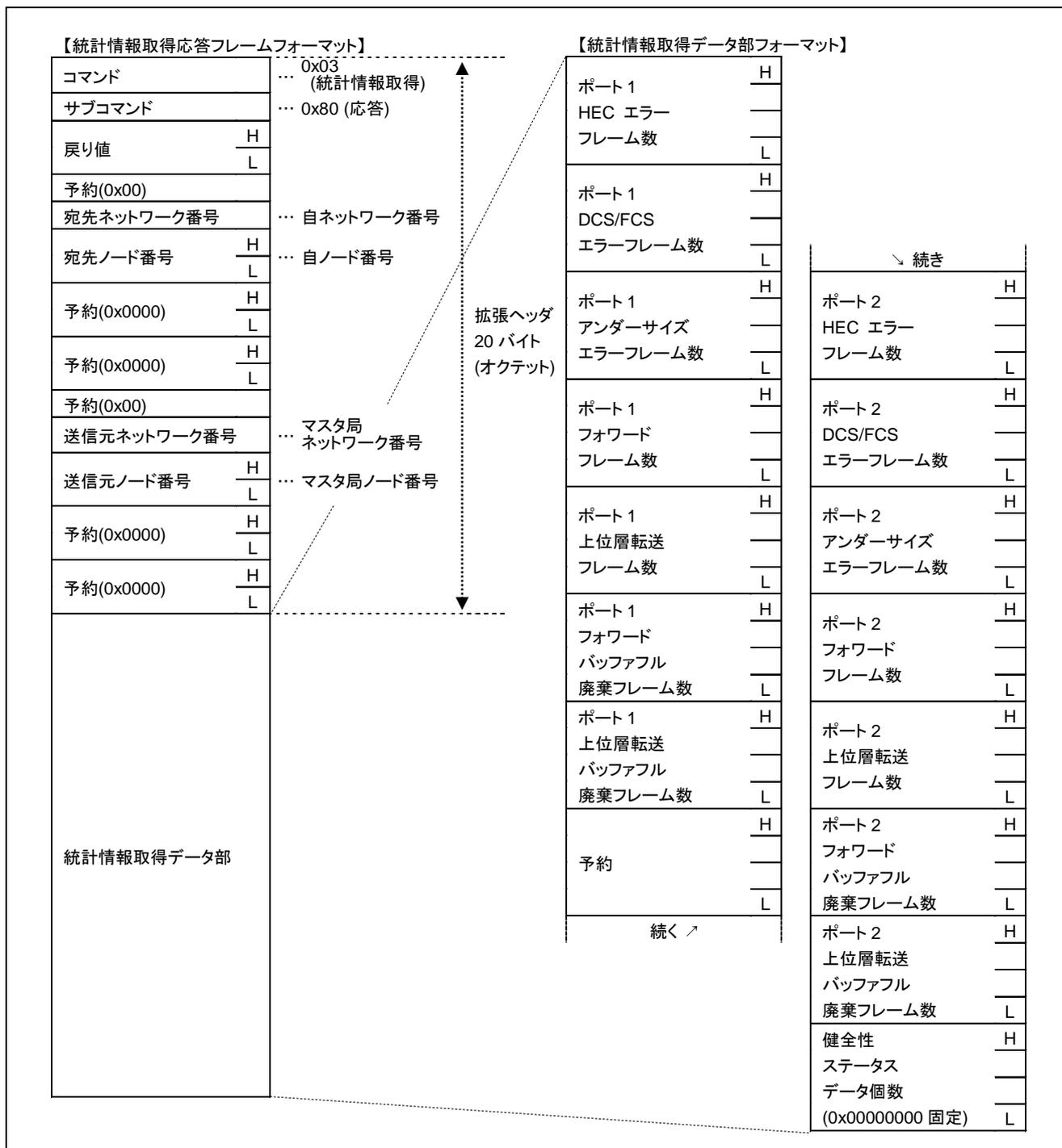


図5.17 Transient1 データ部 統計情報取得応答

下表に示す統計情報取得応答フレームのデータ部各項目は、gerR\_IN32\_GetMIB 関数で取得した値を設定しますので「6.4.6(6) gerR\_IN32\_GetMIB」を参照してください。

表5.18 統計情報取得応答データ項目一覧

| 項目                         | 内容                              | 値              | 備考  |
|----------------------------|---------------------------------|----------------|---|
| ポート1HEC<br>エラーフレーム数        | ポート1のHEC<br>エラーフレーム数            | 0~4294967295   | 受信フレームのHECエラーを<br>カウントする。                   |
| ポート1DCS/FCS<br>エラーフレーム数    | ポート1のDCS/FCS<br>エラーフレーム数        | 0~4294967295   | 受信フレームのDCS/FCSエラ<br>ーをカウントする。               |
| ポート1アンダーサイズ<br>エラーフレーム数    | ポート1のアンダーサイズ<br>エラーフレーム数        | 0~4294967295   | 受信フレームサイズが28バイ<br>ト未満の場合にカウントする。            |
| ポート1フォワード<br>フレーム数         | ポート1のフォワード<br>フレーム数             | 0~4294967295   | フォワードしたフレーム数を<br>カウントする。                    |
| ポート1上位層転送<br>フレーム数         | ポート1の上位層転送<br>フレーム数             | 0~4294967295   | 上位層に転送したフレーム数<br>をカウントする。                   |
| ポート1フォワード<br>バッファフル廃棄フレーム数 | ポート1のフォワード<br>バッファフル廃棄<br>フレーム数 | 0~4294967295   | フォワードバッファフルによ<br>り、破棄されたフレーム数をカ<br>ウントする。   |
| ポート1上位層転送<br>バッファフル廃棄フレーム数 | ポート1の上位層転送<br>バッファフル廃棄<br>フレーム数 | 0~4294967295   | 上位層への転送バッファフル<br>により、破棄されたフレーム数<br>をカウントする。 |
| 予約                         | 予約                              | 固定値：0x00000000 | —   |
| ポート2HEC<br>エラーフレーム数        | ポート2のHEC<br>エラーフレーム数            | 0~4294967295   | 受信フレームのHECエラーを<br>カウントする。                   |
| ポート2DCS/FCS<br>エラーフレーム数    | ポート2のDCS/FCS<br>エラーフレーム数        | 0~4294967295   | 受信フレームのDCS/FCSエラ<br>ーをカウントする。               |
| ポート2アンダーサイズ<br>エラーフレーム数    | ポート2のアンダーサイズ<br>エラーフレーム数        | 0~4294967295   | 受信フレームサイズが28バイ<br>ト未満の場合にカウントする。            |
| ポート2フォワード<br>フレーム数         | ポート2のフォワード<br>フレーム数             | 0~4294967295   | フォワードしたフレーム数を<br>カウントする。                    |
| ポート2上位層転送<br>フレーム数         | ポート2の上位層転送<br>フレーム数             | 0~4294967295   | 上位層に転送したフレーム数<br>をカウントする。                   |
| ポート2フォワード<br>バッファフル廃棄フレーム数 | ポート2のフォワード<br>バッファフル廃棄フレーム数     | 0~4294967295   | フォワードバッファフルによ<br>り、破棄されたフレーム数をカ<br>ウントする。   |
| ポート2上位層転送<br>バッファフル廃棄フレーム数 | ポート2の上位層転送<br>バッファフル廃棄フレーム数     | 0~4294967295   | 上位層への転送バッファフル<br>により、破棄されたフレーム数<br>をカウントする。 |
| 健全性ステータスデータ個数              | 健全性ステータスデータ個数                   | 固定値：0x00000000 | —   |

(c) ノード詳細情報取得

ノード詳細情報取得は、マスタ局がスレーブ局のノード詳細情報を収集するために使用します。

ノード詳細情報取得要求フレームフォーマットを以下に示します。

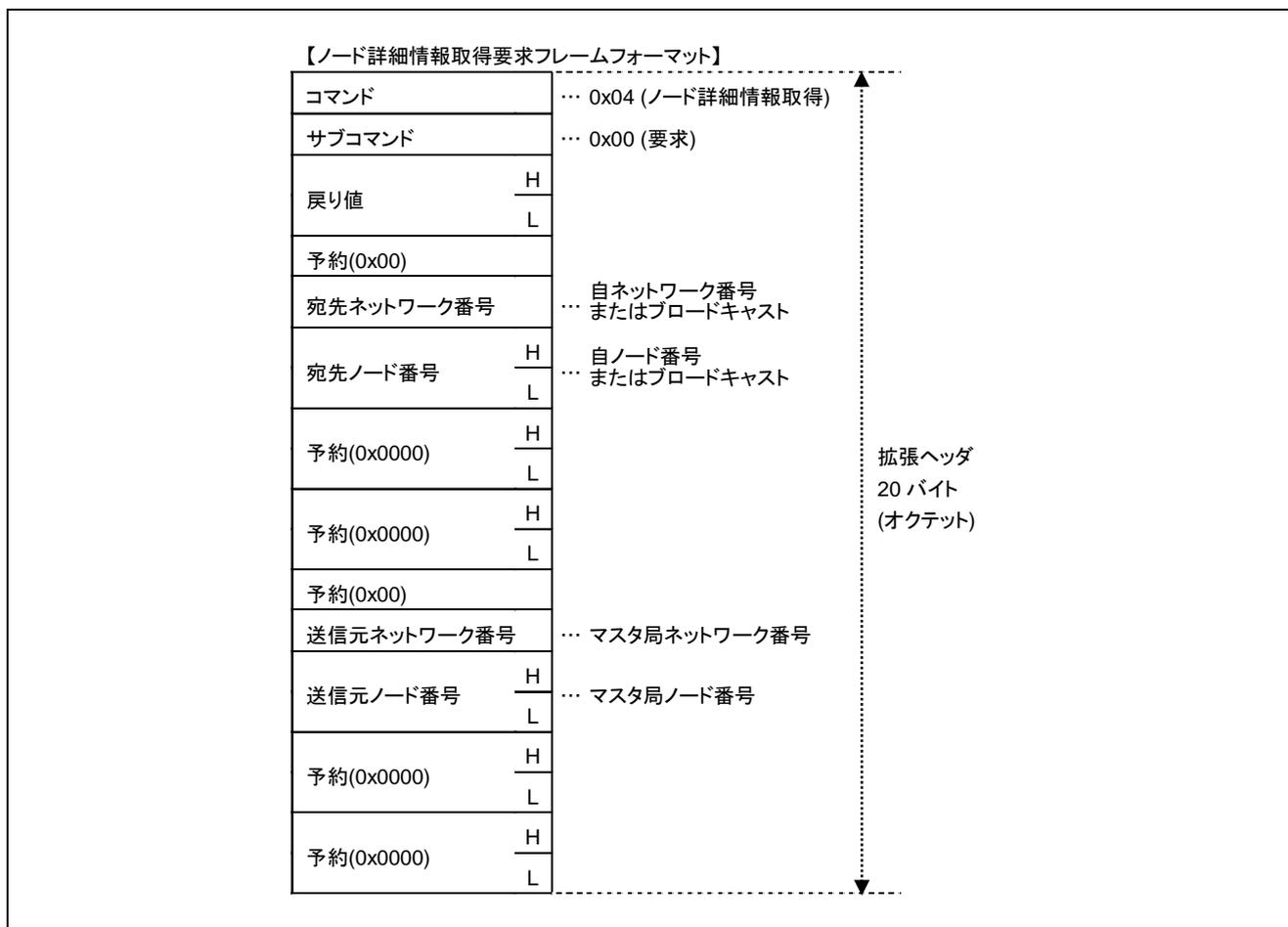


図5.18 Transient1 データ部 ノード詳細情報取得要求

上図の各項目については、「表5.12 拡張ヘッダ項目一覧」を参照してください。

ノード詳細情報取得応答フレームフォーマットを以下に示します。

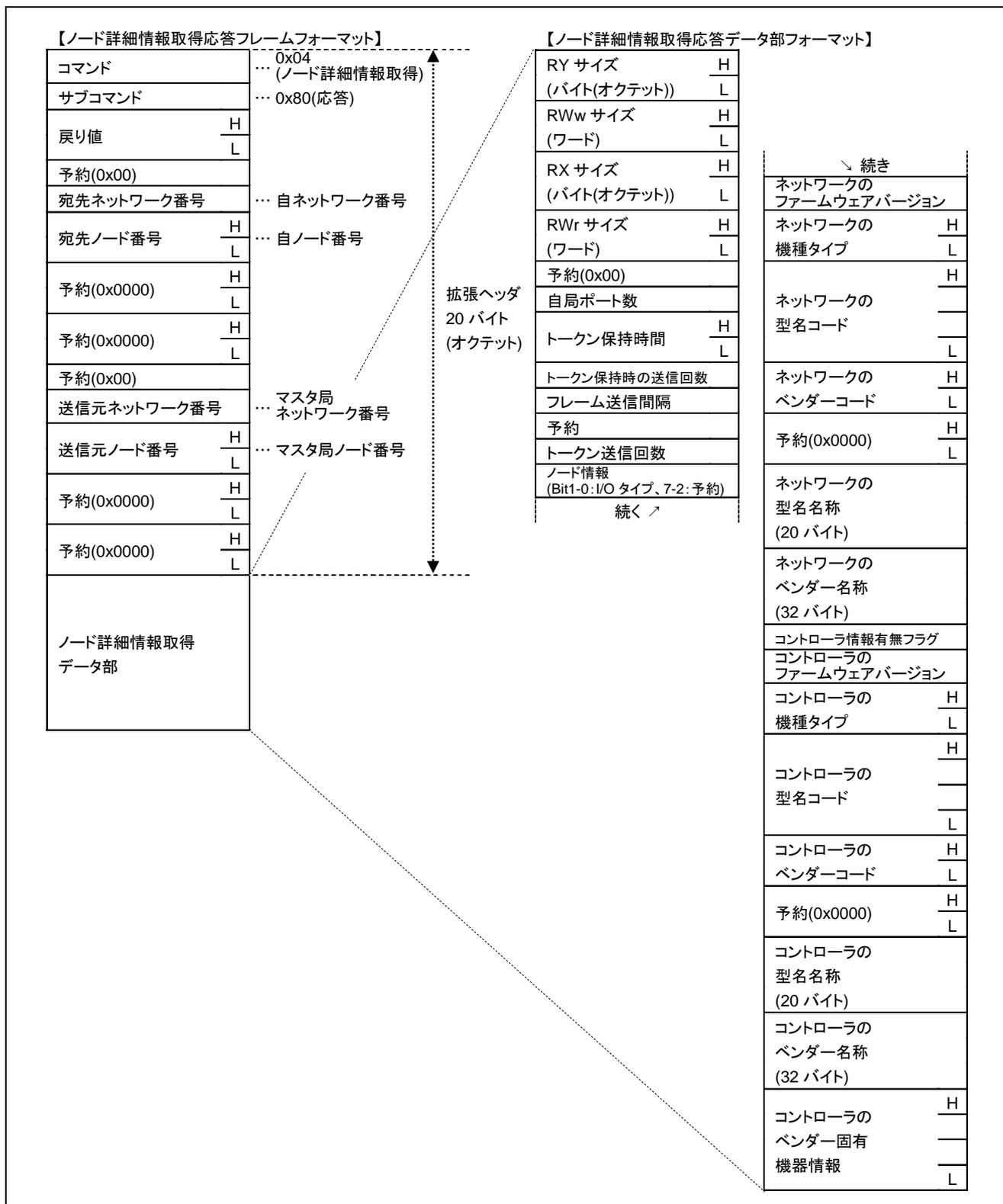


図5.19 Transient1 データ部 ノード詳細情報取得応答

下表に示すノード詳細情報取得応答フレームのデータ部各項目は、gerR\_IN32\_GetUnitInformation 関数で取得した値を設定しますので、「6.4.1(2) gerR\_IN32\_Initialize」を参照してください。

表5.19 ノード詳細情報取得応答データ部項目一覧

| 項目                   | 内容  | 値  | 備考 |
|----------------------|---|--|----|
| RY サイズ (バイト (オクテット)) | 自局の RY サイズ  | 最小値 : 0 最大値 : 256                                  | —  |
| RWw サイズ (ワード)        | 自局の RWw サイズ   | 最小値 : 0 最大値 : 1024                                 | —  |
| RX サイズ (バイト (オクテット)) | 自局の RX サイズ  | 最小値 : 0 最大値 : 256                                  | —  |
| RWr サイズ (ワード)        | 自局の RWr サイズ   | 最小値 : 0 最大値 : 1024                                 | —  |
| 予約                   | 予約  | 固定値 : 0x00   | —  |
| 自局ポート数               | 自局のポート数   | 1~2  | —  |
| トークン保持時間             | 自局がトークンを保持する時間の最大値 (us)   | 1~32767  | —  |
| トークン保持時の送信回数設定       | トークン保持時の Token フレーム以外のフレーム送信回数                                    | 1~255  | —  |
| フレーム送信間隔設定           | Token フレーム受信後、MyStatus フレームを送信するまでのフレーム間隔                         | 1~255  | —  |
| 予約                   | 予約  | 固定値 : 0x00   | —  |
| トークン送信回数設定           | トークン保持時に送信する Token フレームの送信繰返し回数                                   | 1~255  | —  |
| ノード情報 (I/O タイプ)      | I/O タイプ   | 裏表混在 : 0x00<br>入力 : 0x01<br>出力 : 0x02<br>混在 : 0x03 | —  |
| 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   | 0~255  | —  |
| ネットワークの機種タイプ         | ネットワークの機種タイプ  | 0x0001~0xFFFF                                      | —  |
| ネットワークの型名コード         | ネットワークの型名コード  | 0x00000000~0xFFFFFFFF                              | —  |
| ネットワークのベンダーコード       | ネットワークのベンダーコード  | 0x0000~0xFFFF                                      | —  |
| 予約                   | 予約  | 固定値 : 0x0000                                       | —  |
| ネットワークの型名名称          | ネットワークの型名名称   | 型名名称 20 バイト  | —  |
| ネットワークのベンダー名称        | ネットワークのベンダー名称   | ベンダー名称 32 バイト                                      | —  |
| コントローラ情報有無フラグ        | コントローラ情報 (「コントローラのファームウェアバージョン」から「コントローラのベンダー機器固有情報」まで) の有効/無効フラグ | 無効 : 0<br>有効 : 1                                   | —  |
| 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   | 0~255  | —  |
| コントローラの機種タイプ         | コントローラの機種タイプ  | 0x0001~0xFFFF                                      | —  |
| コントローラの型名コード         | コントローラの型名コード  | 0x00000000~0xFFFFFFFF                              | —  |
| コントローラのベンダーコード       | コントローラのベンダーコード  | 0x0000~0xFFFF                                      | —  |
| 予約                   | 予約  | 固定値 : 0x0000                                       | —  |
| コントローラの型名名称          | コントローラの型名名称   | 型名名称 20 バイト  | —  |
| コントローラのベンダー名称        | コントローラのベンダー名称   | ベンダー名称 32 バイト                                      | —  |
| コントローラのベンダー機器固有情報    | コントローラのベンダー機器固有情報   | 0x00000000~0xFFFFFFFF                              | —  |

### 5.3.3 TransientAck フレームフォーマット

TransientAck フレームフォーマットの概要を下表に示します。

表5.20 TransientAck フレームフォーマット概要

| No. | 項目                | サイズ (バイト) | 備考                                    |
|-----|-------------------|-----------|---------------------------------------|
| 1   | MAC ヘッダ           | 14        | 「5.3.1 トランジェントフレーム共通フォーマット」を参照してください。 |
|     | CC-Link IE ヘッダ    | 14        |                                       |
| 2   | TransientAck データ部 | 28        | 固定値 0x00000001 注                      |
| 3   | DCS               | 4         | Data Check Sequence 注                 |
| 4   | FCS               | 4         | Frame Check Sequence 注                |

注. R-IN32M3-CL が自動的に計算し付加します。

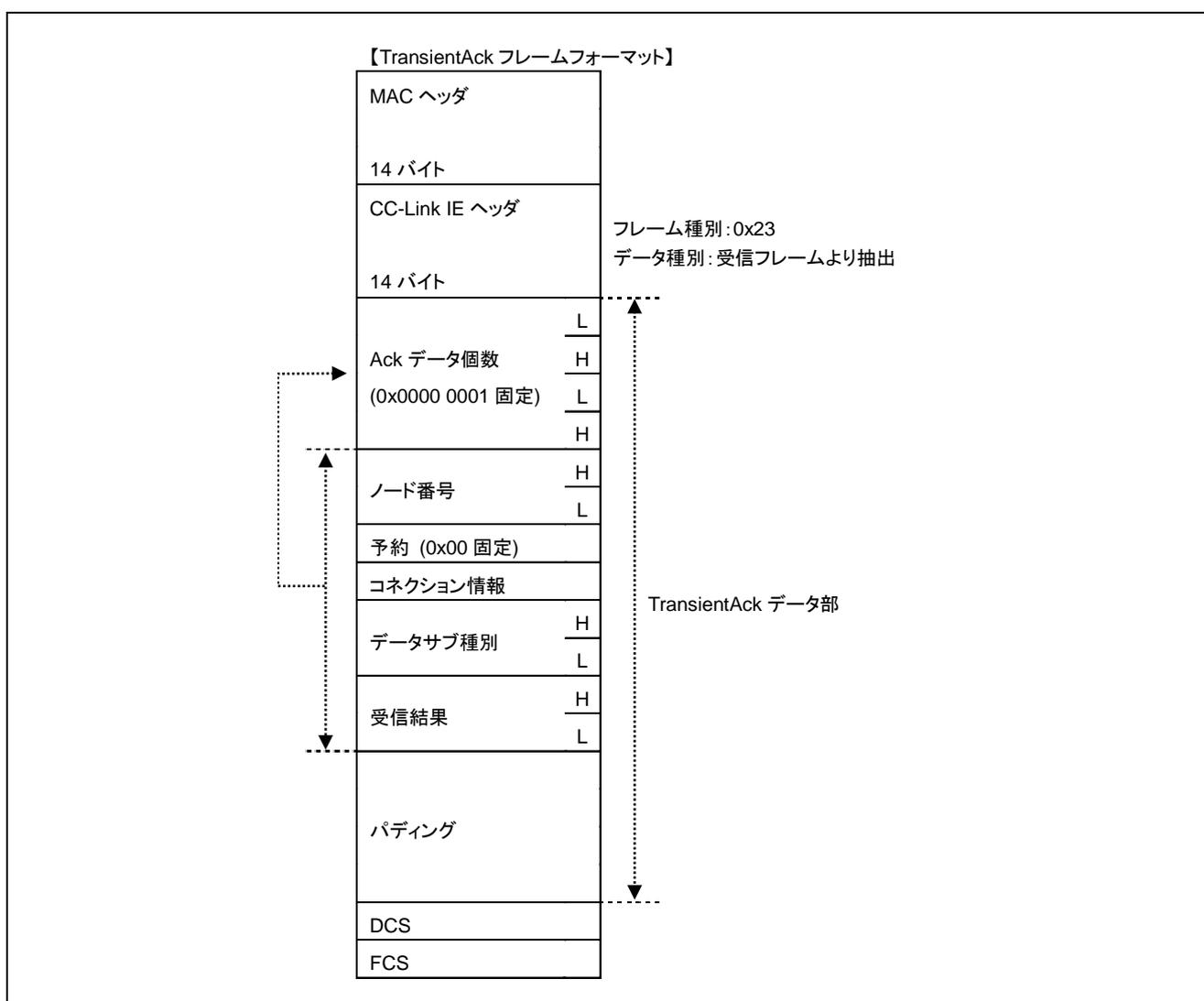


図5.20 TransientAck フレームフォーマット概略

## (1) MAC ヘッダ、CC-Link IE ヘッダ

「5.3.1 トランジェントフレーム共通フォーマット」を参照してください。

## (2) TransientAck データ部

表5.21 TransientAck データ部項目一覧

| 項目        | 内容   | 値  | 備考  |
|-----------|--|--|---|
| Ack データ個数 | ノード番号～受信結果までの個数                                    | 0x0000 0001 固定                               | —   |
| ノード番号     | TransientAck フレーム送信先ノード番号                          | 受信した Transient1 または Transient2 フレームの送信元ノード番号 | マスタ局からトランジェントフレームを受信した場合（送信元ノード番号が「0x0000」の場合）は、送信先ノード番号を「0x007D」に変換して設定してください。 |
| 予約        | 予約   | 0x00 固定                                      | —   |
| 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 Transient1 または Transient2 フレームのコネクション情報 | —   |
| データサブ種別   | 受信した Transient1 フレームのデータサブ種別                       | 0x0002 : Transient1<br>0x0000 : Transient2   | Transient2 はデータサブ種別に固定値 : 0x0000 を設定してください。                                     |
| 受信結果      | Transient1 フレームまたは Transient2 フレームの受信結果 (RET)      | 0x0000 : 正常<br>0x0000 以外 : 異常                | —   |
| パディング     | パディング 16 バイト                                       | —  | Ethernet フレーム最小サイズ 64 バイトを満たすため、R-IN32M3-CL が自動的にパディングします。                      |

### 5.3.4 CC-Link 互換トランジェントフレームフォーマット

CC-Link 互換トランジェントフレームフォーマットの概要を下表に示します。

表5.22 CC-Link 互換トランジェントフレームフォーマット概要

| No. | 項目              | サイズ (バイト) | 備考                                    |
|-----|-----------------|-----------|---------------------------------------|
| 1   | MAC ヘッダ         | 14        | 「5.3.1 トランジェントフレーム共通フォーマット」を参照してください。 |
|     | CC-Link IE ヘッダ  | 14        |                                       |
| 2   | Transient2 ヘッダ  | 要求        | —                                     |
|     |                 | 応答        | —                                     |
| 3   | Transient2 データ部 | 要求        | 0~960                                 |
|     |                 | 応答        | 0~960                                 |
| 4   | DCS             | 4         | Data Check Sequence <sup>注</sup>      |
| 5   | FCS             | 4         | Frame Check Sequence <sup>注</sup>     |

**注. R-IN32M3-CL が自動的に計算し付加します。**

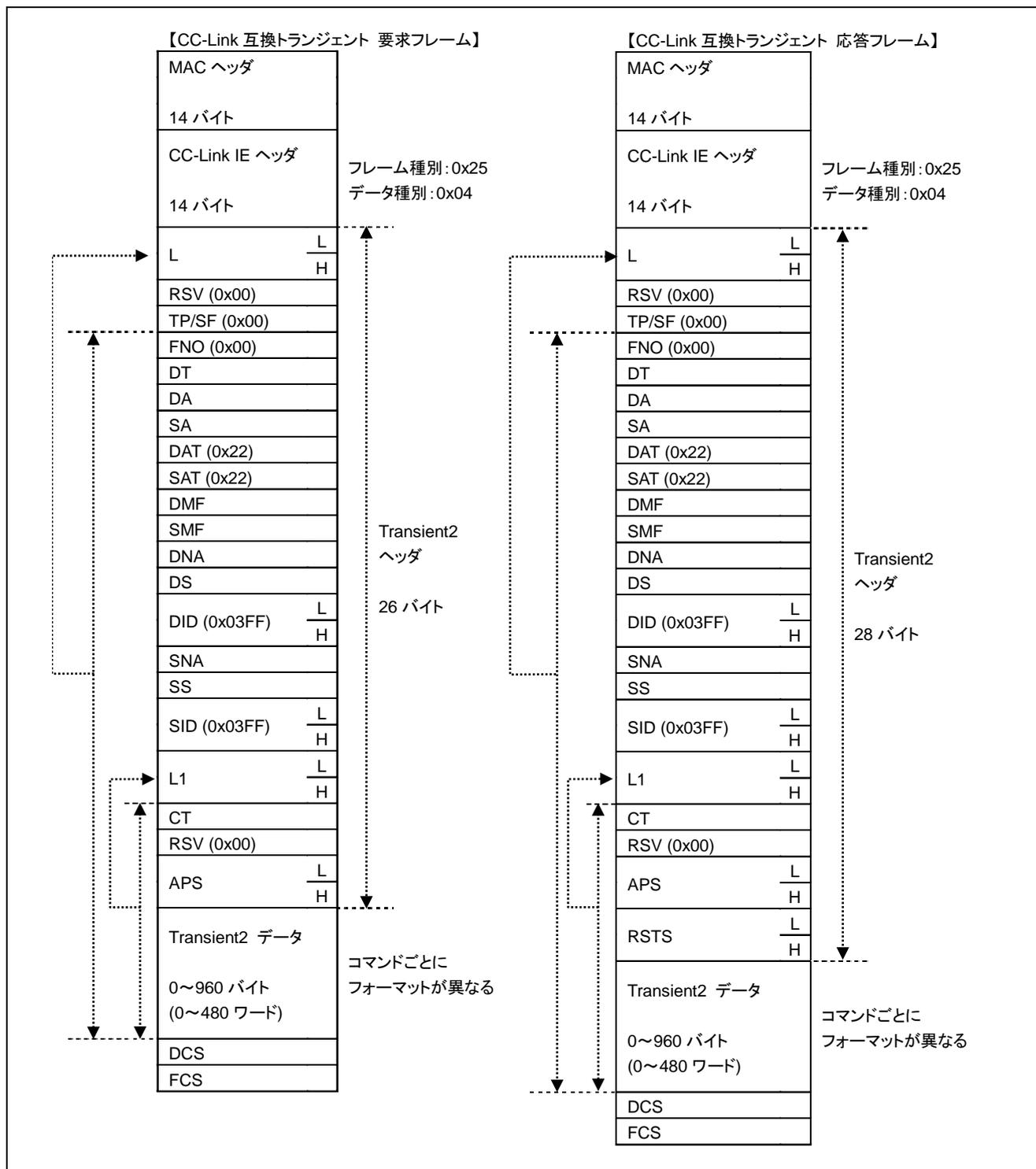


図5.21 CC-Link 互換トランジェントフレームフォーマット概略

(1) MAC ヘッダ、CC-Link IE ヘッダ

「5.3.1 トランジェントフレーム共通フォーマット」を参照してください。

## (2) Transient2 ヘッダ

表5.23 Transient2 ヘッダ項目

| 項目    | 内容                 |         | 値  | 備考   |
|-------|--------------------|---------|--|--|
| L     | フレーム長 (バイト)        |         | 22~982 : CC-Link 互換トランジェント<br>41~1440 : SLMP                                       | CC-Link 互換トランジェント :<br>FNO~Transient2 データまで<br>SLMP : RSV~SLMP データまで   |
| RSV   | 予約                 |         | 0x00 固定  | —  |
| TP/SF | 未使用 (タイプ/シーケンス No) |         | 0x00 固定  | —  |
| FNO   | 未使用 (分割フレーム No)    |         | 0x00 固定  | —  |
| DT    | 未使用 (データフレームタイプ)   |         | 0x00 固定  | —  |
| DA    | 宛先ノード番号            |         | 0x01~0x78 (1~120) : 局番<br>0x7D : 指定管理局/マスタ局<br>0x7E : 現在管理局/マスタ局<br>0xFF : グローバル要求 | DS と同じ値です。   |
| SA    | 送信元ノード番号           |         | 0x01~0x78 (1~120) : 局番   | SS と同じ値です。   |
| DAT   | 宛先アプリケーションタイプ      |         | 0x22 固定  | —  |
| SAT   | 送信元アプリケーションタイプ     |         | 0x22 固定  | —  |
| DMF   | 宛先モジュールフラグ         |         | 0x00 : CC-Link 互換トランジェント<br>0x03 : SLMP  | —  |
| SMF   | 送信元モジュールフラグ        |         | 0x00 : CC-Link 互換トランジェント<br>0x03 : SLMP  | —  |
| DNA   | 宛先ネットワーク番号         |         | 0x01~0xEF(1~239)   | —  |
| DS    | 宛先ノード番号            |         | 0x01~0x78 (1~120) : 局番<br>0x7D : 指定管理局/マスタ局<br>0x7E : 現在管理局/マスタ局<br>0xFF : グローバル要求 | DA と同じ値です。   |
| DID   | 宛先識別番号             |         | 0x03FF 固定  | —  |
| SNA   | 送信元ネットワーク番号        |         | 0x01~0xEF (1~239)  | —  |
| SS    | 送信元ノード番号           |         | 0x01~0x78 (1~120)  | SA と同じ値です。   |
| SID   | 送信元識別番号            |         | 0x03FF 固定  | —  |
| L1    | データ長 (バイト)         |         | 4~972  | CT~DATA までのサイズ (バイト)   |
| CT    | コマンドタイプ            |         | 0x04~0x1F :<br>CC-Link 互換トランジェント<br>0x30 : SLMP 要求<br>0xB0 : SLMP 応答               | CC-Link 互換トランジェントのコマンドタイプは「表5.25 CC-Link 互換トランジェントコマンドタイプ一覧」を参照してください。 |
| RSV   | 予約                 |         | 0x00 固定  | —  |
| APS   | アプリケーション No.       | Bit15-8 | 0x00 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。                            |
|       |                    | Bit7-0  | 0x00~0xFF  |  |
| RSTS  | リターンコード            |         | 0x0000 : 正常<br>0x0000 以外 : エラーコード  | 応答時のみ<br>リターンコードの詳細は本項 (a) を参照してください。                                  |

### (a) リターンコード (RSTS)

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せん。

要求先機器の仕様に従い、適切な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下表に示します。

表5.24 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                                 | 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         | 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                  | 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           | 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                 | 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               | 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                  | 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。<br>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。<br>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。<br>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            |   |

(b) コマンドタイプ (CT)

CC-Link 互換トランジェントコマンドタイプ (CT) のデータ構成を下記に示します。

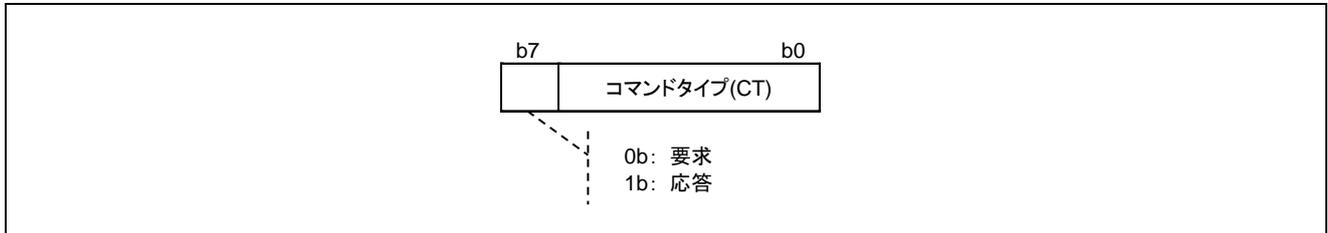


図5.22 コマンドタイプ (CT) のデータ構成

表5.25 CC-Link 互換トランジェントコマンドタイプ一覧

| CT   | コマンド種別        |
|------|---------------|
| 0x04 | メモリアクセス情報取得要求 |
| 0x84 | メモリアクセス情報取得応答 |
| 0x08 | RUN 要求        |
| 0x88 | RUN 応答        |
| 0x09 | STOP 要求       |
| 0x89 | STOP 応答       |
| 0x10 | メモリ読出し要求      |
| 0x90 | メモリ読出し応答      |
| 0x12 | メモリ書込み要求      |
| 0x92 | メモリ書込み応答      |



(b) RUN

他局を RUN 状態にします。

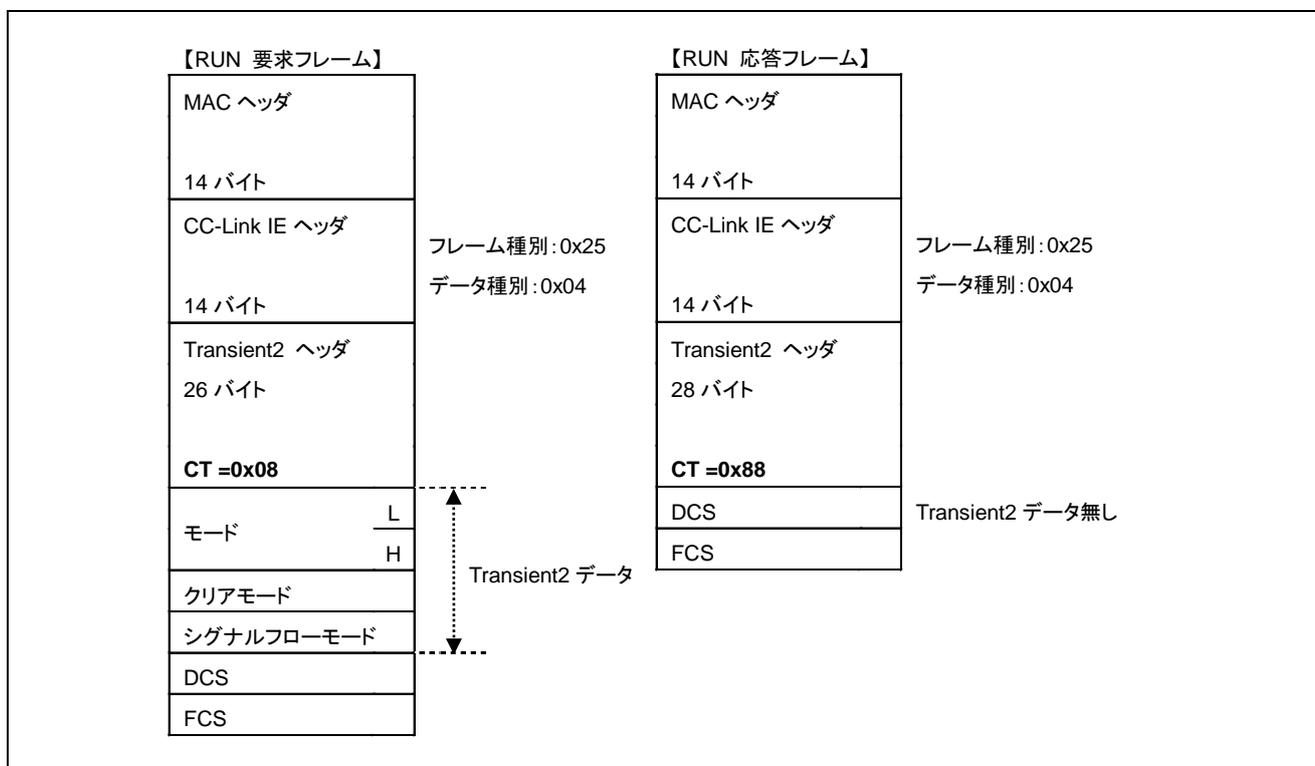


図5.24 RUN フレームフォーマット概略

表5.26 RUN 要求設定一覧

| 項目         | 設定            | 値      |
|------------|---------------|--------|
| モード        | 通常 RUN        | 0x0003 |
|            | 強制 RUN        | 0x0001 |
| クリアモード     | オールクリア        | 0x02   |
|            | ラッチ範囲以外をクリアする | 0x01   |
|            | デバイスクリアしない    | 0x00   |
| シグナルフローモード | 固定値           | 0x00   |

(c) STOP

他局を STOP 状態にします。

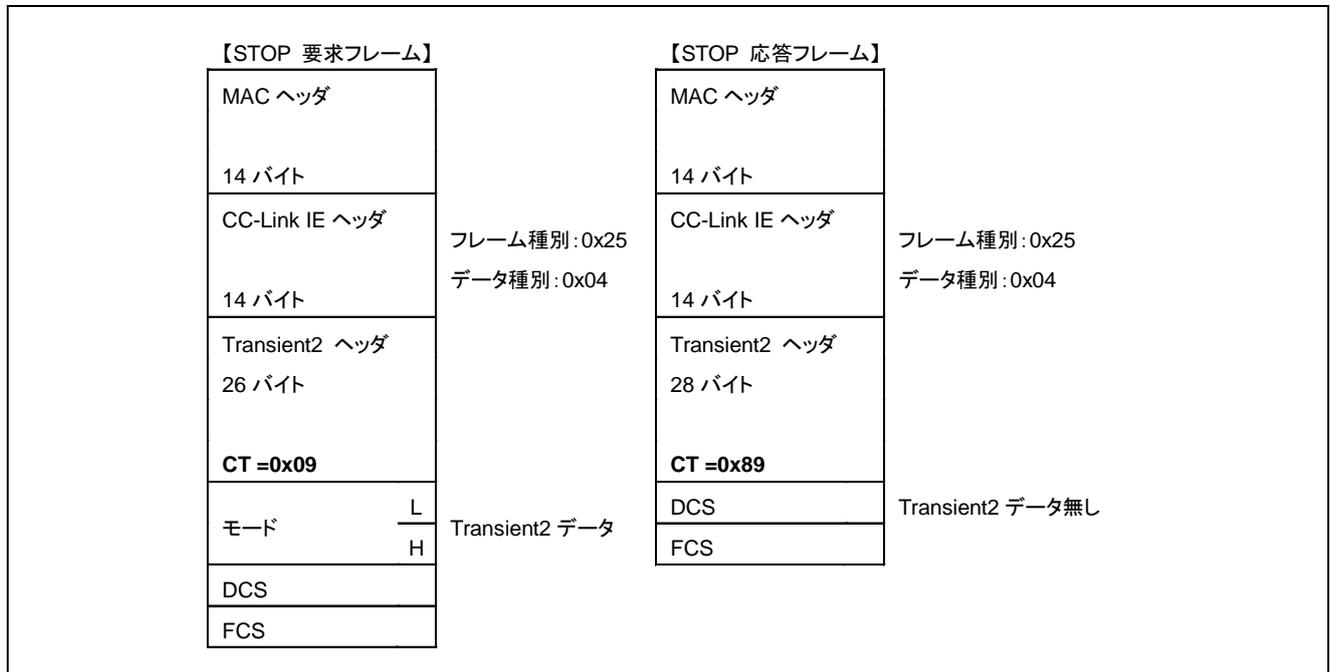


図5.25 STOP フレームフォーマット概略

表5.27 STOP 要求設定一覧

| 項目  | 設定      | 値      |
|-----|---------|--------|
| モード | 通常 STOP | 0x0003 |
|     | 強制 STOP | 0x0001 |

(d) メモリ読出し

他局のデバイスを読み出します。

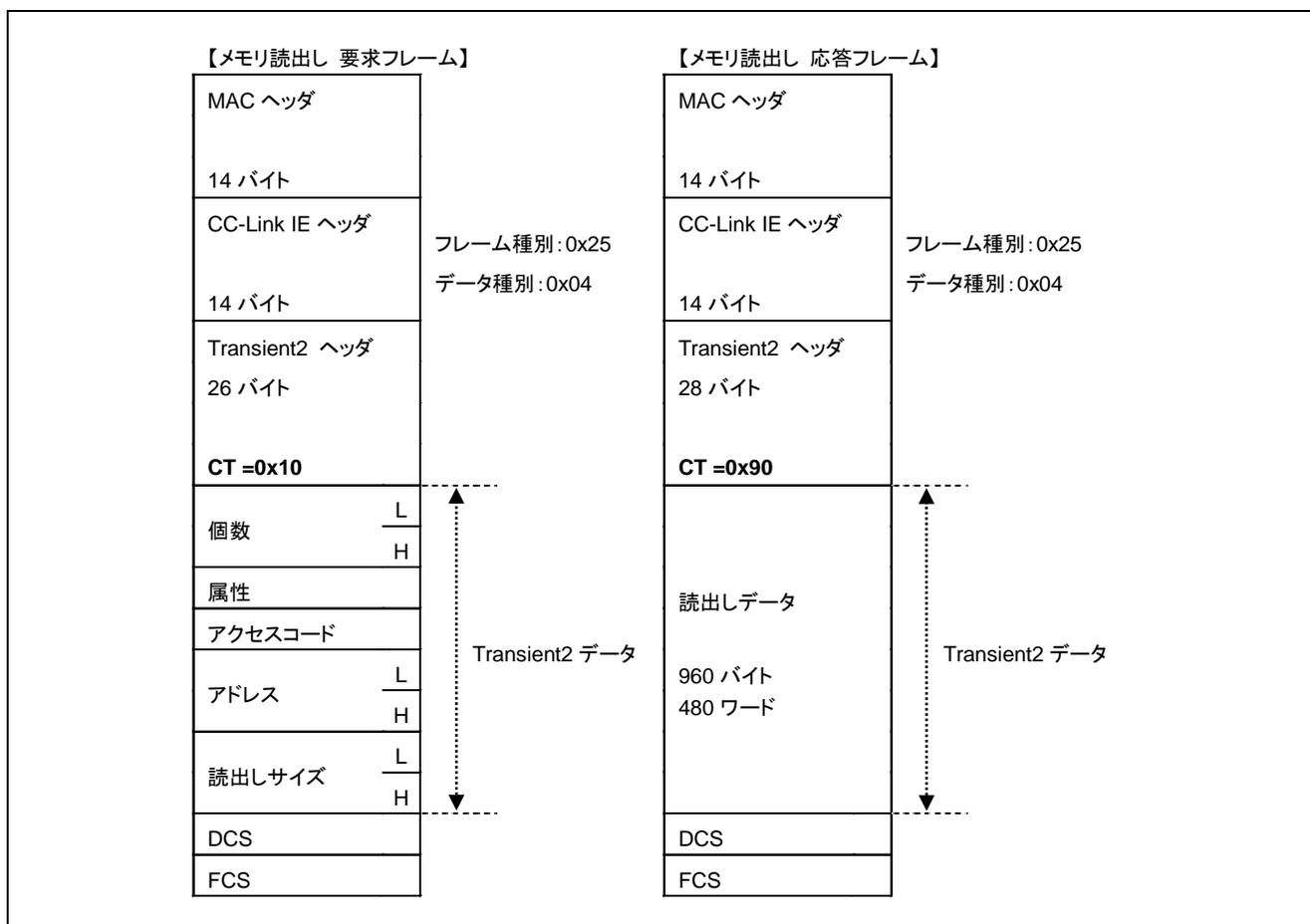


図5.26 メモリ読出しフレームフォーマット概略

表5.28 メモリ読出し設定一覧

| 項目      | 設定                   | 値         |
|---------|----------------------|-----------|
| 個数      | 属性～読出しサイズまでのブロック個数   | 0x0001 固定 |
| 属性      | 「図5.29 属性定義」を参照      | —         |
| アクセスコード | 「図5.28 アクセスコード定義」を参照 | —         |
| アドレス    | デバイスのアドレス            | 0～65535   |
| 読出しサイズ  | ワード単位                | 1～480     |

備考・三菱電機のシーケンサで専用命令「RIRD」を実行した場合、本フレームが送信されます。

・三菱電機製品に要求を送信する場合、属性を 0x05 に設定、アクセスコードを「表5.30 三菱電機製品のアクセスコード一覧」にしたがって設定してください。

(e) メモリ書込み

他局のデバイスに書き込みます。

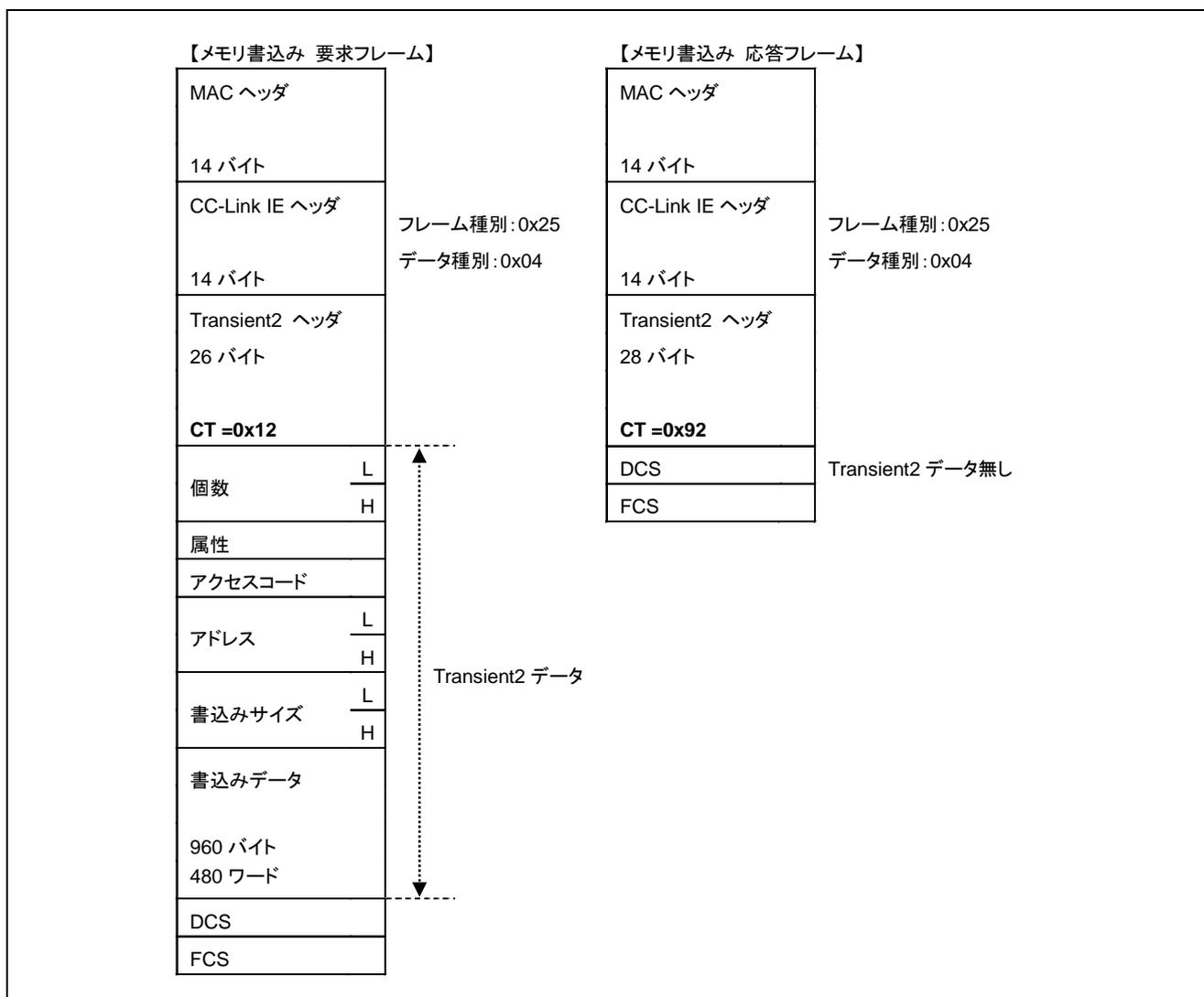


図5.27 メモリ書込みフレームフォーマット概略

表5.29 メモリ書込み要求設定一覧

| 項目      | 設定                   | 値         |
|---------|----------------------|-----------|
| 個数      | 属性～書込みサイズまでのブロック個数   | 0x0001 固定 |
| 属性      | 「図5.29 属性定義」を参照      | —         |
| アクセスコード | 「図5.28 アクセスコード定義」を参照 | —         |
| アドレス    | デバイスのアドレス            | 0～65535   |
| 書込みサイズ  | ワード単位                | 1～480     |

**備考** ・三菱電機のシーケンサで専用命令「RIWT」を実行した場合、本フレームが送信されます。

・三菱電機製品に要求を送信する場合、属性を 0x05 に設定、アクセスコードを「表5.30 三菱電機製品のアクセスコード一覧」にしたがって設定してください。

## (f) 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。

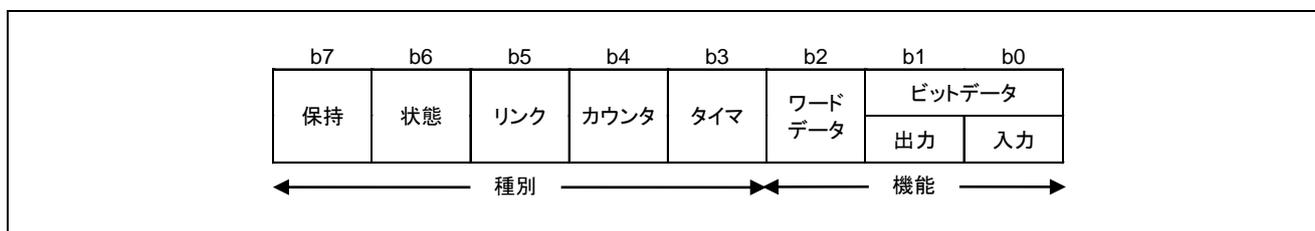


図5.28 アクセスコード定義

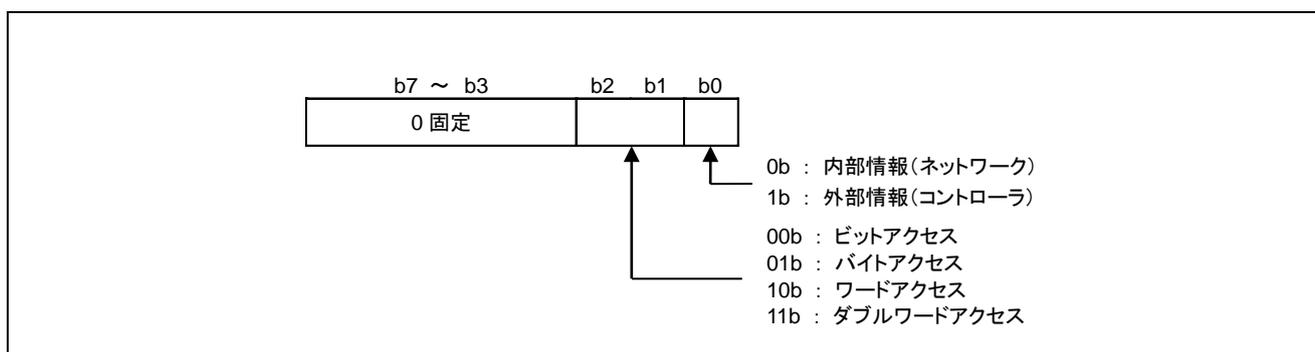


図5.29 属性定義

## 【自局がサーバのとき】

他局（三菱電機製品または開発機器）がメモリ読出し／書込みコマンドを使って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。

## 【自局がクライアントのとき】

自局がメモリ読出し／書込みコマンドを使って他局（三菱電機製品）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。

三菱電機製品以外にアクセスする場合、対象局のユーザーズマニュアルを参照してください。

表5.30 三菱電機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 <sup>注1</sup> | 属性コード <sup>注1</sup> |
|------------|-----------------|---------|-----|------|-----------------------|---------------------|
|            |                 | ビット     | ワード |      |                       |                     |
| 入力リレー      | X               | ○       | —   | 16進数 | 0x01                  | 0x05                |
| 出力リレー      | Y               | ○       | —   | 16進数 | 0x02                  |                     |
| 特殊リレー      | SM              | ○       | —   | 10進数 | 0x43                  |                     |
| 特殊レジスタ     | SD              | —       | ○   | 10進数 | 0x44                  |                     |
| 内部リレー      | M               | ○       | —   | 10進数 | 0x03                  |                     |
| ラッチリレー     | L               | ○       | —   | 10進数 | 0x83                  |                     |
| タイマ（接点）    | T               | ○       | —   | 10進数 | 0x09                  |                     |
| タイマ（コイル）   | T               | ○       | —   | 10進数 | 0x0A                  |                     |
| タイマ（現在値）   | T               | —       | ○   | 10進数 | 0x0C                  |                     |
| 積算タイマ（接点）  | ST              | ○       | —   | 10進数 | 0x89                  |                     |
| 積算タイマ（コイル） | ST              | ○       | —   | 10進数 | 0x8A                  |                     |
| 積算タイマ（現在値） | ST              | —       | ○   | 10進数 | 0x8C                  |                     |
| カウンタ（接点）   | C               | ○       | —   | 10進数 | 0x11                  |                     |
| カウンタ（コイル）  | C               | ○       | —   | 10進数 | 0x12                  |                     |
| カウンタ（現在値）  | C               | —       | ○   | 10進数 | 0x14                  |                     |
| データレジスタ    | D <sup>注2</sup> | —       | ○   | 10進数 | 0x04                  |                     |
| ファイルレジスタ   | R               | —       | ○   | 10進数 | 0x84                  |                     |
| リンクリレー     | B               | ○       | —   | 16進数 | 0x23                  |                     |
| リンクレジスタ    | W <sup>注2</sup> | —       | ○   | 16進数 | 0x24                  |                     |
| リンク特殊リレー   | SB              | ○       | —   | 16進数 | 0x63                  |                     |
| リンク特殊レジスタ  | SW              | —       | ○   | 16進数 | 0x64                  |                     |

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。

### 5.3.5 SLMP フレームフォーマット

SLMP フレームフォーマットの概要を下表に示します。

表5.31 SLMP フレームフォーマット概要

| No. | 項目                               |          | サイズ (バイト) | 備考                                    |   |
|-----|----------------------------------|----------|-----------|---------------------------------------|---|
| 1   | MAC ヘッダ                          |          | 14        | 「5.3.1 トランジェントフレーム共通フォーマット」を参照してください。 |   |
|     | CC-Link IE ヘッダ                   |          | 14        |                                       |   |
| 2   | Transient1 ヘッダ                   |          | 16        | 「5.3.2(2) Transient1ヘッダ」を参照してください。    |   |
| 3   | Transient2 ヘッダ                   | 要求       | 26        | 「5.3.4(2) Transient2ヘッダ」を参照してください。    |   |
|     |                                  | 応答       | 28        |                                       |   |
| 4   | Transient1<br>データ部 <sup>注2</sup> | SLMP ヘッダ |           | 15                                    | — |
| 5   |                                  | SLMP データ | 要求        | 0~1425                                | — |
|     | 応答                               |          | 0~1423    |                                       |   |
| 6   | DCS                              |          | 4         | Data Check Sequence <sup>注1</sup>     |   |
| 7   | FCS                              |          | 4         | Frame Check Sequence <sup>注1</sup>    |   |

注 1. R-IN32M3-CL が自動的に計算し付加します。

2. Transient1 データ部を“SLMP”として使う場合を示します。Transient1 データ部を“CC-Link IE フィールド固有トランジェント”として使う場合、「5.3.2 CC-Link IEフィールド固有トランジェントフレームフォーマット」を参照してください。

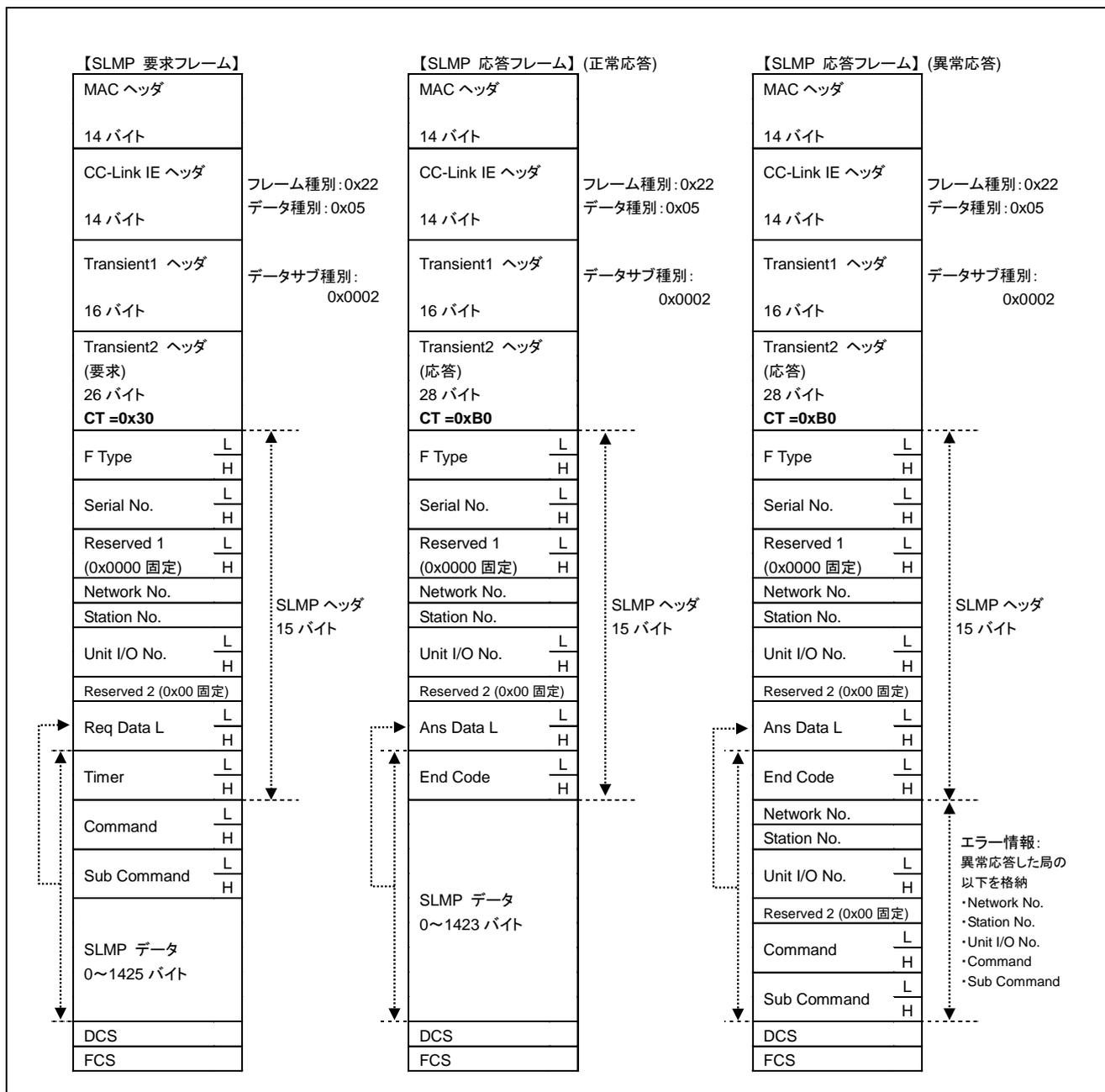


図5.30 SLMP フレームフォーマット概略

(1) MAC ヘッダ、CC-Link IE ヘッダ

「5.3.1 トランジェントフレーム共通フォーマット」を参照してください。

(2) Transient1 ヘッダ

「5.3.2(2) Transient1ヘッダ」を参照してください。

## (3) Transient2 ヘッダ

「5.3.4(2) Transient2ヘッダ」を参照してください。

## (4) SLMP ヘッダ

表5.32 SLMP ヘッダ項目

| 項目           | 内容            | 値  | 備考  |
|--------------|---------------|--|---|
| F Type       | フレームの種別       | 0x0054 : 要求時<br>0x00D4 : 応答時   | —   |
| Serial No.   | シリアル番号        | 0x0000～0xFFFF  | フレームを区別するための任意の番号を設定します。要求時と応答時で同値に設定します。   |
| Network No.  | 宛先ネットワーク番号    | 0x00 : 自局<br>0x01～0xEF(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.  | 宛先局番          | 0x01～0x78 (1～120) : 局番<br>0x7D : 指定管理局／マスタ局<br>0x7E : 現在管理局／マスタ局<br>0xFF : 自局 <sup>注</sup> | 対象となる局番を設定してください。   |
| Unit I/O No. | 宛先ユニット I/O 番号 | 0x03FF 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を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をバイト単位で設定してください。  |
| Timer        | 監視タイマ         | 0x0001～0xFFFF<br>0x0000 : 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)<br>推奨値<br>自局 : 0001h～0028h (0.25～10 秒)<br>他局 : 0002h～00F0h (0.5～60 秒) |
| End Code     | 終了コード         | 0x0000 : 正常終了<br>0x0000 以外 : エラーコード  | 応答時のみ。<br>終了コードの詳細は、本項(a)を参照してください。   |

**注. Network No.が 0x00 の場合のみ有効**

## (a) 終了コード (End Code)

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せん。

要求先機器の仕様を参照して、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下表に示します。

表5.33 終了コードに格納するエラーコード例一覧

| No. | カテゴリ    | 終了コード | 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。<br>・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。              |
| 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 | 書き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。                    | 書き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。                |

(5) SLMP データ部

(a) SLMP メモリ読出し

SLMP メモリ読出しは、他局 (SLMP 対応機器) のバッファメモリのデータを読み出す際に使用します。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。

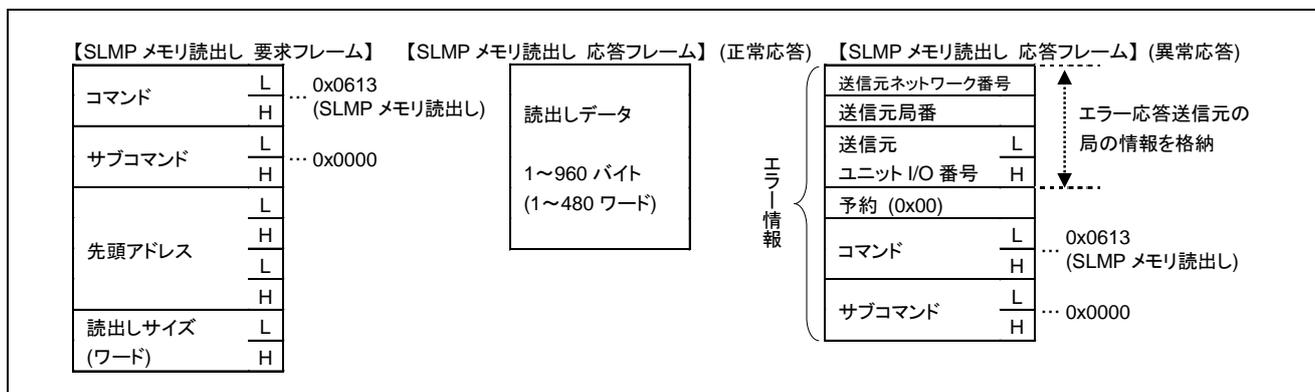


図5.31 SLMP メモリ読出しフレーム

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5.34 SLMP メモリ読出しフレームフォーマットの詳細

| 内容            | 値  | 備考                        |
|---------------|--|---------------------------|
| コマンド          | 0x0613   | —                         |
| サブコマンド        | 0x0000   | —                         |
| 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ます。 |
| 読出しサイズ (ワード)  | 0x1~0x1E0 (1~480)                                | 読み出すバッファメモリのワード長を指定します。   |
| 送信元ネットワーク番号   | 0x00 : 自局<br>0x01~0xEF (1~239) : 他局              | 応答送信元のネットワーク No.を設定します。   |
| 送信元局番         | 0x01~0x78 (1~120) : 局番<br>0xFF : 自局 <sup>注</sup> | 応答送信元の局番を設定します。           |
| 宛先ユニット I/O 番号 | 0x03FF : 固定                                      | アクセス先の CPU ユニットを設定します。    |

**注. Network No.が 0x00 の場合のみ有効**

(b) SLMP メモリ書込み

SLMP メモリ書込みは、他局 (SLMP 対応機器) のバッファメモリのデータに書き込む際に使用します。SLMP メモリ書込みフレームフォーマットを以下に示します。

なお、正常応答の場合、SLMP データ部はありません。(SLMP ヘッダ、および DCS、FCS は必要です。)

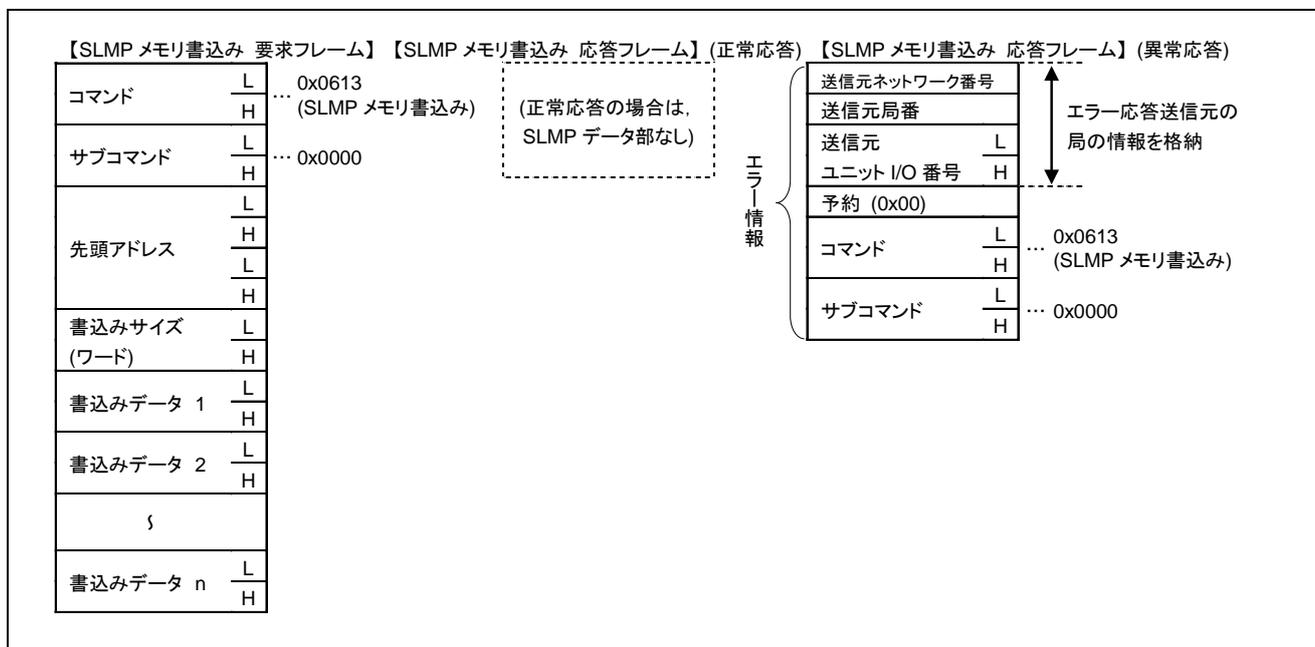


図5.32 SLMP メモリ書込みフレーム

SLMP 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5.35 SLMP メモリ書込みフレームフォーマットの詳細

| 内容            | 値  | 備考                        |
|---------------|--|---------------------------|
| コマンド          | 0x1613   | —                         |
| サブコマンド        | 0x0000   | —                         |
| 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ます。 |
| 書込みサイズ (ワード)  | 0x0001~0x01E0 (1~480)  | 書き込むバッファメモリのワード長を指定します。   |
| 書込みデータ        | —  | 書き込むデータを設定します。            |
| 送信元ネットワーク番号   | 0x00 : 自局<br>0x01~0xEF (1~239) : 他局  | 応答送信元のネットワーク No.を設定します。   |
| 送信元局番         | 0x01~0x78 (1~120) : 局番<br>0x7D : 指定管理局/マスタ局<br>0x7E : 現在管理局/マスタ局<br>0xFF : 自局 <sup>注</sup> | 応答送信元の局番を設定します。           |
| 宛先ユニット I/O 番号 | 0x03FF : 固定  | アクセス先の CPU ユニットを設定します。    |

注. Network No.が 0x00 の場合のみ有効

## 5.4 MyStatus 概要

MyStatus はネットワークに接続するノードの状態通知に使用されます。

R-IN32M3-CL は、自局の情報を MyStatus フレームにセットしてマスタ局に通知します。また、マスタ局からの MyStatus フレームを受信して、マスタ局の状態を監視します。

### 5.4.1 MyStatus 送信

ユーザプログラムが自局の情報を `gerR_IN32_SetNodeStatus` 関数の引数へセットすることで、R-IN32M3-CL ドライバが MyStatus フレームに情報をセットしてマスタ局へ送信します。UserSendMyStatus (MyStatus 送信処理) にて MyStatus フレームにセットする自局情報を以下に示します。

表5.36 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。<br>0000h : アプリケーション詳細動作状態通知未サポート<br>0001h : アプリケーションが停止中<br>0002h : アプリケーションが動作中<br>0003h : アプリケーション実体が存在しない<br>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のエラー発生時のエラー状態を格納する。<br>0000h : エラーなし<br>0001h : 軽度異常<br>0002h : 中度異常<br>0003h : 重度異常<br>上記以外 : 未使用                                     |

## 5.4.2 MyStatus 受信

マスタ局からの MyStatus フレームを R-IN32M3-CL ドライバが受信します。

UserReceiveCyclic (マスタ局からの MyStatus およびサイクリック受信処理) にて、MyStatus フレームで得られるマスタ局の情報を以下に示します。

表5.37 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。 <sup>注1</sup><br>0b : アプリケーションが停止中<br>1b : アプリケーションが実行中 |
| 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。 <sup>注2</sup><br>0b : エラーなし<br>1b : エラーあり              |

**注 1. 三菱電機製マスタ局の場合、シーケンサ CPU ユニットの以下の状態が格納されます。**

**【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. 三菱電機製マスタ局の場合、シーケンサ CPU ユニットの以下の状態が格納されます。**

**【エラーなし】**

エラーがない場合、およびバッテリーエラーなどの CPU ユニットの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 ユニットの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 ユニットの演算を停止するエラー (重度異常)

## 6. ファームウェアの開発

### 6.1 開発の手順

本節では、サンプルコードを使用したファームウェアの開発手順を説明します。

サンプルコードは、「表6.1サンプルコードに含まれるプログラム部品一覧」に示すプログラム部品で構成します。R-IN32M3-CL ドライバ本体のカスタマイズは不要ですが、その他のプログラム部品は開発する機器（ターゲット）のハードウェアに合わせてカスタマイズする必要があります。

表6.1 サンプルコードに含まれるプログラム部品一覧

| プログラム部品名                      | 概要  | 変更要否   |
|-------------------------------|---|--------|
| ユーザプログラム                      | ユーザプログラムは、ユーザが作成するアプリケーションプログラムです。インテリジェントデバイス局の通信機能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）ですので、ユーザの要求仕様に応じてカスタマイズしてください。 | カスタマイズ |
| R-IN32M3-CL ドライバ<br>インタフェース関数 | R-IN32M3-CL 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コールする関数です。   | 変更不要   |
| R-IN32M3-CL ドライバ<br>ターゲット依存関数 | ターゲットとなるユーザの H/W 環境に合わせてカスタマイズする必要のある処理または関数です。   | カスタマイズ |
| R-IN32M3-CL ドライバ<br>コールバック関数  | ユーザプログラムから R-IN32M3-CL 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です。<br>R-IN32M3-CL 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します。    | カスタマイズ |
| R-IN32M3-CL ドライバ<br>本体        | インタフェース関数にコールされ、R-IN32M3-CL を制御するドライバ部の本体です。  | 変更不要   |

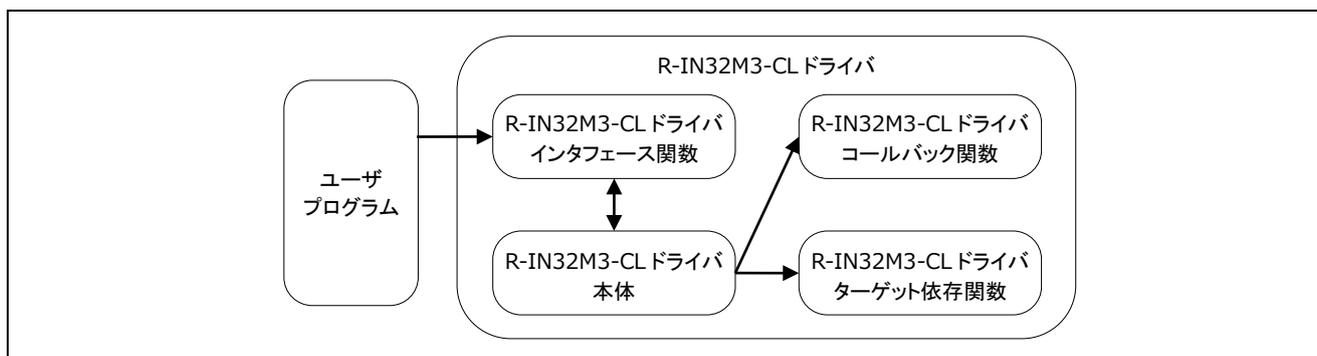
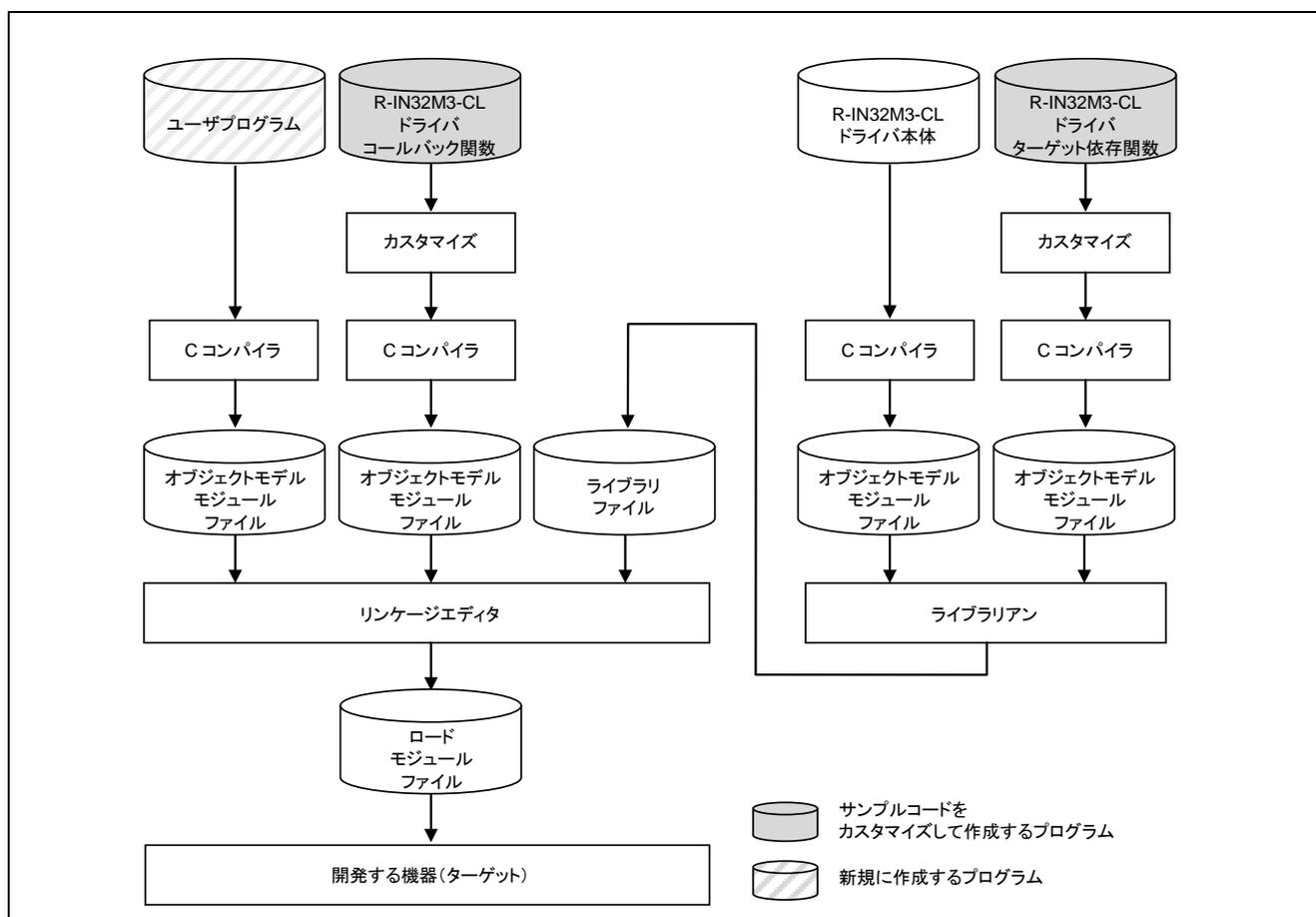


図6.1 サンプルコードの構成

**注意.** 本リファレンスマニュアルで提供するサンプルコードは「GCC (GNU C Compiler) バージョン 4.3.4」でコンパイルエラーが出ないことを確認しています。オペレーティングシステムや MPU に依存しませんので、ユーザの環境に合わせてカスタマイズしてください。

ファームウェア開発手順を以下に示します。

- Step1 : ユーザプログラムの作成  
ベンダーは、「6.2.1 メイン処理」を参考にユーザプログラムを作成します。
- Step2 : R-IN32M3-CL ドライバターゲット依存関数のカスタマイズ  
ベンダーは開発する機器のハードウェアに合わせて、R-IN32M3-CL ドライバターゲット依存関数をカスタマイズします。詳細は、「6.5 R-IN32M3-CL ドライバターゲット依存関数のカスタマイズ」を参照してください。
- Step3 : R-IN32M3-CL ドライバコールバック関数のカスタマイズ  
ベンダーは開発する機器で実現する機能に合わせて、R-IN32M3-CL ドライバコールバック関数をカスタマイズします。詳細は、「6.6 R-IN32M3-CL ドライバコールバック関数のカスタマイズ」を参照してください。
- Step4 : R-IN32M3-CL ライブラリの作成  
ベンダーはR-IN32M3-CL ドライバ本体とR-IN32M3-CL ドライバターゲット依存関数をコンパイル後、ライブラリアンを実行し、ライブラリファイルを作成します。
- Step5 : ユーザプログラムとライブラリファイルの結合  
ベンダーは、ユーザプログラム、カスタマイズしたR-IN32M3-CL ドライバコールバック関数およびライブラリファイルを結合し、ロードモジュールファイルを作成します。
- Step6 : ベンダーはロードモジュールファイルを、開発する機器（ターゲット）にロードしてください。



### 6.1.1 サンプルコードのファイルリスト

サンプルコードのファイルリストを以下に示します。

表6.2 サンプルコードのファイルリスト (1/2)

| フォルダ                 | ファイル                            | 内容                         |   |
|----------------------|---------------------------------|----------------------------|---|
| sample               | include                         | R_IN32M3Callback.h         | R-IN32M3-CL ドライバコールバック関数ヘッダファイル (6.6節 参照)   |
|                      | obj                             | makefile                   | メイクファイル (ユーザプログラム構築用)                       |
|                      | src                             | R_IN32M3_Callback.c        | R-IN32M3-CL ドライバコールバック関数ソースファイル (6.6節 参照)   |
|                      |                                 | R_IN32M3_HWTest.c          | ユーザプログラム (ハードウェアテスト処理) ソースファイル              |
|                      |                                 | R_IN32M3_HWTest.h          | ユーザプログラム (ハードウェアテスト処理) ヘッダファイル              |
|                      |                                 | R_IN32M3_sample.c          | ユーザプログラム (メイン処理など) ソースファイル                  |
|                      |                                 | R_IN32M3_sample.h          | ユーザプログラム (メイン処理など) ヘッダファイル                  |
|                      |                                 | R_IN32M3_Transient.c       | ユーザプログラム (トランジェント送受信処理) ソースファイル             |
| R_IN32M3_Transient.h | ユーザプログラム (トランジェント送受信処理) ヘッダファイル |                            |   |
| driver               | include                         | R_IN32M3Driver.h           | R-IN32M3-CL ドライバヘッダファイル                     |
|                      |                                 | R_IN32M3Function.h         | R-IN32M3-CL ドライバターゲット依存関数ヘッダファイル (6.5.1項参照) |
|                      |                                 | R_IN32M3Types.h            | R-IN32M3-CL ドライバインタフェース関数ヘッダファイル (6.4節参照)   |
|                      | obj                             | makefile                   | メイクファイル (R_IN32M3Driver.a 構築用)              |
|                      | src                             | R_IN32_Interface.c         | R-IN32M3-CL ドライバインタフェース関数ソースファイル (6.3節参照)   |
|                      |                                 | R_IN32R.c                  | R-IN32M3-CL ドライバターゲット依存関数ソースファイル (6.5.2項参照) |
|                      |                                 | R_IN32R.h                  | R-IN32M3-CL ドライバターゲット依存関数ヘッダファイル (6.5.2項参照) |
|                      |                                 | R_IN32C_I.h                | R-IN32M3-CL ドライバ本体                          |
|                      |                                 | R_IN32C_Library.c          |   |
|                      |                                 | R_IN32C_MainState.c        |   |
|                      |                                 | R_IN32C_PortState.c        |   |
|                      |                                 | R_IN32C_R_IN32DInterface.c |   |
|                      |                                 | R_IN32C_Time.c             |   |
|                      |                                 | R_IN32D.h                  |   |
|                      |                                 | R_IN32D_cyc.c              |   |
|                      |                                 | R_IN32D_cyc_I.h            |   |
|                      |                                 | R_IN32D_ihnd.c             |   |
|                      |                                 | R_IN32D_ini.c              |   |
|                      |                                 | R_IN32D_intr.c             |   |
|                      |                                 | R_IN32D_intr_I.h           |   |
|                      |                                 | R_IN32D_led.c              |   |
|                      |                                 | R_IN32D_phy.c              |   |
|                      |                                 | R_IN32D_phy_I.h            |   |
| R_IN32D_RcvCnt.c     |                                 |                            |   |
| R_IN32D_RcvCnt_I.h   |                                 |                            |   |
| R_IN32D_RcvPrm.c     |                                 |                            |   |
| R_IN32D_RcvPrm_I.h   |                                 |                            |   |
| R_IN32D_reg.c        |                                 |                            |   |
| R_IN32D_reg_I.h      |                                 |                            |   |

表6.2 サンプルコードのファイルリスト (2/2)

| フォルダ                 |     | ファイル                  | 内容                 |
|----------------------|-----|-----------------------|--------------------|
| driver               | src | R_IN32D_sub.c         | R-IN32M3-CL ドライバ本体 |
|                      |     | R_IN32D_sub_l.h       |                    |
|                      |     | R_IN32D_tran.c        |                    |
|                      |     | R_IN32D_tran_l.h      |                    |
|                      |     | R_IN32M3.h            |                    |
|                      |     | R_IN32M3_0.h          |                    |
|                      |     | R_IN32M3_1.h          |                    |
|                      |     | R_IN32M3_2.h          |                    |
|                      |     | R_IN32M3_3.h          |                    |
|                      |     | R_IN32S.c             |                    |
|                      |     | R_IN32S.h             |                    |
|                      |     | R_IN32T.h             |                    |
|                      |     | R_IN32T_ASIC.c        |                    |
|                      |     | R_IN32T_ASIC.h        |                    |
|                      |     | R_IN32T_Cmu.h         |                    |
|                      |     | R_IN32T_CmuNCycRcv.c  |                    |
|                      |     | R_IN32T_CmuOutLpBak.c |                    |
|                      |     | R_IN32T_CmuSub.h      |                    |
|                      |     | R_IN32T_CmuSub3.c     |                    |
|                      |     | R_IN32T_Com.c         |                    |
|                      |     | R_IN32T_Com.h         |                    |
|                      |     | R_IN32T_Data.c        |                    |
|                      |     | R_IN32T_Data.h        |                    |
|                      |     | R_IN32T_FrmForm.h     |                    |
|                      |     | R_IN32T_MACIP.c       |                    |
|                      |     | R_IN32T_MACIP.h       |                    |
|                      |     | R_IN32T_RegChk.c      |                    |
|                      |     | R_IN32T_RegChk.h      |                    |
|                      |     | R_IN32T_RING.c        |                    |
|                      |     | R_IN32T_RING.h        |                    |
|                      |     | R_IN32T_TxFrame.c     |                    |
|                      |     | R_IN32T_TxFrame.h     |                    |
|                      |     | R_IN32U.h             |                    |
|                      |     | R_IN32U_Init.c        |                    |
| R_IN32.h             |     |                       |                    |
| R_IN32_Frame.h       |     |                       |                    |
| R_IN32C.h            |     |                       |                    |
| R_IN32C_Cyclic.c     |     |                       |                    |
| R_IN32C_Data.c       |     |                       |                    |
| R_IN32C_Indication.c |     |                       |                    |
| R_IN32C_Init.c       |     |                       |                    |

## 6.2 サンプルフローチャート

ユーザプログラムについてのサンプルフローチャートの一覧を示します。各フローチャートに記述している処理は、インテリジェントデバイス局の機能を実現するためのサンプル処理です。ベンダーは、サンプルフローチャートを参考にユーザプログラムをカスタマイズしてください。

表6.3 イニシャル・サイクリック伝送関連サンプルフローチャート一覧 (R\_IN32M3\_sample.c)

| No | 概要                             | 記載項    | 実装要否 | 備考                  |
|----|--------------------------------|--------|------|---------------------|
| 1  | メイン処理                          | 6.2.1  | ◎    |                     |
| 2  | 初期化処理                          | 6.2.2  | ◎    |                     |
| 3  | 通信開始処理                         | 6.2.3  | ◎    |                     |
| 4  | PHY チェック処理                     | 6.2.4  | △    | PHYによって実装の要否が異なります。 |
| 5  | PHY 設定変更処理                     | 6.2.5  | △    |                     |
| 6  | 自局エラー処理                        | 6.2.6  | △    |                     |
| 7  | サイクリック伝送停止処理                   | 6.2.7  | △    |                     |
| 8  | イベント処理                         | 6.2.8  | ◎    |                     |
| 9  | マスタ局からの MyStatus およびサイクリック受信処理 | 6.2.9  | ◎    |                     |
| 10 | MyStatus 送信処理                  | 6.2.10 | ◎    |                     |
| 11 | サイクリック送信処理                     | 6.2.11 | ◎    |                     |
| 12 | 通信状態更新処理                       | 6.2.12 | ◎    |                     |
| 13 | サイクリック伝送状態更新処理                 | 6.2.13 | △    |                     |
| 14 | MIB 情報取得処理                     | 6.2.14 | △    |                     |

備考. ◎ : 必須 ○ : 推奨 △ : 任意

表6.4 トランジェント伝送関連サンプルフローチャート一覧 (R\_IN32M3\_Transient.c) (1/2)

| No | 概要                                      | 記載項    | 実装要否 | 備考                                      |
|----|---|--------|------|---|
| 1  | Transient1、Transient2、TransientAck 受信処理 | 6.2.15 | ◎    |   |
| 2  | Transient2 要求フレーム作成処理                   | 6.2.16 | △    | Transient2 のクライアントになる場合、必要です。           |
| 3  | Transient1、Transient2、TransientAck 送信処理 | 6.2.17 | ◎    |   |
| 4  | Transient1 受信データ処理                      | 6.2.18 | ◎    |   |
| 5  | Transient1 受信データ組立て開始処理                 | 6.2.19 | ◎    |   |
| 6  | Transient1 受信データ組立て処理                   | 6.2.20 | ◎    |   |
| 7  | ノード情報配信フレーム受信対応処理                       | 6.2.21 | △    | Transient2 または SLMP のクライアントになる場合、必要です。  |
| 8  | ノード情報配信フレームチェック処理                       | 6.2.22 | △    |   |
| 9  | 統計情報取得要求フレーム受信対応処理                      | 6.2.23 | △    |   |
| 10 | 統計情報取得応答フレーム作成処理                        | 6.2.24 | △    |   |
| 11 | ノード詳細情報取得要求フレーム受信対応処理                   | 6.2.25 | ◎    |   |
| 12 | ノード詳細情報取得応答フレーム作成処理                     | 6.2.26 | ◎    |   |
| 13 | オプション情報取得要求フレーム受信対応処理                   | 6.2.27 | ○    | 拡張機能 <sup>※</sup> に対応する場合、必要です。         |
| 14 | 選択局情報取得要求フレーム受信対応処理                     | 6.2.28 | ○    | 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、必要です。 |
| 15 | 交信テスト要求フレーム受信対応処理                       | 6.2.29 | ○    |   |
| 16 | ケーブルテスト要求フレーム受信対応処理                     | 6.2.30 | ○    |   |

表6.4 トランジェント伝送関連サンプルフローチャート一覧 (R\_IN32M3\_Transient.c) (2/2)

| No | 概要                          | 記載項    | 実装要否 | 備考                                  |
|----|-----------------------------|--------|------|-------------------------------------|
| 17 | Transient2 受信データ処理          | 6.2.31 | △    | Transient2 のサーバまたはクライアントになる場合、必要です。 |
| 18 | Transient2 受信データチェック処理      | 6.2.32 | △    |                                     |
| 19 | TransientAck 受信データ処理        | 6.2.33 | ◎    |                                     |
| 20 | TransientAck フレーム作成処理       | 6.2.34 | ◎    |                                     |
| 21 | Transient2 応答フレーム作成処理       | 6.2.35 | △    | Transient2 のサーバになる場合、必要です。          |
| 22 | Transient2 メモリ読出し要求フレーム作成処理 | 6.2.36 | △    | Transient2 メモリ読出しのクライアントになる場合、必要です。 |
| 23 | Transient2 メモリ書込み要求受信処理     | 6.2.37 | △    | Transient2 メモリ書込みのサーバになる場合、必要です。    |
| 24 | Transient2 メモリ読出し応答受信処理     | 6.2.38 | △    | Transient2 メモリ読出しのクライアントになる場合、必要です。 |
| 25 | SLMP メモリ読出し要求フレーム受信対応処理     | 6.2.39 | △    | SLMP メモリ読出しのサーバになる場合、必要です。          |
| 26 | SLMP メモリ書込み要求フレーム受信対応処理     | 6.2.40 | △    | SLMP メモリ書込みのサーバになる場合、必要です。          |
| 27 | SLMP メモリ読出し要求フレーム作成処理       | 6.2.41 | △    | SLMP メモリ読出しのクライアントになる場合、必要です。       |
| 28 | Transient1 要求送信分割判定処理       | 6.2.42 | △    | 1518バイト以上のSLMP要求フレームを送信する場合、必要です。   |
| 29 | Transient1 要求フレーム作成処理       | 6.2.43 | △    | SLMP のクライアントになる場合、必要です。             |
| 30 | SLMP メモリ読出し応答受信処理           | 6.2.44 | △    | SLMP メモリ読出しのクライアントになる場合、必要です。       |

備考. ◎ : 必須 ○ : 推奨 △ : 任意

注. 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です。

注意. R\_IN32RM3\_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/書込み、および SLMP メモリ読出し/書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、本書「5.2 トランジェント伝送概要」と、関連マニュアル「SLMP リファレンスマニュアル (BAP-C3002-001)」を参照して、コマンドごとの処理を追加してください。

表6.5 ハードウェアテスト関連サンプルフローチャート一覧 (R\_IN32M3\_HWTTest.c)

| No | 概要                                     | 記載項    | 実装要否 | 備考 |
|----|--|--------|------|----|
| 1  | ハードウェアテスト<br>(IEEE802.3ab コンプライアンステスト) | 6.2.45 | ◎    |    |
| 2  | ハードウェアテスト (折り返し通信テスト)                  | 6.2.46 | ○    |    |

備考. ◎ : 必須 ○ : 推奨 △ : 任意

6.2.1 メイン処理

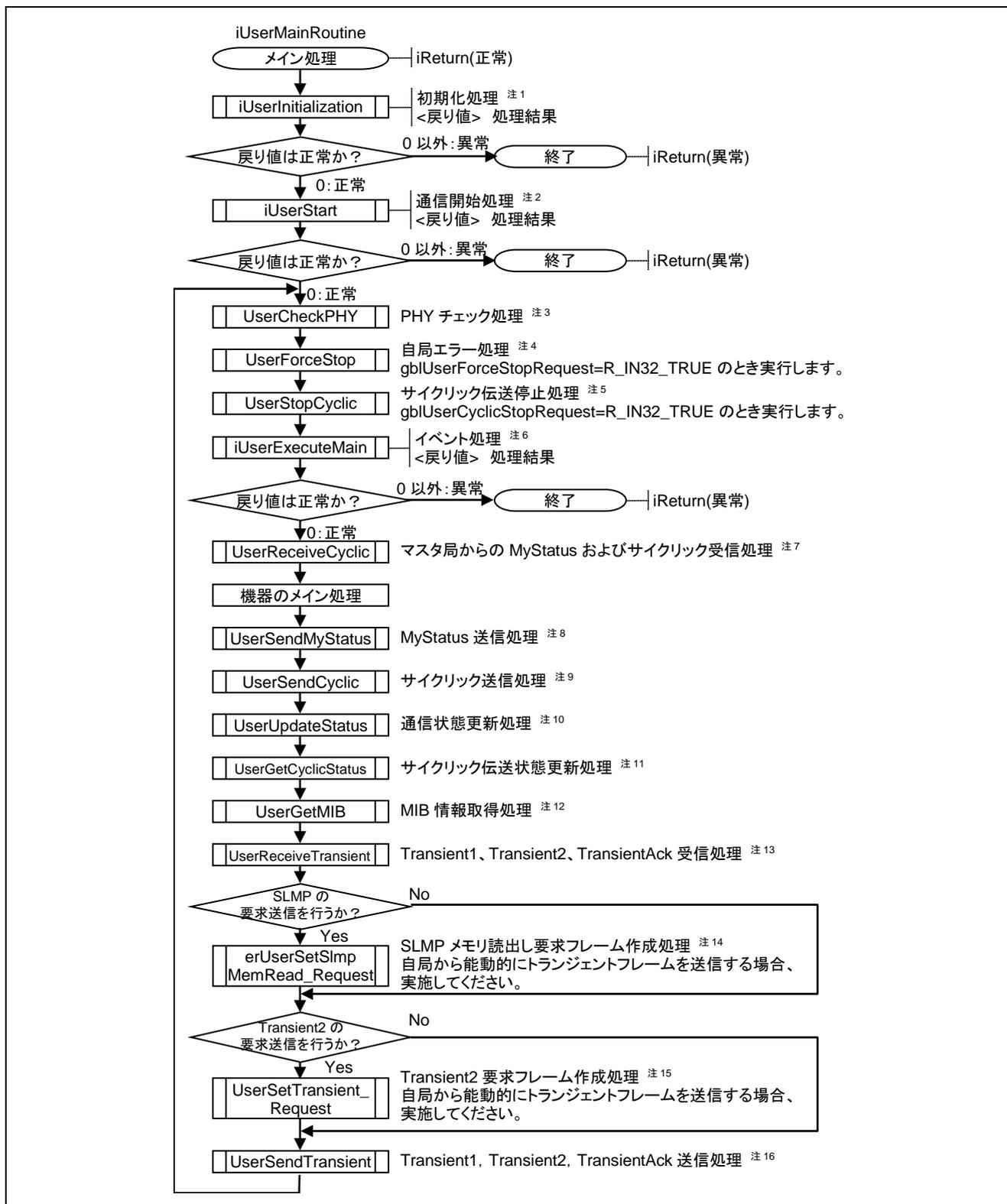


図6.3 メイン処理 フロー図

- 注 1. 詳細は、「6.2.2 初期化処理」を参照してください。
2. 詳細は、「6.2.3 通信開始処理」を参照してください。
3. 詳細は、「6.2.4 PHYチェック処理」を参照してください。
4. 詳細は、「6.2.6 自局エラー処理」を参照してください。
5. 詳細は、「6.2.7 サイクリック伝送停止処理」を参照してください。
6. 詳細は、「6.2.8 イベント処理」を参照してください。
7. 詳細は、「6.2.9 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。
8. 詳細は、「6.2.10 MyStatus送信処理」を参照してください。
9. 詳細は、「6.2.11 サイクリック送信処理」を参照してください。
10. 詳細は、「6.2.12 通信状態更新処理」を参照してください。
11. 詳細は、「6.2.13 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、「6.2.14 MIB情報取得処理」を参照してください。
13. 詳細は、「6.2.15 Transient1、Transient2、TransientAck受信処理」を参照してください。
14. 詳細は、「6.2.41 SLMPメモリ読み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、「6.2.16 Transient2要求フレーム作成処理」を参照してください。
16. 詳細は、「6.2.17 Transient1、Transient2、TransientAck送信処理」を参照してください。

## 6.2.2 初期化処理

R-IN32M3-CL の初期化、R-IN32M3-CL 内部 WDT の有効/無効およびノード番号/ネットワーク番号を設定します。

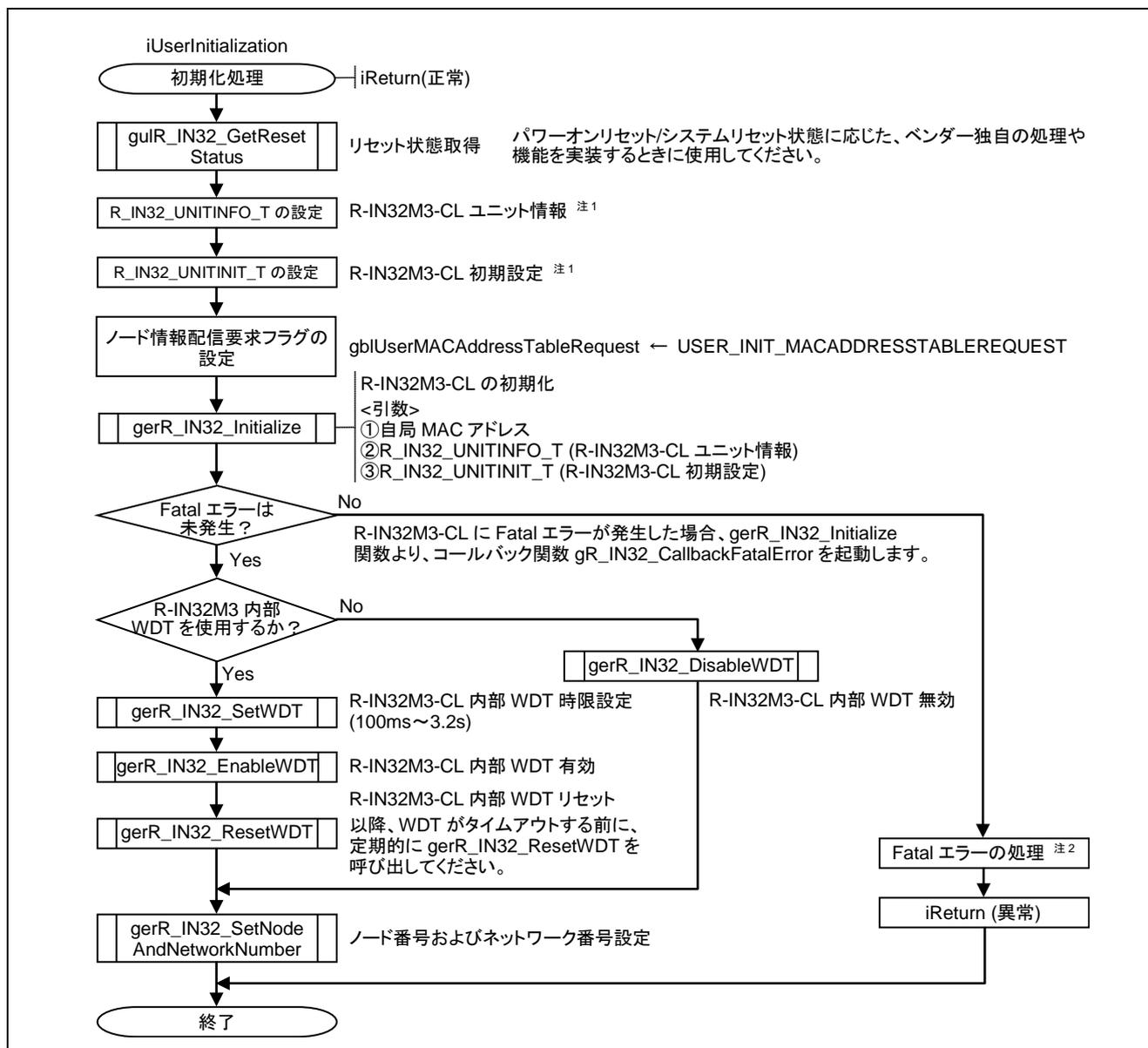


図6.4 初期化処理 フロー図

注 1. 詳細は、「6.4.1(2) gerR\_IN32\_Initialize」を参照してください。

2. たと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。

**注意. 【“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。

- ・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。

### 6.2.3 通信開始処理

R-IN32M3-CL に通信開始を指示する処理です。

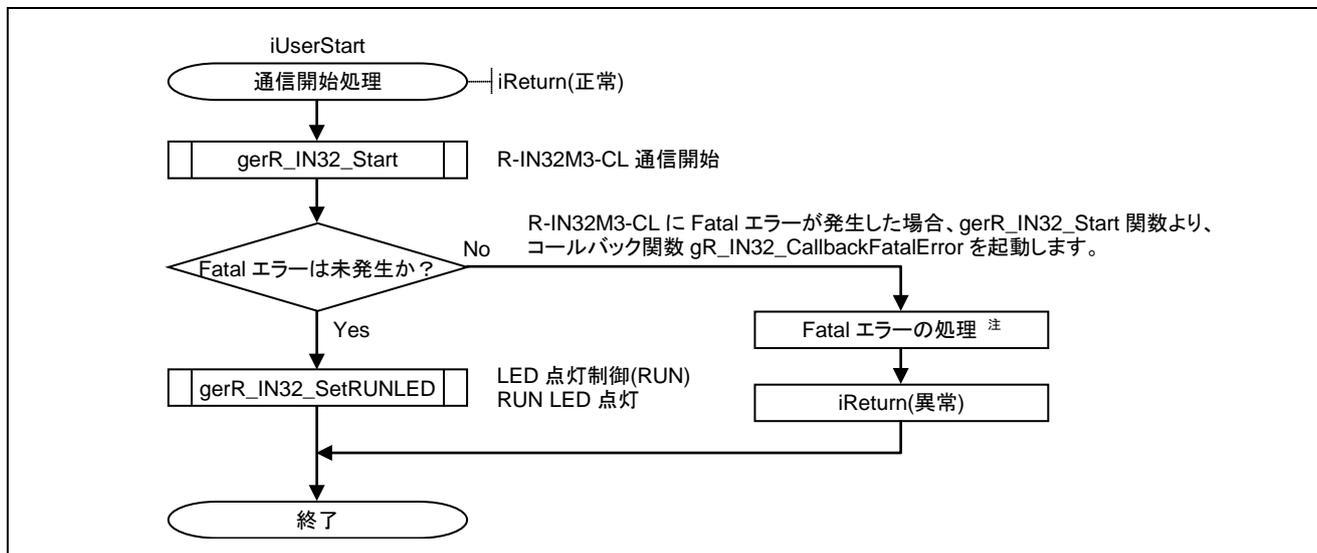


図6.5 通信開始処理 フロー図

注. たと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。

## 6.2.4 PHY チェック処理

R-IN32M3-CL は 1Gbps/全二重でリンクアップしている必要があるため、PHY が 1Gbps/全二重以外でリンクアップしていないかチェックします。

1Gbps/全二重以外でリンクアップしている場合は、PHY の設定を変更します。

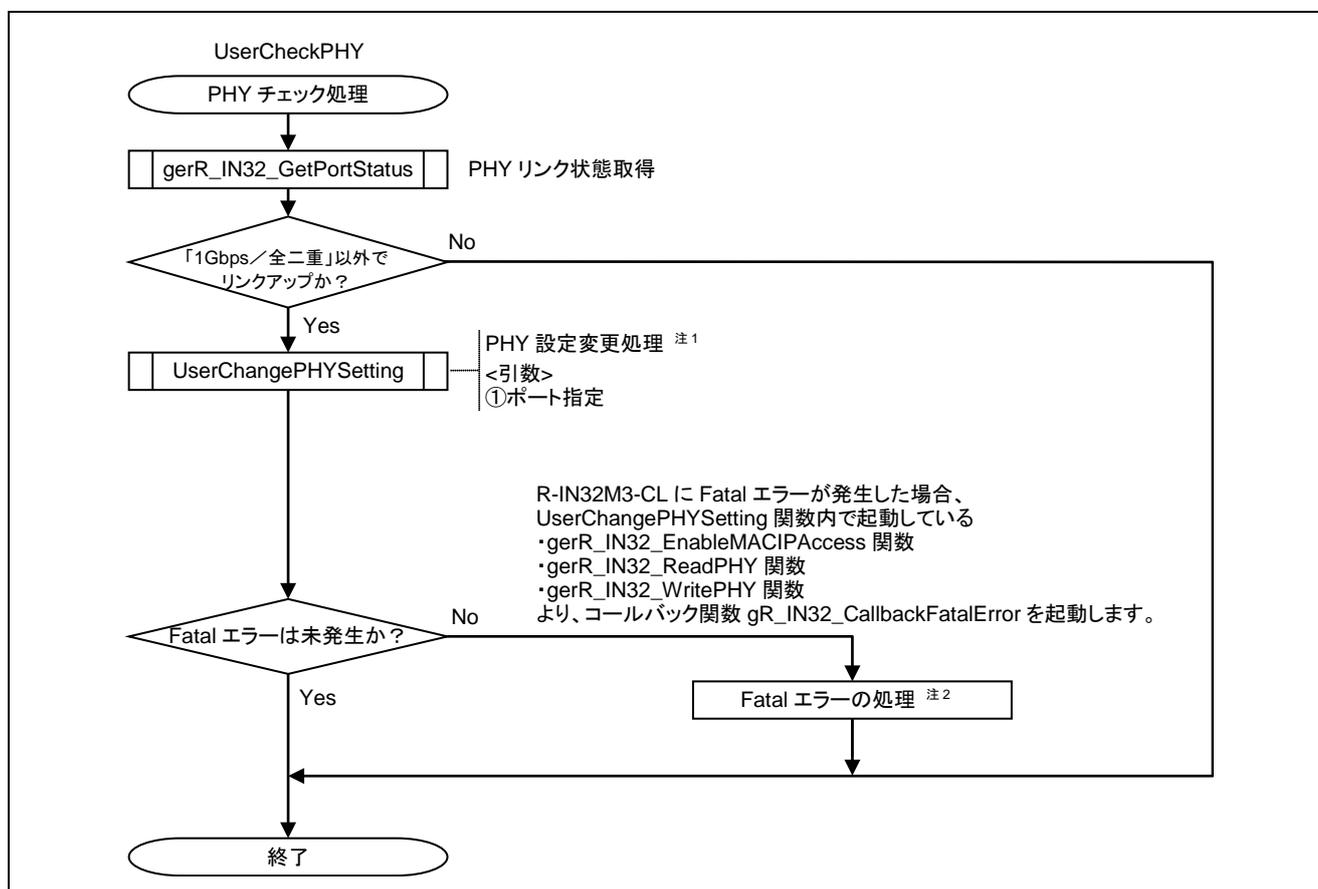


図6.6 PHY チェック処理 フロー図

注 1. 詳細は、「6.2.5 PHY設定変更」を参照してください。

2. たと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。

注意. 上記処理をポート 1 とポート 2 の両方に対して実施してください。

使用する PHY が、ハードウェア設定によって 1Gbps/全二重固定でリンクアップ可能であれば、実装は不要です。

### 6.2.5 PHY 設定変更処理

PHY を 1Gbps/全二重でしかリンクアップしないように設定します。

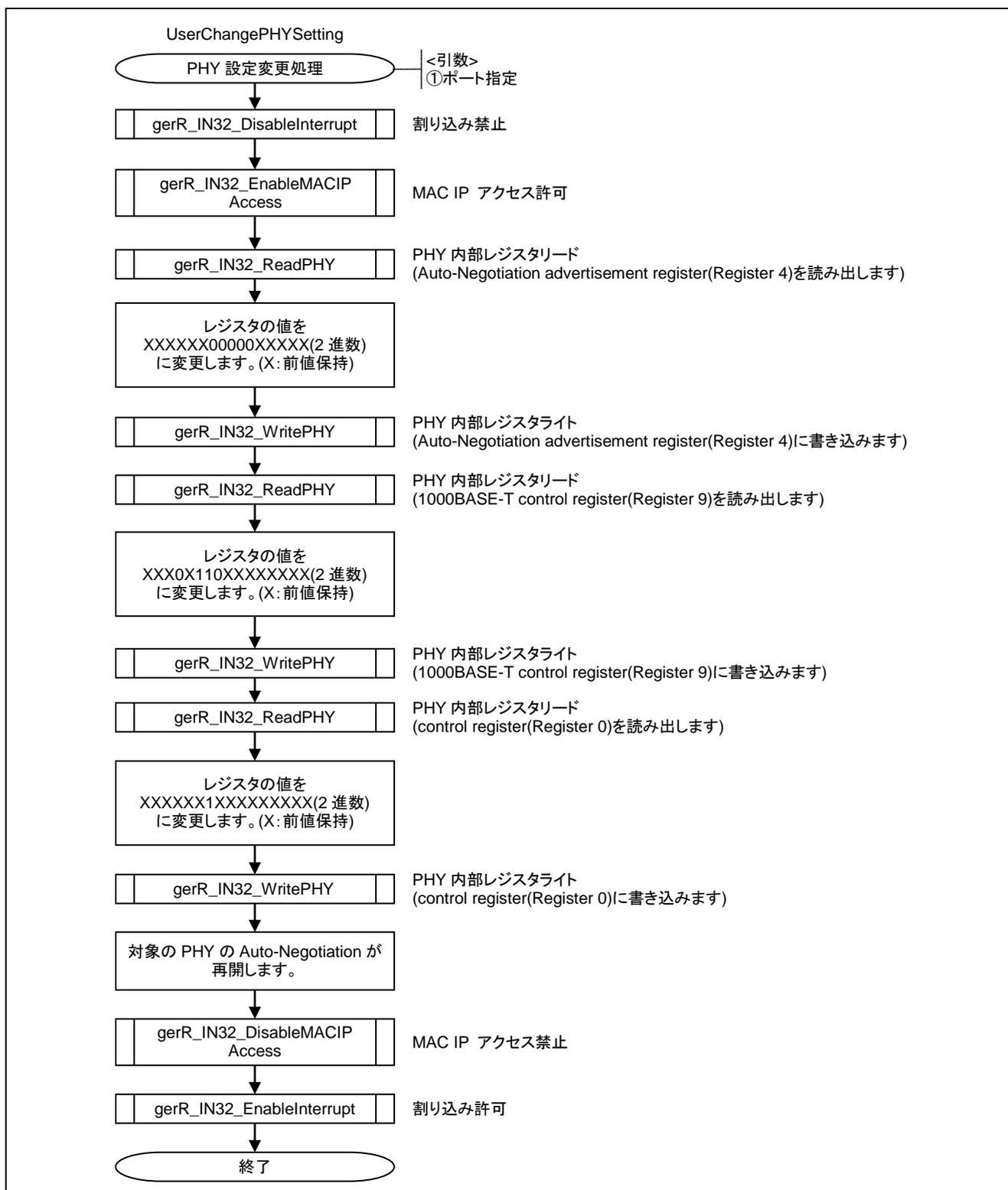


図6.7 PHY 設定変更処理 フロー図

### 6.2.6 自局エラー処理

ベンダー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。（任意で実装する処理です。）

R-IN32M3-CL 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-IN32M3-CL 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。

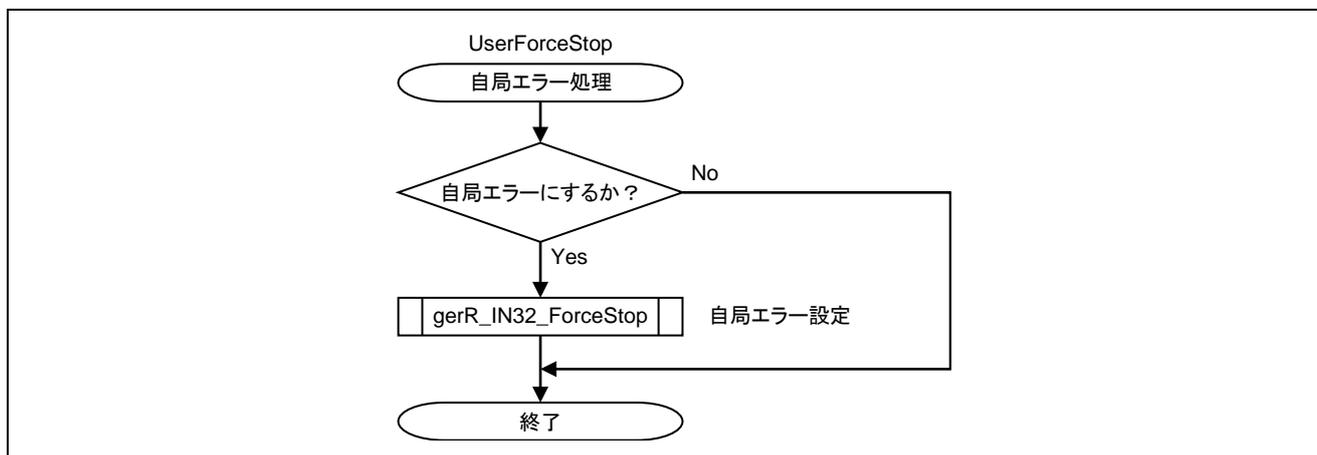


図6.8 自局エラー処理 フロー図

### 6.2.7 サイクリック伝送停止処理

機器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。（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。）

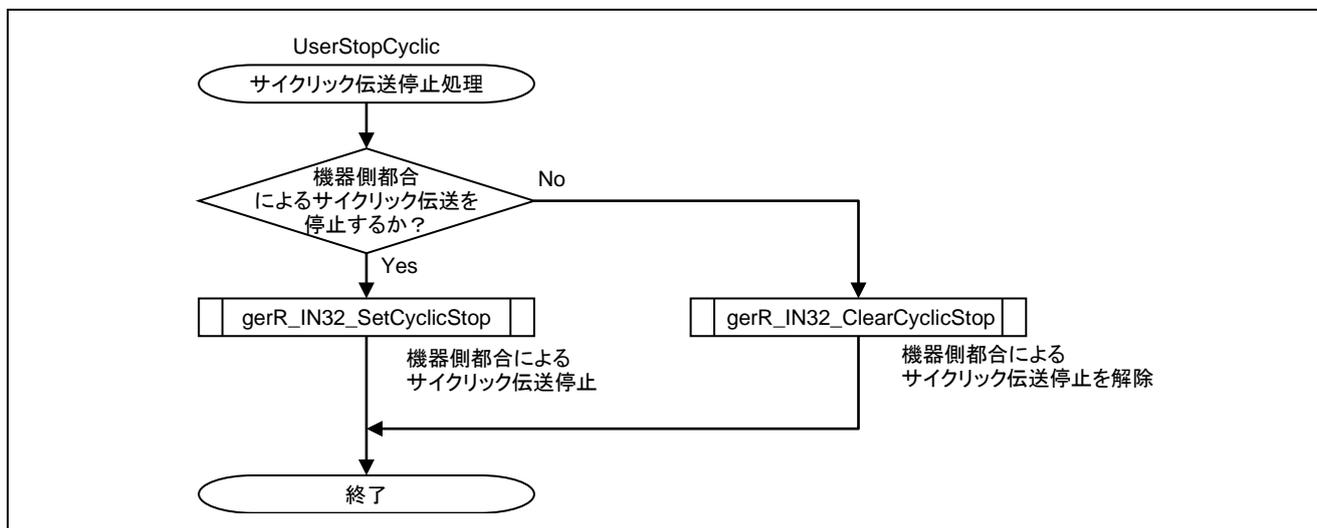


図6.9 サイクリック伝送停止処理 フロー図

## 6.2.8 イベント処理

MPU 割込み (R-IN32M3-CL イベント) を検出し、イベントの処理と MIB 情報の更新を行います。

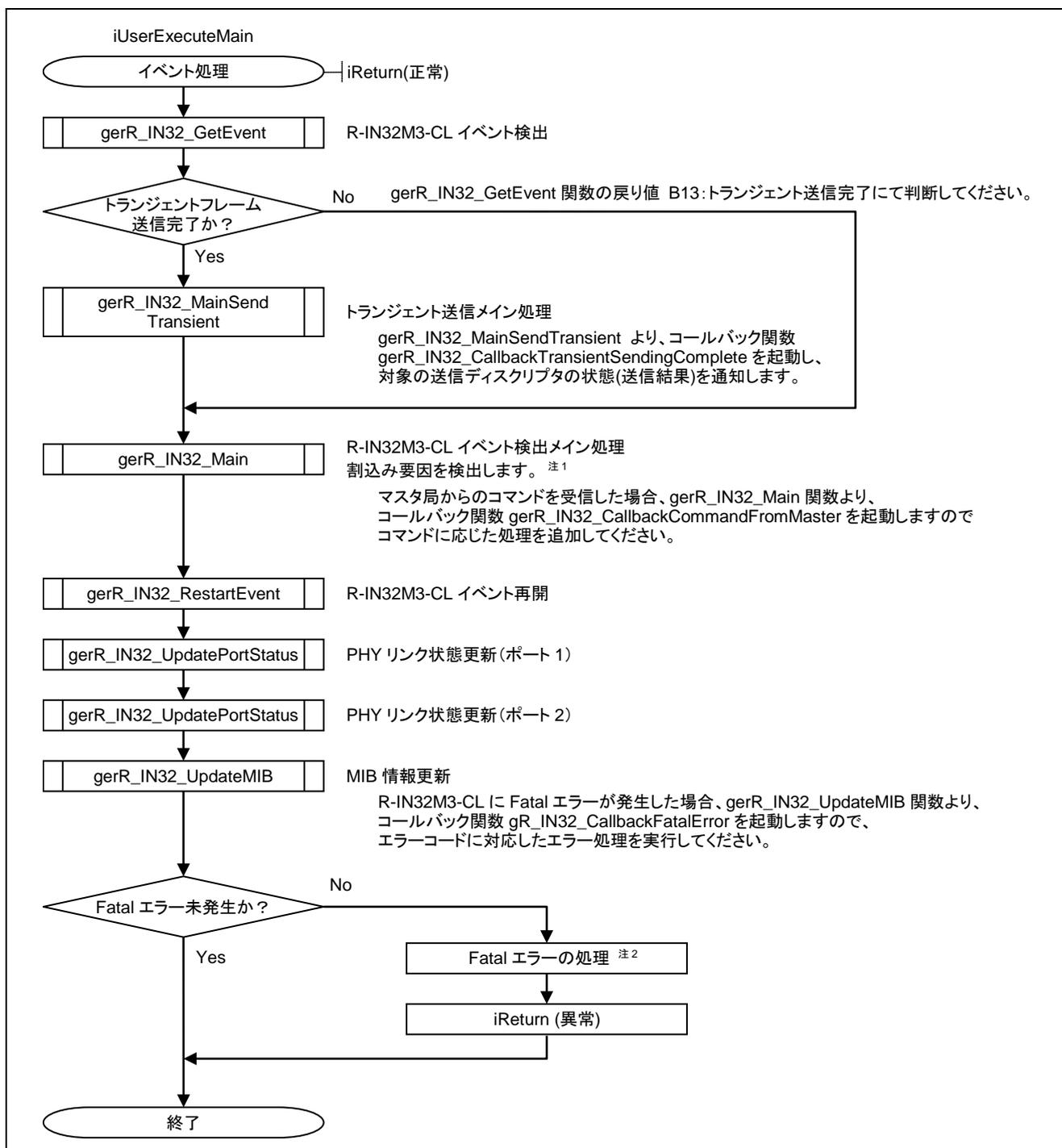


図6.10 イベント処理 フロー図

注 1. 詳細は、「6.4.3 イベント」を参照してください。

2. たと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。

### 6.2.9 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 (RY・RWw) 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、開発機器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) する処理です。

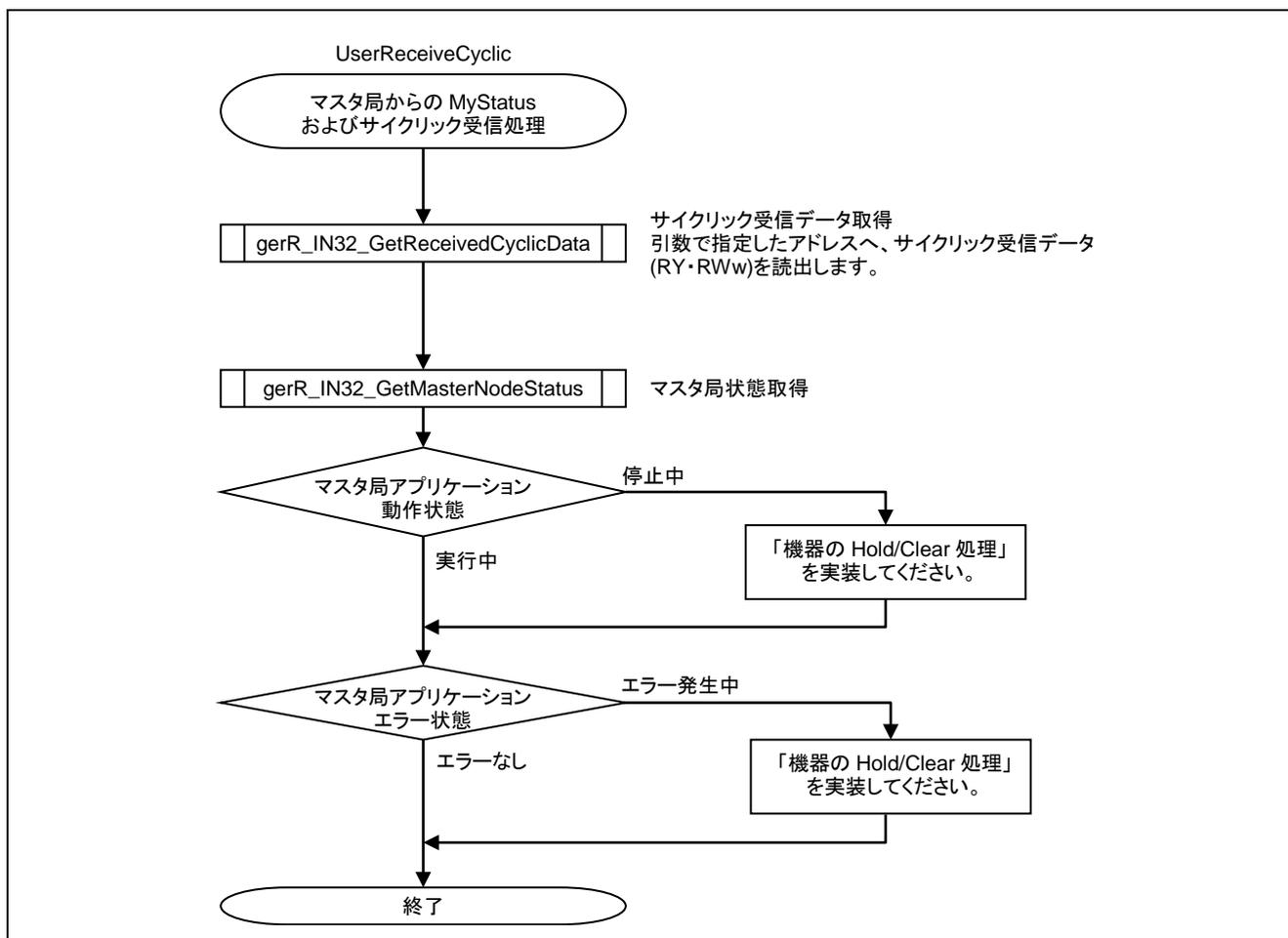


図6.11 マスタ局からの MyStatus およびサイクリック受信処理 フロー図

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 (RY、RWw) について

マスタ局アプリケーション停止/異常のとき、マスタ局が送信するサイクリックデータは、マスタ局の設定によって Hold または Clear されています。(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-IN32M3-CL 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-IN32M3-CL 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-IN32M3-CL ドライバが取得する<br>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 <sup>※</sup> | エラーあり <sup>※</sup> | 取得しない<br>(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 ユニットの場合、“実行中かつエラーあり”の状態は、存在しません。

### 6.2.10 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-IN32M3-CL によって自動的に送信されます。

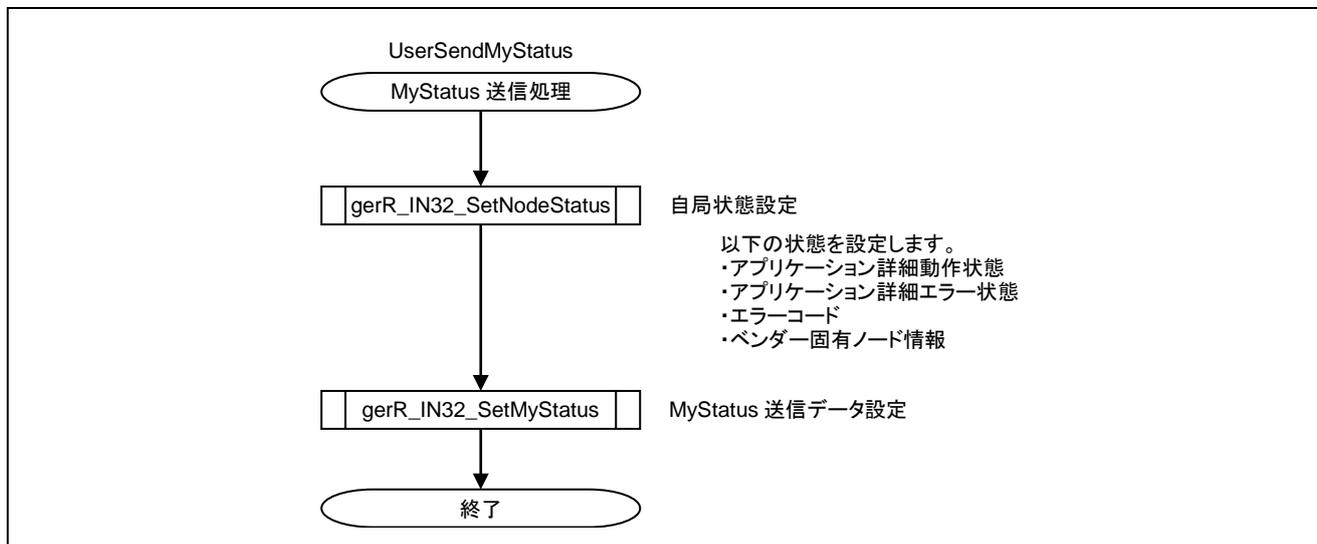


図6.12 MyStatus 送信処理 フロー図

### 6.2.11 サイクリック送信処理

サイクリック送信データ (RX・RW<sub>r</sub>) を送信します。

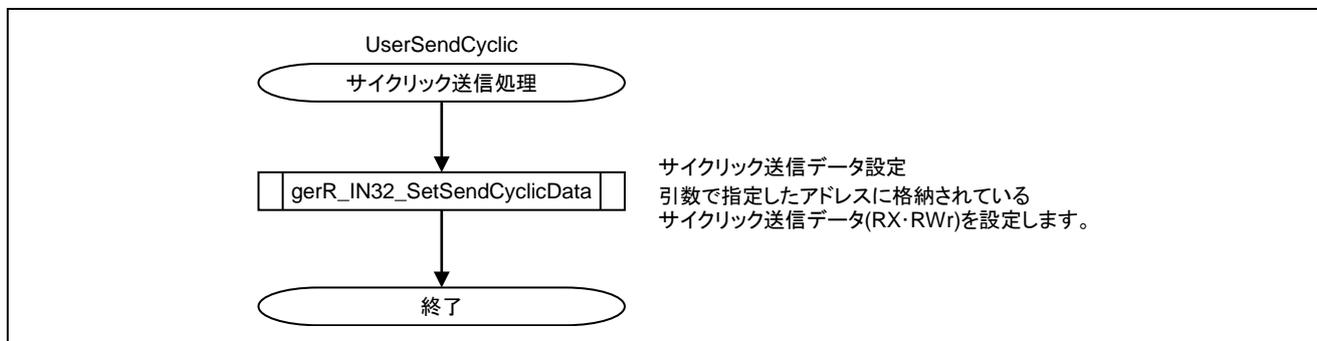


図6.13 サイクリック送信処理 フロー図

### 6.2.12 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、D LINK LED および ERR.LED の点灯/消灯の制御を行います。

Hold/Clear 処理は、開発機器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) する処理です。

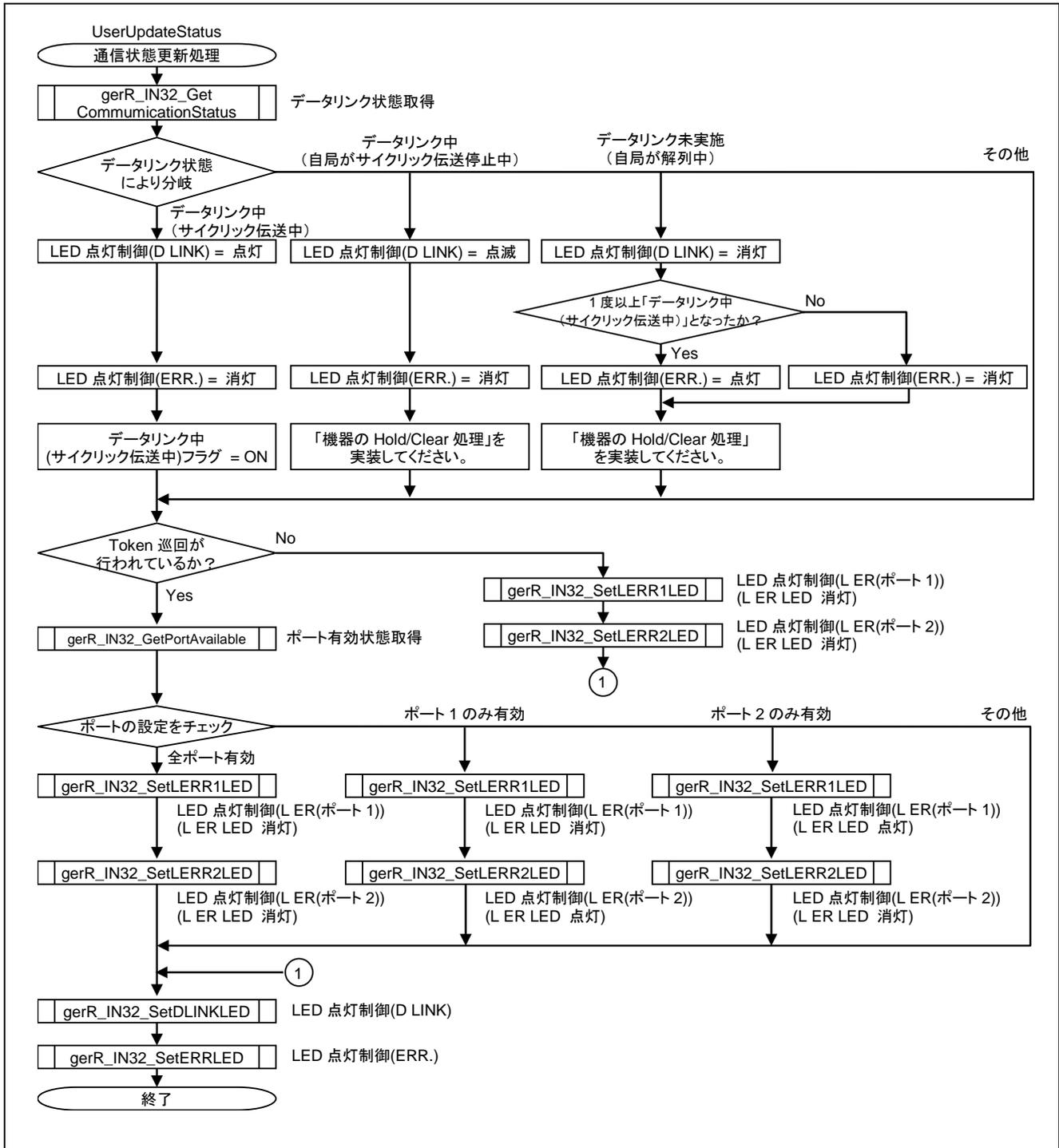


図6.14 通信状態更新処理 フロー図

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-IN32M3-CL 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-IN32M3-CL 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-IN32M3-CL 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<br>（自局が解列中）                    | 取得しない<br>（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<br>（自局がサイクリック伝送停止中） <sup>※</sup> | マスター局が“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。

### 6.2.13 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できます。

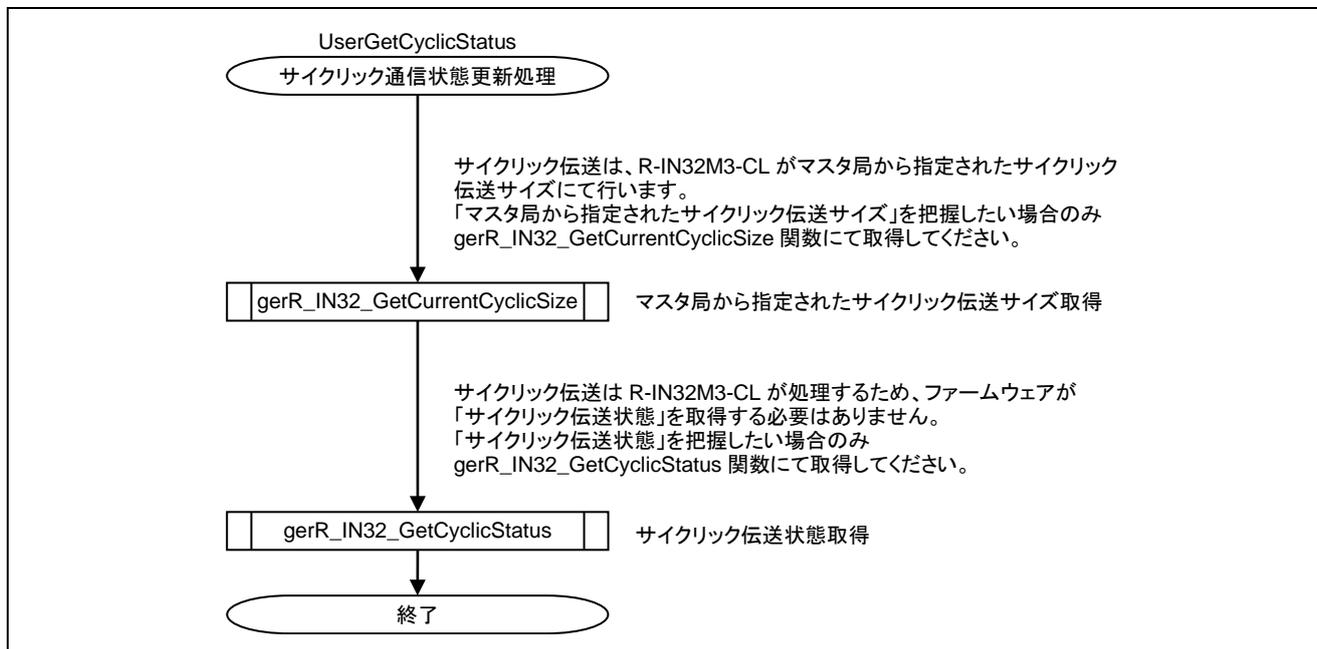


図6.15 サイクリック伝送状態更新処理 フロー図

## 6.2.14 MIB 情報取得処理

MIB 情報を取得またはクリアします。

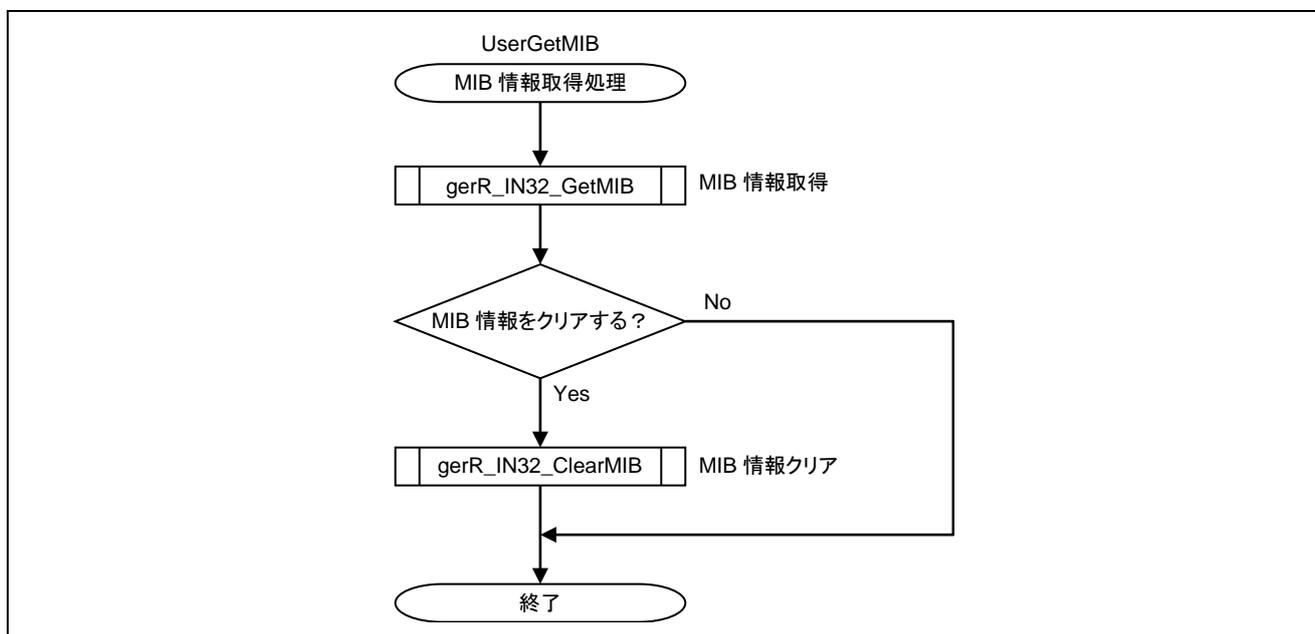


図6.16 MIB 情報取得処理 フロー図

**注意.** MIB 情報は非公開情報ですので、エンドユーザー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6.6 リング制御部 MIB 情報一覧表

| No | MIB 情報             | 説明                                  |
|----|--------------------|-------------------------------------|
| 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6.7 MAC IP 部 MIB 情報一覧表

| No | MIB 情報           | 説明  |
|----|------------------|---|
| 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。<br>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。<br>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 <sup>注</sup> )、カウントアップする。<br>アイドル中 (RECV_*_DV=1 <sup>注</sup> ) に RECV_*_ERR <sup>注</sup> が複数回 1 となっても、カウントは 1 回になる。<br>注：「*」はワイルド文字です。(A：ポート 1、B：ポート 2) |
| 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。<br>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。<br>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。   |

## (3) その他の MIB 情報一覧

表6.8 その他 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    | R-IN32M3-CL が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   | R-IN32M3-CL が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 | R-IN32M3-CL が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。 |

### 6.2.15 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。

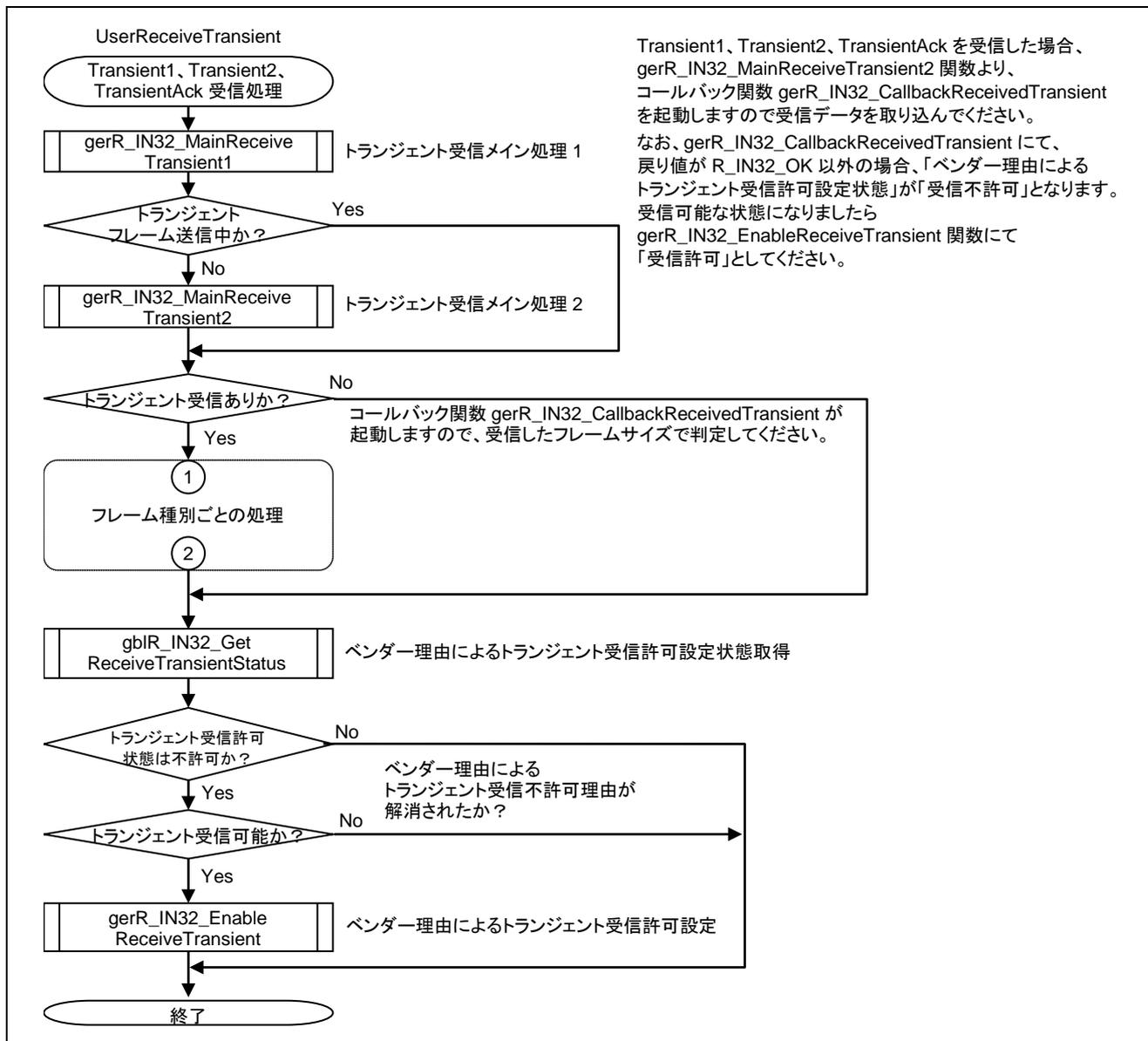


図6.17 Transient1、Transient2、TransientAck 受信処理 フロー図 (1/2)

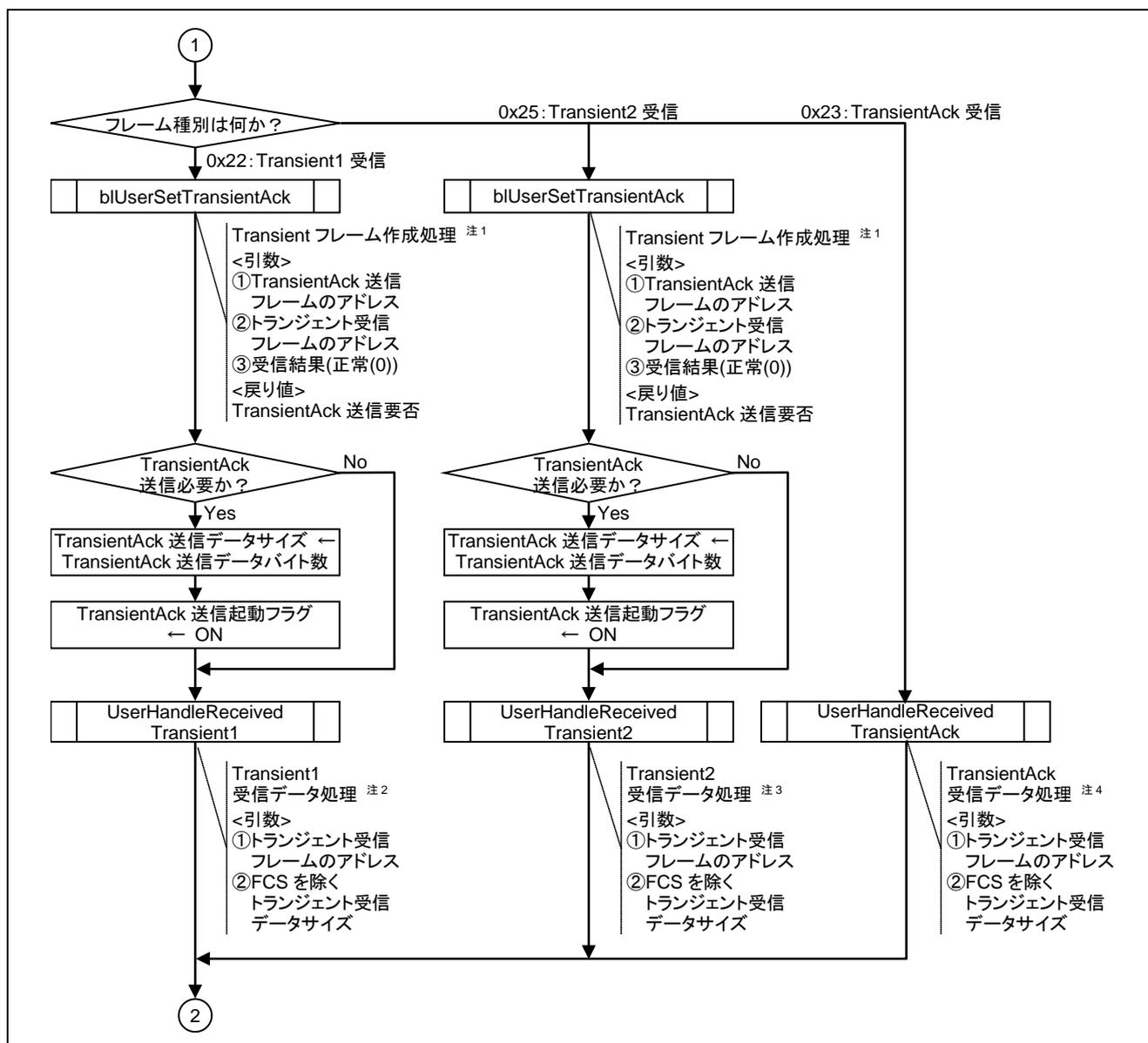


図6.17 Transient1、Transient2、TransientAck 受信処理 フロー図 (2/2)

- 注 1. 詳細は、「6.2.34 TransientAckフレーム作成」を参照してください。
2. 詳細は、「6.2.18 Transient1受信データ処理」を参照してください。
3. 詳細は、「6.2.31 Transient2受信データ処理」を参照してください。
4. 詳細は、「6.2.33 TransientAck受信データ処理」を参照してください。

### 6.2.16 Transient2 要求フレーム作成処理

Transient2 要求フレームを作成する処理手順を、Transient2 メモリ読出し要求の例で示します。

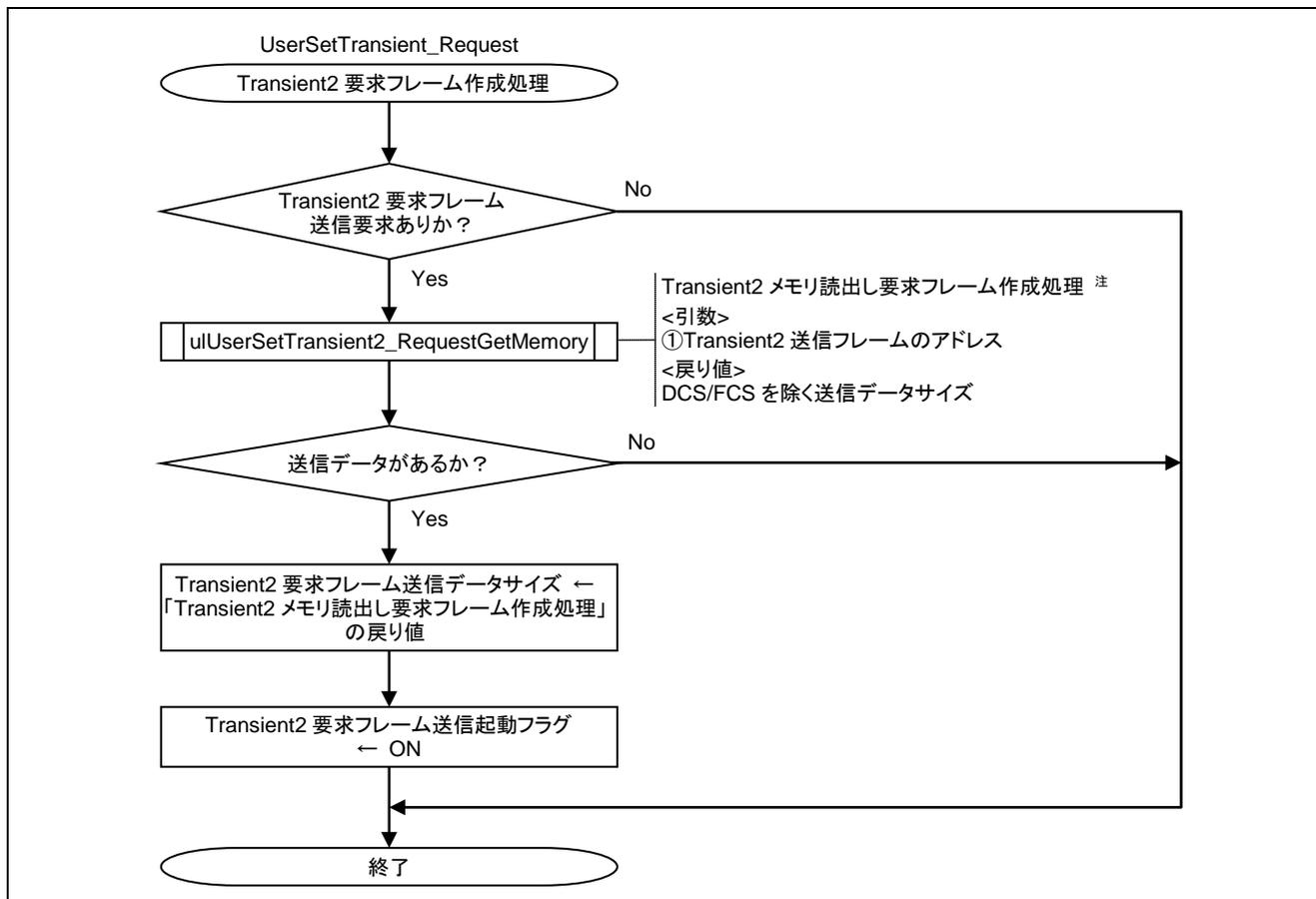


図6.18 Transient2 要求フレーム作成処理 フロー図

**注. 詳細は、「6.2.36 Transient2メモリ読出し要求フレーム作成」を参照してください。**

本フローでは、「Transient2 メモリ読出し要求」を例にしています。  
必要に応じて処理を実装してください。

### 6.2.17 Transient1、Transient2、TransientAck 送信処理

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

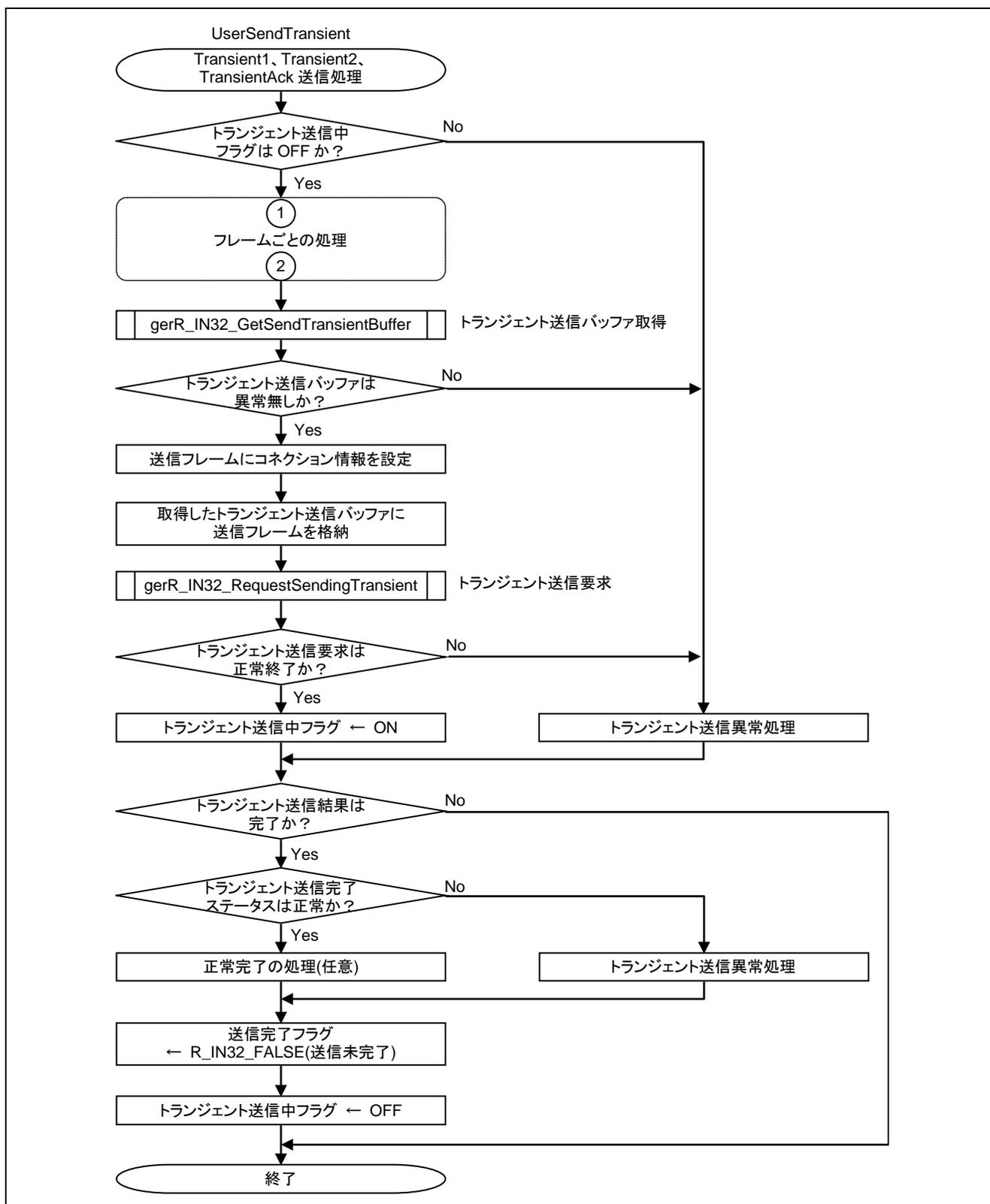


図6.19 Transient1、Transient2、TransientAck 送信処理 フロー図 (1/2)

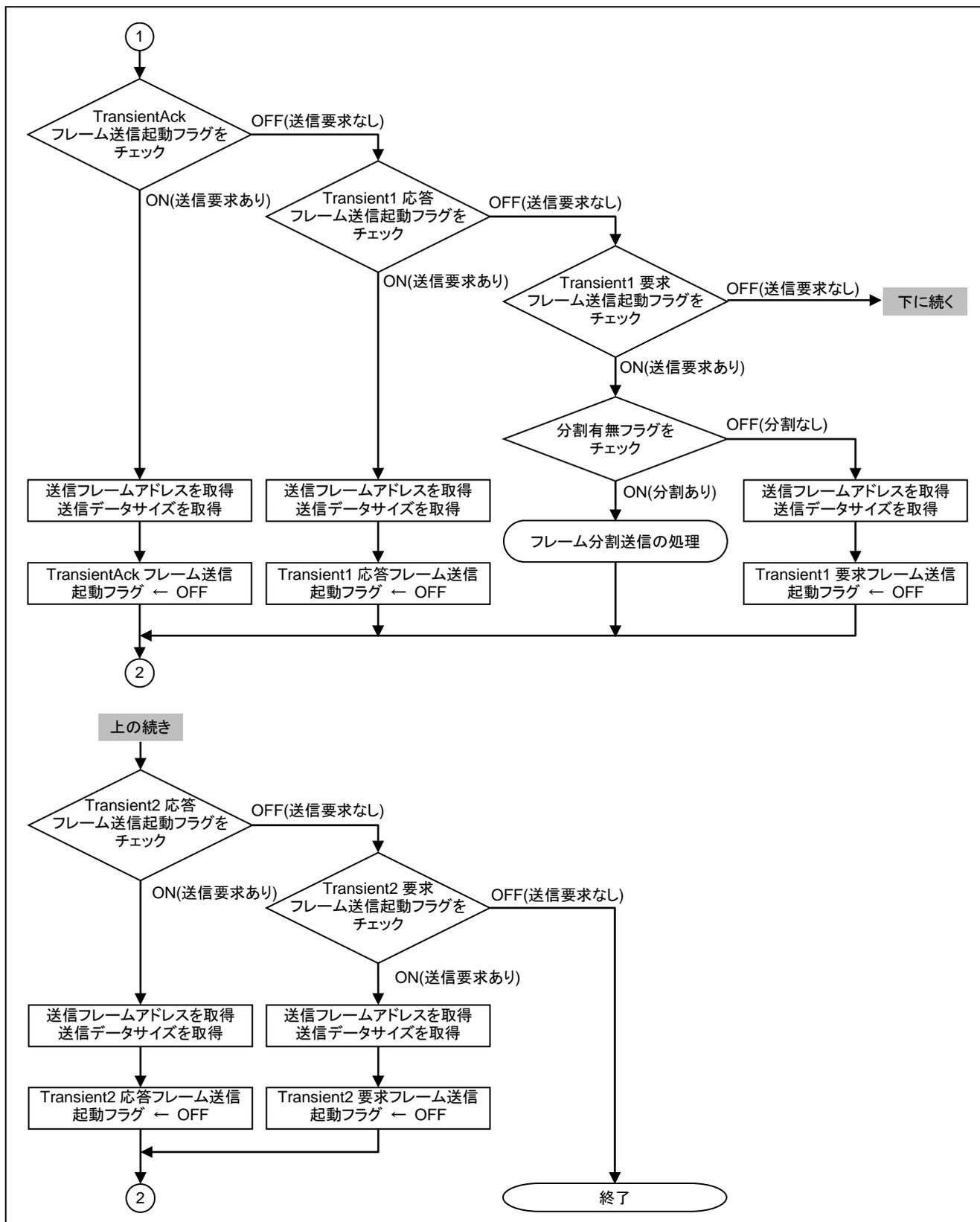


図6.19 Transient1、Transient2、TransientAck 送信処理 フロー図 (2/2)

【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。開発機器の仕様に応じて、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「5.3.2 CC-Link IEフィールド固有トランジェントフレームフォーマット」を参照してください。

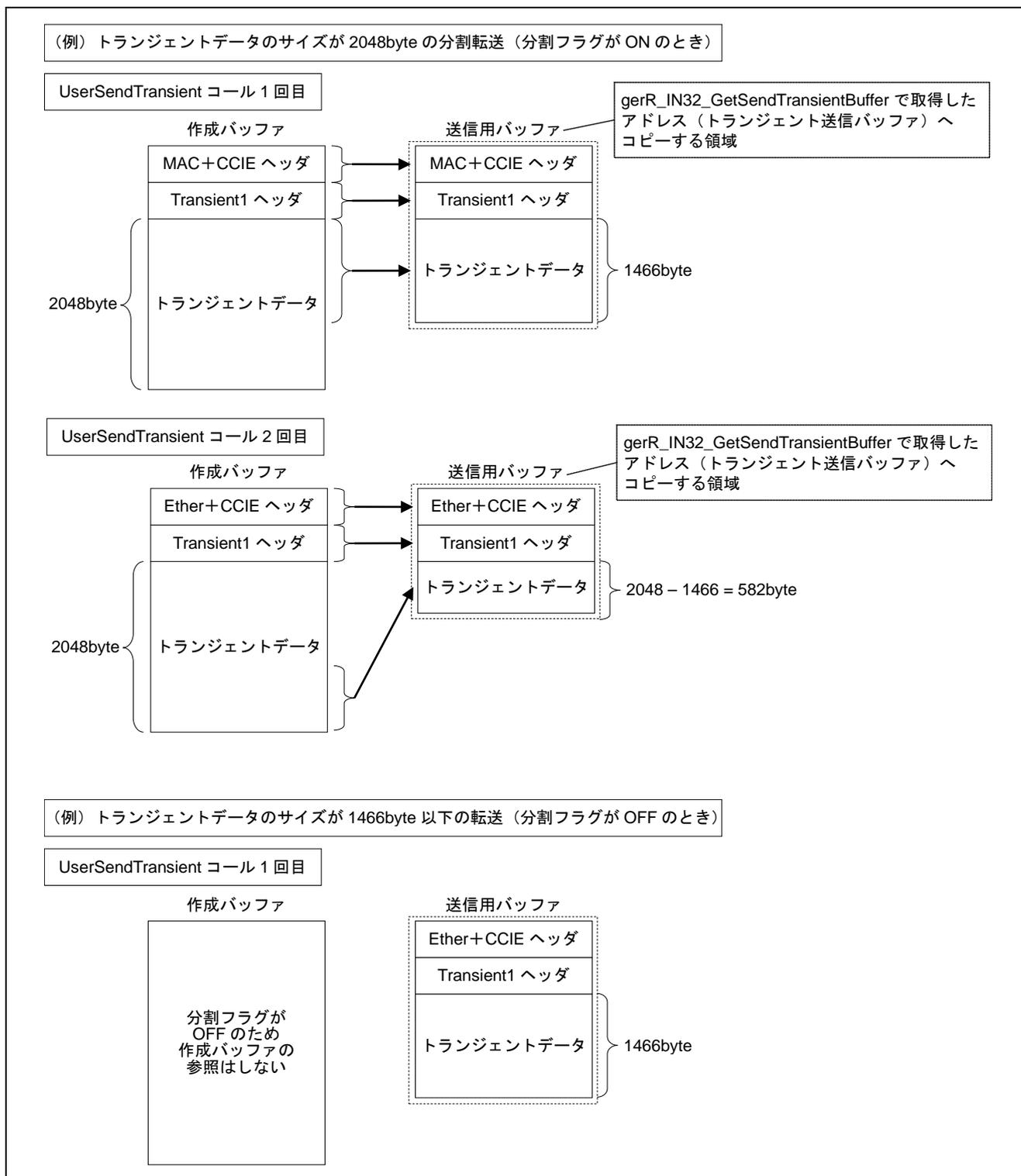


図6.20 トランジェントフレーム分割送信処理のイメージ

### 6.2.18 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。

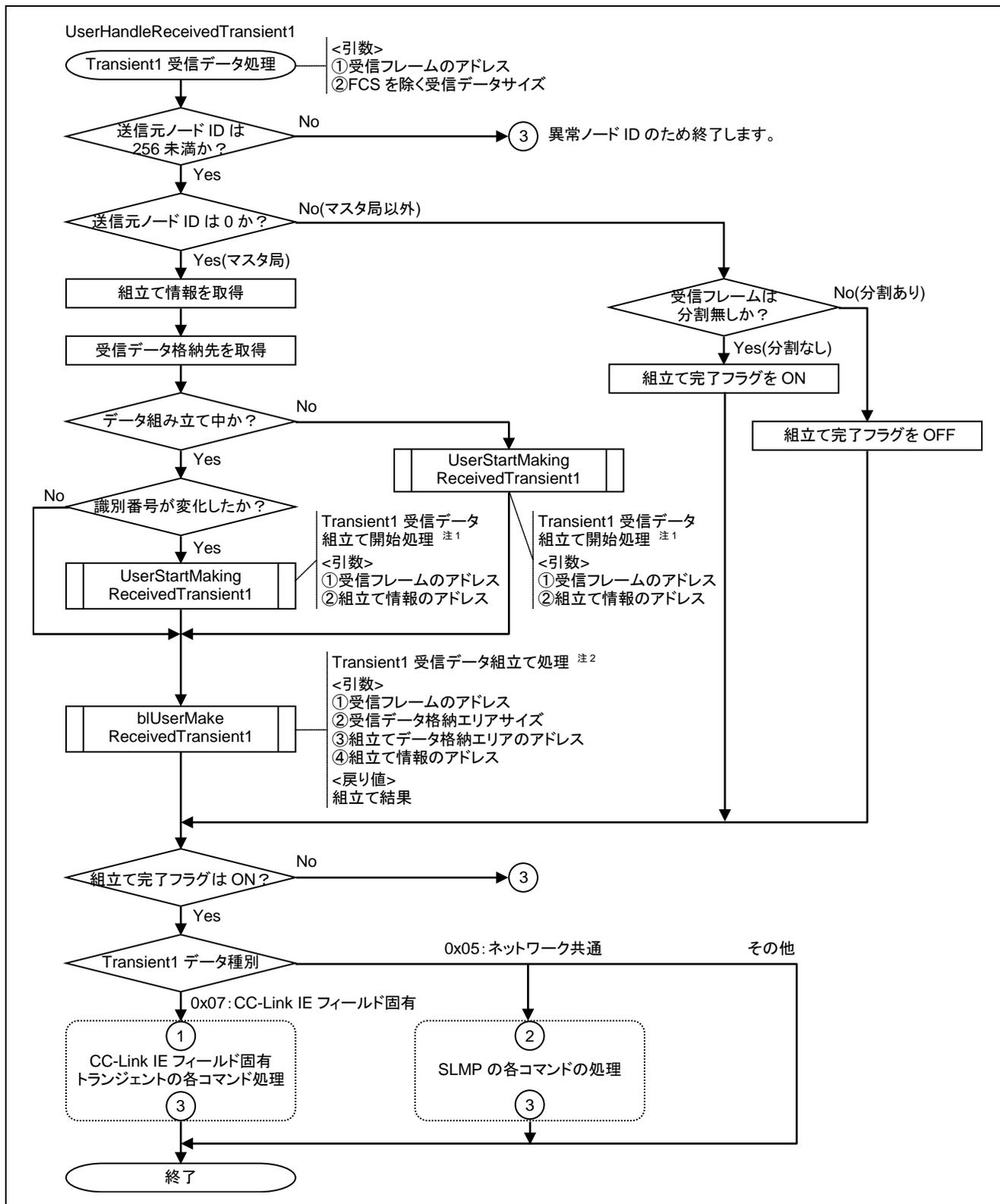


図6.21 Transient1 受信データ処理 フロー図 (1/3)

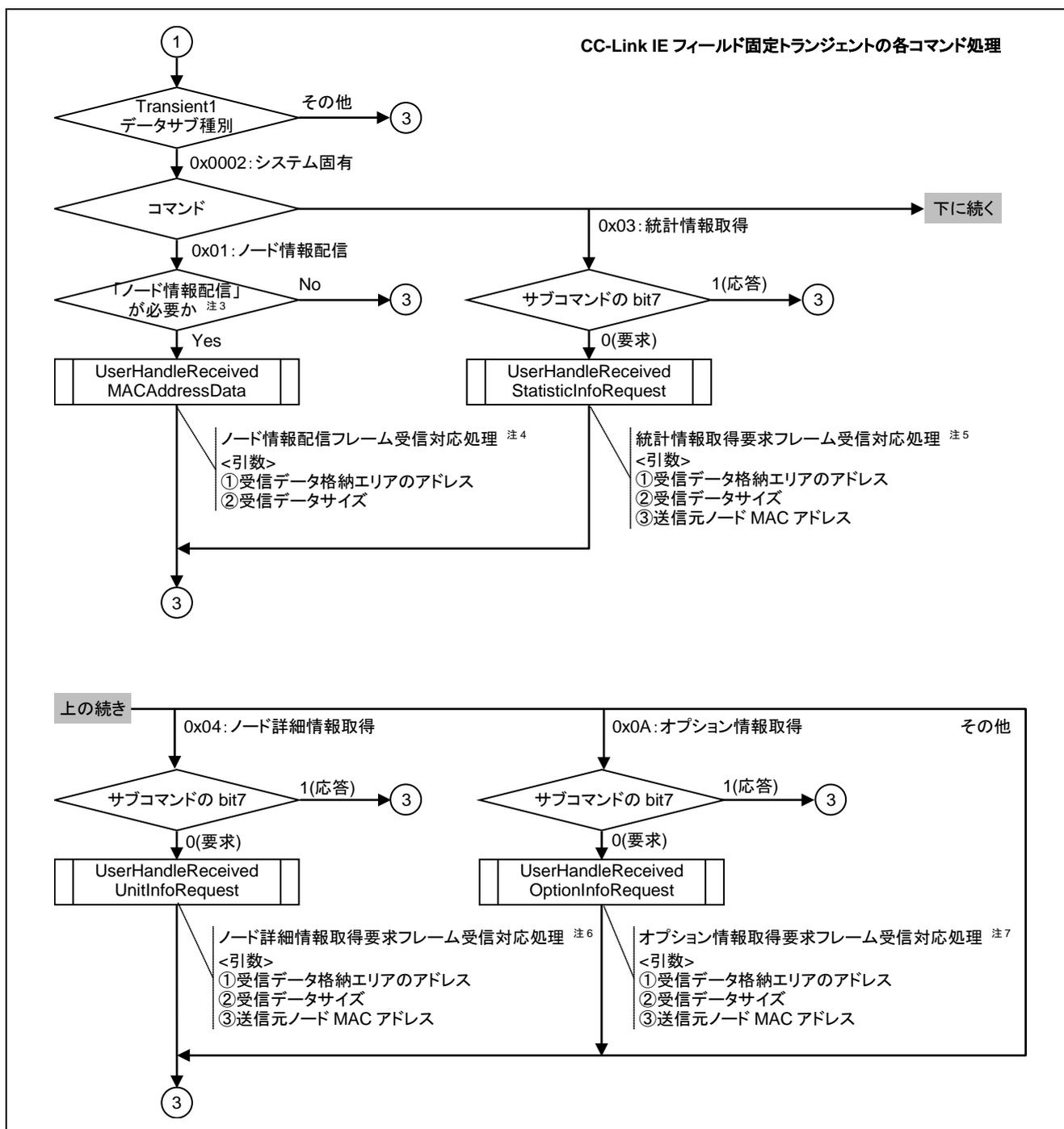


図6.21 Transient1 受信データ処理 フロー図 (2/3)

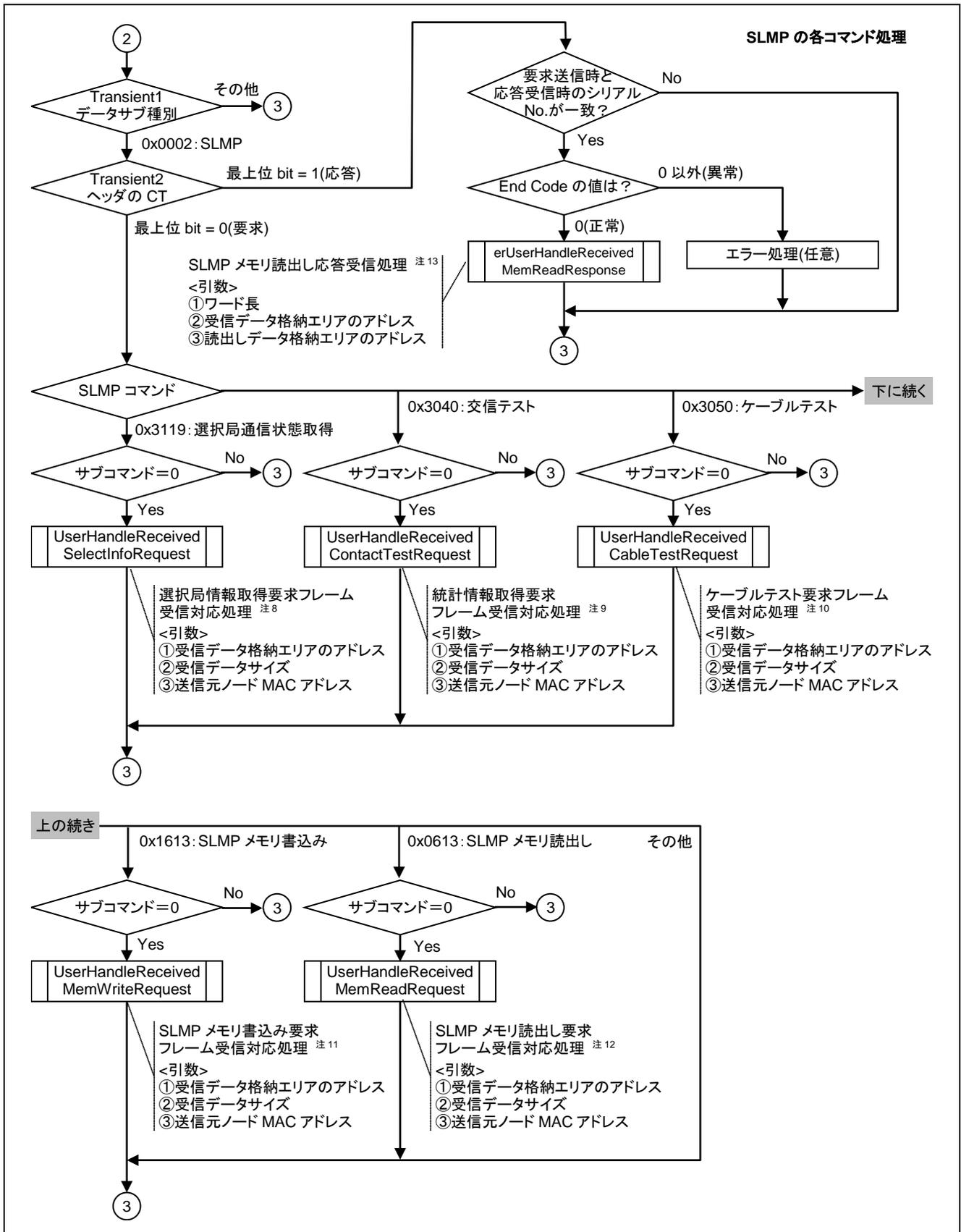


図6.21 Transient1 受信データ処理 フロー図 (3/3)

- 注 1. 詳細は、「6.2.19 Transient1受信データ組立て開始」を参照してください。
2. 詳細は、「6.2.20 Transient1受信データ組立て」を参照してください。
3. gerR\_IN32\_InitializeのB)R\_IN32\_UNITINIT\_T初期設定 (g)ノード情報配信の要求でR\_IN32\_FALSEと設定した場合は、「ノード情報配信」が不要です。  
この場合、gblUserMACAddressTableRequestもR\_IN32\_FALSEと設定してください。
4. 詳細は、「6.2.21 ノード情報配信フレーム受信対応処理」を参照してください。
5. 詳細は、「6.2.23 統計情報取得要求フレーム受信対応処理」を参照してください。
6. 詳細は、「6.2.25 ノード詳細情報取得要求フレーム受信対応処理」を参照してください。
7. 詳細は、「6.2.27 オプション情報取得要求フレーム受信対応処理」を参照してください。
8. 詳細は、「6.2.28 選択局情報取得要求フレーム受信対応処理」を参照してください。
9. 詳細は、「6.2.29 交信テスト要求フレーム受信対応処理」を参照してください。
10. 詳細は、「6.2.30 ケーブルテスト要求フレーム受信対応処理」を参照してください。
11. 詳細は、「6.2.40 SLMPメモリ書き込み要求フレーム受信対応処理」を参照してください。
12. 詳細は、「6.2.39 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。
13. 詳細は、「6.2.44 SLMPメモリ読出し応答受信処理」を参照してください。

#### 【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。

マスタ局からの SLMP 要求フレームに対して、サーバが SLMP 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。

ここでは選択局情報を例に示します。交信テスト、ケーブルテスト、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。

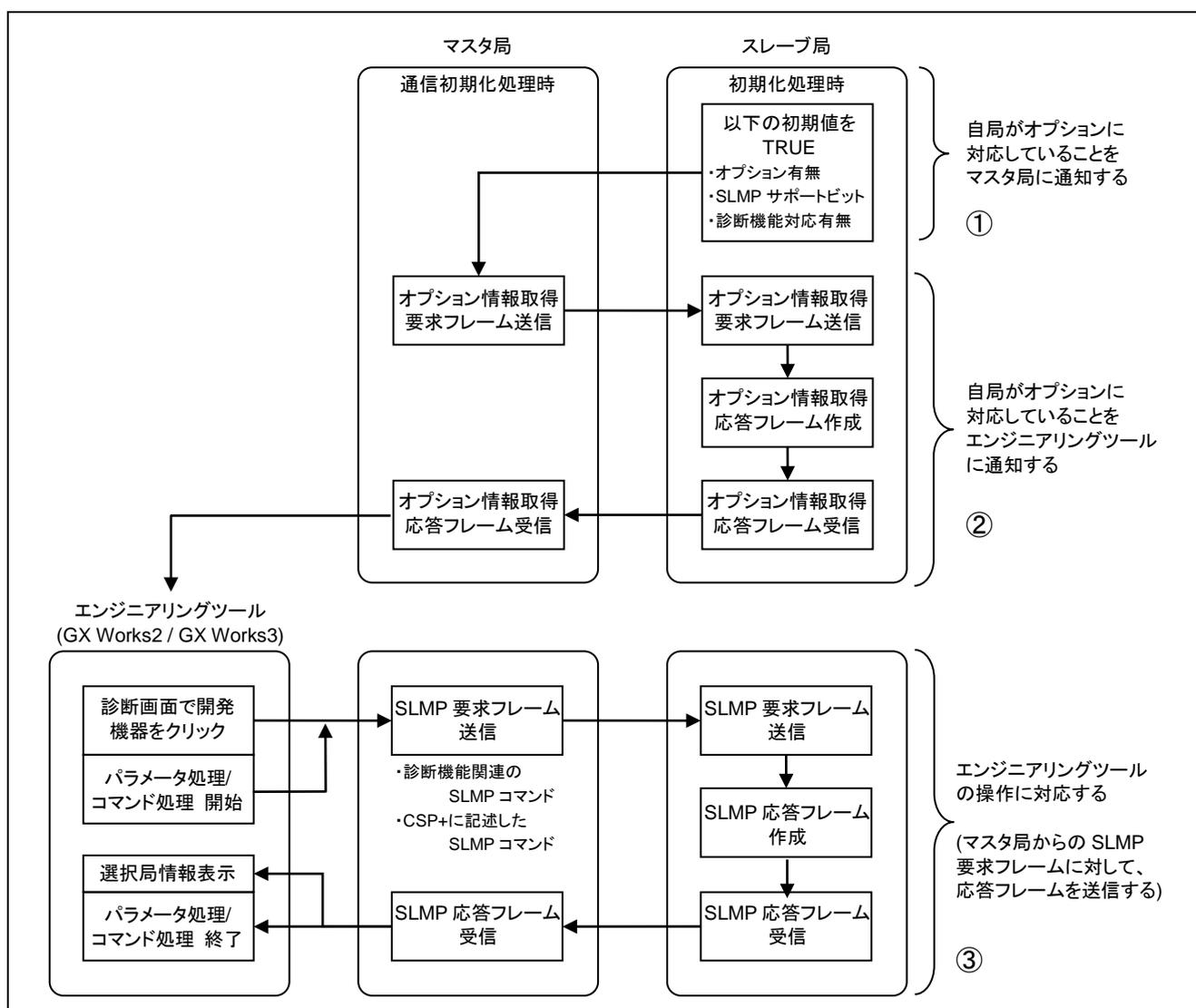


図6.22 マスタ局からの SLMP 要求処理手順 (例)

## ① R\_IN32\_UNITINIT\_Tの設定 (R-IN32M3-CL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。(「6.4.1(2) gerR\_IN32\_Initialize」参照)

- ・ 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回答

UserHandleReceivedOptionInfoRequest (オプション情報取得要求フレーム受信対応処理) にて、マスタ局からのオプション情報取得要求フレームに回答してください。

## ③ 選択局情報取得要求フレームに回答

UserHandleReceivedSelectInfoRequest (選択局情報取得要求フレーム受信対応処理) にて、マスタ局からの選択局情報取得要求フレームに回答してください。

### 6.2.19 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。

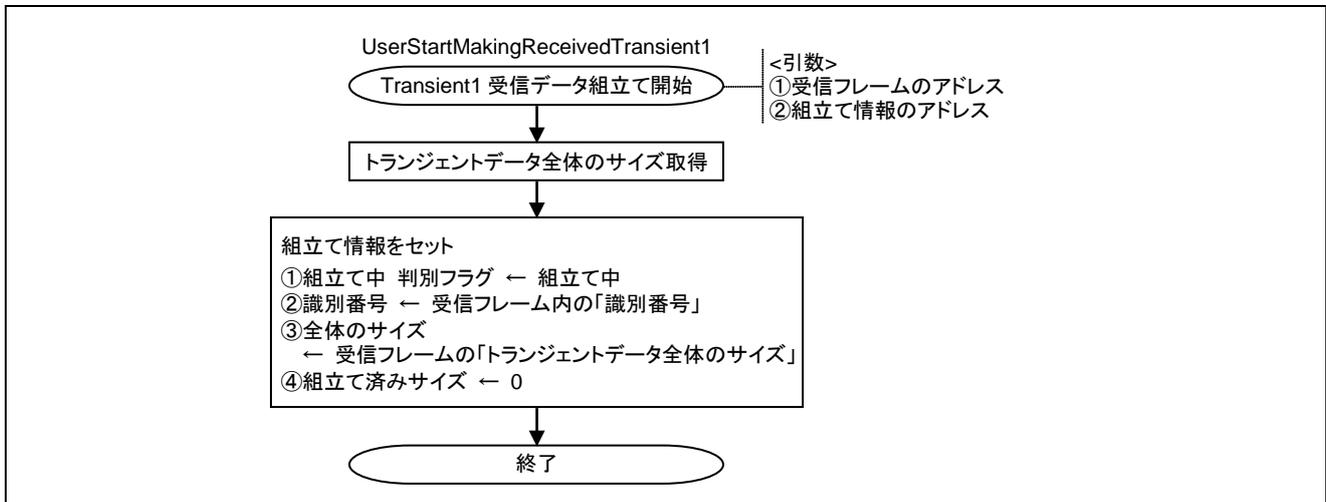


図6.23 Transient1 受信データ組立て開始処理 フロー図

### 6.2.20 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。

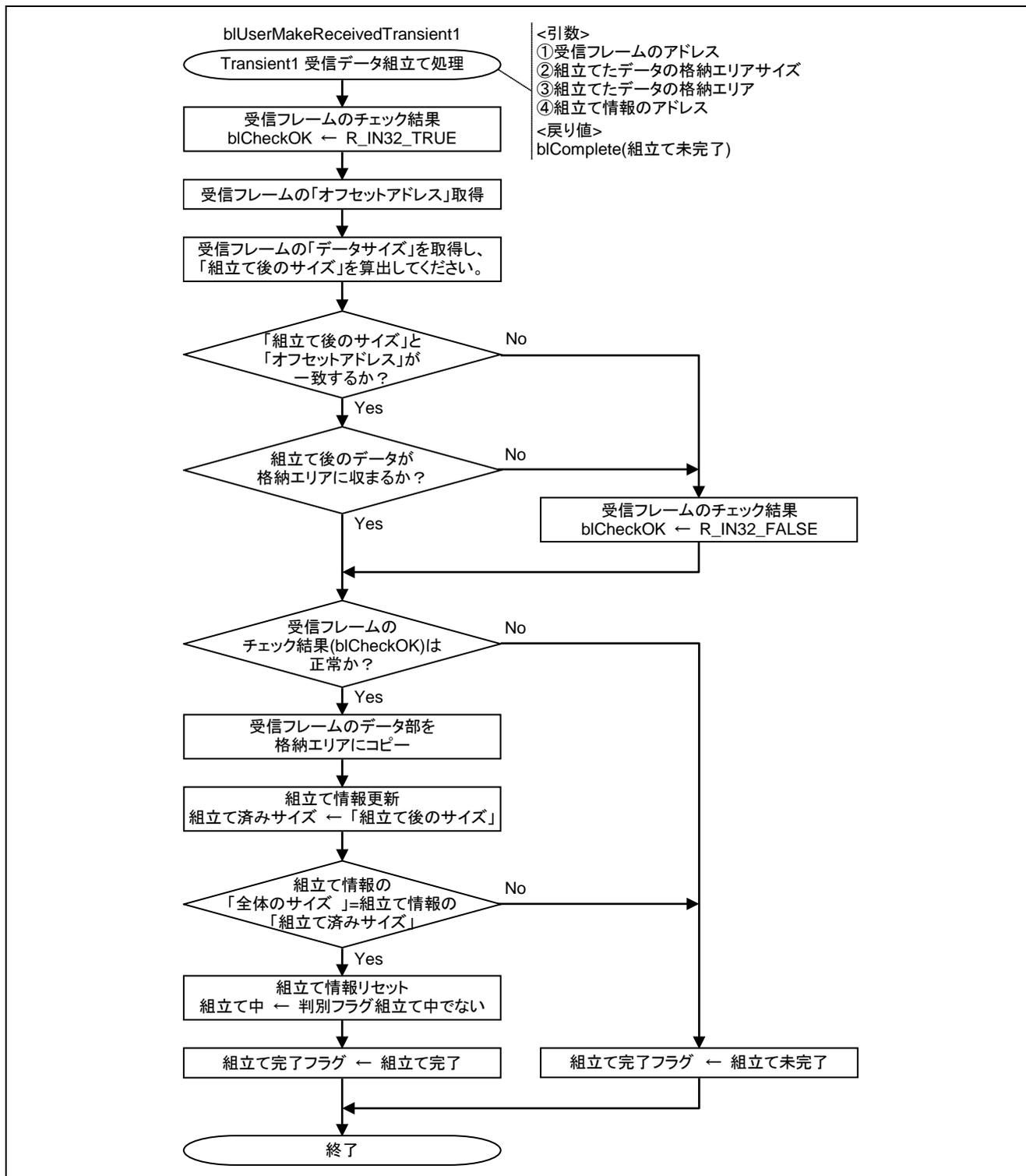


図6.24 Transient1 受信データ組立て処理 フロー図

### 6.2.21 ノード情報配信フレーム受信対応処理

ノード情報配信フレームを受信し、ノードごとの情報を登録します。

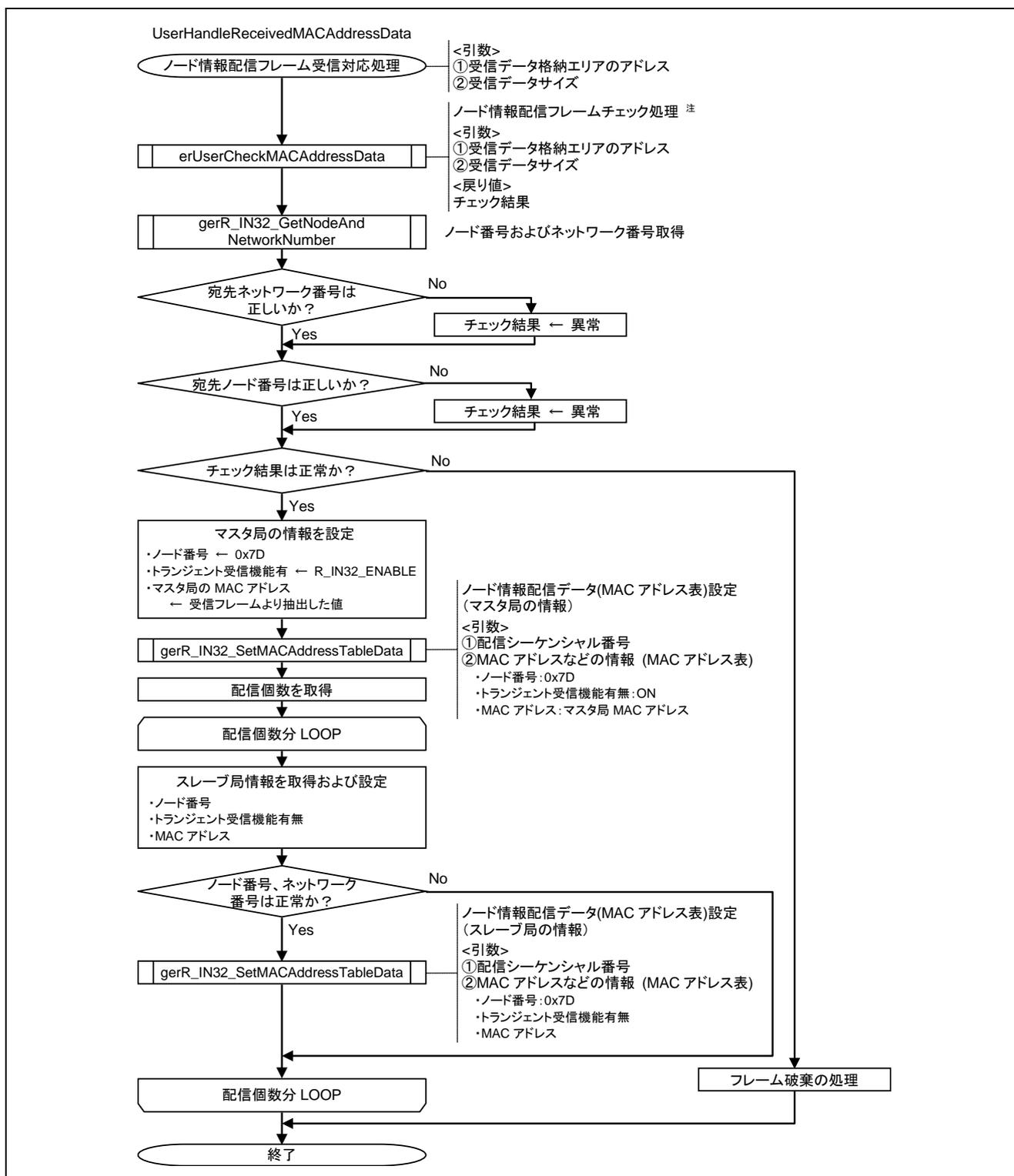


図6.25 ノード情報配信フレーム受信対応処理 フロー図

注. 詳細は、「6.2.22 ノード情報配信フレームチェック」を参照してください。

### 6.2.22 ノード情報配信フレームチェック処理

ノード情報配信フレーム内のデータをチェックします。

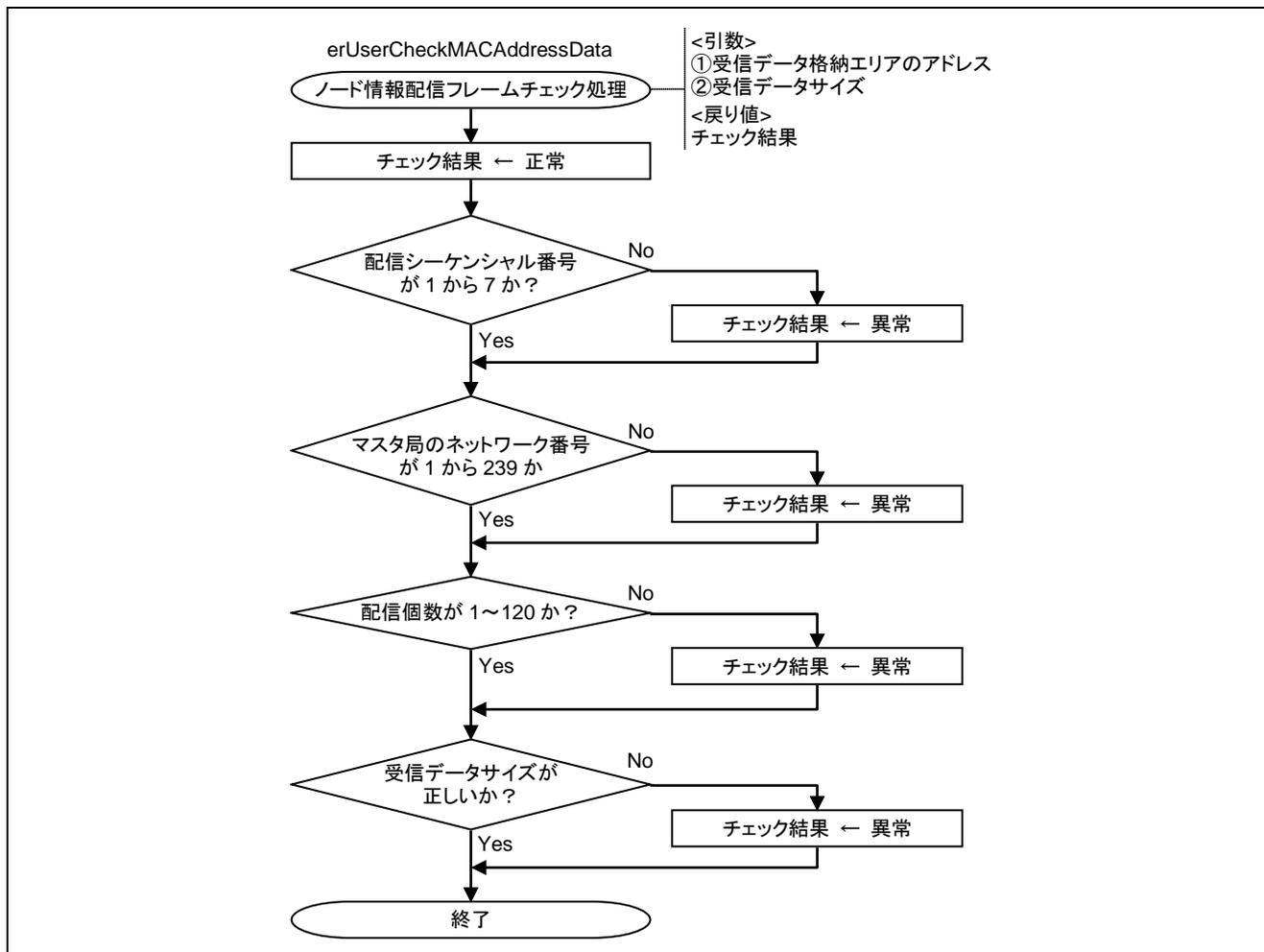


図6.26 ノード情報配信フレームチェック処理 フロー図

### 6.2.23 統計情報取得要求フレーム受信対応処理

統計情報取得要求フレーム受信時の処理を行います。

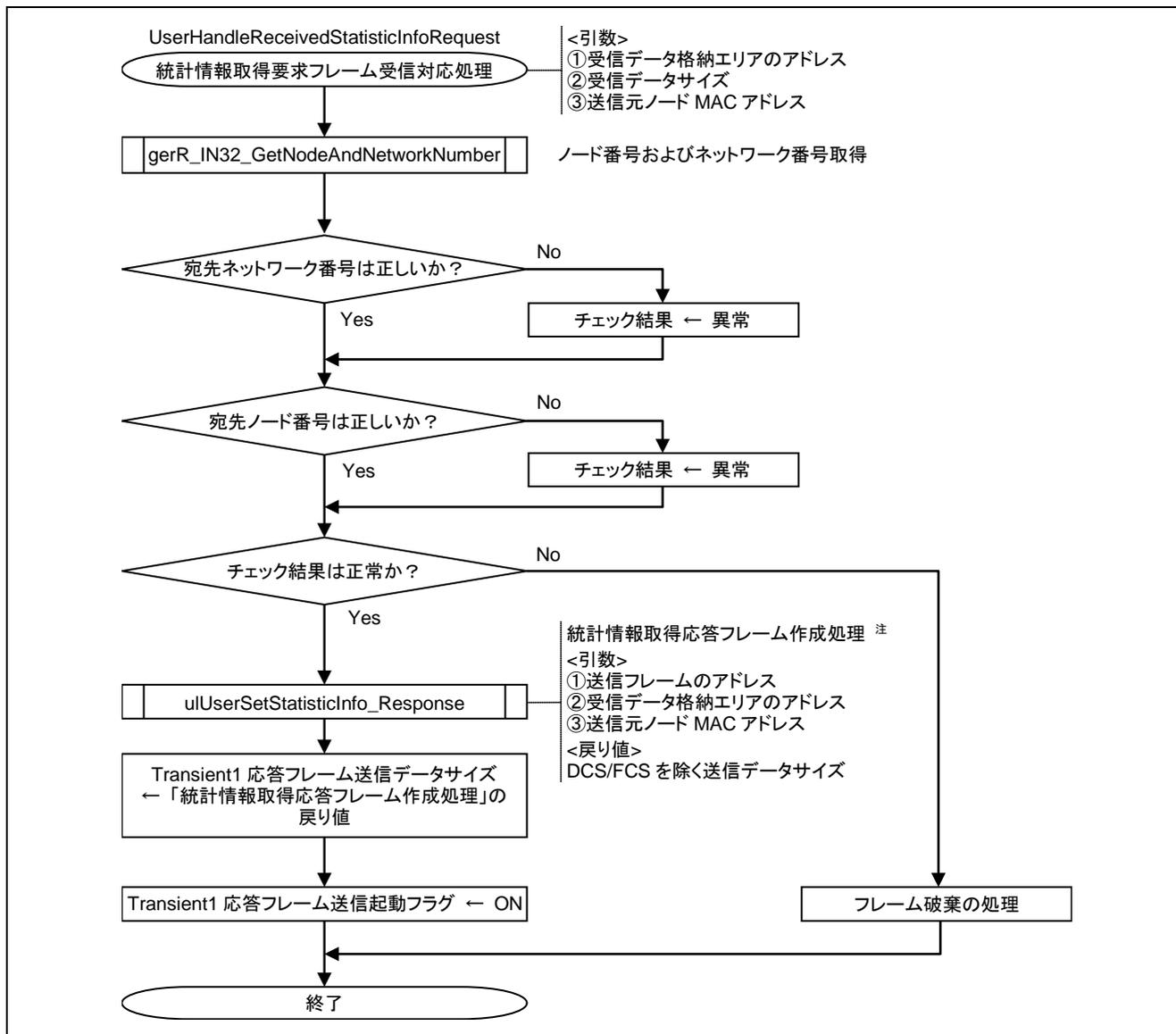


図6.27 統計情報取得要求フレーム受信対応処理 フロー図

注. 詳細は、「6.2.24 統計情報取得応答フレーム作成」を参照してください。

### 6.2.24 統計情報取得応答フレーム作成処理

統計情報取得応答フレームを作成します。

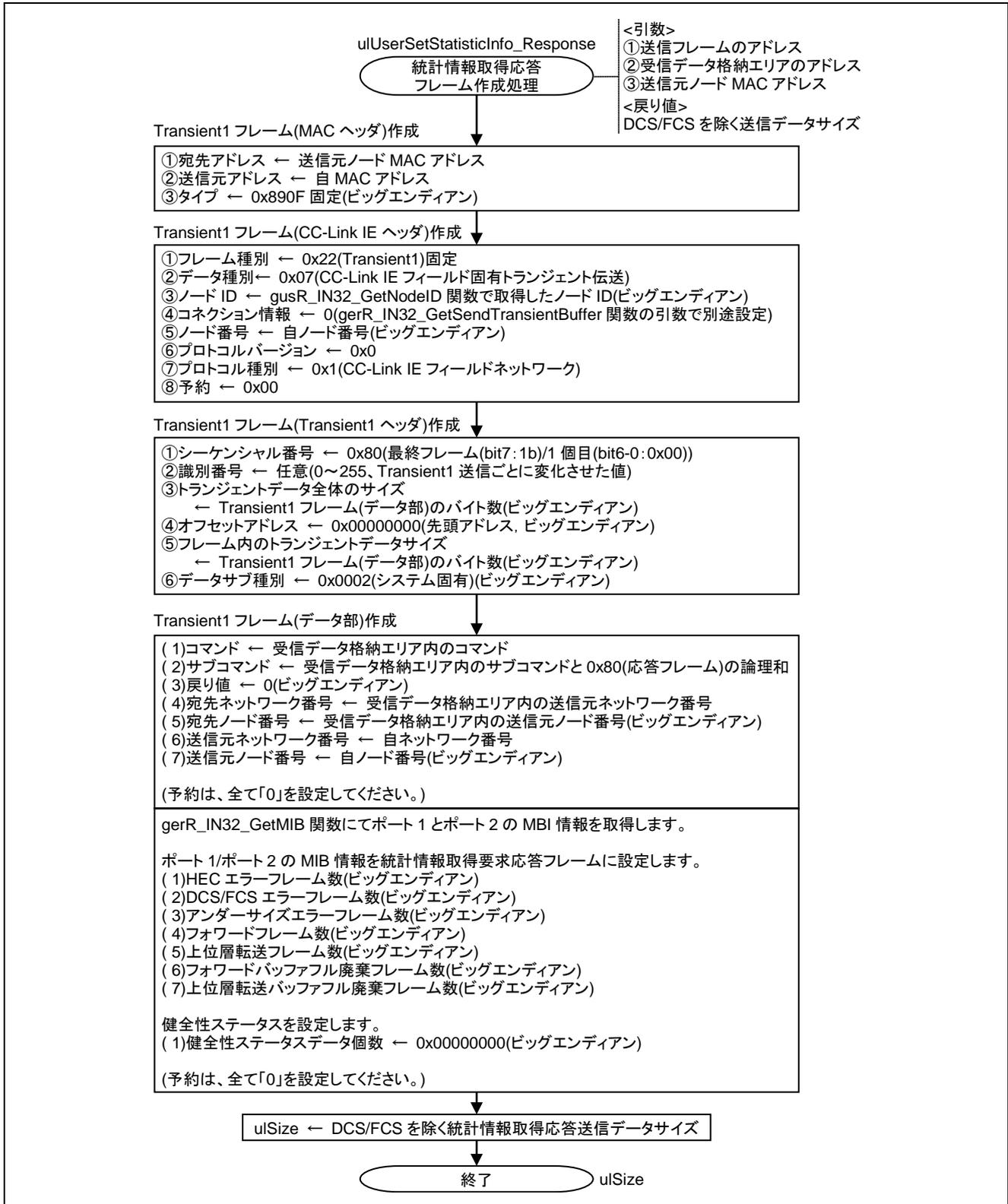


図6.28 統計情報取得応答フレーム作成処理 フロー図

## 6.2.25 ノード詳細情報取得要求フレーム受信対応処理

ノード詳細情報取得要求フレーム受信時の処理を行います。

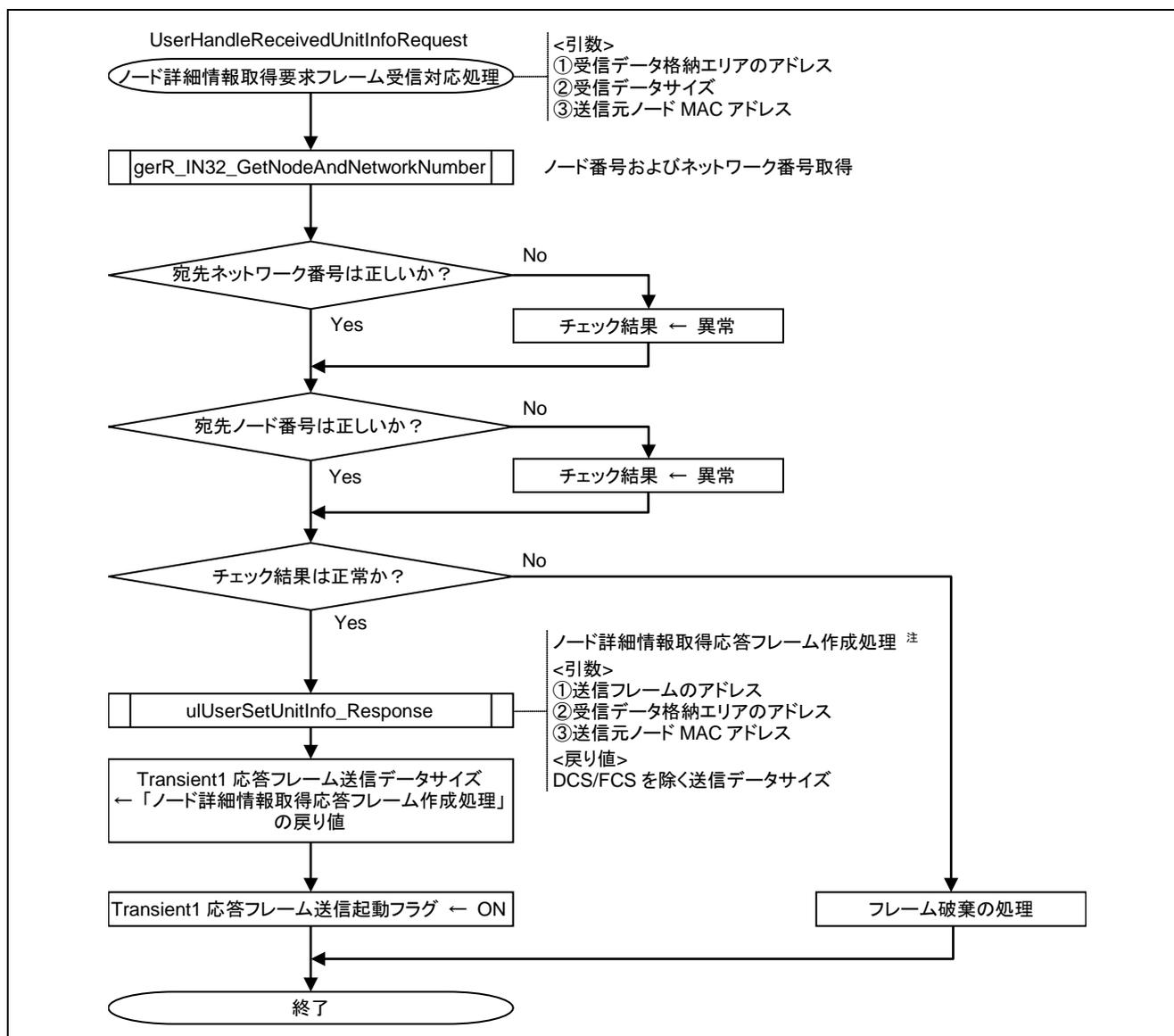


図6.29 ノード詳細情報取得要求フレーム受信対応処理 フロー図

注. 詳細は、「6.2.26 ノード詳細情報取得応答フレーム作成」を参照してください。

6.2.26 ノード詳細情報取得応答フレーム作成処理

ノード詳細情報取得応答フレームを作成します。

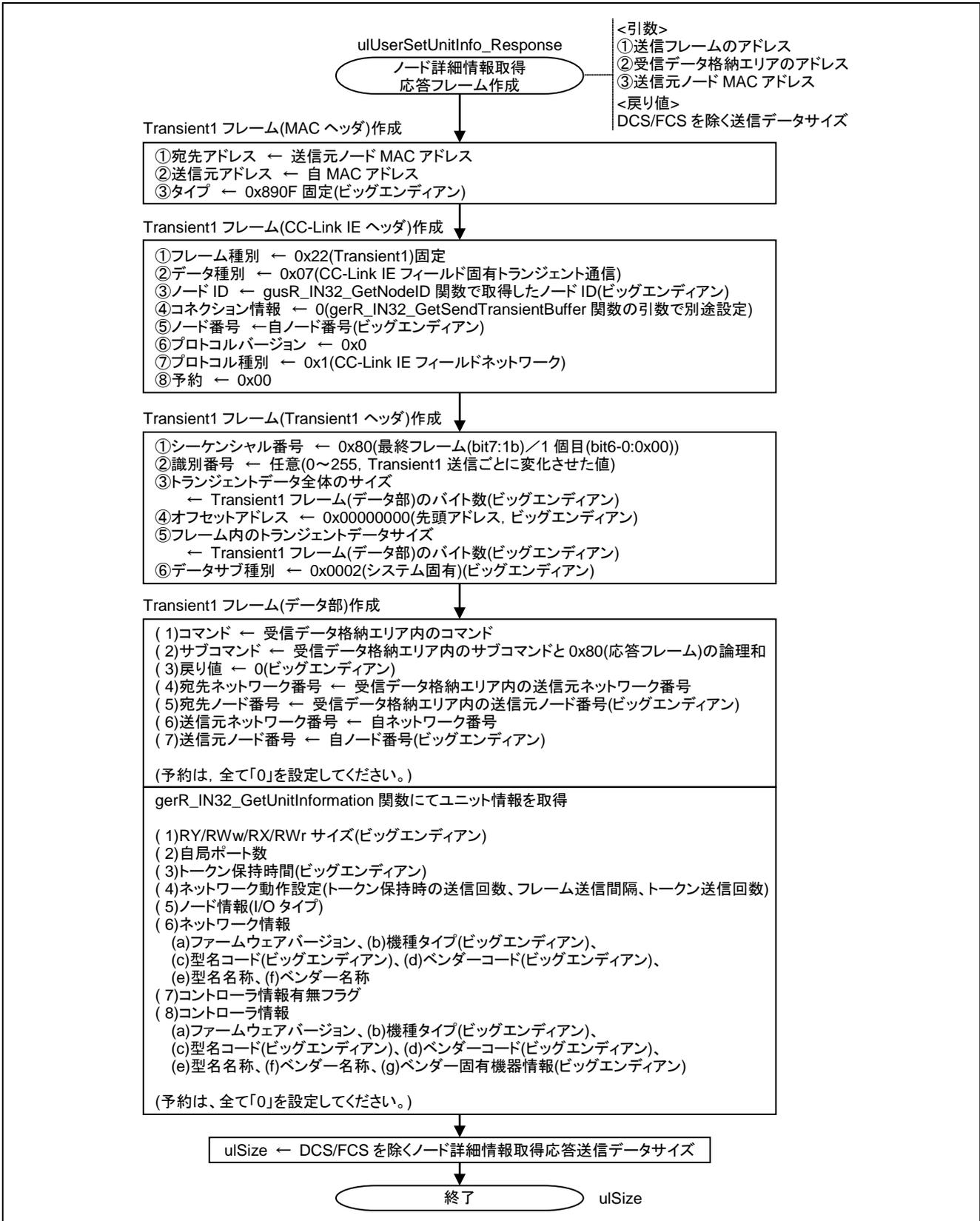


図6.30 ノード詳細情報取得応答フレーム作成処理 フロー図

### 6.2.27 オプション情報取得要求フレーム受信対応処理

オプション情報取得要求フレーム受信時の処理を行います。  
 本処理は自局が SLMP フレームに対応していることを、マスタ局に通知するための処理です。

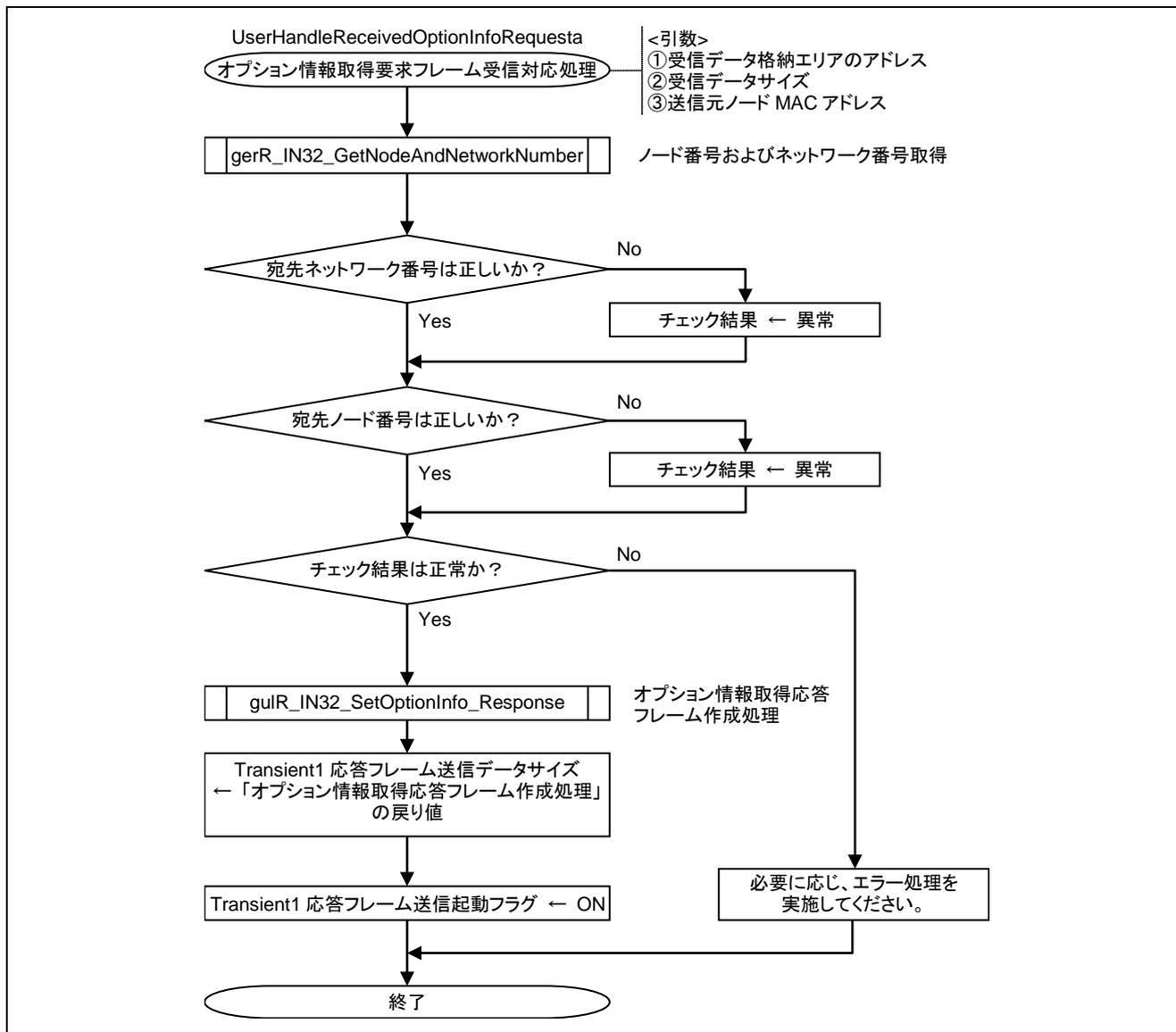


図6.31 オプション情報取得要求フレーム受信対応処理 フロー図

### 6.2.28 選択局情報取得要求フレーム受信対応処理

選択局情報取得要求フレーム受信時の処理を行います。  
 本処理は CC IE フィールド診断機能の「選択局通信状態モニタ」に対応するための処理です。

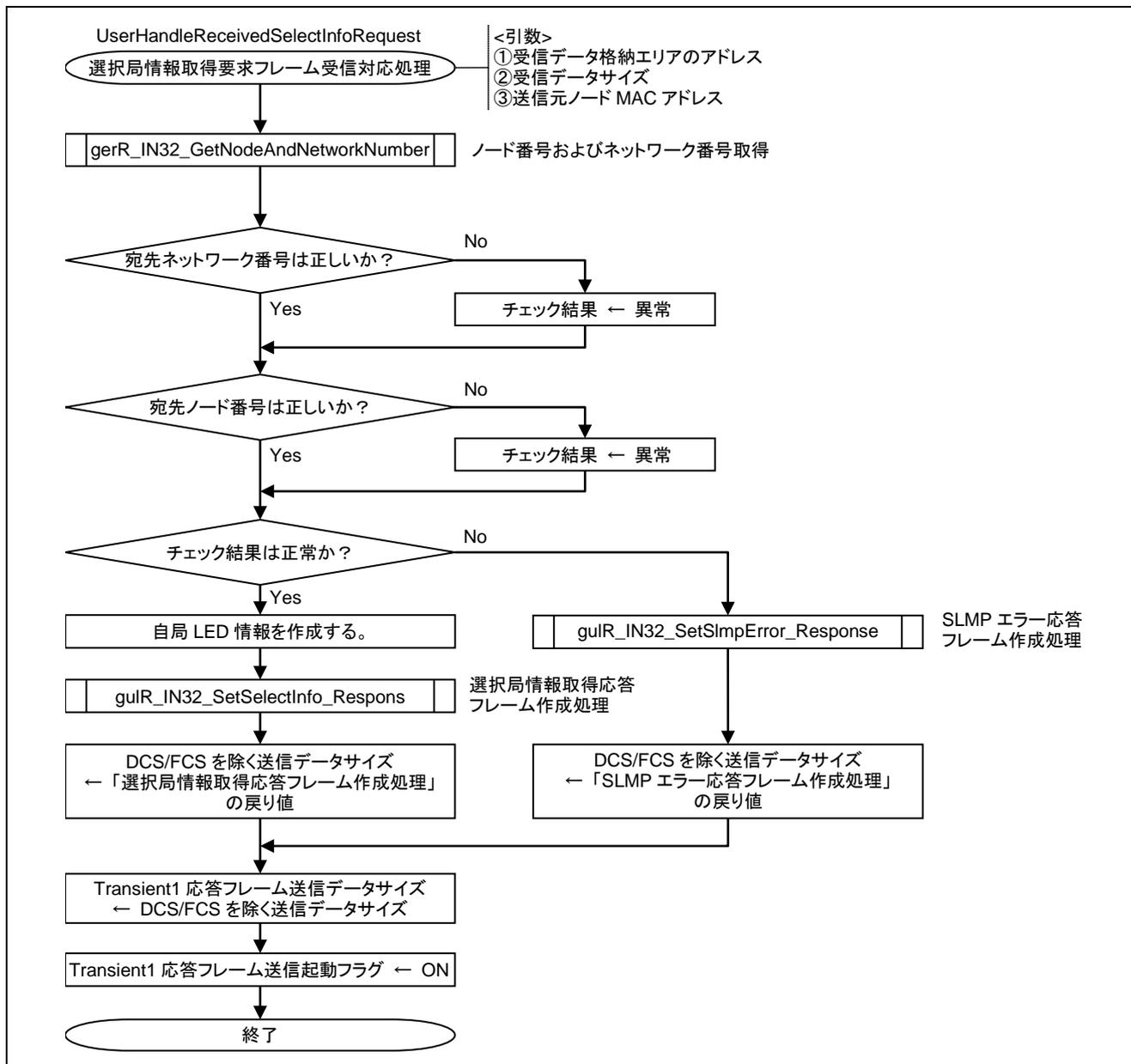


図6.32 選択局情報取得要求フレーム受信対応処理 フロー図

### 6.2.29 交信テスト要求フレーム受信対応処理

交信テスト要求フレーム受信時の処理を行います。

本処理は CC IE フィールド診断機能の「交信テスト」に対応するための処理です。

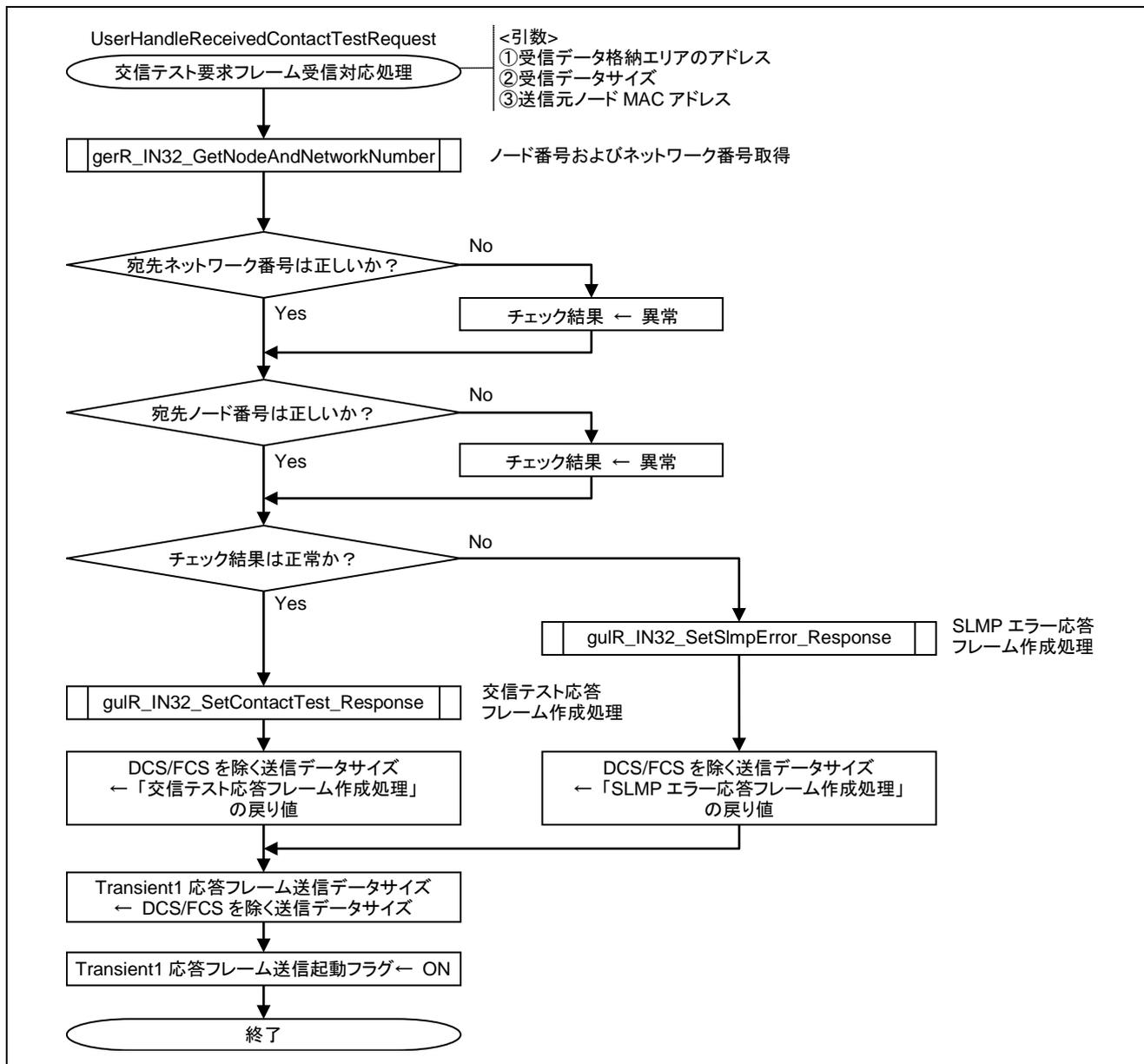


図6.33 交信テスト要求フレーム受信対応処理 フロー図

### 6.2.30 ケーブルテスト要求フレーム受信対応処理

ケーブルテスト要求フレーム受信時の処理を行います。

本処理は CC IE フィールド診断機能の「ケーブルテスト」に対応するための処理です。

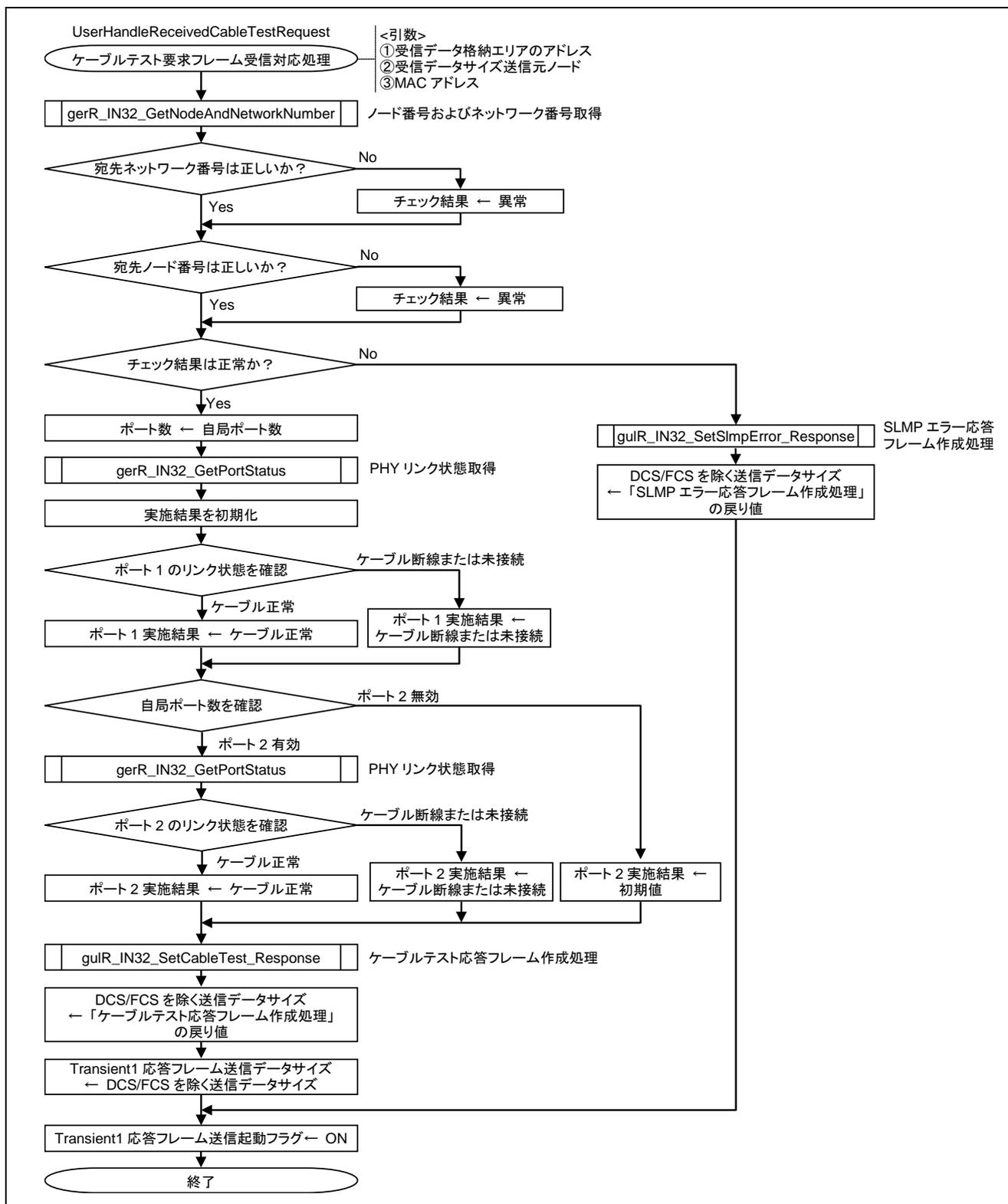


図6.34 ケーブルテスト要求フレーム受信対応処理 フロー図

### 6.2.31 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。

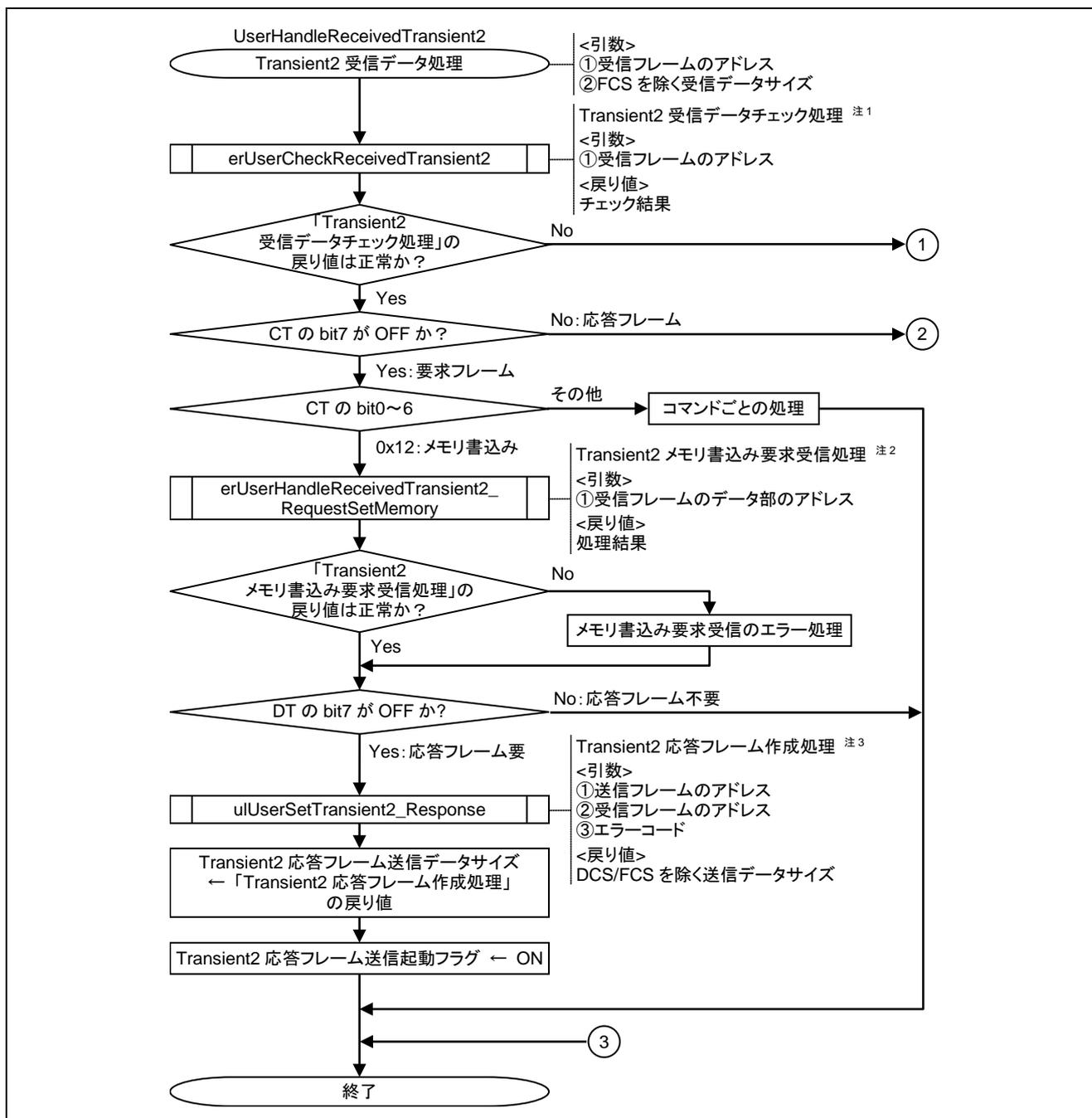


図6.35 Transient2 受信データ処理 フロー図 (1/2)

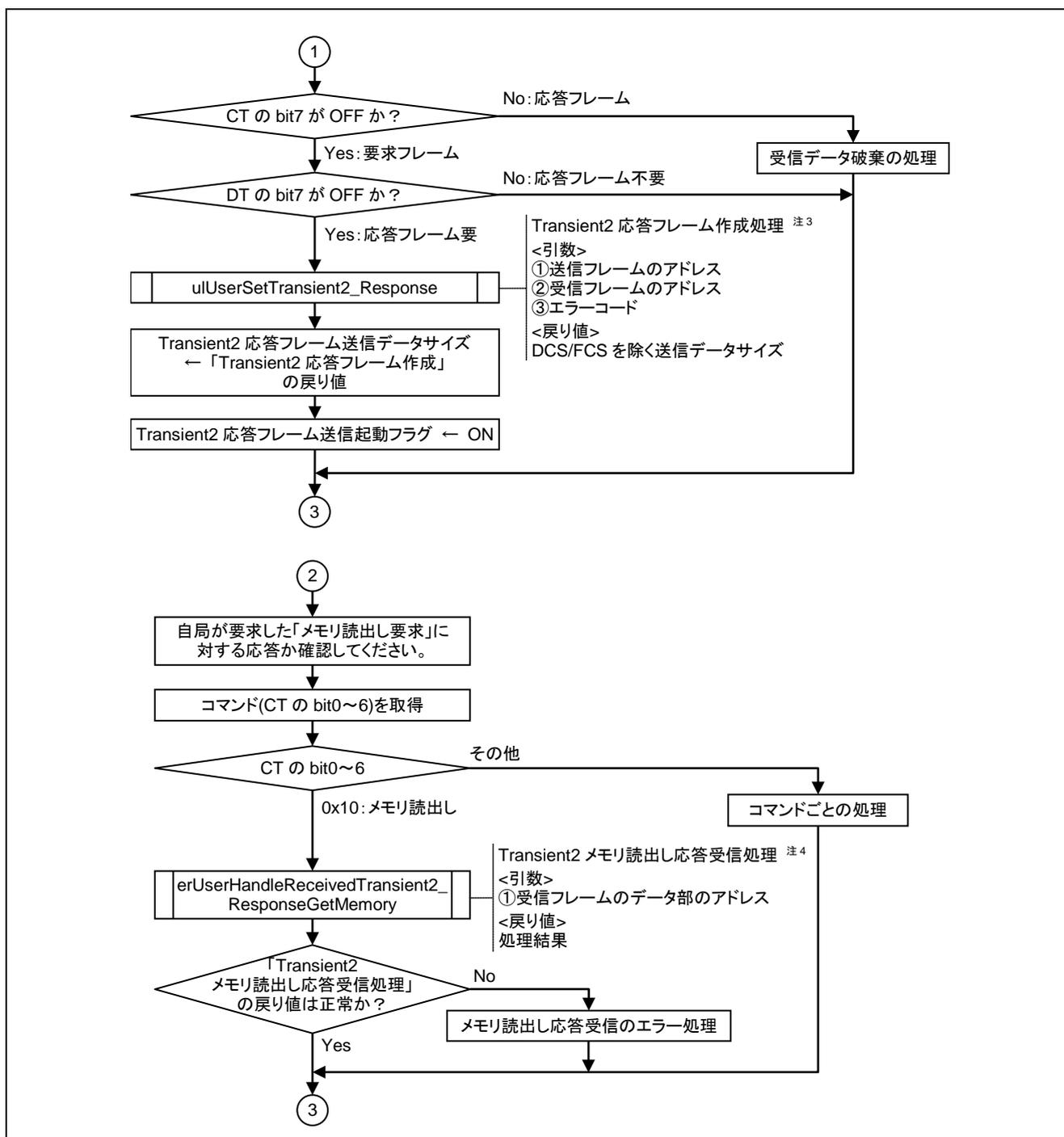


図6.35 Transient2 受信データ処理 フロー図 (2/2)

- 注 1. 詳細は、「6.2.32 Transient2受信データチェック」を参照してください。
- 注 2. 詳細は、「6.2.37 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、「6.2.35 Transient2応答フレーム作成」を参照してください。
- 注 4. 詳細は、「6.2.38 Transient2メモリ読出し応答受信処理」を参照してください。

### 6.2.32 Transient2 受信データチェック処理

受信した Transient2 フレームが自局宛か、宛先ノード番号 (DA (Distination Address) ・ DS (Distination Station no.)) および宛先ネットワーク番号 (DNA (Distination Network Address)) をチェックします。

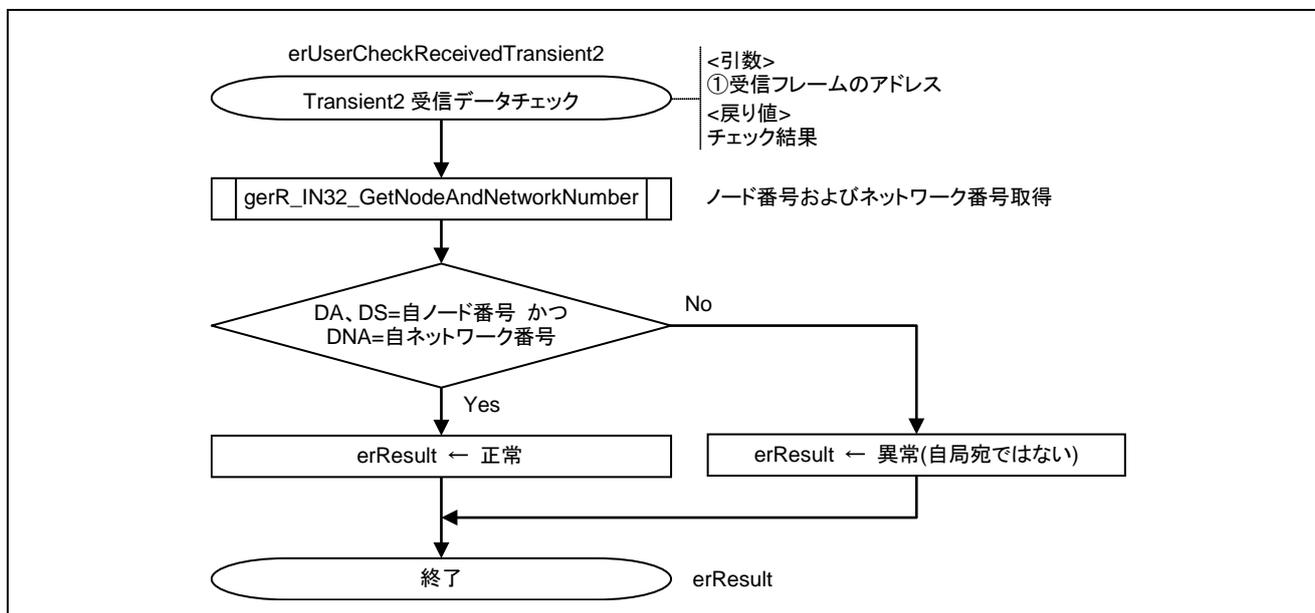


図6.36 Transient2 受信データチェック処理 フロー図

### 6.2.33 TransientAck 受信データ処理

受信した TransientAck フレームの処理を行います。

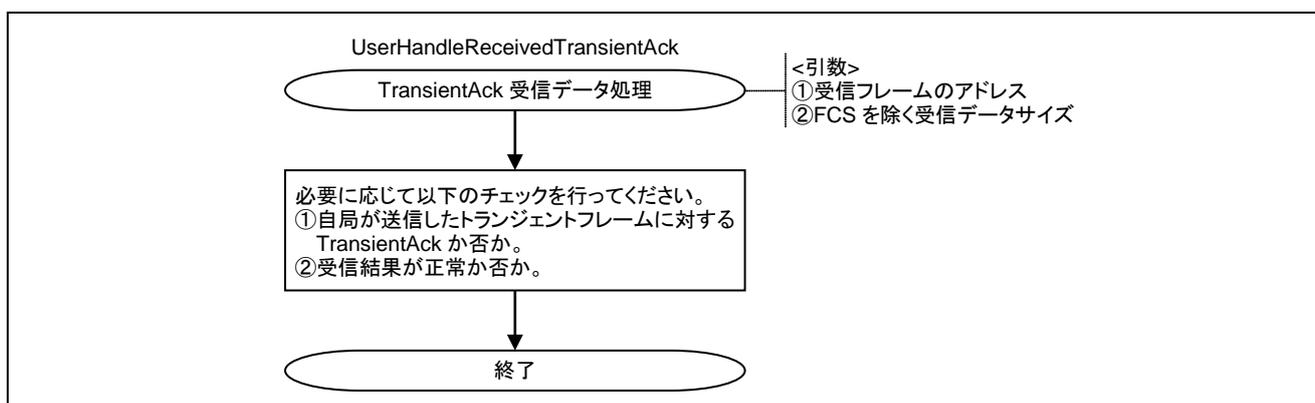


図6.37 TransientAck 受信データ処理 フロー図

6.2.34 TransientAck フレーム作成処理

TransientAck フレームを作成します。

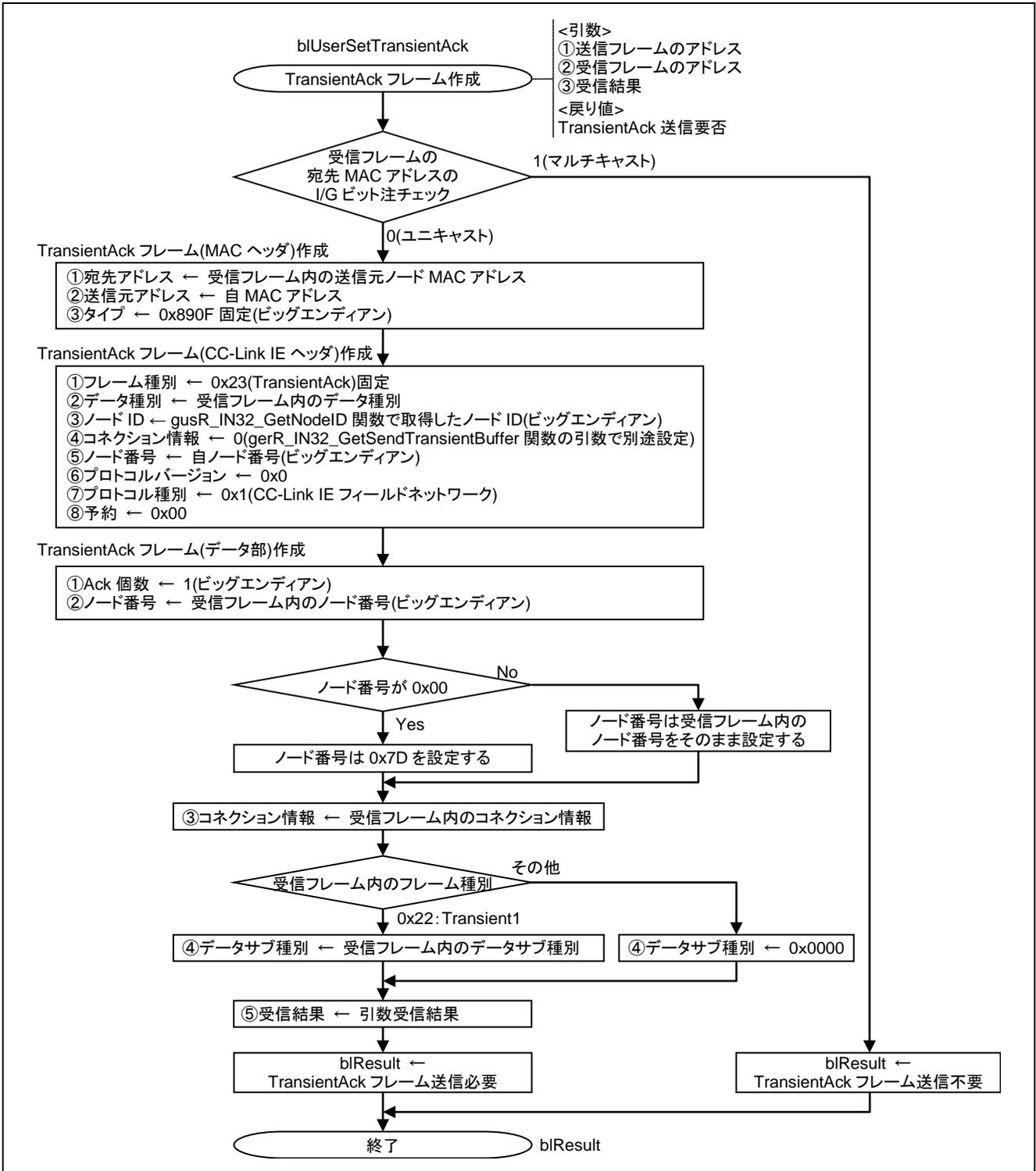


図6.38 TransientAck フレーム作成処理 フロー図

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。

## 6.2.35 Transient2 応答フレーム作成処理

Transient2 応答フレームを作成します。

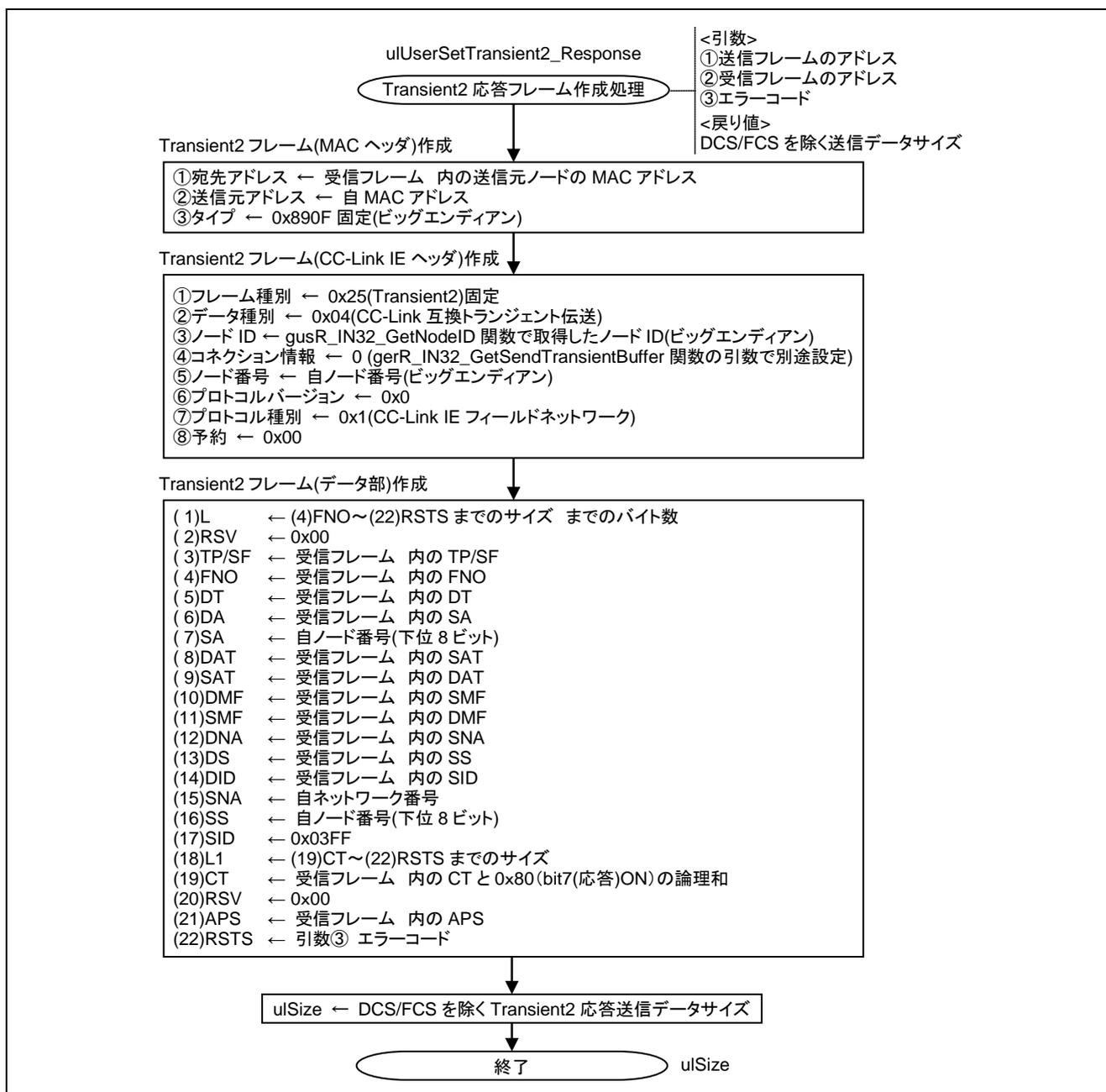


図6.39 Transient2 応答フレーム作成処理 フロー図

### 6.2.36 Transient2 メモリ読出し要求フレーム作成処理

Transient2 メモリ読出し要求フレームを作成します。

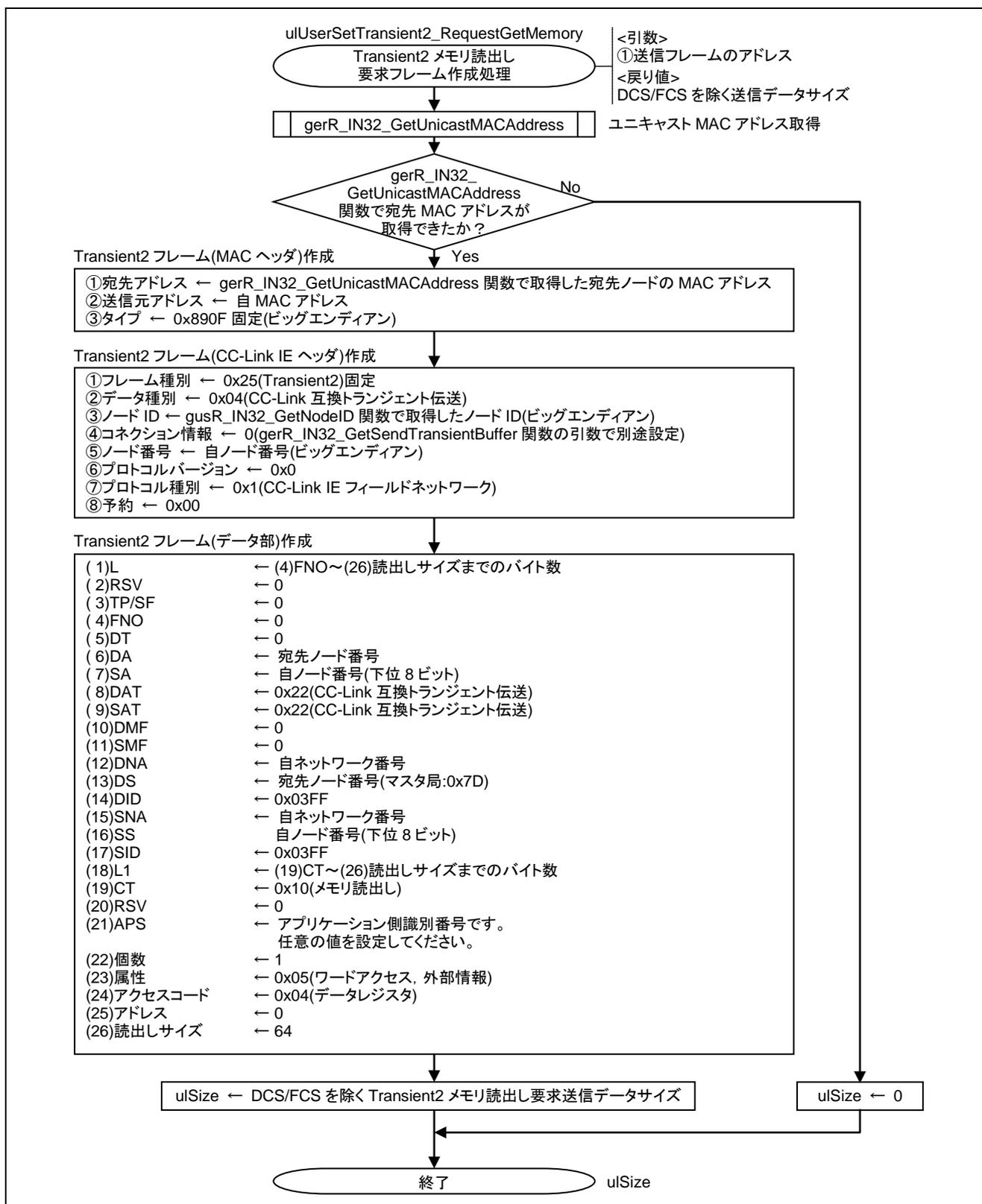


図6.40 Transient2 メモリ読出し要求フレーム作成処理 フロー図

本フローで記載するのは、メモリ読出し機能における以下の処理について記載しています。

- (1) 宛先ノード番号 0x7D (マスタ局)
- (2) アクセスコード 0x04 (データレジスタ)
- (3) 属性 0x05 (ワードアクセス、外部情報)
- (4) アドレス 0
- (5) 読出しサイズ 64

### 6.2.37 Transient2 メモリ書き込み要求受信処理

Transient2 メモリ書き込み要求フレーム受信時の処理を行います。

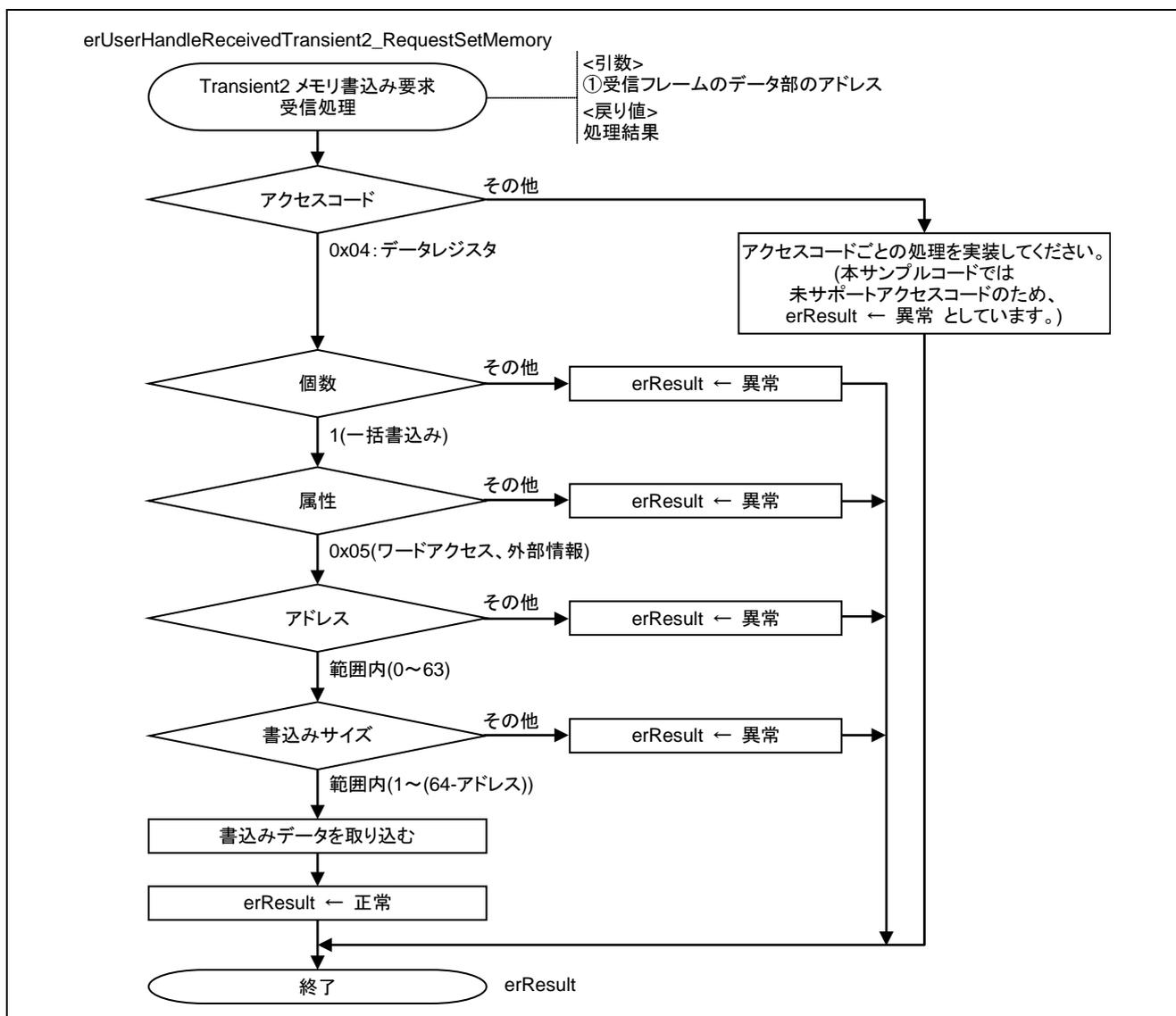


図6.41 Transient2 メモリ書き込み要求受信処理 フロー図

本フローで記載するのは、メモリ書き込み機能における以下の処理について記載しています。

下記以外の場合、異常とします。

- (1) メモリー一括書き込み
- (2) アクセスコード 0x04 (データレジスタ)
- (3) 属性 0x05 (ワードアクセス、外部情報)
- (4) アドレス 0~63
- (5) 書き込みサイズ 1~(64-アドレス)

## 6.2.38 Transient2 メモリ読出し応答受信処理

Transient2 メモリ読出し応答フレーム受信時の処理を行います。

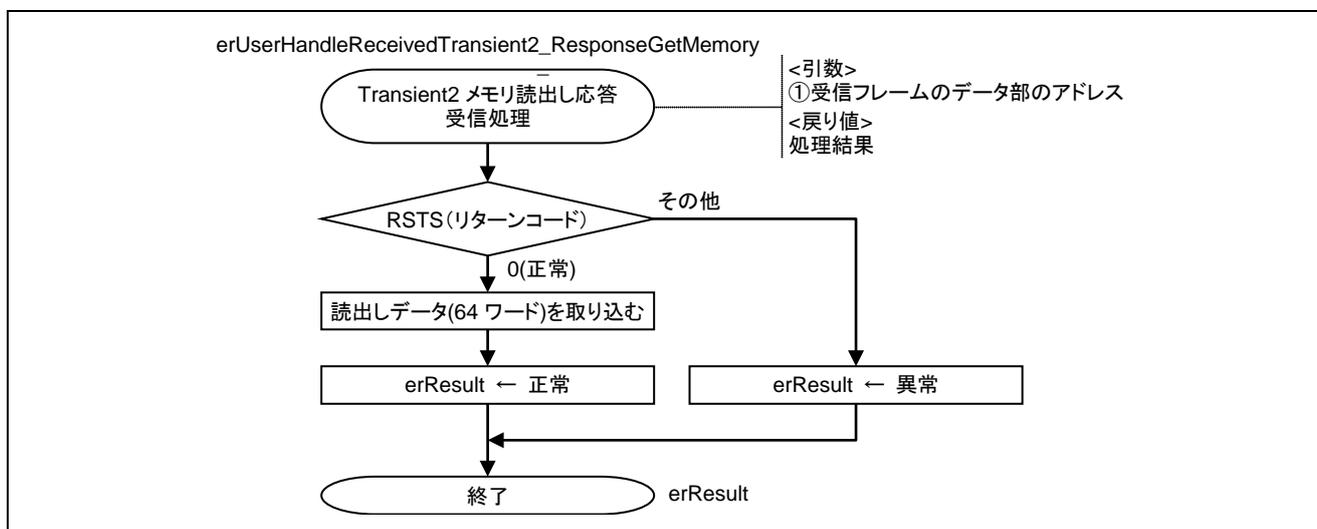


図6.42 Transient2 メモリ読出し応答受信処理 フロー図

本フローで記載するのは、メモリ読出し機能における以下の要求に対する受信処理について記載しています。

- (1) 宛先ノード番号 0x7D (マスタ局)
- (2) アクセスコード 0x04 (データレジスタ)
- (3) 属性 0x05 (ワードアクセス、外部情報)
- (4) アドレス 0
- (5) 読出しサイズ 64

### 6.2.39 SLMP メモリ読出し要求フレーム受信対応処理

SLMP メモリ読出し要求フレーム受信時の処理を行います。

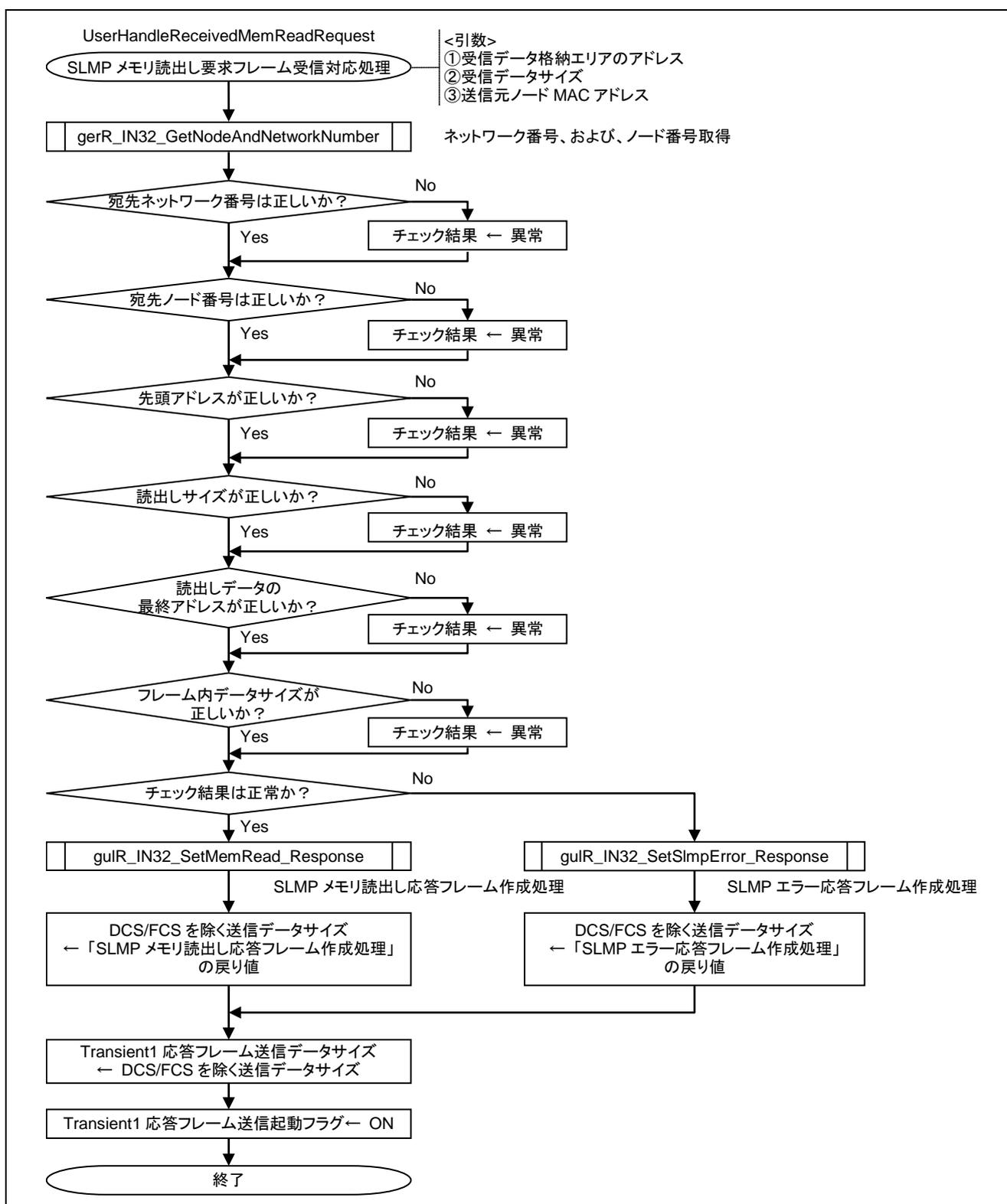


図6.43 SLMP メモリ読出し要求フレーム受信対応処理 フロー図

### 6.2.40 SLMP メモリ書き込み要求フレーム受信対応処理

SLMP メモリ書き込み要求フレーム受信時の処理を行います。

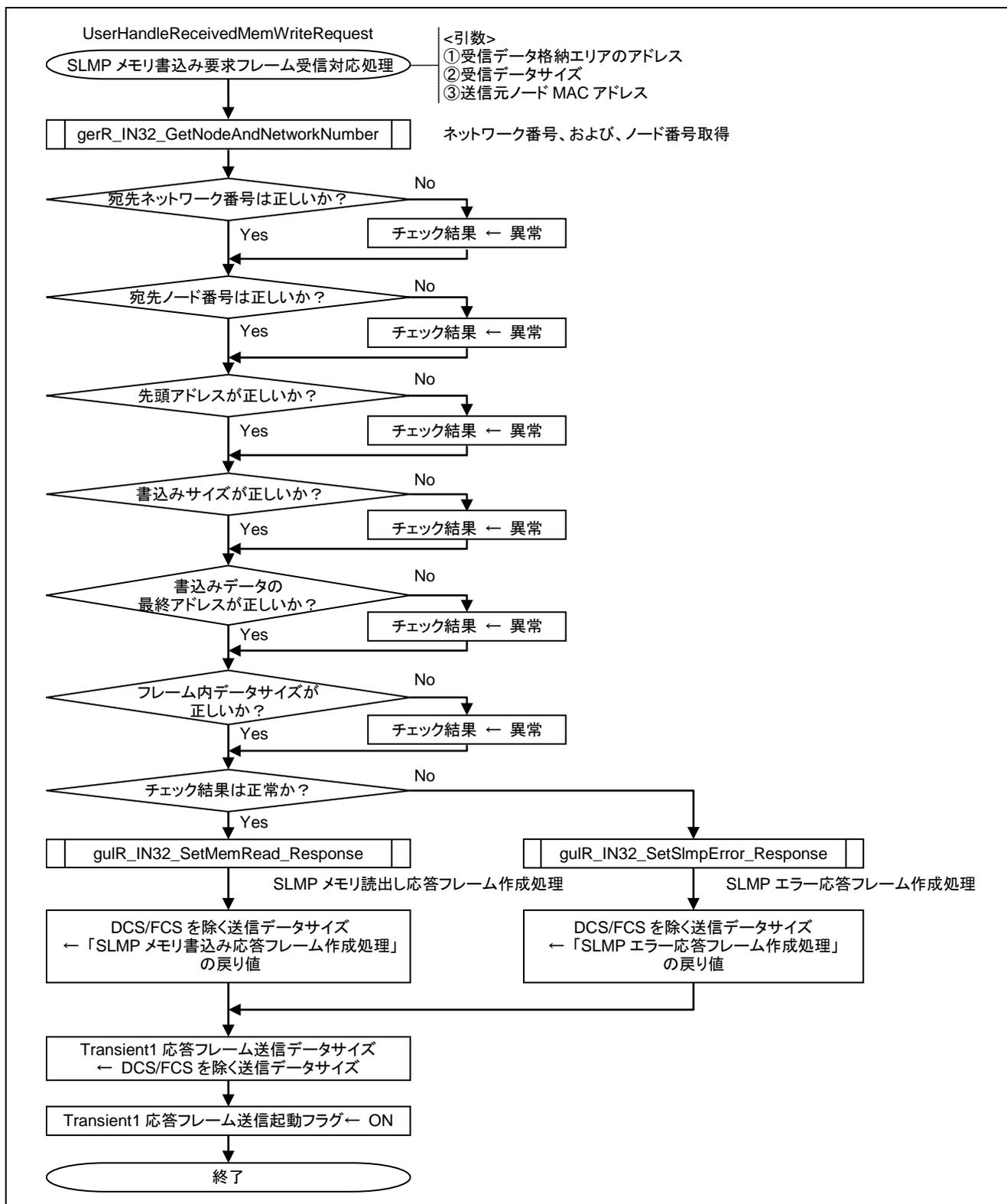


図6.44 SLMP メモリ書き込み要求フレーム受信対応処理 フロー図

6.2.41 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 要求フレームを送信するときの、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。

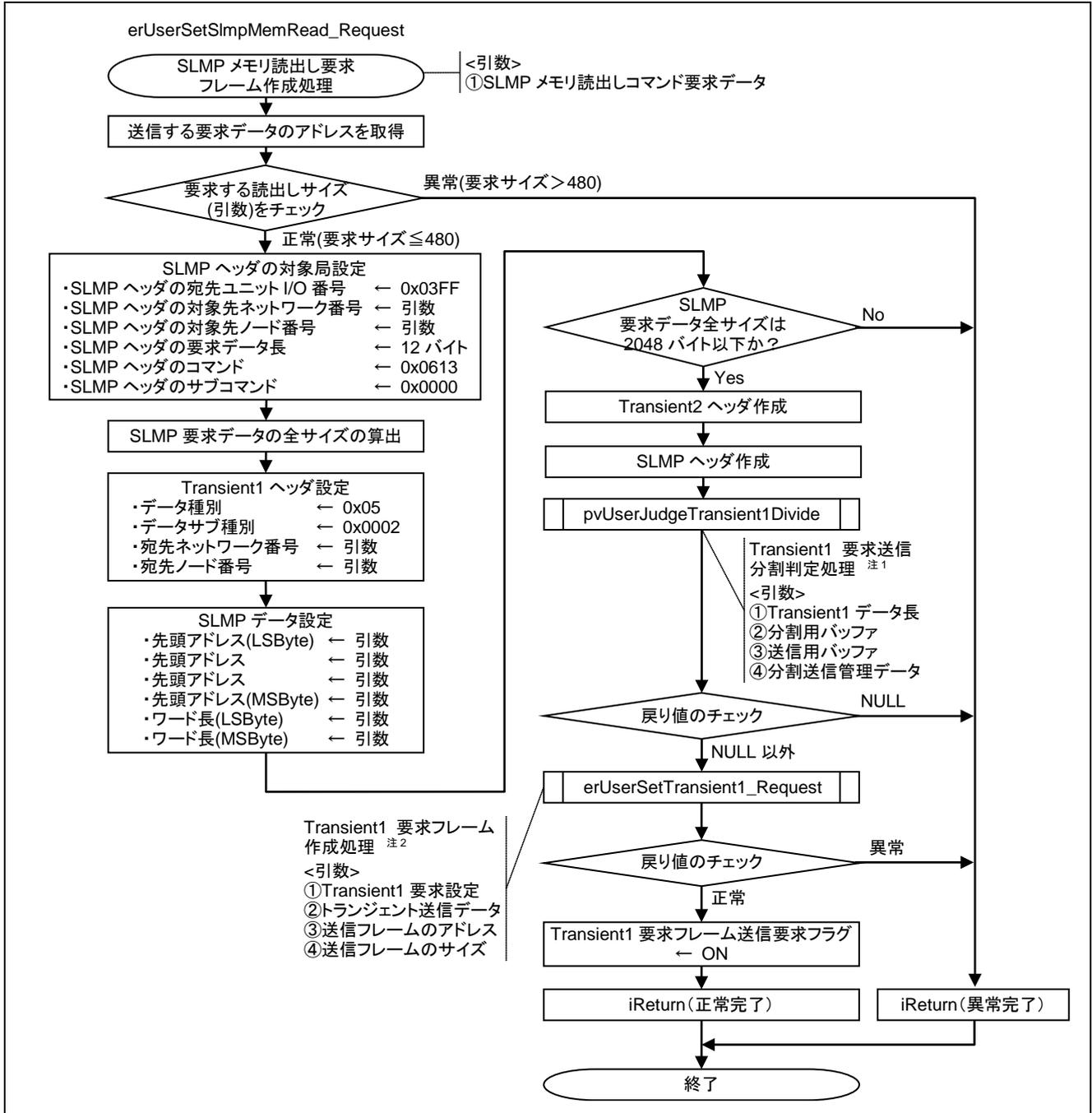


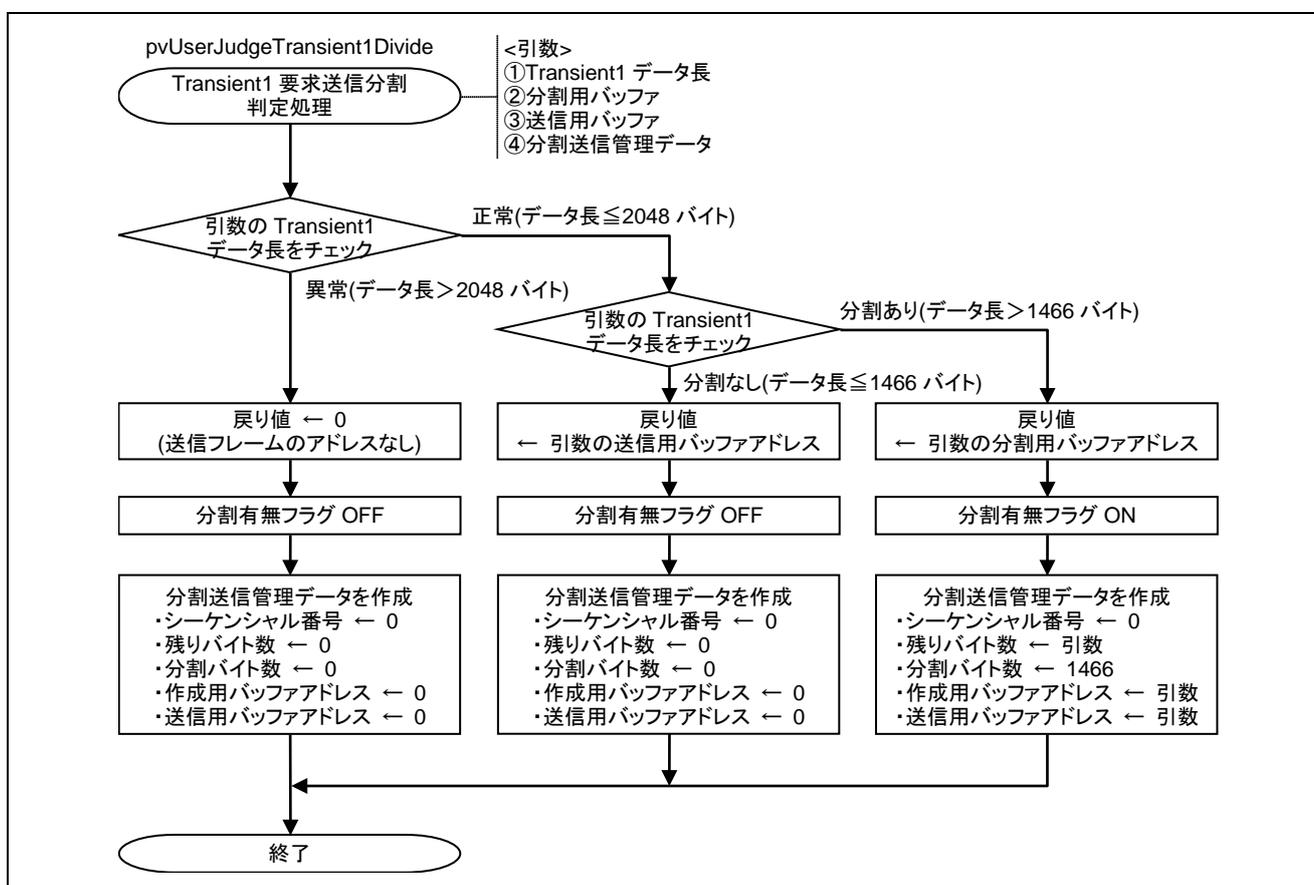
図6.45 SLMP メモリ読出し要求フレーム作成処理 フロー図

注 1. 詳細は、「6.2.42 Transient1要求送信分割判定処理」を参照してください。

注 2. 詳細は、「6.2.43 Transient1要求フレーム作成処理」を参照してください。

## 6.2.42 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。



### 6.2.43 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。

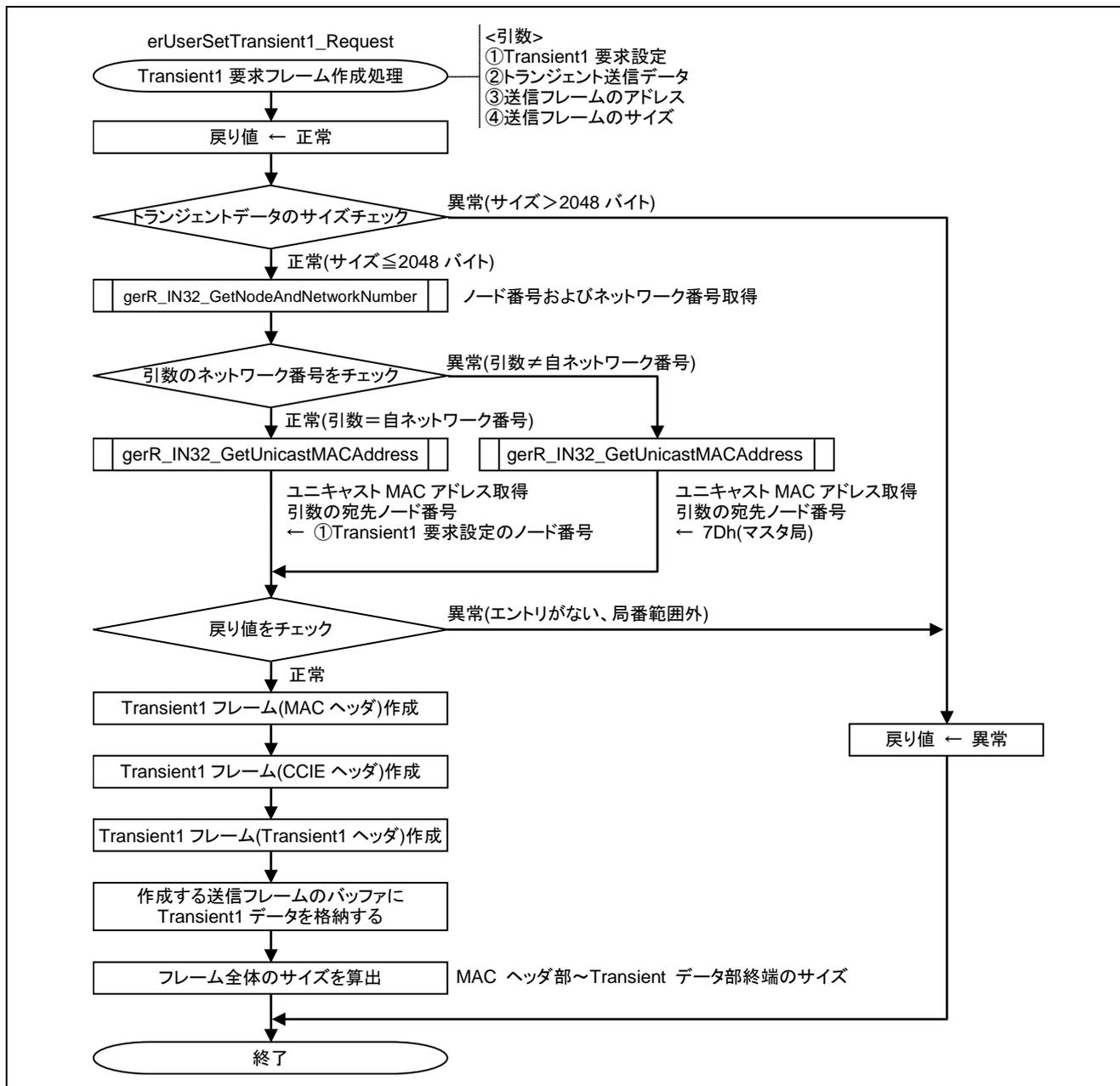


図6.47 Transient1 要求フレーム作成処理 フロー図

## 6.2.44 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。

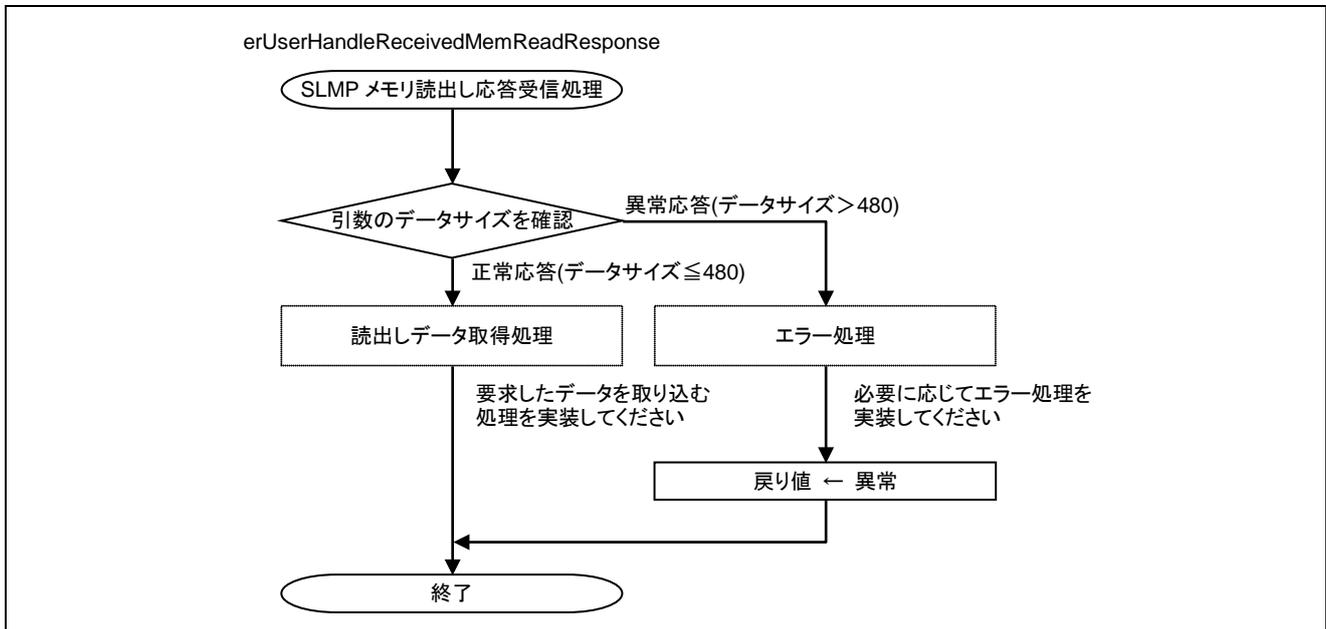


図6.48 SLMP メモリ読出し応答受信処理 フロー図

## 6.2.45 ハードウェアテスト (IEEE802.3ab コンプライアンステスト)

IEEE802.3ab コンプライアンステストを行います。

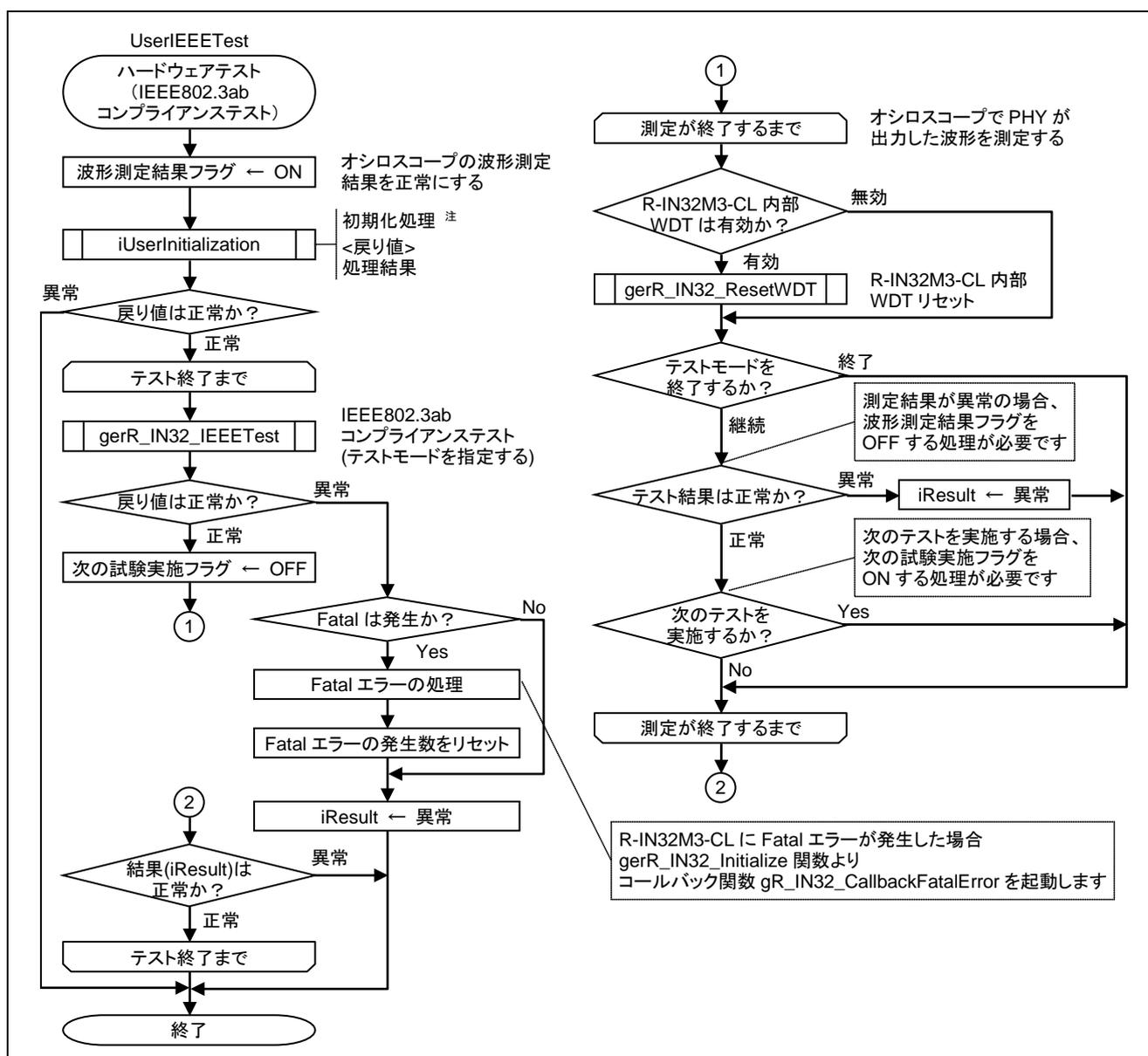


図6.49 ハードウェアテスト (IEEE802.3ab コンプライアンステスト) フロー図

注. 詳細は、「6.2.2 初期化処理」を参照してください。

注意. 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-C041-037) の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。

gerR\_IN32\_IEEEtest の処理内で、gerR\_IN32R\_IEEEtest (「6.5.2 R-IN32M3-CL ドライバターゲット依存関数の作成」を参照) を呼び出しますので、ご使用の PHY の仕様に合わせて gerR\_IN32R\_IEEEtest をカスタマイズしてください。

## 6.2.46 ハードウェアテスト（折り返し通信テスト）

折り返し通信テストには、内部折り返し通信テストと外部折り返し通信テストがあります。それぞれのテスト結果によって、故障の疑いのあるポートをトラブルシュートできます。

表6.9 折り返し通信テストによるトラブルシュート

| 内部折り返し通信テストで<br>R_IN32_ERR となった対象ポート<br>(gerR_IN32_InternalLoopBackTest) | 外部折り返し通信テストで<br>R_IN32_ERR となった送信元ポート<br>(gerR_IN32_ExternalLoopBackTest) | 故障の疑いのあるポート |
|--|---|-------------|
| ポート 1  | ポート 1   | ポート 1 XMIT  |
|  | ポート 2   | ポート 1 RECV  |
| ポート 2  | ポート 1   | ポート 2 RECV  |
|  | ポート 2   | ポート 2 XMIT  |

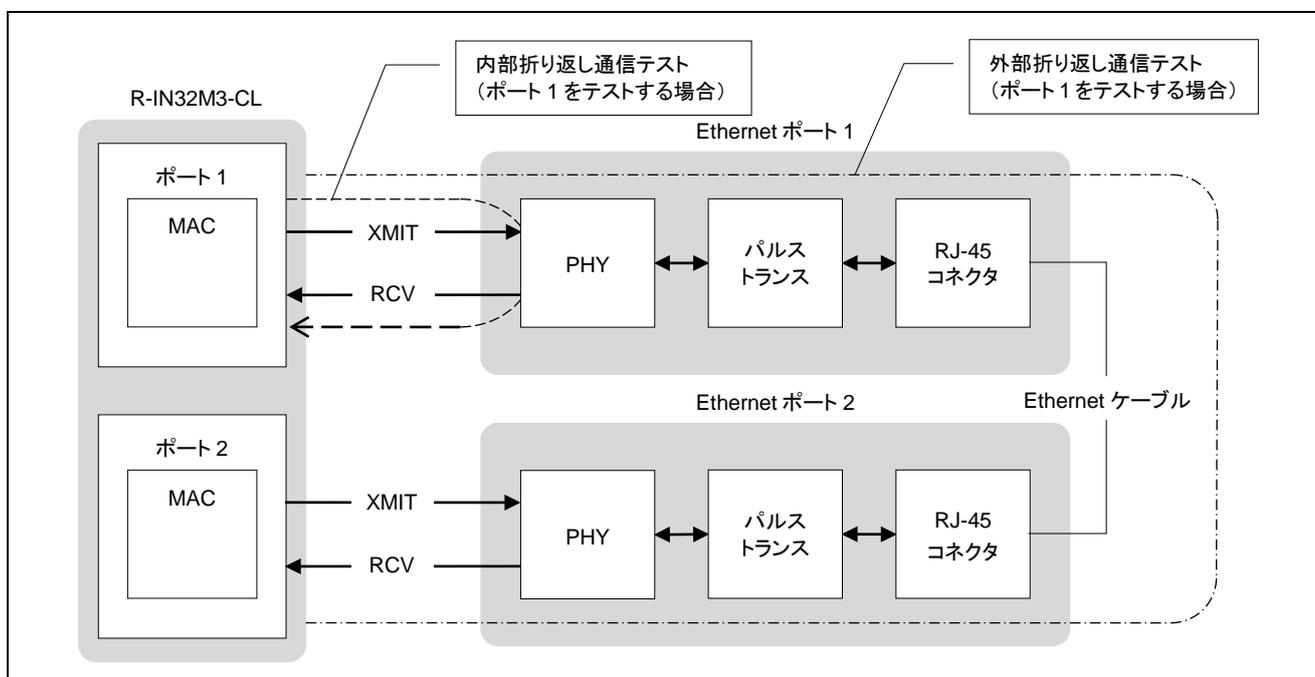


図6.50 ポート概略図

それぞれのテスト項目の注意事項にしたがってテストを実施してください。

表6.10 テスト項目の注意事項

| No. | テスト項目       | 注意事項   |
|-----|-------------|--|
| 1   | 内部折り返し通信テスト | <ul style="list-style-type: none"> <li>内部折り返し通信テストを実施すると、PHY がリンクダウンします。PHY がリンクアップするまで3秒以上かかりますので、WDT がタイムアウトしないようリセット処理をしてください。(R-IN32M3-CL 内部 WDT を使用時は、gerR_IN32_ResetWDT 関数を起動してください)</li> <li>内部折り返し通信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。<br/>(例：メイン処理を起動する通常稼働モードと、内部折り返し通信テストモードを分けて実装する。)</li> </ul> |
| 2   | 外部折り返し通信テスト | ポート 1 とポート 2 間を Ethernet ケーブルで接続してください。  |

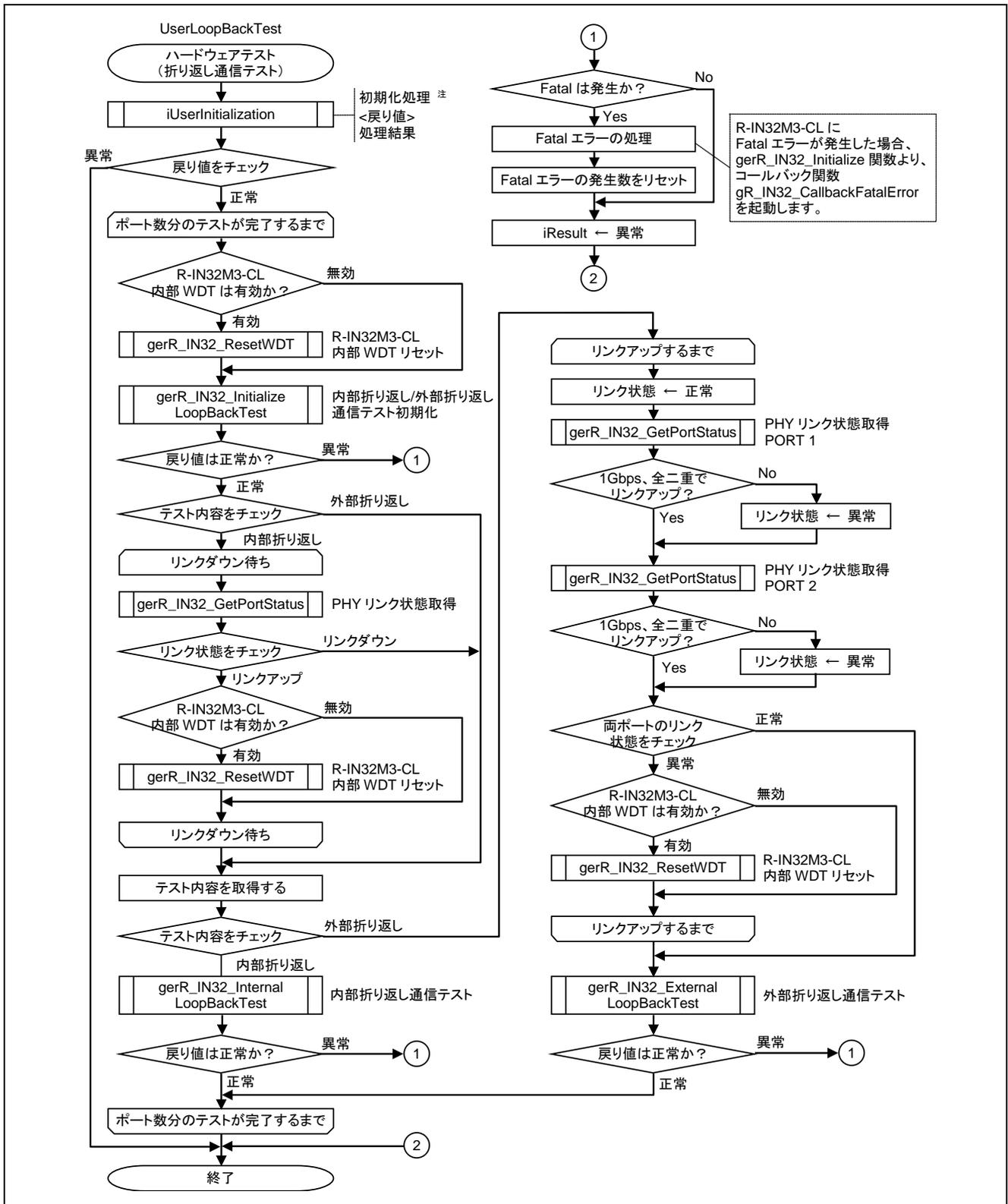


図6.51 ハードウェアテスト (折り返し通信テスト) フロー図

注. 詳細は、「6.2.2 初期化処理」を参照してください。

### 6.3 R-IN32M3-CL ドライバインタフェース関数一覧

R-IN32M3-CL ドライバのインタフェース関数一覧を以下に示します。

表6.11 R-IN32M3-CL ドライバインタフェース関数一覧表 (1/2)

| 機能分類       | 関数名                               | 関数型     | 概要                       |
|------------|-----------------------------------|---------|--------------------------|
| 初期設定       | gulR_IN32_GetResetStatus          | ULONG   | リセット状態取得                 |
|            | gerR_IN32_Initialize              | ERRCODE | R-IN32M3-CL の初期化         |
|            | gerR_IN32_SetNodeAndNetworkNumber | ERRCODE | ノード番号およびネットワーク番号設定       |
|            | gerR_IN32_Start                   | ERRCODE | R-IN32M3-CL 通信開始         |
| ウォッチドッグタイマ | gerR_IN32_ResetWDT                | ERRCODE | R-IN32M3-CL 内部 WDT リセット  |
|            | gerR_IN32_DisableWDT              | ERRCODE | R-IN32M3-CL 内部 WDT 無効    |
|            | gerR_IN32_EnableWDT               | ERRCODE | R-IN32M3-CL 内部 WDT 有効    |
|            | gerR_IN32_SetWDT                  | ERRCODE | R-IN32M3-CL 内部 WDT 時限設定  |
| イベント       | gerR_IN32_GetEvent                | ERRCODE | R-IN32M3-CL イベント検出       |
|            | gerR_IN32_Main                    | ERRCODE | R-IN32M3-CL イベント検出メイン処理  |
|            | gerR_IN32_RestartEvent            | ERRCODE | R-IN32M3-CL イベント再開       |
|            | gerR_IN32_UpdatePortStatus        | ERRCODE | PHY リンク状態更新              |
|            | gerR_IN32_UpdateMIB               | ERRCODE | MIB 情報更新                 |
| サイクリック伝送   | gerR_IN32_SetCyclicStop           | ERRCODE | 機器側都合によるサイクリック伝送停止       |
|            | gerR_IN32_ClearCyclicStop         | ERRCODE | 機器側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     | gerR_IN32_SetNodeStatus           | ERRCODE | 自局状態設定                   |
|            | gerR_IN32_ForceStop               | ERRCODE | 自局エラー設定                  |
| 自局状態取得     | gerR_IN32_GetNodeAndNetworkNumber | ERRCODE | ノード番号およびネットワーク番号取得       |
|            | 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                 |
|            | gerR_IN32_ClearMIB                | ERRCODE | MIB 情報クリア                |
|            | gerR_IN32_GetPortAvailable        | ERRCODE | ポート有効状態取得                |
| LED 制御     | gerR_IN32_SetLERR1LED             | ERRCODE | LED 点灯制御 (L ER (ポート 1))  |
|            | gerR_IN32_SetLERR2LED             | ERRCODE | LED 点灯制御 (L ER (ポート 2))  |
|            | gerR_IN32_SetERRLED               | ERRCODE | LED 点灯制御 (ERR.)          |
|            | gerR_IN32_SetDLINKLED             | ERRCODE | LED 点灯制御 (D LINK)        |
|            | 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               |

表6.11 R-IN32M3-CL ドライバインタフェース関数一覧表 (2/2)

| 機能分類      | 関数名                                 | 関数型     | 概要                         |
|-----------|-------------------------------------|---------|----------------------------|
| 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 | 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              |
| トランジェント受信 | 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 | ベンダー理由によるトランジェント受信許可設定     |
|           | gblR_IN32_GetReceiveTransientStatus | BOOL    | ベンダー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    |
| トランジェント送信 | gerR_IN32_GetUnitInformation        | ERRCODE | ユニット情報取得                   |
|           | gusR_IN32_GetNodeID                 | USHORT  | ノード ID 取得                  |
|           | gerR_IN32_GetMulticastMACAddresses  | 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                |
|           | gerR_IN32_MainSendTransient         | ERRCODE | トランジェント送信メイン処理             |
|           | gulR_IN32_SetOptionInfo_Response    | ULONG   | オプション情報取得応答フレーム作成処理        |
|           | gulR_IN32_SetSelectInfo_Response    | ULONG   | 選択局情報取得応答フレーム作成処理          |
|           | gulR_IN32_SetSlmpError_Response     | ULONG   | SLMP エラー応答フレーム作成処理         |
|           | gulR_IN32_SetContactTest_Response   | ULONG   | 交信テスト応答フレーム作成処理            |
|           | gulR_IN32_SetCableTest_Response     | ULONG   | ケーブルテスト応答フレーム作成処理          |
|           | gulR_IN32_SetMemRead_Response       | ULONG   | SLMP メモリ読み出し応答フレーム作成処理     |
|           | gulR_IN32_SetMemWrite_Response      | ULONG   | SLMP メモリ書き込み応答フレーム作成処理     |
| 割込み       | gerR_IN32_DisableInterrupt          | ERRCODE | 割込み禁止                      |
|           | gerR_IN32_EnableInterrupt           | ERRCODE | 割込み許可                      |
| ハードウェアテスト | gerR_IN32_IEEETest                  | 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_InternalLoopBackTest      | ERRCODE | 内部折り返し通信テスト                |
|           | gerR_IN32_ExternalLoopBackTest      | ERRCODE | 外部折り返し通信テスト                |

## 6.4 R-IN32M3-CL ドライバ インタフェース関数詳細

R-IN32M3-CL ドライバ インタフェース関数は、C 言語で記述したユーザプログラムから起動します。本節では、R-IN32M3-CL 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記述します。

本節では、サンプルコードを元に下記の定義を使用しています。

### (1) パラメータのデータ型とサイズ

パラメータのデータ型とタイプは以下のものを使用します。

```
#define VOID                void;
typedef char                CHAR;
typedef unsigned char      UCHAR;
typedef short              SHORT;
typedef unsigned short     USHORT;
typedef int                 INT;
typedef unsigned int       UINT;
typedef long               LONG;
typedef unsigned long      ULONG;
typedef int                 ERRCODE;
typedef int                 BOOL;
```

### (2) エラーコード定義

戻り値として返すエラーコードは、以下のものを使用します。

```
#define R_IN32_OK           0          /*!< 正常 */
#define R_IN32_ERR         (-1)       /*!< 異常終了 */
#define R_IN32_ERR_OTHER   (-2)       /*!< 異常終了(ライブラリ内ドライバで異常発生) */
#define R_IN32_ERR_OUTOFRANGE (-3)    /*!< 範囲外 */
#define R_IN32_ERR_EMPTY   (-4)       /*!< エンプティ */
#define R_IN32_ERR_OVERFLOW (-5)      /*!< オーバーフロー */
#define R_IN32_ERR_NOENTRY (-6)       /*!< エントリがない */
#define R_IN32_ERR_NOPERMIT (-7)      /*!< 許可されていない */
#define R_IN32_ERR_NODATA  (-8)       /*!< データがない */
#define R_IN32_ERR_NOMYSTATUS (-9)    /*!< 有効な MyStatus がない */
```

### (3) その他定義

```
#define R_IN32_TRUE        1
#define R_IN32_FALSE       0
```

## 6.4.1 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 |
|        | なし   |     |    |     |
| 戻り値    | R_IN32_RESET_PWRON(1) : パワーオンリセット<br>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。<br>本関数の起動は、gerR_IN32_Initialize 関数を起動する前に実施してください。       |     |    |     |

## (2) gerR\_IN32\_Initialize

|        |   |              |  |     |
|--------|---|--------------|--|-----|
| 機能     | R-IN32M3-CL 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<br>12-34-56-78-90-AB の場合、以下のように設定してください。<br>puchMACAddr[0] : 0x12<br>puchMACAddr[1] : 0x34<br>puchMACAddr[2] : 0x56<br>puchMACAddr[3] : 0x78<br>puchMACAddr[4] : 0x90<br>puchMACAddr[5] : 0xAB | 入力  |
|        | const R_IN32_UNITINFO_T   | *pstUnitInfo | R-IN32M3-CL ユニット情報初期設定<br>詳細は A) R_IN32_UNITINFO_T 初期設定を参照してください。  | 入力  |
|        | const R_IN32_UNITINIT_T   | *pstUnitInit | R-IN32M3-CL 初期設定<br>詳細は B) R_IN32_UNITINIT_T 初期設定を参照してください。  | 入力  |
| 戻り値    | R_IN32_OK : 正常終了  |              |  |     |
| 説明     | R-IN32M3-CL の初期化および PHY のリセットを行います。<br>本関数を起動すると R-IN32M3-CL 内部 WDT が無効になりますので、R-IN32M3-CL 内部 WDT を使用する場合は gerR_IN32_EnableWDT 関数を起動してください。詳細は、「6.4.2 ウォッチドグタイマ」を参照してください。<br><br>※R-IN32M3-CL に Fatal エラーが発生した場合、本関数より、ベンダー作成の以下の関数を起動しますので、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。<br>gR_IN32_CallbackFatalError |              |  |     |

gerR\_IN32\_Initialize の引数

R\_IN32\_UNITINFO\_T の構成を、サンプルコードを元に下記に示します。

```

/* R-IN32M3-CL ユニット情報 */
typedef struct R_IN32_UNITINFO_TAG          {
    /* サイクリック伝送サイズ最大値 */
    ULONG    ulMaxRySize;                    /*!< RY サイズ(バイト (オクテット)) */
    ULONG    ulMaxRWwSize;                  /*!< RWw サイズ(ワード) */
    ULONG    ulMaxRxSize;                   /*!< RX サイズ(バイト (オクテット)) */
    ULONG    ulMaxRWrSize;                  /*!< RWr サイズ(ワード) */

    /* 局情報 1 */
    ULONG    ulMyStationPortTotalNumber;    /*!< 自局ポート数 */
    ULONG    ulTokenHoldTime;              /*!< トークン保持時間 */

    /* 局情報 2 */
    ULONG    ulIOType;                      /*!< ノード情報(I/O タイプ) */

    /* ネットワーク情報 */
    ULONG    ulNetVersion;                  /*!< ネットワークのファームウェアバージョン */
    ULONG    ulNetModelType;               /*!< ネットワークの機種タイプ */
    ULONG    ulNetUnitModelCode;           /*!< ネットワークの型名コード */
    ULONG    ulNetVendorCode;              /*!< ネットワークのベンダーコード */
    UCHAR    auchNetUnitModelName[20];     /*!< ネットワークの型名名称 */
    UCHAR    auchNetVendorName[32];        /*!< ネットワークのベンダー名称 */
    USHORT   usHwVersion;                   /*!< ネットワークのハードウェアバージョン */
    USHORT   usDeviceVersion;              /*!< ネットワークの機器バージョン */

    /* コントローラ情報 */
    BOOL     blInfomationFlag;              /*!< コントローラ情報有無フラグ */
    ULONG    ulCtrlVersion;                 /*!< コントローラのファームウェアバージョン */
    ULONG    ulCtrlModelType;               /*!< コントローラの機種タイプ */
    ULONG    ulCtrlUnitModelCode;           /*!< コントローラの型名コード */
    ULONG    ulCtrlVendorCode;              /*!< コントローラのベンダーコード */
    UCHAR    auchCtrlUnitModelName[20];     /*!< コントローラの型名名称 */
    UCHAR    auchCtrlVendorName[32];        /*!< コントローラのベンダー名称 */
    ULONG    ulVendorInformation;           /*!< コントローラのベンダー機器固有情報 */
} R_IN32_UNITINFO_T;

```

#### A) R\_IN32\_UNITINFO\_T 初期設定

R\_IN32\_UNITINFO\_T にて初期設定する内容は以下のとおりです。

##### (a) RY サイズ (バイト (オクテット))

自局が通信可能な RY のサイズ (バイト) を 1 バイト単位 (1 の倍数) 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。

(b) RW<sub>w</sub> サイズ (ワード)

自局が通信可能な RW<sub>w</sub> のサイズ (ワード) を 2 ワード単位 (2 の倍数) 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。

## (c) RX サイズ (バイト (オクテット))

自局が通信可能な RX のサイズ (バイト) を 1 バイト単位 (1 の倍数) 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。

(d) RW<sub>r</sub> サイズ (ワード)

自局が通信可能な RW<sub>r</sub> のサイズ (ワード) を 2 ワード単位 (2 の倍数) 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。

## (e) 自局ポート数

自局が保有する物理的な通信ポート数を設定します。  
R-IN32M3-CL で開発するインテリジェントデバイス局の場合、2 または 1 を設定してください。

## (f) 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  $\mu\text{s}$  単位で設定します。  
R-IN32M3-CL で開発するインテリジェントデバイス局の場合、23 ( $\mu\text{sec}$ ) を設定してください。

## (g) ノード情報 (I/O タイプ)

I/O タイプを設定します。  
00b(0x0)は裏表混在、01b(0x1)は入力、10b(0x2)は出力、11b(0x3)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、かつ入力と出力が同じアドレスを使用していない場合です。

## (h) 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。  
ファームウェアバージョンは、ベンダーが任意に定義してください。

## (i) 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を設定します。

## (j) 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、ベンダーが任意に定義したコードである。同一ベンダーコード内で一意になるように管理してください。

## (k) ネットワークのベンダーコード

CC-Link 協会入会時に取得したベンダーコードを BCD で設定します。  
(ベンダーコードが 5678 の場合、0x5678 を設定します。)

## (l) 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、ベンダーが任意に定義した名称である。同一ベンダーコード内で一意になるように管理してください。

- (m) ネットワークのベンダー名称  
ネットワークのベンダー名称を設定します。(32バイトの文字列(ASCIIコード))  
ベンダー名称は、ベンダーが任意に定義してください。
- (n) 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。  
ハードウェアバージョンは、ベンダーが任意に定義してください。
- (o) ネットワークの機器バージョン  
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。<sup>注</sup>
- (p) コントローラ情報有無フラグ  
コントローラの情報 ((q) コントローラのファームウェアバージョン~(w) コントローラのベンダー機器  
固有情報) の有効/無効を設定します。  
R\_IN32\_FLASE 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、コントローラ情報  
有無フラグを無効にします。
- (q) 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。  
ファームウェアバージョンは、ベンダーが任意に定義してください。
- (r) 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を設定します。
- (s) 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、ベンダーが任意に定義したコードである。同一ベンダーコード内で一意になるように管理  
してください。
- (t) コントローラのベンダーコード  
CC-Link 協会入会時に取得したベンダーコードを BCD で設定します。  
(ベンダーコードが 5678 の場合、0x5678 を設定します。)
- (u) 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、ベンダーが任意に定義した名称である。同一ベンダーコード内で一意になるように管理して  
ください。
- (v) コントローラのベンダー名称  
コントローラのベンダー名称を設定します。(32バイトの文字列(ASCIIコード))  
ベンダー名称は、ベンダーが任意に定義してください。

- (w) コントローラのベンダー機器固有情報  
 コントローラのベンダー機器固有情報を設定します。  
 ベンダー機器固有情報は、ベンダーが任意に定義してください。

**注. 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仕様書」の「DEVICE\_INFO パート」を参照してください。**

#### 【ネットワークとコントローラの補足】

##### ① ネットワークとコントローラの定義

- ネットワーク： 自局のうち、R-IN32M3-CL および周辺回路で構成した通信機能部分を指します。  
 コントローラ： 自局のうち、ベンダー独自の機能部分（I/O 部分、温調部分、ロボット部分など）を指します。

以下にイメージを例示します。

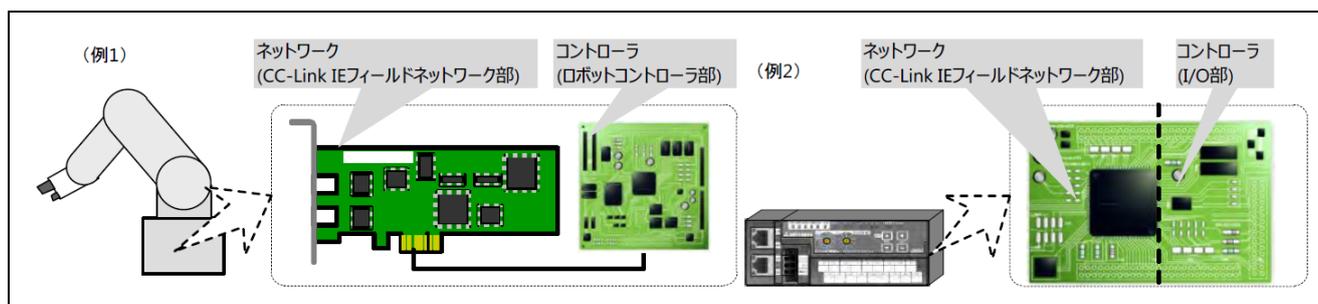


図6.52 ネットワークとコントローラの例

##### ② ネットワークの設定

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

- (h) ネットワークのファームウェアバージョン
- (i) ネットワークの機種タイプ
- (j) ネットワークの型名コード
- (k) ネットワークのベンダーコード

##### ③ コントローラの設定

コントローラの設定は任意です。

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

- ・ CSP+に記述されたベンダーコード/型名コードと、接続しているスレーブ局のコントローラの情報とを照合したうえで、スレーブ局のパラメータ処理・コマンド実行を行いたい場合
- ・ 開発機器（ネットワーク）はシリーズ製品（コントローラ）などの通信オプション品である場合
- ・ コントローラとネットワークのベンダーが異なる場合

## 【機器バージョンの補足】

## ① 背景

R-IN32M3-CL 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-IN32M3-CL 適用製品の仕様変更を行うと、CSP+ファイルも R-IN32M3-CL 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。

## ② 機器バージョンの目的

仕様変更前と後を識別する情報が機器バージョンであり、各 CSP+ファイルがどの R-IN32M3-CL 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-IN32M3-CL 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-IN32M3-CL 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、サンプルコードを元に下記に示します。

```

/* R-IN32M3-CL 初期設定 */
typedef struct R_IN32_UNITINIT_TAG {
    BOOL        blNMIUse;                /*!< NMI 割込み使用 */
    BOOL        blInterruptUse;         /*!< MPU 割込み機能使用 */
    BOOL        blFailedProcess1;      /*!< 故障処理設定 1 */
    BOOL        blFailedProcess2;      /*!< 故障処理設定 2 */
    ULONG       ulNodeType;             /*!< ノード種別 */
    BOOL        blTransientReceiveEnable; /*!< トランジェント受信機能 */
    BOOL        blMACAddressTableRequest; /*!< ノード情報配信の要求 */
    ULONG       ulRunStatus;            /*!< アプリケーション詳細動作状態 初期値 */
    ULONG       ulErrorStatus;          /*!< アプリケーション詳細エラー状態 初期値 */
    ULONG       ulErrorCode;            /*!< アプリケーションエラーコード 初期値 */
    ULONG       ulUserInformation;      /*!< ベンダー固有のノード情報 初期値 */
    ULONG       ulOptionSupport;        /*!< オプション有無 */
    ULONG       ulSlmpSupport;          /*!< SLMP サポートビット */
    ULONG       ulSlmpDiagnosisSupport; /*!< 診断機能対応有無 */
} R_IN32_UNITINIT_T;

```

## B) R\_IN32\_UNITINIT\_T 初期設定

R\_IN32\_UNITINIT\_T にて初期設定する内容は以下のとおりです。

## (a) NMI 割込み使用 (R-IN32M3-CL 内部 WDT 機能使用時のみ意味がある)

R-IN32M3-CL 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-IN32M3-CL の内部 WDT がオーバーフローしたときに NMIL 端子が “Low” になります。

## (b) MPU 割込み機能使用

R-IN32M3-CL の MPU 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-IN32M3-CL の割込みが発生したとき、INTL 端子が “Low” になります。

## (c) 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-IN32M3-CL はバイパスモードとなります。(通信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。)

① WDTIL 端子が真 (Low)

② R-IN32M3-CL 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。

## (d) 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 関数を起動) すると、R-IN32M3-CL はバイパスモードとなります。(通信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 関数については、「6.4.5 自局状態設定 (2) gerR\_IN32\_ForceStop」を参照してください。

## (e) ノード種別

自局のノード種別を設定します。インテリジェントデバイス局 (0x33) を設定してください。

## (f) 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\_FALSE” は機能なし、“R\_IN32\_TRUE” は機能ありを表します。

## (g) ノード情報配信の要求

ノード情報は、ノード番号と MAC アドレスの対応を示した情報です。“R\_IN32\_TRUE” にするとマスター局からノード情報がマルチキャスト配信されます。

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。

**注意 1.** 本設定を“R\_IN32\_FALSE”と設定した場合、Transient1 受信データ処理（「6.2.17 Transient1、Transient2、TransientAck送信処理」参照）でノード情報配信の必要、不要を判定する“gblUserMACAddressTableRequest”も“R\_IN32\_FALSE”に設定してください。

**2.** 送信元に応答を返す場合は、送信元 MAC アドレスを利用して返信できます。  
能動的に送信する場合は、MAC アドレス表を利用します。  
MAC アドレス表はマスタ局から配信されたノード情報配信要求フレーム（Transient1 フレーム）を使用して作成します。

## (h) 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。

表6.12 アプリケーション詳細動作状態の初期値一覧

| 値                         | 通信動作                  |
|---------------------------|-----------------------|
| R_IN32_RUNSTS_UNSUPPORTED | 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        | アプリケーションが停止中          |
| R_IN32_RUNSTS_RUN         | アプリケーションが動作中          |
| R_IN32_RUNSTS_NOTEXIST    | アプリケーションユーザ実体が存在しない   |

## (i) アプリケーション詳細エラー状態 初期値

MyStatus フレームの nodeStatus 内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。

表6.13 アプリケーション詳細エラー状態の初期値一覧

| 値                        | 通信動作  |
|--------------------------|-------|
| R_IN32_ERRSTS_NONE       | エラーなし |
| R_IN32_ERRSTS_WARNING    | 軽度異常  |
| R_IN32_ERRSTS_ERROR      | 中度異常  |
| R_IN32_ERRSTS_FATALERROR | 重度異常  |

## (j) アプリケーションエラーコード 初期値

MyStatus フレームの errorCode の初期値を設定します。

## (k) ベンダー固有のノード情報 初期値

MyStatus フレームの vendorSpfNodeInfo の初期値を設定します。

## (l) オプション有無

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。

**備考.** 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。

## (m) SLMP サポートビット

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。

**注意. 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。**

## (n) 診断機能対応有無

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。

**注意. 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。**

## (3) gerR\_IN32\_SetNodeAndNetworkNumber

|        |  |                  |                      |     |
|--------|--|------------------|----------------------|-----|
| 機能     | ノード番号およびネットワーク番号設定   |                  |                      |     |
| 呼び出し形式 | ERRCODE gerR_IN32_SetNodeAndNetworkNumber<br>(UCHAR uchNetworkNumber, USHORT usNodeNumber)   |                  |                      |     |
| 引数     | 型名   | 変数名              | 内容                   | 入出力 |
|        | UCHAR  | uchNetworkNumber | ネットワーク番号 (値域: 1~239) | 入力  |
|        | USHORT   | usNodeNumber     | ノード番号 (値域: 1~120)    | 入力  |
| 戻り値    | R_IN32_OK : 正常終了<br>R_IN32_ERR : 異常終了 (ライブラリ内での状態異常)<br>R_IN32_ERR_OUTOFRANGE : ノード番号範囲外またはネットワーク番号範囲外の場合  |                  |                      |     |
| 説明     | R-IN32M3-CL にノード番号およびネットワーク番号を設定します。<br>戻り値が R_IN32_ERR_OUTOFRANGE の場合、ノード番号とネットワーク番号は設定されませんので、エラー処理を起動元の関数に追加してください。<br><br>※本関数の起動は、「6.2.2 初期化処理」の後、「6.2.3 通信開始処理」で gerR_IN32_Start 関数を起動する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起動すると 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-IN32M3-CL 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 |
|        | なし  |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了<br>R_IN32_ERR : 異常終了   |     |    |     |
| 説明     | R-IN32M3-CL に通信開始を指示します。<br><br>※R-IN32M3-CL に Fatal エラーが発生した場合、本関数より、ベンダー作成の以下の関数を起動しますので、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。gR_IN32_CallbackFatalError |     |    |     |

## 6.4.2 ウォッチドグタイマ

## (1) gerR\_IN32\_ResetWDT

|        |  |     |    |     |
|--------|--|-----|----|-----|
| 機能     | R-IN32M3-CL 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 |
|        | なし   |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了   |     |    |     |
| 説明     | <p>R-IN32M3-CL 内部 WDT をリセットします。</p> <p>※ : 本関数起動後、「6.4.2 ウォッチドグタイマ」内の関数を起動する場合、1.032<math>\mu</math>s 以上あけてください。</p> |     |    |     |

## (2) gerR\_IN32\_DisableWDT

|        |  |     |    |     |
|--------|--|-----|----|-----|
| 機能     | R-IN32M3-CL 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 |
|        | なし   |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了   |     |    |     |
| 説明     | <p>R-IN32M3-CL 内部 WDT を無効にします。</p> <p>※ : 本関数起動後、「6.4.2 ウォッチドグタイマ」内の関数を起動する場合、1.032<math>\mu</math>s 以上あけてください。</p> <p>R-IN32M3-CL は、リセット直後は、R-IN32M3-CL 内部 WDT は有効になっています。(R-IN32M3-CL 内部 WDT 時限設定の初期値は、3.2 秒)</p> <p>gerR_IN32_Initialize 関数起動にて R-IN32M3-CL 内部 WDT が無効になりますが、gerR_IN32_Initialize 関数起動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。</p> <ul style="list-style-type: none"> <li>・ 本関数を起動し、R-IN32M3-CL 内部 WDT を無効にします。</li> <li>・ gerR_IN32_ResetWDT 関数を起動し、R-IN32M3-CL 内部 WDT をリセットします。(R-IN32M3-CL 内部 WDT がタイムアウトしないようにしてください。)</li> </ul> |     |    |     |

## (3) gerR\_IN32\_EnableWDT

|        |  |     |    |     |
|--------|--|-----|----|-----|
| 機能     | R-IN32M3-CL 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 |
|        | なし   |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了   |     |    |     |
| 説明     | <p>R-IN32M3-CL 内部 WDT を有効にします。</p> <p>※ : 本関数起動後、「6.4.2 ウォッチドグタイマ」内の関数を起動する場合、1.032<math>\mu</math>s 以上あけてください。</p> <p>gerR_IN32_Initialize 関数起動にて、R-IN32M3-CL 内部 WDT が無効となりますので R-IN32M3-CL 内部 WDT を使用する場合は、本関数を起動してください。</p> |     |    |     |

## (4) gerR\_IN32\_SetWDT

|        |  |            |  |     |
|--------|--|------------|--|-----|
| 機能     | R-IN32M3-CL 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-IN32M3-CL 内部 WDT 時限設定<br>0x0000 : 100ms<br>0x0001 : 200ms<br>0x0002 : 300ms<br>:<br>0x001F : 3.2 秒 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了   |            |  |     |
| 説明     | <p>R-IN32M3-CL 内部 WDT の時限を設定します。</p> <p>※ : 本関数起動後、「6.4.2 ウォッチドグタイマ」内の関数を起動する場合、1.032<math>\mu</math>s 以上あけてください。</p> <p>R-IN32M3-CL 内部 WDT 起動中 (gerR_IN32_EnableWDT 関数起動後) に本関数で R-IN32M3-CL 内部 WDT 時限設定を変更した場合、gerR_IN32_ResetWDT 関数を起動したときに変更した時限設定で R-IN32M3-CL 内部 WDT が動作します。</p> <p>(gerR_IN32_ResetWDT 関数を起動するまでは変更前の R-IN32M3-CL 内部 WDT 時限設定で動作します。)</p> |            |  |     |

## 6.4.3 イベント

## (1) gerR\_IN32\_GetEvent

|        |  |           |       |     |
|--------|--|-----------|-------|-----|
| 機能     | R-IN32M3-CL 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 | 出力  |
| 戻り値    | R_IN32_OK : 正常終了   |           |       |     |
| 説明     | R-IN32M3-CL のイベントを検出します。   |           |       |     |

gerR\_IN32\_GetEvent の引数

R\_IN32\_EVTPRM\_INTERRUPT\_T の構成を、サンプルコードを元に下記に示します。

```

/* 割込み要因 */
typedef struct R_IN32_EVTPRM_INTERRUPT_TAG          {
    union {
        ULONG    ulAll;
        struct {
            ULONG    b1ZCommConnect:           1;    /* b0 : 通信参加 */
            ULONG    b1ZCommDisconnect:        1;    /* b1 : 通信解列 */
            ULONG    b1ZCommConnectToDisconnect: 1;    /* b2 : 通信参加→通信解列 */
            ULONG    b1ZCommDisconnectToConnect: 1;    /* b3 : 通信解列→通信参加 */
            ULONG    b1ZChangeStNoNetNo:        1;    /* b4 : ノード番号、ネットワーク番号変更 */
            ULONG    b1ZChangeActCommand:       1;    /* b5 : 動作コマンド変更 */
            ULONG    b1ZPrmFrmRcv_OK:           1;    /* b6 : Parameter フレーム受信 */
            ULONG    b1ZReserve1:               1;    /* b7 : 予約 */
            ULONG    b1ZPrmChkFrmRcv_OK:        1;    /* b8 : ParamCheck フレーム受信
                (パラメーター一致時) */
            ULONG    b3ZReserve2:               3;    /* b9-11 : 予約 */
            ULONG    b1ZRecvNonCyclic:          1;    /* b12 : トランジェント受信 */
            ULONG    b1ZSendFinNonCyclic:       1;    /* b13 : トランジェント送信完了 */
            ULONG    b7ZReserve3:               7;    /* b14-20 : 予約 */
            ULONG    b1ZMasterWatchTimeout:     1; /* b21 : マスタウォッチタイマ タイムアウト発生 */
            ULONG    bAZReserve4:               10;   /* b22-31 : 予約 */
        } stBit;
    } uniFlag;
} R_IN32_EVTPRM_INTERRUPT_T;

```

## (2) gerR\_IN32\_Main

|        |   |           |       |     |
|--------|---|-----------|-------|-----|
| 機能     | R-IN32M3-CL 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<br>R_IN32_EVTPRM_<br>INTERRUPT_T  | *pstEvent | 割込み要因 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了<br>R_IN32_ERR : 異常終了 (ライブラリ内での状態異常)  |           |       |     |
| 説明     | R-IN32M3-CL イベントに対する処理を行います。<br><br>※ : 本関数は、「6.2.2 初期化処理」と「6.2.3 通信開始処理」の後で起動する必要があります。<br>前述の処理を実行する前に本関数を起動すると<br>R_IN32_ERR : 異常終了 (ライブラリ内での状態異常)<br>となります。 |           |       |     |

## (3) gerR\_IN32\_RestartEvent

|        |   |     |    |     |
|--------|---|-----|----|-----|
| 機能     | R-IN32M3-CL 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 |
|        | なし  |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了  |     |    |     |
| 説明     | R-IN32M3-CL イベント検出 (gerR_IN32_GetEvent 関数) で停止したイベントを再開します。 |     |    |     |

## (4) gerR\_IN32\_UpdatePortStatus

|        |   |        |   |     |
|--------|---|--------|---|-----|
| 機能     | PHY リンク状態更新                                       |        |   |     |
| 呼び出し形式 | ERRCODE gerR_IN32_UpdatePortStatus (ULONG ulPort) |        |   |     |
| 引数     | 型名  | 変数名    | 内容  | 入出力 |
|        | ULONG   | ulPort | ポート指定<br>R_IN32_PORT1(0) : ポート 1<br>R_IN32_PORT2(1) : ポート 2 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了                                  |        |   |     |
| 説明     | PHY リンク状態を更新します。                                  |        |   |     |

## (5) 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 |
|        | なし  |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了<br>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))<br>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。<br><br>※ : 本関数の戻り値が R_IN32_OK 以外の場合、ベンダー作成の以下の関数を起動しますので、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。<br>gR_IN32_CallbackFatalError |     |    |     |

## 6.4.4 サイクリック伝送

## (1) gerR\_IN32\_SetCyclicStop

|        |  |     |    |     |
|--------|--|-----|----|-----|
| 機能     | 機器側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 |
|        | なし   |     |    |     |
| 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。<br>解除する場合、gerR_IN32_ClearCyclicStop 関数を起動してください。 |     |    |     |

## (2) gerR\_IN32\_ClearCyclicStop

|        |   |     |    |     |
|--------|---|-----|----|-----|
| 機能     | 機器側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 |
|        | なし  |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了                                  |     |    |     |
| 説明     | gerR_IN32_SetCyclicStop 関数の起動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 <sup>※1</sup>                              | 出力  |
|        | BOOL   | blEnable | コピー有効/無効<br>R_IN32_TRUE : 有効<br>R_IN32_FALSE : 無効 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了 (受信データあり)<br>R_IN32_ERR : 異常終了 (受信データなし)  |          |   |     |
| 説明     | <p>マスタ局から受信したサイクリック受信データを pRyDst および pRWwDst の示すアドレスに格納します。</p> <p>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 となります。)</p> <p>※ : R_IN32_ERR : 異常終了 (受信データなし)</p> <p>前回の gerR_IN32_GetReceivedCyclicData 関数起動から、今回の gerR_IN32_GetReceivedCyclicData 関数起動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。</p> <p>※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。</p> |          |   |     |

## (4) gerR\_IN32\_GetMasterNodeStatus

|        |   |             |   |     |
|--------|---|-------------|---|-----|
| 機能     | マスタ局状態取得  |             |   |     |
| 呼び出し形式 | ERRCODE gerR_IN32_GetMasterNodeStatus<br>(BOOL *pblRunSts, BOOL *pblErrSts, ULONG *pulErrCode)  |             |   |     |
| 引数     | 型名  | 変数名         | 内容  | 入出力 |
|        | BOOL  | *pblRunSts  | アプリケーション動作状態<br>R_IN32_TRUE : 実行中<br>R_IN32_FALSE : 停止中       | 出力  |
|        | BOOL  | *pblErrSts  | アプリケーションエラー状態<br>R_IN32_TRUE : エラー発生中<br>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)<br>R_IN32_ERR : 異常終了<br>(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。<br>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。<br>pblRunStsR_IN32_FALSE<br>pblErrStsR_IN32_FALSE<br>pulErrCode0 |             |   |     |

## (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 |
|        | なし   |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了   |     |    |     |
| 説明     | gerR_IN32_SetNodeStatus 関数で設定した自局状態を R-IN32M3-CL 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<br>(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 <sup>※1</sup>                              | 入力  |
|        | BOOL  | blEnable | 更新 有効/無効<br>R_IN32_TRUE : 有効<br>R_IN32_FALSE : 無効 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了  |          |   |     |
| 説明     | <p>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-IN32M3-CLに設定します。</p> <p>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。<br/>(戻り値は、R_IN32_ERRとなります。)</p> <p>※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。</p> |          |   |     |

## 6.4.5 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<br>(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<br>R_IN32_RUNSTS_UNSUPPORTED(0) :<br>アプリケーション詳細動作状態通知未サポート<br>R_IN32_RUNSTS_STOP(1) :<br>アプリケーションが停止中<br>R_IN32_RUNSTS_RUN(2) :<br>アプリケーションが動作中<br>R_IN32_RUNSTS_NOTEXIST(3) :<br>アプリケーションユーザ実体が存在しない | 入力  |
|        | ULONG   | ulErrSts          | アプリケーションエラー状態<br>R_IN32_ERRSTS_NONE(0) : エラーなし<br>R_IN32_ERRSTS_WARNING(1) : 軽度異常<br>R_IN32_ERRSTS_ERROR(2) : 中度異常<br>R_IN32_ERRSTS_FATALERROR(3) : 重度異常   | 入力  |
|        | ULONG   | ulErrCode         | エラーコード   | 入力  |
|        | ULONG   | ulUserInformation | ベンダー固有ノード情報  | 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 |
|        | なし   |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了   |     |    |     |
| 説明     | R-IN32M3-CL を自局エラーに設定します。<br>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。 |     |    |     |

## 6.4.6 自局状態取得

## (1) gerR\_IN32\_GetNodeAndNetworkNumber

|        |  |                    |          |     |
|--------|--|--------------------|----------|-----|
| 機能     | ノード番号およびネットワーク番号取得   |                    |          |     |
| 呼び出し形式 | ERRCODE gerR_IN32_GetNodeAndNetworkNumber<br>(USHORT *pusNodeNumber, UCHAR *puchNetworkNumber) |                    |          |     |
| 引数     | 型名   | 変数名                | 内容       | 入出力 |
|        | USHORT   | *pusNodeNumber     | ノード番号    | 出力  |
|        | UCHAR  | *puchNetworkNumber | ネットワーク番号 | 出力  |
| 戻り値    | R_IN32_OK : 正常終了   |                    |          |     |
| 説明     | ノード番号およびネットワーク番号を取得します。  |                    |          |     |

## (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<br>ulRySize :<br>RY サイズ (バイト (オクテット))<br>ulRWwSize :<br>RWw サイズ (バイト (オクテット))<br>ulRxSize :<br>RX サイズ (バイト (オクテット))<br>ulRWrSize :<br>RWr サイズ (バイト (オクテット)) | 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。<br>gerR_IN32_GetReceivedCyclicData 関数および gerR_IN32_SetSendCyclicData 関数は、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。 |                |   |     |

gerR\_IN32\_GetCurrentCyclicSize の引数

R\_IN32\_CYCLIC\_SIZE\_T の構成を、サンプルコードを元に下記に示します。

```

/* サイクリック伝送サイズ */
typedef struct R_IN32_CYCLIC_SIZE_TAG {
    ULONG    ulRySize;          /*!< RY サイズ(バイト(オクテット)) */
    ULONG    ulRWwSize;        /*!< RWw サイズ(バイト(オクテット)) */
    ULONG    ulRxSize;         /*!< RX サイズ(バイト(オクテット)) */
    ULONG    ulRWrSize;        /*!< RWr サイズ(バイト(オクテット)) */
} R_IN32_CYCLIC_SIZE_T;

```

## (3) gerR\_IN32\_GetCommunicationStatus

| 機能     | データリンク状態取得   |             |  |     |
|--------|--|-------------|--|-----|
| 呼び出し形式 | ERRCODE gerR_IN32_GetCommunicationStatus (ULONG *pulCommSts)   |             |  |     |
| 引数     | 型名   | 変数名         | 内容   | 入出力 |
|        | ULONG  | *pulCommSts | データリンク状態<br>R_IN32_COMMSTS_CYC_DLINK(2) :<br>データリンク中 (サイクリック伝送中)<br>R_IN32_COMMSTS_TOKEN_PASS(1) :<br>データリンク中 (サイクリック伝送停止中)<br>R_IN32_COMMSTS_DISCONNECT(0) :<br>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。<br>データリンク状態により、D LINK LED を点消灯させてください。<br>R_IN32_COMMSTS_CYC_DLINK : LED 点灯<br>その他 : LED 消灯<br>※ : D LINK LED の点灯/消灯の制御は、「6.2.12 通信状態更新処理」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<br>(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<br>R_IN32_PORT1(0) : ポート 1<br>R_IN32_PORT2(1) : ポート 2   | 入力  |
|        | ULONG  | *pulLinkStatus   | リンク状態<br>R_IN32_LINKUP(1) : リンクアップ<br>R_IN32_LINKDOWN(0) : リンクダウン   | 出力  |
|        | ULONG  | *pulSpeed  | 速度<br>R_IN32_SPEED_1G(0) : 1Gbps<br>R_IN32_SPEED_100M(1) : 100Mbps<br>R_IN32_SPEED_10M(2) : 10Mbps<br>(第 2 引数*pulLinkStatus が R_IN32_LINKUP(1)のときに有効ですので、R_IN32_LINKDOWN(0)のときは使用しないでください。) | 出力  |
| ULONG  | *pulDuplex   | 全二重/半二重<br>R_IN32_DUPLEX_FULL(0) : 全二重<br>R_IN32_DUPLEX_HALF(1) : 半二重<br>(第 2 引数*pulLinkStatus が R_IN32_LINKUP(1)のときに有効ですので、R_IN32_LINKDOWN(0)のときは使用しないでください。) | 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。   |  |   |     |

## (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) |                  |  |     |
|        | 型名   | 変数名              | 内容   | 入出力 |
| 引数     | R_IN32_CYCLIC_STA_T  | *pstCyclicStatus | サイクリック伝送状態<br>Bit2-0 サイクリック伝送パラメータ保有状態<br>001b : パラメータ正常受信済み<br>010b : 未受信または ID 不一致<br>011b : 確認中<br>100b : パラメータ異常受信済み<br>Bit3 サイクリック伝送パラメータ確認状態<br>0 : 確認完了<br>1 : 確認中<br>Bit4 ノード番号不正設定状態<br>0 : 範囲内<br>1 : 範囲外<br>Bit5 予約ノード設定状態<br>0 : 非予約ノード<br>1 : 予約ノード<br>Bit6 サイクリック伝送実施指示 (一括) 設定状態<br>0 : 動作設定<br>1 : 停止設定<br>Bit7 サイクリック伝送実施指示 (個別) 設定状態<br>0 : 動作設定<br>1 : 停止設定<br>Bit8 予約<br>Bit9 サイクリック伝送継続不可異常状態<br>0 : 異常なし<br>1 : サイクリック伝送継続不可異常有<br>Bit10 ノード番号重複状態<br>0 : 重複なし<br>1 : 重複している<br>Bit11 予約<br>Bit12 ノード種別不正・指定サイズ不正状態<br>0 : 正常<br>1 : 不正<br>Bit13 予約<br>Bit14 解列発生状態<br>0 : サイクリック交信実施中<br>またはトークンパス実施中<br>1 : 解列中<br>Bit15 自己理由による停止状態<br>0 : 非停止<br>1 : 上記以外の理由でサイクリック伝送停止中 | 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。  |                  |  |     |

gerR\_IN32\_GetCyclicStatus の引数

R\_IN32\_CYCLIC\_STA\_T の構成を、サンプルコードを元に下記に示します。

```
/* サイクリック伝送状態 */
typedef struct R_IN32_CYCLIC_STA_TAG    {
    union {
        USHORT UsAll;
        struct {
            USHORT b3ZComonParamkeepCond:    3; /* b2-0 : サイクリック伝送パラメータ保有状態 */
            USHORT b1ZParamCheckCond:        1; /* b3 : サイクリック伝送パラメータ確認状態 */
            USHORT b1ZMyNodeNoRangeOut:      1; /* b4 : ノード番号不正設定状態 */
            USHORT b1ZMyNodeReserveSetup:    1; /* b5 : 予約ノード設定状態 */
            USHORT b1ZCyclicOpeInstructPackage: 1;
                                                    /* b6 : サイクリック伝送実施指示（一括）設定状態 */
            USHORT b1ZCyclicOpeInstructVarious: 1;
                                                    /* b7 : サイクリック伝送実施指示（個別）設定状態 */
            USHORT b1ZReserved1:              1; /* b8 : 予約 */
            USHORT b1ZMyMpuAbnomal:          1; /* b9 : サイクリック伝送継続不可異常状態 */
            USHORT b1ZMyNodeNumberDuplicate:  1; /* b10 : ノード番号重複状態 */
            USHORT b1ZReserved2:              1; /* b11 : 予約 */
            USHORT b1ZNodeTypeWrong:         1; /* b12 : ノード種別不正・指定サイズ不正状態 */
            USHORT b1ZReserved3:              1; /* b13 : 予約 */
            USHORT b1ZDLinkState:             1; /* b14 : 解列発生状態 */
            USHORT b1ZCyclicState:           1; /* b15 : 自己理由による停止状態 */
        } stBit;
    } uniCycSta;
} R_IN32_CYCLIC_STA_T;
```

## (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-IN32M3-CL 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。                                   |         |                    |     |

**注意. MIB 情報は非公開情報ですので、ベンダーのみの展開としてください。**

gerR\_IN32\_GetMIB の引数

R\_IN32\_MIB\_T の構成を、サンプルコードを元に下記に示します。

```

/* MIB 情報 */
typedef struct R_IN32_MIB_TAG {
    R_IN32_MIBSDRD_T    stSDRD;           /*!< 送受信部カウンタ値 */
    R_IN32_MIBMACIP_T   stMACIP1;        /*!< MAC IP 部カウンタ値(ポート 1) */
    R_IN32_MIBMACIP_T   stMACIP2;        /*!< MAC IP 部カウンタ値(ポート 2) */
    R_IN32_MIBRGCNT_T   stRING1;         /*!< リング制御部カウンタ(ポート 1) */
    R_IN32_MIBRGCNT_T   stRING2;         /*!< リング制御部カウンタ(ポート 2) */
    ULONG               ulP1DownCounter; /*!< リンクダウン数カウンタ(ポート 1) */
    ULONG               ulP2DownCounter; /*!< リンクダウン数カウンタ(ポート 2) */
    ULONG               ulMasterWatchCount; /*!< マスタウォッチタイマエラー数カウンタ*/
} R_IN32_MIB_T;

```

上記 R\_IN32\_MIB\_T に含まれるタグの構成を下記に示します。

```

/* MIB 情報(カウンタ) */
typedef struct R_IN32_MIBSDRD_TAG {
    ULONG   ulCyclicRecNomalFrameCnt; /*!< サイクリック受信フレーム数カウンタ */
    ULONG   ulNonCyclicRecValidCnt;   /*!< トランジェント受信フレーム数カウンタ */
    ULONG   ulNonCyclicRecRejectCnt;  /*!< トランジェント受信フレーム破棄数カウンタ */
} R_IN32_MIBSDRD_T;

/* MIB 情報(リング制御部) */
typedef struct R_IN32_MIBRGCNT_TAG {
    ULONG   ulHecErr; /*!< MIB1 : HEC エラーフレーム数 */
    ULONG   ulDcsFcsErr; /*!< MIB2 : DCS/FCS エラーフレーム数 */
    ULONG   ulUnderErr; /*!< MIB3 : アンダーサイズエラーフレーム数 */
    ULONG   ulRpt; /*!< MIB4 : フォワードフレーム数 */
    ULONG   ulUp; /*!< MIB5 : 上位層転送フレーム数 */
    ULONG   ulRptFullDrop; /*!< MIB6 : フォワードバッファフル廃棄フレーム数 */
    ULONG   ulUpFullDrop; /*!< MIB7 : 上位層転送バッファフル廃棄フレーム数 */
} R_IN32_MIBRGCNT_T;

```

```

/* MIB 情報 (MAC IP) */
typedef struct R_IN32_MIBMACIP_TAG {
    ULONG    ulRFrm;                /*!< 受信フレーム数カウンタ */
    ULONG    ulTFrm;                /*!< 送信フレーム数カウンタ */
    ULONG    ulRUnd;                /*!< 受信アンダーサイズフレーム数カウンタ */
    ULONG    ulROvr;                /*!< 受信オーバーサイズフレーム数カウンタ */
    ULONG    ulRFcs;                /*!< 受信フレーム FCS エラー数カウンタ */
    ULONG    ulRFgm;                /*!< 受信フレームフラグメントエラー数カウンタ */
    ULONG    ulRIFGErr;            /*!< 最小 IFG フレーム検出数カウンタ */
    ULONG    ulREps;                /*!< 受信フレーム SFD 以下検出数カウンタ */
    ULONG    ulRCde;                /*!< 受信コードエラー数カウンタ */
    ULONG    ulRFce;                /*!< 受信不正キャリアエラー数カウンタ */
    ULONG    ulRCEE;                /*!< 受信キャリア拡張エラー数カウンタ */
} R_IN32_MIBMACIP_T;

```

## (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 |
|        | なし                                |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了                  |     |    |     |
| 説明     | MIB 情報をクリアします。                    |     |    |     |

## (8) gerR\_IN32\_GetPortAvailable

|        |  |                   |  |     |
|--------|--|-------------------|--|-----|
| 機能     | ポート有効状態取得  |                   |  |     |
| 呼び出し形式 | ERRCODE gerR_IN32_GetPortAvailable (ULONG* pulPortAvailable) |                   |  |     |
| 引数     | 型名   | 変数名               | 内容   | 入出力 |
|        | ULONG  | *pulPortAvailable | ポート有効状態<br>R_IN32_MYPORT_PORTALL(0x00) :<br>保有する全ポート有効<br>R_IN32_MYPORT_PORT_1(0x01) :<br>ポート 1 のみ有効<br>R_IN32_MYPORT_PORT_2(0x02) :<br>ポート 2 のみ有効 | 出力  |
| 戻り値    | R_IN32_OK : 正常終了   |                   |  |     |
| 説明     | マスタ局から設定されるポートごとの有効状態を取得します。                                 |                   |  |     |

## 6.4.7 LED 制御

## (1) gerR\_IN32\_SetLERR1LED

|        |  |        |   |     |
|--------|--|--------|---|-----|
| 機能     | LED 点灯制御 (L ER (ポート 1))  |        |   |     |
| 呼び出し形式 | ERRCODE gerR_IN32_SetLERR1LED (ULONG ulCtrl)   |        |   |     |
| 引数     | 型名   | 変数名    | 内容  | 入出力 |
|        | ULONG  | ulCtrl | LED 点灯制御<br>R_IN32_LED_OFFLED 消灯<br>R_IN32_LED_ONLED 点灯 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了   |        |   |     |
| 説明     | ポート 1 の L ER LED の点消灯を行います。<br>※ : R-IN32M3-CL 内部 WDT / 外部 WDT / 自局エラー発生中は、点灯にできません。 |        |   |     |

## (2) gerR\_IN32\_SetLERR2LED

|        |  |        |   |     |
|--------|--|--------|---|-----|
| 機能     | LED 点灯制御 (L ER (ポート 2))  |        |   |     |
| 呼び出し形式 | ERRCODE gerR_IN32_SetLERR2LED (ULONG ulCtrl)   |        |   |     |
| 引数     | 型名   | 変数名    | 内容  | 入出力 |
|        | ULONG  | ulCtrl | LED 点灯制御<br>R_IN32_LED_OFFLED 消灯<br>R_IN32_LED_ONLED 点灯 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了   |        |   |     |
| 説明     | ポート 2 の L ER LED の点消灯を行います。<br>※ : R-IN32M3-CL 内部 WDT / 外部 WDT / 自局エラー発生中は、点灯にできません。 |        |   |     |

## (3) 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<br>R_IN32_LED_OFFLED 消灯<br>R_IN32_LED_ONLED 点灯<br>R_IN32_LED_BLINKLED 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。<br>※ : R-IN32M3-CL 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\_SetDLINKLED

|        |   |        |   |     |
|--------|---|--------|---|-----|
| 機能     | LED 点灯制御 (D LINK)   |        |   |     |
| 呼び出し形式 | ERRCODE gerR_IN32_SetDLINKLED (ULONG ulCtrl)                                    |        |   |     |
| 引数     | 型名  | 変数名    | 内容  | 入出力 |
|        | ULONG   | ulCtrl | LED 点灯制御<br>R_IN32_LED_OFFLED 消灯<br>R_IN32_LED_ONLED 点灯<br>R_IN32_LED_BLINKLED 点滅 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了  |        |   |     |
| 説明     | D LINK LED の点消灯を行います。<br>※ : R-IN32M3-CL 内部 WDT/外部 WDT/自局エラー発生中は、点灯または点滅にできません。 |        |   |     |

## (5) 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<br>R_IN32_LED_OFFLED 消灯<br>R_IN32_LED_ONLED 点灯<br>R_IN32_LED_BLINKLED 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。<br>※ : R-IN32M3-CL 内部 WDT/外部 WDT/自局エラー発生中は、点灯または点滅にできません。 |        |   |     |

## (6) 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<br>R_IN32_LED_OFFLED 消灯<br>R_IN32_LED_ONLED 点灯<br>R_IN32_LED_BLINKLED 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。<br>※ : R-IN32M3-CL 内部 WDT/外部 WDT/自局エラー発生中は、点灯または点滅にできません。 |        |   |     |

## (7) 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<br>R_IN32_LED_OFFLED 消灯<br>R_IN32_LED_ONLED 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。<br>※ : R-IN32M3-CL 内部 WDT / 外部 WDT / 自局エラー発生中は、点灯にできません。 |        |   |     |

## (8) 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<br>(ON : 無効、OFF : 前置保持)<br>Bit 0 : RUN LED 無効<br>Bit 2 : User LED 2 無効<br>Bit 4 : User LED 1 無効<br>Bit 6 : D LINK LED 無効<br>Bit 8 : ERR. LED 無効<br>Bit10 : ポート 1 L ER LED 無効<br>Bit11 : ポート 2 L ER LED 無効<br>(Bit1、3、5、7、9、12~15 : 使用しません) | 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。<br>※ : R-IN32M3-CL 内部 WDT / 外部 WDT / 自局エラー発生中は、無効にできません。 |              |   |     |

## (9) 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<br>(ON : 有効設定、OFF : 前置保持)<br>Bit 0 : RUN LED 有効<br>Bit 2 : User LED 2 有効<br>Bit 4 : User LED 1 有効<br>Bit 6 : D LINK LED 有効<br>Bit 8 : ERR. LED 有効<br>Bit10 : ポート 1 L ER LED 有効<br>Bit11 : ポート 2 L ER LED 有効<br>(Bit1、3、5、7、9、12~15 : 使用しません) | 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。                                 |              |   |     |

## 6.4.8 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<br>pusSerial[0] : ネットワーク時刻 (bit15-0)<br>pusSerial[1] : ネットワーク時刻 (bit31-16)<br>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、15.2587890625μsec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<br>pusSerial[0] : ネットワーク時刻 (bit15-0)<br>pusSerial[1] : ネットワーク時刻 (bit31-16)<br>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、15.2587890625μsec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<br>(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   | 出力  |
|        | const USHORT   | *pusSerial   | ネットワーク時刻<br>pusSerial[0] : ネットワーク時刻 (bit31-16)<br>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、sec単位のシリアル値）を時計情報（西暦・月・日・時・分・秒・msec（0固定）・曜日）に変換します。               |              |  |     |

gerR\_IN32\_NetworkTimeToDate の引数

R\_IN32\_TIMEINFO\_T の構成を、サンプルコードを元に下記に示します。

```

/* 時計情報 */
typedef struct R_IN32_TIMEINFO_TAG {
    USHORT    usYear;           /*!< 西暦 (2000 - 2136)*/
    USHORT    usMonth;         /*!< 月 ( 1 - 12)*/
    USHORT    usDay;           /*!< 日 ( 1 - 31)*/
    USHORT    usHour;          /*!< 時 ( 0 - 23)*/
    USHORT    usMin;           /*!< 分 ( 0 - 59)*/
    USHORT    usSec;           /*!< 秒 ( 0 - 59)*/
    USHORT    usMsec;          /*!< msec ( 0 - 999)*/
    USHORT    usWday;          /*!< 曜日 (0(日)- 6(土))*/
} R_IN32_TIMEINFO_T;

```

#### (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<br>R_IN32_TIMEINFO_T   | *pstTimeInfo                          | 時計情報  | 入力  |
| 引数     | USHORT   | *pusSerial                            | ネットワーク時刻<br>pusSerial[0] : ネットワーク時刻 (bit15-0)<br>pusSerial[1] : ネットワーク時刻 (bit31-16)<br>pusSerial[2] : ネットワーク時刻 (bit47-32) | 出力  |
|        | 戻り値  | R_IN32_OK : 正常終了<br>R_IN32_ERR : 異常終了 |   |     |
| 説明     | <p>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sec単位のシリアル値）に変換します。</p> <p>（ausSerial[0] : ネットワーク時刻（bit15-0）は、「0固定」となります。）</p> <p>※：西暦が、2000年～2136年以外の場合、R_IN32_ERRとなります。</p> <p>R-IN32M3-CLドライバは、上記以外のエラーチェックは行っていませんので、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。</p> |                                       |   |     |

## 6.4.9 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 |
|        | なし   |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了<br>R_IN32_ERR : 異常終了 (MDIO コマンド終了待ち異常)  |     |    |     |
| 説明     | <p>MAC IP のアクセスを許可します。</p> <p>※ : MAC IP アクセス許可 (gerR_IN32_EnableMACIPAccess 関数) ~MAC IP アクセス禁止 (gerR_IN32_DisableMACIPAccess 関数) の間は、出来るだけ短い間としてください。<br/>(ベンダーが割込みを使用される場合、MAC IP アクセス許可~MAC IP アクセス禁止の間は割込み禁止状態で使用してください。)</p> <p>本関数の戻り値が R_IN32_OK 以外の場合、ベンダー作成の以下の関数を起動しますので、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。<br/>gR_IN32_CallbackFatalError</p> |     |    |     |

## (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 |
|        | なし  |     |    |     |
| 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<br>R_IN32_PORT1(0) : ポート 1<br>R_IN32_PORT2(1) : ポート 2 | 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<br>R_IN32_ERR : 異常終了 (MDIO コマンド終了待ち異常)  |        |  |     |
| 説明     | <p>MDIO にて PHY 内部レジスタをライトします。</p> <p>※ : MAC IP アクセス許可 (gerR_IN32_EnableMACIPAccess 関数) ~MAC IP アクセス禁止 (gerR_IN32_DisableMACIPAccess 関数) の間で使用してください。</p> <p>本関数の戻り値が R_IN32_OK 以外の場合、ベンダー作成の以下の関数を起動しますので、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。<br/>gR_IN32_CallbackFatalError</p> |        |  |     |

## (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<br>R_IN32_PORT1(0) : ポート 1<br>R_IN32_PORT2(1) : ポート 2 | 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<br>R_IN32_ERR : 異常終了 (MDIO コマンド終了待ち異常)  |         |  |     |
| 説明     | MDIO にて PHY 内部レジスタをリードします。<br>※ : MAC IP アクセス許可 (gerR_IN32_EnableMACIPAccess 関数) ~MAC IP アクセス禁止 (gerR_IN32_DisableMACIPAccess 関数) の間で使用してください。<br>本関数の戻り値が R_IN32_OK 以外の場合、ベンダー作成の以下の関数を起動しますので、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。<br>gR_IN32_CallbackFatalError |         |  |     |

## 6.4.10 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 |
|        | なし   |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了                               |     |    |     |
| 説明     | R-IN32M3-CL 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 |
|        | なし  |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了  |     |    |     |
| 説明     | gerR_IN32_MainReceiveTransient1 関数で取得したトランジェント受信フレームをコールバック関数 gerR_IN32_CallbackReceivedTransient 関数でユーザプログラムに渡します。 |     |    |     |

## (3) gerR\_IN32\_EnableReceiveTransient

|        |   |          |  |     |
|--------|---|----------|--|-----|
| 機能     | ベンダー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<br>R_IN32_TRUE : 受信許可<br>R_IN32_FALSE : 受信不許可 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了  |          |  |     |
| 説明     | ベンダー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。<br><br>ベンダー作成の以下の関数の戻り値が R_IN32_ERR の場合、「ベンダー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。<br>gerR_IN32_CallbackReceivedTransient |          |  |     |

## (4) gblR\_IN32\_GetReceiveTransientStatus

|        |  |     |    |     |
|--------|--|-----|----|-----|
| 機能     | ベンダー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 |
|        | なし   |     |    |     |
| 戻り値    | 受信許可設定状態<br>R_IN32_TRUE : 受信許可<br>R_IN32_FALSE : 受信不許可 |     |    |     |
| 説明     | ベンダー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<br>(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<br>R_IN32_ERR_OUTOFRANGE : ノード番号範囲外または配信シーケンシャル番号範囲外  |                |                           |     |
| 説明     | <p>マスタ局から受信した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。</p> <p>※ : マスタ局のノード番号は、0x7D として登録してください。</p> <p>gerR_IN32_Initialize 関数の B) R_IN32_UNITINIT_T 初期設定 (g) ノード情報配信の要求で R_IN32_FALSE と設定した場合、本関数の起動は不要です。</p> |                |                           |     |

gerR\_IN32\_SetMACAddressTableData の引数

R\_IN32\_MACADDRESSDATA\_T の構成を、サンプルコードを元に下記に示します。

```

/* MAC アドレスなどの情報 (MAC アドレス表) */
typedef struct _R_IN32_MACADDRESSDATA_TAG {
    USHORT    usNodeNumber;          /*!<ノード番号(1~120、マスタ局:0x7D) */
    UCHAR     uchTransientReceiveEnable;
                                           /*!<トランジェント受信機能(R_IN32_ENABLE/R_IN32_DISABLE) */
    UCHAR     auchMacAddress[6];     /*!< MAC アドレス */
} R_IN32_MACADDRESSDATA_T;

```

## 6.4.11 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。<br>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。  |                        |            |     |

gerR\_IN32\_GetUnitInformation の引数

R\_IN32\_UNITNETWORKSETTING\_T の構成を、サンプルコードを元に下記に示します。

```

/* ネットワーク動作設定 */
typedef struct R_IN32_UNITNETWORKSETTING_TAG {
    ULONG    ulFrameSendCount;           /*!< トークン保持時の送信回数設定 */
    ULONG    ulFrameSendInterval;       /*!< フレーム送信間隔設定 */
    ULONG    ulTokenSendCount;          /*!< トークン送信回数設定 */
} R_IN32_UNITNETWORKSETTING_T;

```

## (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 |
|        | なし  |     |    |     |
| 戻り値    | ノード ID  |     |    |     |
| 説明     | ノード ID を取得します。<br>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<br>13-34-56-78-90-AB が設定されている場合、<br>以下のアドレスが返ります。<br>puchMACAddr[0] : 0x13<br>puchMACAddr[1] : 0x34<br>puchMACAddr[2] : 0x56<br>puchMACAddr[3] : 0x78<br>puchMACAddr[4] : 0x90<br>puchMACAddr[5] : 0xAB | 出力  |
| 戻り値    | R_IN32_OK : 正常終了<br>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。<br>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全てのノード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   | ノード番号 (値域: 1~120、マスタ局: 0x7D) | 入力  |
| UCHAR  | *puchMACAddr   | ユニキャストアドレス<br>12-34-56-78-90-AB が設定されている場合、<br>以下のアドレスが返ります。<br>puchMACAddr[0] : 0x12<br>puchMACAddr[1] : 0x34<br>puchMACAddr[2] : 0x56<br>puchMACAddr[3] : 0x78<br>puchMACAddr[4] : 0x90<br>puchMACAddr[5] : 0xAB | 出力                           |     |
| 戻り値    | R_IN32_OK : 正常終了<br>R_IN32_ERR_NOENTRY : エントリがない<br>R_IN32_ERR_OUTOFRANGE : ノード番号範囲外   |  |                              |     |
| 説明     | マスタ局から受信したノード情報配信よりノード番号に対応したユニキャスト MAC アドレスを取得します。<br><br>※ : 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。<br>(R_IN32_ERR_NOENTRY となります。)<br>マスタ局のノード番号は、0x7D としてください。 |  |                              |     |

## (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)<br>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     | <p>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。</p> <ul style="list-style-type: none"> <li>・トランジェント送信バッファアドレス</li> <li>・トランジェント送信バッファ No</li> <li>・トランジェントコネクション情報</li> </ul> <p>※：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。</p> <ul style="list-style-type: none"> <li>・データリンク未実施 (解列中) の場合</li> <li>・送信データサイズが 1510 バイトより大きい場合</li> </ul> <p>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。</p> <ul style="list-style-type: none"> <li>・本関数にてトランジェント送信バッファ No を取得</li> <li>・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</li> <li>・gerR_IN32_RequestSendingTransient 関数にてトランジェント送信要求</li> </ul> |                     |                     |     |

## (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<br>R_IN32_ERR : 異常終了 (トランジェント送信要求異常)   |               |                     |     |
| 説明     | <p>gerR_IN32_GetSendTransientBuffer 関数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。</p> <p>本関数を実行する前に、以下の処理を行ってください。</p> <ul style="list-style-type: none"> <li>・ gerR_IN32_GetSendTransientBuffer 関数で、トランジェント送信バッファを取得</li> <li>・ 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</li> </ul> <p>※：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。</p> <ul style="list-style-type: none"> <li>・ データリンク未実施 (解列中) の場合</li> </ul> <p>本関数にて送信要求後に発生した異常は、gerR_IN32_MainSendTransient 関数の戻り値で通知します。</p> <p>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。</p> |               |                     |     |

## (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 |
|        | なし   |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了   |     |    |     |
| 説明     | <p>トランジェント送信完了結果を取得します。</p> <p>本関数から、gerR_IN32_CallbackTransientSendingComplete 関数を起動して、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。</p> |     |    |     |

## (8) gulR\_IN32\_SetOptionInfo\_Response

|        |   |                   |   |     |
|--------|---|-------------------|---|-----|
| 機能     | オプション情報取得応答フレーム作成処理   |                   |   |     |
| 呼び出し形式 | ULONG gulR_IN32_SetOptionInfo_Response(VOID* pvSendFrame, const VOID* pvReceivedData, const UCHAR* puchSA, const USHORT usSupportFunction)                          |                   |   |     |
| 引数     | 型名  | 変数名               | 内容  | 入出力 |
|        | VOID*   | pvSendFrame       | 送信フレームのアドレス   | 出力  |
|        | const VOID*   | pvReceivedData    | 受信データ格納エリアのアドレス                                     | 入力  |
|        | const UCHAR*  | puchSA            | 送信元ノード MAC アドレス                                     | 入力  |
|        | const USHORT  | usSupportFunction | SLMP 対応有無<br>USER_SUPPORT_FUNCTION (1) :<br>SLMP 対応 | 入力  |
| 戻り値    | 送信データサイズ (DCS/FCS を除く)  |                   |   |     |
| 説明     | オプション情報取得の応答フレームを作成します。<br>SLMP 対応有無 (usSupportFunction) には、USER_SUPPORT_FUNCTION (1)を設定してください。<br>サンプルコードでは、デフォルトで USER_SUPPORT_FUNCTION (1)に設定していますので、変更しないでください。 |                   |   |     |

## (9) gulR\_IN32\_SetSelectInfo\_Response

|        |   |                |   |     |
|--------|---|----------------|---|-----|
| 機能     | 選択局情報取得応答フレーム作成処理   |                |   |     |
| 呼び出し形式 | ULONG gulR_IN32_SetSelectInfo_Response(VOID* pvSendFrame, const VOID* pvReceivedData, const UCHAR* puchSA, const USER_LED_INFO* pstUserLedInfo) |                |   |     |
| 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  |
|        | const VOID*   | pvReceivedData | 受信データ格納エリアのアドレス   | 入力  |
|        | const UCHAR*  | puchSA         | 送信元ノード MAC アドレス   | 入力  |
|        | const USER_SELECTINFO_LED_INFO_T*   | pstUserLedInfo | 自局 LED 情報<br><b>【LED 色】</b><br>USER_SELECTINFO_LED_UNUSED(0) :<br>LED 未使用<br>USER_SELECTINFO_LED_GREEN(1) : 緑<br>USER_SELECTINFO_LED_RED(2) : 赤<br>USER_SELECTINFO_LED_ORANGE(3) : 橙<br><b>【LED 状態】</b><br>USER_SELECTINFO_LED_UNUSED(0) :<br>LED 未使用<br>USER_SELECTINFO_LED_OFF(1) : 消灯<br>USER_SELECTINFO_LED_ON(2) : 点灯<br>USER_SELECTINFO_LED_BLINK(3) : 点滅 | 入力  |
| 戻り値    | 送信データサイズ (DCS/FCS を除く)  |                |   |     |
| 説明     | 選択局情報取得の応答フレームを作成します。<br>自局の LED 情報を表示させるために、自局の状態に応じた自局 LED 情報を設定してください。   |                |   |     |

gulR\_IN32\_SetSelectInfo\_Response の引数

USER\_SELECTINFO\_LED\_INFO\_T の構成を、サンプルコードを元に下記に示します。

```

/* 自局 LED 情報 */
typedef struct USER_SELECTINFO_LED_INFO_TAG {
    UCHAR          uchRow;          /* LED 配列行数 */
    UCHAR          uchColumn;      /* LED 配列列数 */
    USER_LED_INFO_T stLedInf[8];  /* LED 情報 1~8 */
} USER_SELECTINFO_LED_INFO_T;
    
```

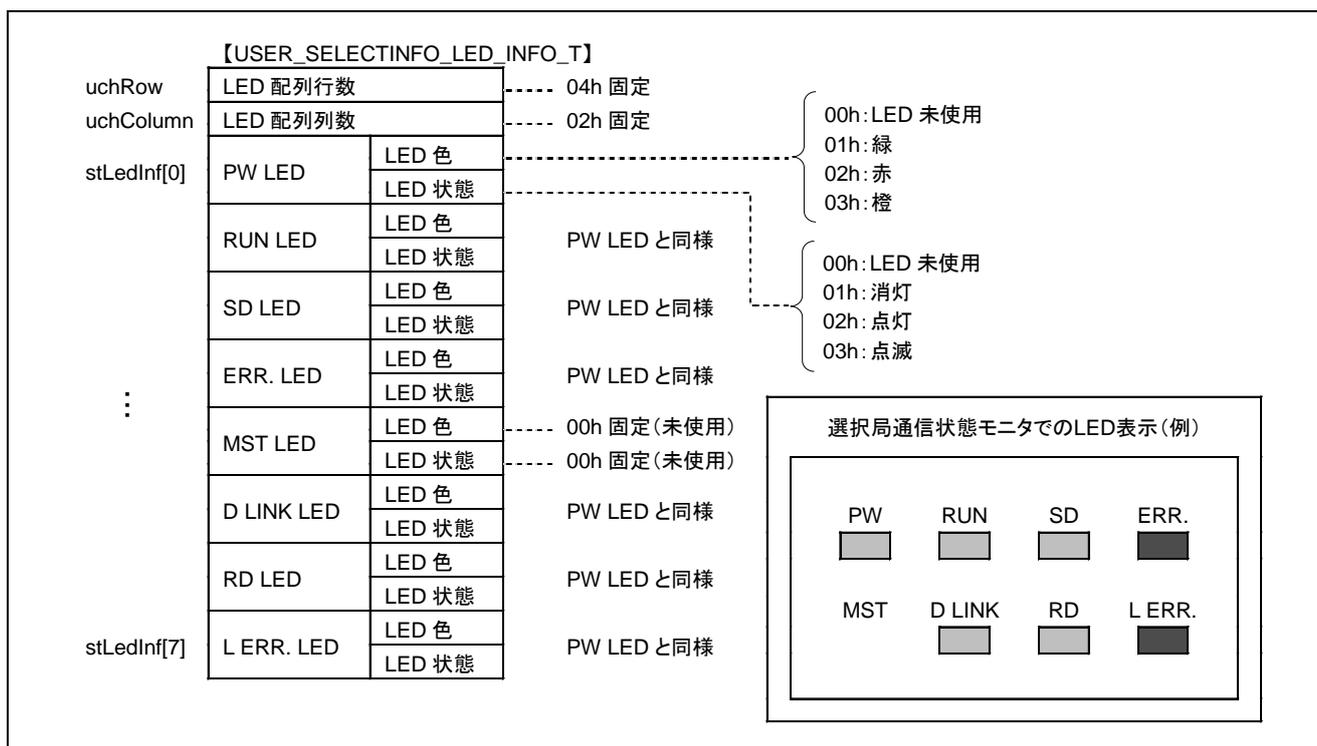


図6.53 自局 LED 情報

**注意.** 選択局情報取得の交信間隔よりも短い間隔で、実際の 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。  
 (三菱電機製のエンジニアリングツールを使用した場合、選択局情報取得の交信間隔は約 5 秒です。)  
 この場合、診断画面の LED 表示が実際の LED 状態と異なります。

(例) SD、RD など高速に点灯/消灯を繰り返す LED は、選択局情報取得の交信間隔よりも短い間隔で実際の LED 状態が変化します。そのため、診断画面の LED 表示が実際の LED 状態と異なります。

## (10) gulR\_IN32\_SetSlmpError\_Response

|        |   |                |  |     |
|--------|---|----------------|--|-----|
| 機能     | SLMP エラー応答フレーム作成処理  |                |  |     |
| 呼び出し形式 | ULONG gulR_IN32_SetSlmpError_Response(VOID* pvSendFrame, const VOID* pvReceivedData, const UCHAR* puchSA, const USHORT usFinCode)   |                |  |     |
| 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  |
|        | const VOID*   | pvReceivedData | 受信データ格納エリアのアドレス  | 入力  |
|        | const UCHAR*  | puchSA         | 送信元ノード MAC アドレス  | 入力  |
|        | const USHORT  | usFinCode      | 終了コード<br>0x0000 : 正常終了<br>0x0001~0xFFFF : エラーコード (ユーザ定義) | 入力  |
| 戻り値    | 送信データサイズ (DCS/FCS を除く)  |                |  |     |
| 説明     | <p>SLMP コマンドのエラー応答フレームを作成します。</p> <p>終了コードは、クライアントが送信した要求フレームに対して、サーバがエラーコードを設定します。</p> <p>① 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。</p> <p>② 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 (ユーザ定義) を設定してください。</p> |                |  |     |

## (11) gulR\_IN32\_SetContactTest\_Response

|        |  |                |                 |     |
|--------|--|----------------|-----------------|-----|
| 機能     | 交信テスト応答フレーム作成処理  |                |                 |     |
| 呼び出し形式 | ULONG gulR_IN32_SetContactTest_Response (VOID* pvSendFrame, const VOID* pvReceivedData, const UCHAR* puchSA) |                |                 |     |
| 引数     | 型名   | 変数名            | 内容              | 入出力 |
|        | VOID*  | pvSendFrame    | 送信フレームのアドレス     | 出力  |
|        | const VOID*  | pvReceivedData | 受信データ格納エリアのアドレス | 入力  |
|        | const UCHAR*   | puchSA         | 送信元ノード MAC アドレス | 入力  |
| 戻り値    | 送信データサイズ (DCS/FCS を除く)   |                |                 |     |
| 説明     | 交信テストの応答フレームを作成します。  |                |                 |     |

## (12) gulR\_IN32\_SetCableTest\_Response

|        |  |                |   |     |
|--------|--|----------------|---|-----|
| 機能     | ケーブルテスト応答フレーム作成処理  |                |   |     |
| 呼び出し形式 | ULONG gulR_IN32_SetCableTest_Response (VOID* pvSendFrame, const VOID* pvReceivedData, const UCHAR* puchSA, const USER_CABLETEST_RESULT_T* pstTestResult) |                |   |     |
| 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  |
|        | const VOID*  | pvReceivedData | 受信データ格納エリアのアドレス   | 入力  |
|        | const UCHAR*   | puchSA         | 送信元ノード MAC アドレス   | 入力  |
|        | const USER_CABLETEST_RESULT_T*   | pstTestResult  | 実施結果<br>【ポート数】<br>gulR_IN32U_MAX_PORT_NUMBER :<br>自局のポート数<br>【ケーブルテスト実施結果】<br>USER_CABLE_TEST_OK(0) :<br>ケーブル正常<br>USER_CABLE_TEST_NG(2) :<br>ケーブル断線、または未接続 | 入力  |
| 戻り値    | 送信データサイズ (DCS/FCS を除く)   |                |   |     |
| 説明     | ケーブルテストの応答フレームを作成します。<br>実施結果にはポート数とケーブルテスト実施結果を設定してください。  |                |   |     |

gulR\_IN32\_SetCableTest\_Response の引数

USER\_CABLETEST\_RESULT\_T の構成を、サンプルコードを元に下記に示します。

```

/* SLMP ケーブルテスト(応答用)フレームフォーマット*/
typedef struct _USER_CABLETEST_RESULT_TAG {
    USHORT usPortNum; /* ポート数 */
    USHORT auchPortResult[USER_CABLE_TEST_RESULT_MAX]; /* 実施結果 */
} USER_CABLETEST_RESULT_T;

```

## (13) gulR\_IN32\_SetMemRead\_Response

|        |  |                |                 |     |
|--------|--|----------------|-----------------|-----|
| 機能     | SLMP メモリ読出し応答フレーム作成処理  |                |                 |     |
| 呼び出し形式 | ULONG gulR_IN32_SetMemRead_Response (VOID* pvSendFrame, VOID* pvBufferMemory, const VOID* pvReceivedData, const UCHAR* puchSA) |                |                 |     |
| 引数     | 型名   | 変数名            | 内容              | 入出力 |
|        | VOID*  | pvSendFrame    | 送信フレームのアドレス     | 出力  |
|        | VOID*  | pvBufferMemory | バッファメモリのアドレス    | 出力  |
|        | const VOID*  | pvReceivedData | 受信データ格納エリアのアドレス | 入力  |
|        | const UCHAR*   | puchSA         | 送信元ノード MAC アドレス | 入力  |
| 戻り値    | 送信データサイズ (DCS/FCS を除く)   |                |                 |     |
| 説明     | SLMP メモリ読出しの応答フレームを作成します。  |                |                 |     |

## (14) gulR\_IN32\_SetMemWrite\_Response

|        |   |                |                 |     |
|--------|---|----------------|-----------------|-----|
| 機能     | SLMP メモリ書き込み応答フレーム作成処理  |                |                 |     |
| 呼び出し形式 | ULONG gulR_IN32_SetMemWrite_Response (VOID* pvSendFrame, const VOID* pvReceivedData, const UCHAR* puchSA) |                |                 |     |
| 引数     | 型名  | 変数名            | 内容              | 入出力 |
|        | VOID*   | pvSendFrame    | 送信フレームのアドレス     | 出力  |
|        | const VOID*   | pvReceivedData | 受信データ格納エリアのアドレス | 入力  |
|        | const UCHAR*  | puchSA         | 送信元ノード MAC アドレス | 入力  |
| 戻り値    | 送信データサイズ (DCS/FCS を除く)  |                |                 |     |
| 説明     | SLMP メモリ書き込みの応答フレームを作成します。  |                |                 |     |

## 6.4.12 割込み

## (1) gerR\_IN32\_DisableInterrupt

|        |   |     |    |     |
|--------|---|-----|----|-----|
| 機能     | 割込み禁止                                     |     |    |     |
| 呼び出し形式 | ERRCODE gerR_IN32_DisableInterrupt (VOID) |     |    |     |
| 引数     | 型名  | 変数名 | 内容 | 入出力 |
|        | なし  |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了                          |     |    |     |
| 説明     | 割込み禁止                                     |     |    |     |

## (2) gerR\_IN32\_EnableInterrupt

|        |  |     |    |     |
|--------|--|-----|----|-----|
| 機能     | 割込み許可                                    |     |    |     |
| 呼び出し形式 | ERRCODE gerR_IN32_EnableInterrupt (VOID) |     |    |     |
| 引数     | 型名                                       | 変数名 | 内容 | 入出力 |
|        | なし                                       |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了                         |     |    |     |
| 説明     | 割込み許可                                    |     |    |     |

## 6.4.13 ハードウェアテスト

## (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<br>R_IN32_IEEE_MODE1(1) : MODE1<br>R_IN32_IEEE_MODE2(2) : MODE2<br>R_IN32_IEEE_MODE3(3) : MODE3<br>R_IN32_IEEE_MODE4(4) : MODE4<br>R_IN32_IEEE_END(5) : テスト終了 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了<br>R_IN32_ERR : 異常終了  |        |  |     |
| 説明     | <p>引数の IEEE802.3ab コンプライアンステストモードに従い、テストモード用の波形出力を PHY に設定します。</p> <p>本関数内で、gerR_IN32R_IEEEtest (「6.5.2 R-IN32M3-CL ドライバターゲット依存関数の作成」を参照) を呼び出しますので、ご使用の PHY の仕様に合わせて gerR_IN32R_IEEEtest をカスタマイズしてください。</p> <p>※ : 本関数の戻り値が R_IN32_OK 以外の場合、ベンダー作成の以下の関数を起動しますので、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。</p> <p>gR_IN32_CallbackFatalError</p> |        |  |     |

## (2) gerR\_IN32\_InitializeLoopBackTest

|        |  |     |    |     |
|--------|--|-----|----|-----|
| 機能     | 内部折り返し／外部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 |
|        | なし   |     |    |     |
| 戻り値    | R_IN32_OK : 正常終了<br>R_IN32_ERR : 異常終了  |     |    |     |
| 説明     | <p>内部折り返し／外部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。</p> <p>※ : 本関数の戻り値が R_IN32_OK 以外の場合、ベンダー作成の以下の関数を起動しますので、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。</p> <p>gR_IN32_CallbackFatalError</p> |     |    |     |

## (3) gerR\_IN32\_InternalLoopBackTest

|        |   |        |  |     |
|--------|---|--------|--|-----|
| 機能     | 内部折り返し通信テスト   |        |  |     |
| 呼び出し形式 | ERRCODE gerR_IN32_InternalLoopBackTest (ULONG ulPort)   |        |  |     |
| 引数     | 型名  | 変数名    | 内容   | 入出力 |
|        | ULONG   | ulPort | テスト対象ポート<br>R_IN32_PORT1(0) : ポート 1<br>R_IN32_PORT2(1) : ポート 2 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了<br>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、内部折り返しで受信した結果を確認します。<br>※ : 本関数の戻り値が R_IN32_OK 以外の場合、ベンダー作成の以下の関数を起動しますので、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。<br>gR_IN32_CallbackFatalError |        |  |     |

## (4) gerR\_IN32\_ExternalLoopBackTest

|        |  |        |  |     |
|--------|--|--------|--|-----|
| 機能     | 外部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 | テスト対象ポート<br>R_IN32_PORT1(0) : ポート 1<br>R_IN32_PORT2(1) : ポート 2 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了<br>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。<br><br>本テストを実施する場合、ポート 1 とポート 2 間を Ethernet ケーブルで接続してください。<br><br>※ : 本関数の戻り値が R_IN32_OK 以外の場合、ベンダー作成の以下の関数を起動しますので、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。<br>gR_IN32_CallbackFatalError |        |  |     |

## 6.5 R-IN32M3-CL ドライバターゲット依存関数のカスタマイズ

### 6.5.1 ヘッダファイルの変更

ヘッダファイル“R\_IN32M3Function.h”に定義している各項目をベンダーシステムの環境にあわせて変更してください。

#### (1) R-IN32M3-CL アドレス設定

##### ① R-IN32M3-CL 先頭アドレス

R-IN32M3-CL ドライバが R-IN32M3-CL をアクセスするためのアドレスを設定します。

```
#define R_IN32_BASE_ADR    0x4010000    /* R-IN32M3-CL 先頭アドレス */
```

#### (2) PHY リセット設定

初期化時に PHY リセットを行うための設定を定義します。

##### ① PHY リセットアサート時間設定

R-IN32M3-CL ドライバが PHY リセット信号をアサートする時間を  $\mu\text{s}$  単位で設定します。  
アサートする時間は、ご使用の PHY により異なります。ご使用の PHY のマニュアルをご確認ください。

```
#define R_IN32_WAITUS_PHYRESET_ASSERT    10000UL    /* PHY リセットアサート時間*/
```

##### ② PHY リセット解除後、正常動作するまでの時間設定

R-IN32M3-CL ドライバが PHY リセット解除後に PHY が正常に動作するまでの時間を  $\mu\text{s}$  単位で設定します。

リセット解除後に PHY が正常に動作するまでの時間は、ご使用の PHY により異なります。ご使用の PHY のマニュアルをご確認ください。

```
#define R_IN32_WAITUS_PHYRESET_END    5000UL    /* PHY リセット解除後、  
正常動作するまでの時間*/
```

#### (3) トランジェントの受信バッファ個数設定

トランジェント受信バッファの個数を定義します。

R-IN32M3-CL ドライバは、R\_IN32\_TRANSIENT\_BUFFER\_NUM  $\times$  1520 バイトのエリア（メモリ）を使用します。

2 以上の値を設定してください。

```
#define R_IN32_TRANSIENT_BUFFER_NUM    (64)    /* トランジェント受信バッファ個数 */
```

## 6.5.2 R-IN32M3-CL ドライバターゲット依存関数の作成

**注意.** 「表6.14 R-IN32M3-CLドライバターゲット依存関数一覧表」に記載のターゲット依存関数は必ず実装してください。

R-IN32M3-CL ドライバターゲット依存関数は、ターゲットとなるハードウェア環境に合わせたカスタマイズが必要です。ベンダーがカスタマイズする関数の一覧を以下に示します。

表6.14 R-IN32M3-CL ドライバターゲット依存関数一覧表

| 機能分類      | 関数名                          | 関数型     | 概要                      |
|-----------|------------------------------|---------|-------------------------|
| 待ち処理      | gR_IN32R_WaitUS              | VOID    | 時間待ち                    |
| 時間計測      | gR_IN32R_StartStopwatchTimer | VOID    | 時間計測開始                  |
|           | gR_IN32R_GetElapsedTime      | VOID    | 経過時間取得                  |
| 割込み       | gR_IN32R_DisableInt          | VOID    | 割込み禁止                   |
|           | gR_IN32R_EnableInt           | VOID    | 割込み許可                   |
| ハードウェアテスト | gerR_IN32R_IEEETest          | ERRCODE | IEEE802.3ab コンプライアンステスト |

## (1) gR\_IN32R\_WaitUS

| 機能     | 時間待ち   |            |           |     |
|--------|--|------------|-----------|-----|
| 呼び出し形式 | VOID gR_IN32R_WaitUS (ULONG ulWaitTime)  |            |           |     |
| 引数     | 型名   | 変数名        | 内容        | 入出力 |
|        | ULONG  | ulWaitTime | 待ち時間 (μs) | 入力  |
| 戻り値    | なし   |            |           |     |
| 説明     | <p>引数で指定した待ち時間の経過を待ちます。</p> <p>※：R-IN32M3-CL ドライバが使用する時間待ちの最大値は 10ms (10000UL) です。使用する PHY のアサート時間が、10ms (10000UL) より長い場合、その値をカウントできるように以下の値を変更してください。</p> <pre>#define R_IN32_WAITUS_PHYRESET_ASSERT 10000UL /* PHY リセットアサート時間*/</pre> <p>(詳細は「6.5.1(2) PHYリセット設定」を参照してください。)</p> |            |           |     |

## (2) gR\_IN32R\_StartStopwatchTimer

| 機能     | 時間計測開始  |              |                  |     |
|--------|---|--------------|------------------|-----|
| 呼び出し形式 | VOID gR_IN32R_StartStopwatchTimer (R_IN32R_STOPWATCH_T *pstStopWatch, ULONG ulUnit) |              |                  |     |
| 引数     | 型名  | 変数名          | 内容               | 入出力 |
|        | R_IN32R_STOPWATCH_T*  | pstStopWatch | ストップウォッチワーク領域    | 入出力 |
|        | ULONG   | ulUnit       | 計測単位 (1 : μs 単位) | 入力  |
| 戻り値    | なし  |              |                  |     |
| 説明     | 時間計測を開始します。   |              |                  |     |

gR\_IN32R\_StartStopwatchTimer の引数

R\_IN32R\_STOPWATCH\_T の構成を、サンプルコードを元に下記に示します。

```
typedef struct _R_IN32R_STOPWATCH_TAG {
    ULONG    ulUnit;                /* 計測時間単位          */
    ULONG    ulFirstTmr1Cnt;       /* 汎用タイマ1カウンタ値(起動時)*/
    ULONG    ulLastTmr1Cnt;       /* 汎用タイマ1カウンタ値(前回値)*/
} R_IN32R_STOPWATCH_T;
```

### (3) gR\_IN32R\_GetElapsedTime

|        |   |                 |  |     |
|--------|---|-----------------|--|-----|
| 機能     | 経過時間取得  |                 |  |     |
| 呼び出し形式 | VOID gR_IN32R_GetElapsedTime (R_IN32R_STOPWATCH_T *pstStopWatch, ULONG *pulElapsedTime)   |                 |  |     |
| 引数     | 型名  | 変数名             | 内容   | 入出力 |
|        | R_IN32R_STOPWATCH_T   | *pstStopWatch   | ストップウォッチワーク領域  | 入出力 |
|        | ULONG   | *pulElapsedTime | 経過時間<br>(単位: gR_IN32R_StartStopwatchTimer<br>関数にて設定したもの) | 出力  |
| 戻り値    | なし  |                 |  |     |
| 説明     | <p>時間計測開始 gR_IN32R_StartStopwatchTimer 関数を起動後の経過時間を取得します。<br/>R-IN32M3-CL ドライバは、gR_IN32R_StartStopwatchTimer 関数と gR_IN32R_GetElapsedTime 関数でタイムアウト監視を行います。</p> <p>※: Unsigned Long(0~4294967295)をカウントできるように実装してください。<br/>タイムアウト監視が不要な場合、*pulElapsedTime に「0」(経過時間「0」μs)を設定してください。</p> |                 |  |     |

### (4) gR\_IN32R\_DisableInt

|        |  |     |    |     |
|--------|--|-----|----|-----|
| 機能     | 割込み禁止  |     |    |     |
| 呼び出し形式 | VOID gR_IN32R_DisableInt (VOID)                  |     |    |     |
| 引数     | 型名   | 変数名 | 内容 | 入出力 |
|        | なし   |     |    |     |
| 戻り値    | なし   |     |    |     |
| 説明     | <p>割込み禁止</p> <p>※: 本関数は、ダミー関数ですので無処理としてください。</p> |     |    |     |

## (5) gR\_IN32R\_EnableInt

|        |                                      |     |    |     |
|--------|--------------------------------------|-----|----|-----|
| 機能     | 割込み許可                                |     |    |     |
| 呼び出し形式 | VOID gR_IN32R_EnableInt (VOID)       |     |    |     |
| 引数     | 型名                                   | 変数名 | 内容 | 入出力 |
|        | なし                                   |     |    |     |
| 戻り値    | なし                                   |     |    |     |
| 説明     | 割込み許可<br>※：本関数は、ダミー関数ですので無処理としてください。 |     |    |     |

## (6) gerR\_IN32R\_IEEEtest

|        |  |                |   |     |
|--------|--|----------------|---|-----|
| 機能     | IEEE802.3ab コンプライアンステスト  |                |   |     |
| 呼び出し形式 | ERRCODE gerR_IN32R_IEEEtest (USHORT usIEEEtestMode)  |                |   |     |
| 引数     | 型名   | 変数名            | 内容  | 入出力 |
|        | USHORT   | usIEEEtestMode | IEEE802.3ab コンプライアンステストモード<br>R_IN32R_IEEE_MODE1(1) : MODE1<br>R_IN32R_IEEE_MODE2(2) : MODE2<br>R_IN32R_IEEE_MODE3(3) : MODE3<br>R_IN32R_IEEE_MODE4(4) : MODE4<br>R_IN32R_IEEE_END(5) : テスト終了 | 入力  |
| 戻り値    | R_IN32_OK : 正常終了<br>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従い、テストモード用の波形出力を PHY に設定します。<br>本関数は、Marvell Semiconductor 社の PHY「88E111-B2-BAB1C100」を使用することを前提としています。他の PHY を使用する場合は、PHY の仕様に合わせて本関数をカスタマイズしてください。 |                |   |     |

## 6.6 R-IN32M3-CL ドライバコールバック関数のカスタマイズ

R-IN32M3-CL ドライバコールバック関数の内部処理は、ベンダーのカスタマイズが必要です。R-IN32M3-CL ドライバが起動するコールバック関数一覧を以下に示します。

表6.15 R-IN32M3-CL ドライバが使用するコールバック関数一覧表

| 機能分類    | 関数名  | 関数型     | 概要                     |
|---------|--|---------|------------------------|
| エラー処理   | gR_IN32_CallbackFatalError                 | VOID    | R-IN32M3-CLのFatalエラー取得 |
| 自局状態取得  | gerR_IN32_CallbackCommandFromMaster        | ERRCODE | マスタ局からのコマンド取得          |
| 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        |
| 送受信     | gerR_IN32_CallbackTransientSendingComplete | ERRCODE | トランジェント送信完了状態取得        |

### (1) gR\_IN32\_CallbackFatalError

| 機能     | R-IN32M3-CL 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<br>(エラー発生時関数のアドレス) | 入力  |
| 戻り値    | なし  |             |                                |     |
| 説明     | R-IN32M3-CL の Fatal エラーを取得します。<br>R-IN32M3-CL ドライバは、R-IN32M3-CL の Fatal エラーを検出したとき本関数を起動します。<br>関数内部はベンダーが任意に実装します。 |             |                                |     |

表6.16 gR\_IN32\_CallbackFatalError 関数の Fatal エラーコード一覧

| Fatal エラーコード<br>(ulErrorCode) | Fatal エラー情報<br>(ulErrorInfo)                            | Fatal エラー内容 | 処置   |
|-------------------------------|---|-------------|--|
| D529                          | ドライバ内起動元関数<br>gerR_IN32D_ClearTxRxRAM関数のアドレス            | 通信LSI異常     | <ul style="list-style-type: none"> <li>・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。</li> <li>・ユニットの単体テストを実行してください。再度異常になったときはユニットのハードウェアに異常が考えられます。</li> </ul> |
| D52A                          | ドライバ内起動元関数<br>erR_IN32D_MDIO_WaitCommandComplete関数のアドレス | 通信LSI異常     |  |
| D52B                          | ドライバ内起動元関数<br>erR_IN32D_ResetMAC関数のアドレス                 | 通信LSI異常     |  |
| D52C                          | ドライバ内起動元関数<br>gerR_IN32D_StartRing関数のアドレス               | 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 | <p>マスタ局からのコマンド状態<br/>ulCommand</p> <p>Bit0 : サイクリック伝送停止指示 (ノード番号範囲外)<br/>1 : 停止指示</p> <p>Bit1 : サイクリック伝送停止指示 (予約ノード設定)<br/>1 : 停止指示</p> <p>Bit2 : サイクリック伝送停止指示 (マスタ局指示)<br/>1 : 停止指示</p> <p>Bit3 : サイクリック伝送停止指示 (ノード番号重複)<br/>1 : 停止指示</p> <p>Bit15~4 : 予約</p> <p>Bit16 : ノード種別不正 (マスタ局が指定したノード種別と自局のノード種別が不一致)<br/>1 : ノード種別不正</p> <p>Bit17 : 指定サイズ不正 (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 関数にて設定したサイズ) より大きかった)<br/>1 : 指定サイズ不正</p> <p>Bit31~18 : 予約</p> | 入力  |
| 戻り値    | R_IN32_OK : 正常終了  |            |   |     |
| 説明     | <p>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。</p> <p>R-IN32M3-CL 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起動します。関数内部はベンダーが任意に実装します。</p> |            |   |     |

## (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    | <p>R_IN32_OK : 正常終了</p> <p>R_IN32_ERR : 異常終了</p>  |             |                |     |
| 説明     | <p>受信したトランジェントフレームを取得します。</p> <p>R-IN32M3-CL 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起動します。関数内部はベンダーが任意に実装します。</p> <p>※ : 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、「ベンダー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 gerR_IN32_EnableReceiveTransient 関数にて「受信許可」としてください。</p> |             |                |     |

## (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<br>(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<br>(送信結果)<br>R_IN32_OK :<br>トランジェント送信正常完了<br>R_IN32_ERR :<br>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。<br><br>R-IN32M3-CL 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起動します。<br>関数内部はベンダーが任意に実装します。 |               |  |     |

## 7. 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インテリジェントデバイス局は、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7.1 システム領域のビット割付け（例）」のように割付けてください。

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<sub>r</sub>、RW<sub>w</sub>）に定義する場合は、RXをRW<sub>r</sub>、RYをRW<sub>w</sub>に置き換えてください。

表7.1 システム領域のビット割付け（例）

|        | ビット     | 名称              | ビット     | 名称              |
|--------|---------|-----------------|---------|-----------------|
| システム領域 | 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) |                 |
|        | RX(S+C) | リザーブ            | RY(S+C) | リザーブ            |
|        | 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**

なお、リンクデバイスの一部をシステム領域として定義した場合、Control & Communication システムプロファイルに、リンクデバイスの定義情報を記載してください。

## 7.1 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。

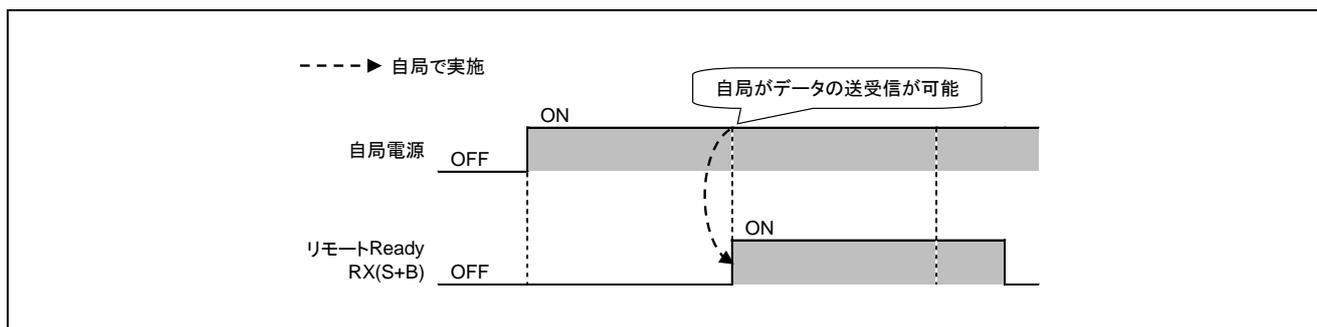


図7.1 リモート Ready タイミング図

## (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。

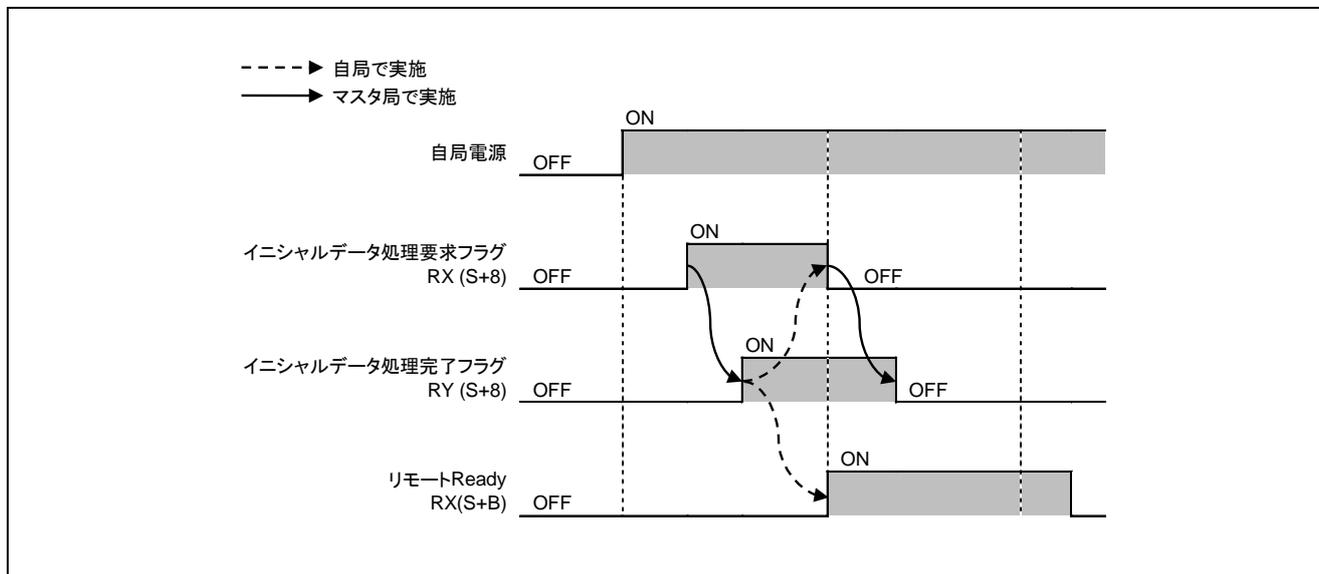


図7.2 イニシャルデータ処理要求/完了フラグ タイミング図

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

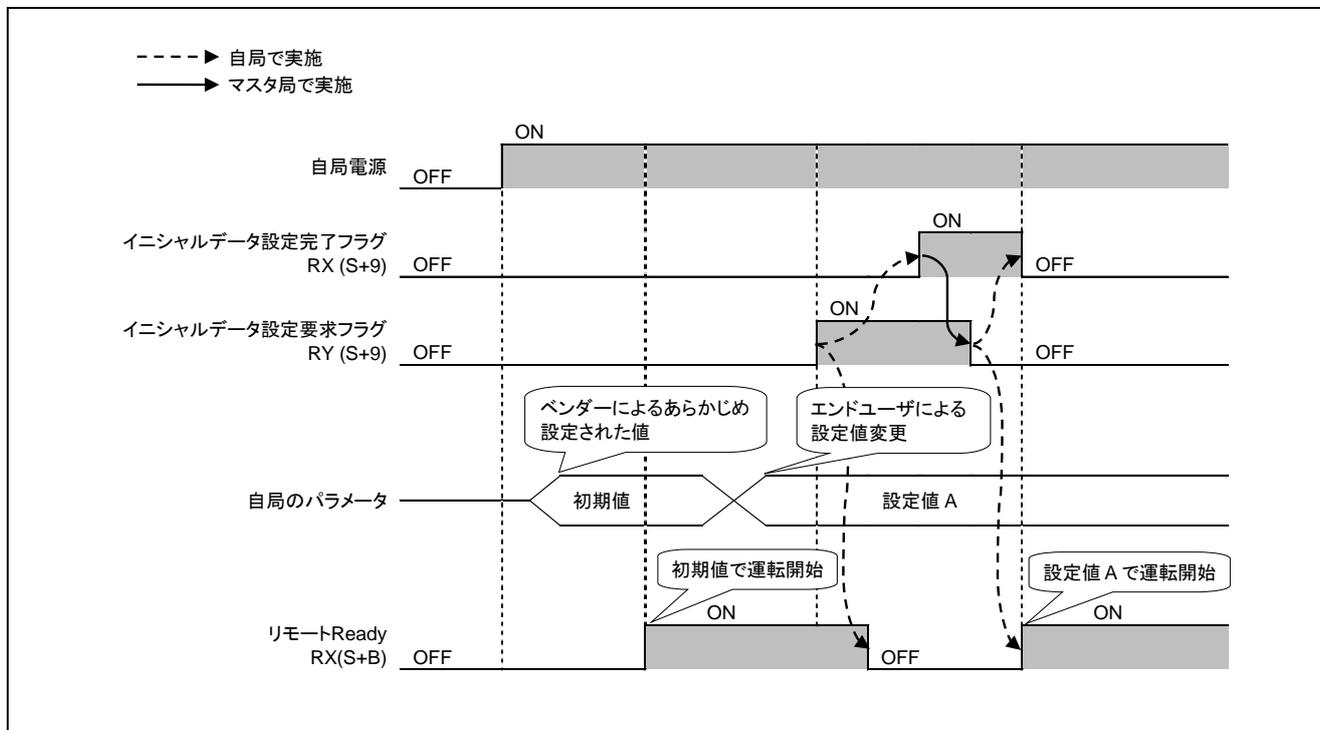


図7.3 イニシャルデータ設定完了/要求フラグ タイミング図

(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。

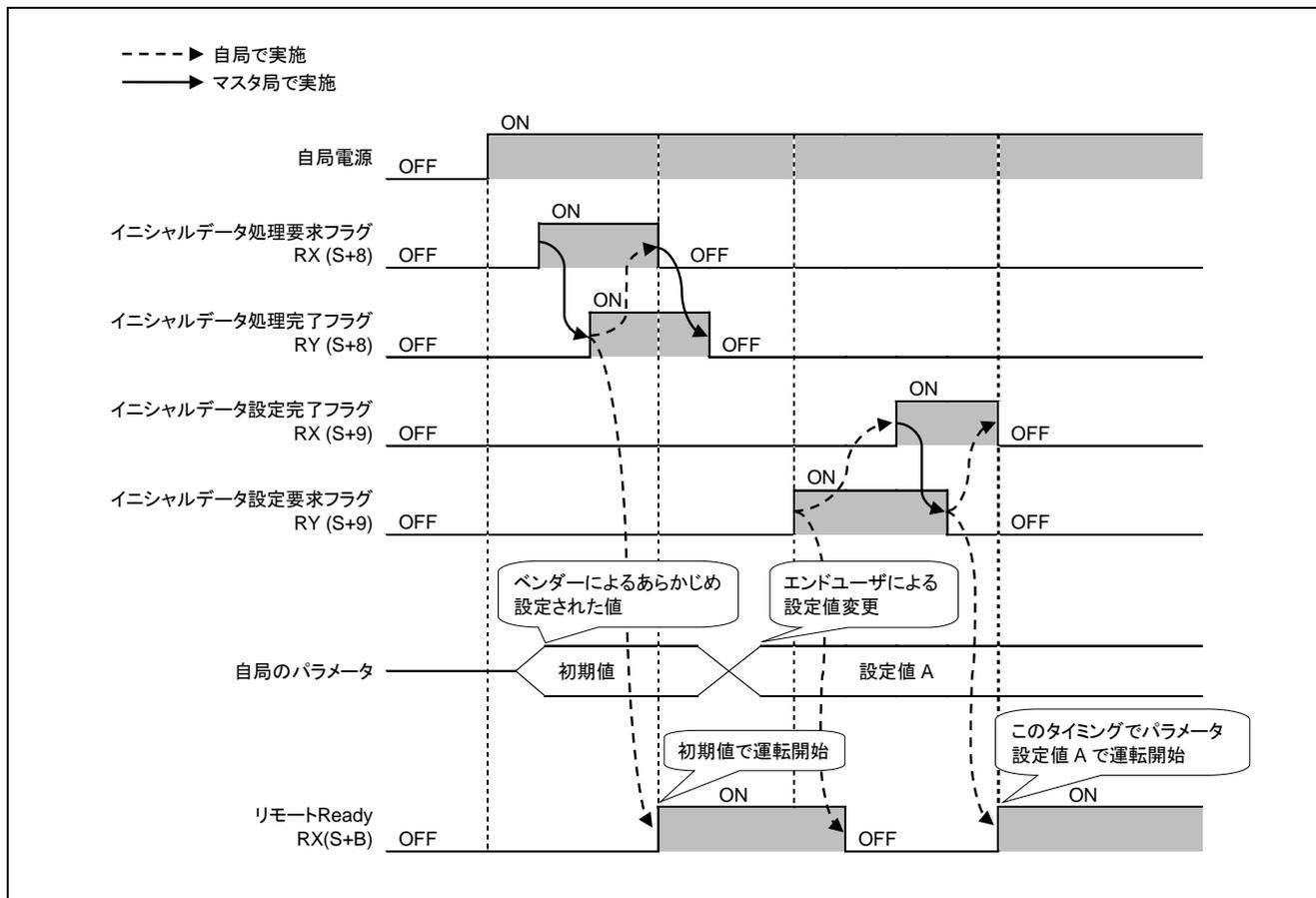


図7.4 イニシャルデータ処理および設定 タイミング図

## (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。

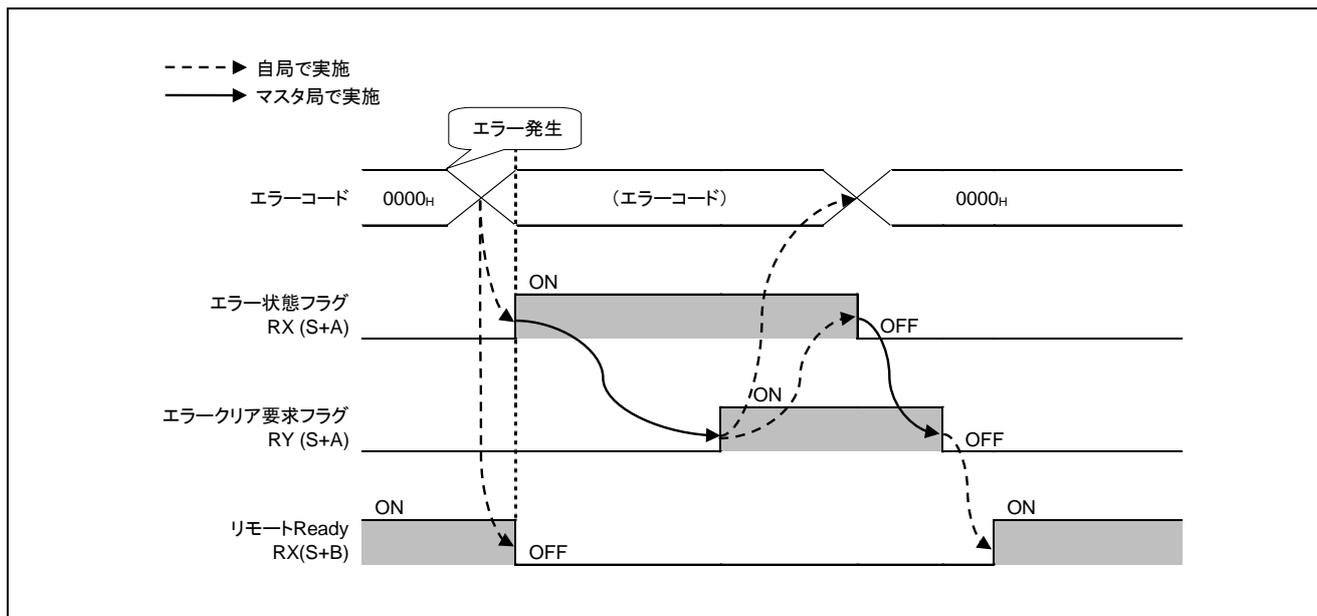


図7.5 エラー状態／リセット要求フラグ タイミング図

## (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。

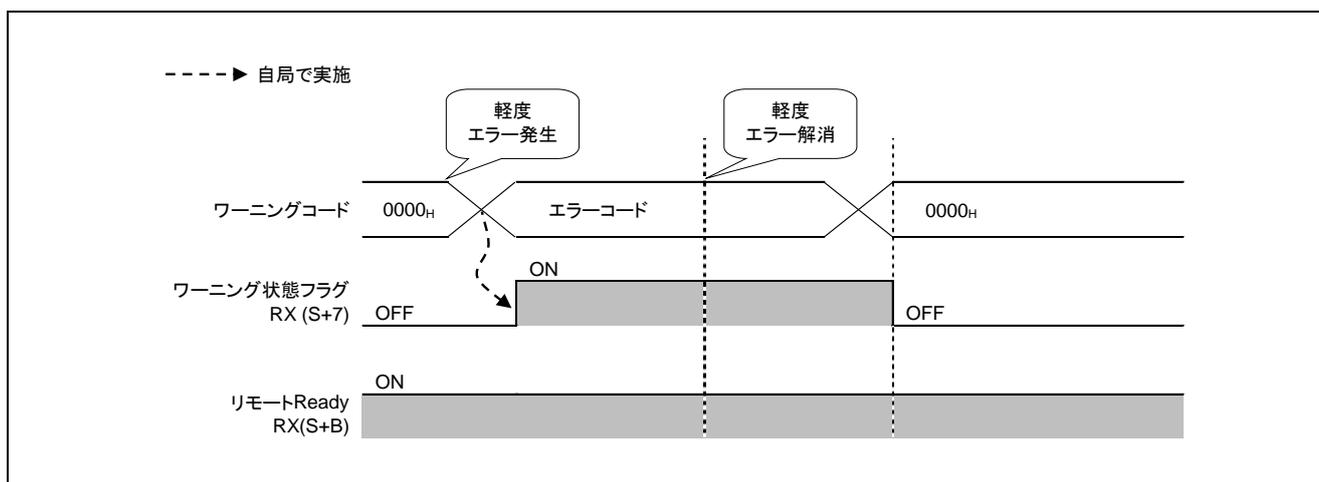


図7.6 ワーニング状態フラグ タイミング図

|      |   |
|------|---|
| 改訂記録 | R-IN32M3 シリーズ CC-Link IE Field インテリジェントデバイス局編 |
|------|---|

| Rev. | 発行日        | 改訂内容   |  |
|------|------------|--------|--|
|      |            | ページ    | ポイント   |
| 1.00 | 2013.07.26 | —      | 初版発行   |
| 2.00 | 2014.12.25 | 138    | 「4.6.1 ヘッダファイルの変更」 ヘッダファイル名と R-IN32M3-CL の先頭アドレスの変更  |
| 5.00 | 2018.1.31  | —      | 全面改訂<br>◇全体の変更<br>・章構成の変更<br>章追加 … 1、2、7 章<br>章削除 … 変更前の 5 章<br>内容の変更あり … 4、5、6 章（変更前の 2、3、4 章）<br>内容の変更なし … 3 章（変更前の 5 章）<br>・書式、文言の統一（内容の変更なし）<br>・記載方法、表現、図を変更（内容の変更なし）   |
|      |            | 21-24  | ◇「4. 状態表示機能」の変更内容<br>・構成の変更<br>4.1 節の注意書きを詳細化し、4.1.1 項を作成<br>・4.1 節、4.2.1 項、4.3 節の記載内容を詳細化   |
|      |            | 25-76  | ◇「5. CC-Link IE フィールドネットワークのデータ通信方式」の変更内容<br>・構成の変更<br>・5.2 節の一部を詳細化し、5.2.1 項、5.2.3 項、5.2.4 項を作成<br>・5.2.2 項追加（Transient1 要求の送信手順の記載を追加）<br>・5.3.5 項追加（SLMP フレームフォーマットの記載を追加）<br>・5.4 節追加（MyStatus に関する記載を追加）<br>・5.3.1 項の内容を詳細化し、5.3.1 項と 5.3.2 項に分離<br>・記載内容を詳細化<br>・Transient1 のオプション情報取得の記載を追加、SLMP、MyStatus の記載を追加<br>・フレーム名称を変更（変更前の Transient1 フレームと SLMP の区別のため）<br>“Transient1” → “CC-Link IE フィールド固有トランジェント伝送”<br>“Transient2” → “CC-Link 互換トランジェント伝送” |
|      |            | 77-80  | ◇「6.1 開発手順」の変更内容<br>・構成の変更<br>変更前の 3 節を削除し、6.1.1 項を作成<br>・記載内容を詳細化   |
|      |            | 81-140 | ◇「6.2 サンプルフローチャート」の変更内容<br>・構成の変更<br>・6.2.27 項追加（オプション情報取得要求フレーム受信対応処理を追加）<br>・6.2.28 項追加（選択局情報取得要求フレーム受信対応処理を追加）<br>・6.2.29 項追加（交信テスト要求フレーム受信対応処理を追加）<br>・6.2.30 項追加（ケーブルテスト要求フレーム受信対応処理を追加）<br>・6.2.39 項追加（SLMP メモリ読出し要求フレーム受信対応処理を追加）<br>・6.2.40 項追加（SLMP メモリ書込み要求フレーム受信対応処理を追加）<br>・6.2.41 項追加（SLMP メモリ読出し要求フレーム作成処理を追加）<br>・6.2.42 項追加（Transient1 要求送信分割判定処理を追加）<br>・6.2.43 項追加（Transient1 要求フレーム作成処理を追加）<br>・6.2.44 項追加（SLMP メモリ読出し応答受信処理を追加）              |
|      |            |        |  |

|         |   |
|---------|---|
|         | <ul style="list-style-type: none"> <li>・ 上記追加処理のサンプルチャート一覧への追加</li> <li>・ “概略フロー” から “メイン処理” に処理名称を変更</li> <li>・ 自局エラーの記載のある各項で、自局エラーの例を注意書きとして追加</li> <li>・ 6.2.2 項に gblUserMACAddressTableRequest についての注意書きを追加</li> <li>・ 6.2.9 項に Hold/Clear 処理の記載と注意書きを追加</li> <li>・ 6.2.12 項に Hold/Clear 処理の記載と注意書きを追加</li> <li>・ 6.2.17 項に分割送信についての記述を追加</li> <li>・ 6.2.18 項にマスタ局からの SLMP 要求受信についての記述を追加</li> <li>・ 6.2.46 項に折り返し通信テストによるトラブルシュートについての記載を追加</li> </ul>  |
| 141-142 | <ul style="list-style-type: none"> <li>◇ 「6.3 R-IN32M3-CL ドライバインタフェース関数一覧」の変更内容</li> <li>・ R-IN32M3-CL ドライバインタフェース関数一覧に以下の関数を追加<br/>gerR_IN32_GetPortAvailable、gulR_IN32_SetOptionInfo_Response、<br/>gulR_IN32_SetSelectInfo_Response、gulR_IN32_SetSmpError_Response、<br/>gulR_IN32_SetContactTest_Response、gulR_IN32_SetCableTest_Response、<br/>gulR_IN32_SetMemRead_Response、gulR_IN32_SetMemWrite_Response</li> </ul>  |
| 143-190 | <ul style="list-style-type: none"> <li>◇ 「6.4 R-IN32M3-CL ドライバインタフェース関数詳細」の変更内容</li> <li>・ 構成の変更 <ul style="list-style-type: none"> <li>・ 6.4.6 項(8)追加 (gerR_IN32_GetPortAvailable 関数追加)</li> <li>・ 6.4.11 項(8)追加 (gulR_IN32_SetOptionInfo_Response 関数追加)</li> <li>・ 6.4.11(9)追加 (gulR_IN32_SetSelectInfo_Response 関数追加)</li> <li>・ 6.4.11(10)追加 (gulR_IN32_SetSmpError_Response 関数追加)</li> <li>・ 6.4.11(11)追加 (gulR_IN32_SetContactTest_Response 関数追加)</li> <li>・ 6.4.11(12)追加 ( gulR_IN32_SetCableTest_Response 関数追加)</li> <li>・ 6.4.11(13)追加 (gulR_IN32_SetMemRead_Response 関数追加)</li> <li>・ 6.4.11(14)追加 (gulR_IN32_SetMemWrite_Response 関数追加)</li> </ul> </li> <li>・ 6.4.1 項(2)の説明欄の修正</li> <li>・ 6.4.1 項(2)の引数 R_IN32_UNITINFO_T に以下の構造体メンバを追加・削除<br/>追加 : usHwVersion、usDeviceVersion<br/>削除 : blNodeAndNetworkNumberFromMasterPermission</li> <li>・ 6.4.1 項(2) A)に上記メンバの初期設定の説明項を追加 ((n)(o)) ・ 削除</li> <li>・ 6.4.1 項(2) A)の以下の初期設定の記載を変更 <ul style="list-style-type: none"> <li>・ (e) … 自局ポート数の設定を “2 または 1” に変更</li> <li>・ (f) … トークン保持時間を 23μsec 固定に変更</li> <li>・ (h) … 説明の追加</li> <li>・ CSP+における機器バージョンについての注意書きを追加</li> <li>・ ネットワークとコントローラの補足を追加</li> <li>・ 機器バージョンの補足を追加</li> </ul> </li> <li>・ 6.4.1 項(2)の引数 R_IN32_UNITINIT_T の以下の構造体メンバの追加・修正<br/>追加 : ulOptionSupport、ulSmpSupport、ulSmpDiagnosisSupport<br/>修正 : ulErrorStatus、ulErrorCode</li> <li>・ 6.4.1 項(2) B)に上記メンバの初期設定の説明項を追加 ((l)(m)(n)) ・ 修正 ((i)(j))</li> <li>・ 6.4.1 項(2) B)の以下の初期設定の記載を変更 <ul style="list-style-type: none"> <li>・ (f) … 設定値を “R_IN32_TRUE” 固定に変更</li> <li>・ (g) … 記載内容を詳細化</li> </ul> </li> <li>・ 6.4.1 項(3)の説明欄の記載内容を詳細化</li> <li>・ 6.4.13 項(1)の説明欄に説明追加</li> </ul> |
| 191-194 | <ul style="list-style-type: none"> <li>◇ 「6.5 R-IN32M3-CL ドライバターゲット依存関数のカスタマイズ」の変更内容</li> <li>・ 6.5.2 項(6) 説明欄の記載を詳細化</li> </ul>  |

|  |  |         |   |
|--|--|---------|---|
|  |  | 195-197 | ◇ 「6.6 R-IN32M3-CL ドライバコールバック関数のカスタマイズ」の変更内容<br>・ R-IN32M3-CL ドライバが使用するコールバック関数一覧の以下の関数を削除<br>gerR_IN32_CallbackNodeAndNetworkNumber<br>・ 上記関数の説明項を削除 |
|--|--|---------|---|

---

R-IN32M3シリーズ ユーザーズマニュアル  
(CC-Link IE Fieldインテリジェントデバイス局編)

発行年月日 2013年7月26日 Rev.1.00  
2018年1月31日 Rev.5.00

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)

---



ルネサスエレクトロニクス株式会社

営業お問合せ窓口

<http://www.renesas.com>

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

ルネサス エレクトロニクス株式会社 〒135-0061 東京都江東区豊洲3-2-24 (豊洲フォレシア)

技術的なお問合せおよび資料のご請求は下記へどうぞ。  
総合お問合せ窓口：<https://www.renesas.com/contact/>

R-IN32M3 シリーズ  
ユーザーズマニュアル  
(CC-Link IE Field インテリジェントデバイス局



ルネサスエレクトロニクス株式会社

R18UZ0014JJ0500