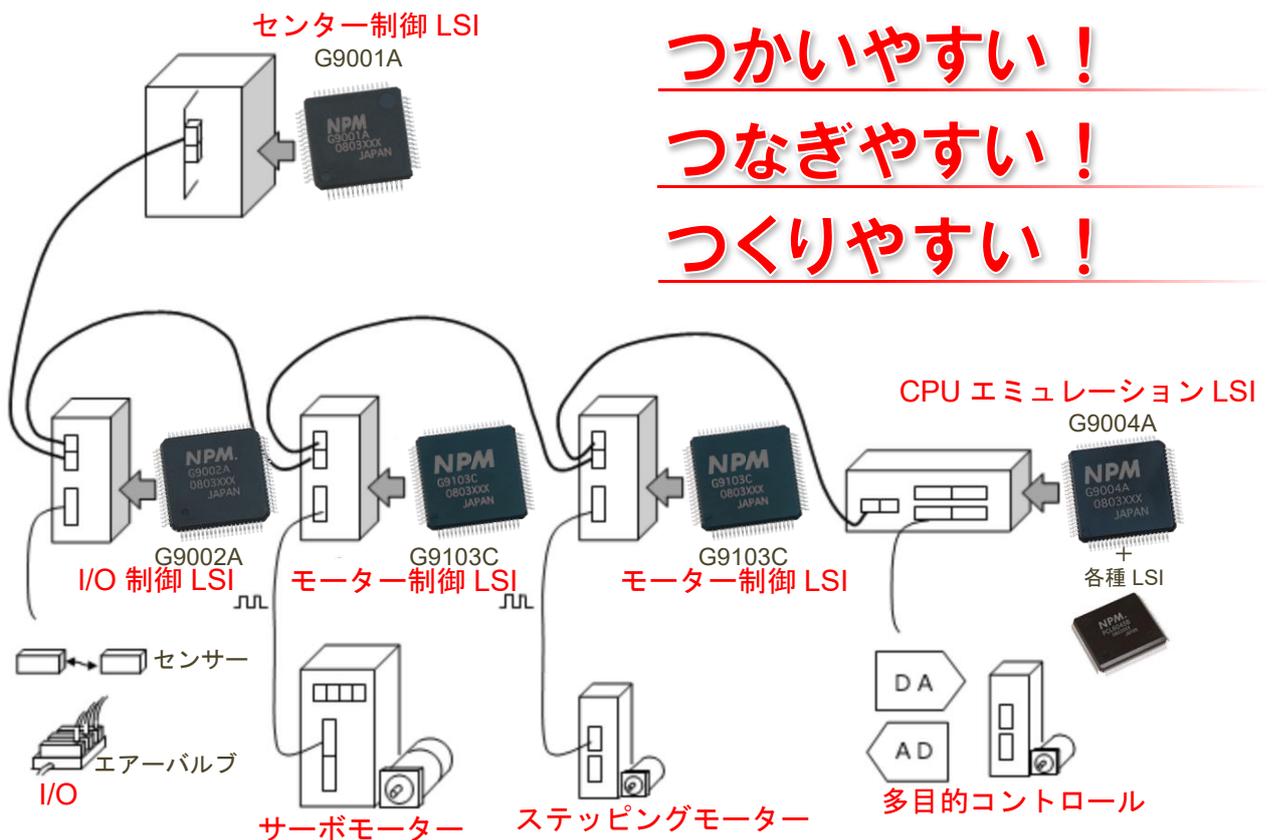


## G9000 シリーズ

Motionnet<sup>®</sup> にできること

G9000 シリーズの機能と性能について



# 目次

1. はじめに .....	1
1.1 本書の取扱い .....	1
1.1.1 記号説明 .....	1
1.1.1.1 負傷レベル .....	1
1.1.1.2 危険レベル .....	1
1.1.1.3 警告図記号 .....	2
1.1.2 専門用語 .....	3
2. Motionnet の基本 .....	4
2.1 G9000 シリーズには、どのような製品があり、その機能は？ .....	4
2.1.1 G9001A : センター制御 LSI .....	4
2.1.2 G9002A : I/O 制御 LSI .....	4
2.1.2.1 入出力ポートの設定方法 .....	4
2.1.3 G9205A : I/O 制御 LSI .....	5
2.1.3.1 センター制御ノーマルモード .....	5
2.1.3.2 センター制御ポート間モード .....	5
2.1.3.3 センターレスポート間モード .....	5
2.1.3.4 入出力ポートの設定方法 (センター制御ノーマルモード、偶数アドレス LSI) .....	6
2.1.3.5 入出力ポートの設定方法 (奇数アドレス LSI) .....	6
2.1.4 G9103C : モーター制御 LSI .....	7
2.1.5 G9004A : CPU エミュレーション LSI .....	8
2.1.5.1 CPU エミュレーションモード .....	8
2.1.5.2 メッセージ通信モード .....	8
2.1.6 G9006 : 機能拡張 LSI .....	9
2.1.6.1 ローカル動作モード .....	9
2.1.6.2 モニター動作モード .....	9
2.1.6.3 I/O の拡張 .....	10
2.1.6.4 状態モニター .....	10
2.1.6.5 CPU 間通信 .....	10
2.1.7 G9H50A : HUB 機能 LSI .....	11
2.1.7.1 HUB 機能 .....	11
2.2 どのように構成すればよいのか？ .....	12
2.3 接続できるローカル制御 LSI はいくつまでか？ .....	12
2.4 通信はどのように行われるのか？ .....	13
2.4.1 Motionnet の通信プロトコル概要 .....	13
2.4.1.1 通信の最小単位とポーリング単位 .....	13
2.4.1.2 ポーリング単位の種類 .....	14
2.4.1.3 サイクリック通信 .....	15
2.4.1.4 トランジェント通信 .....	16
2.5 サイクリック通信で送られるのはどのようなデータか？ .....	17
2.6 ステータスとは何か？ .....	18

2.7	通信周期はどれくらいの時間か？	18
2.8	通信できる距離は？	19
2.9	デバイス番号は連続しなくてもよいか？	19
2.10	RS485 とはなにか？	20
2.11	パルストランス（トランス）が必要な理由は？	20
2.12	終端抵抗の役割は？	21
2.13	G9000 シリーズを制御できる CPU の種類は？	22
2.13.1	G9001A に接続可能な CPU タイプ	22
2.13.2	G9004A に接続可能な CPU タイプ	22
2.13.3	G9103C、G9006 に接続可能な CPU タイプ	22
2.14	センター制御 LSI にシリアル入力端子が 2 本ある理由は？	23
3.	Motionnet の操作	24
3.1	Motionnet はどれだけ簡単に使えるか？	24
3.2	データ通信のやり方は？	25
3.3	送信用 FIFO へのデータの書き込み方法は？	26
3.4	受信用 FIFO からのデータの読み出し方法は？	26
3.5	G9002A の入力ポートのデータ取得方法は？	27
3.6	G9002A の出力ポートへのデータ出力方法は？	28
3.7	G9103C、G9004A のステータス情報はどのように参照するか？	29
3.8	通信ラインに接続されているローカル制御 LSI 種類の確認方法	30
3.9	I/O 制御 LSI の入力ポートの変化を割り込み機能で認識するには？	31
3.9.1	入力値変化を監視するローカル制御 LSI の指定	31
3.9.2	入力値変化のあったローカル制御 LSI の特定	32
3.10	G9103C、G9004A のステータス状態の変化を認識するには？	33
3.10.1	ステータス変化を監視するローカル制御 LSI の指定	33
3.10.2	ポート 0 の監視上の注意（変化割り込みの使用時）	34
3.11	G9103C を動作させるには？	36
3.12	G9004A とのデータのやりとりの方法は？	37
3.12.1	G9004A のステータスに関する注意	37
3.12.1	メッセージ通信	37
3.12.1.1	G9001A から G9004A 側 CPU へデータを渡す	37
3.12.1.2	G9001A から G9004A 側 CPU へデータを渡し、返信データもらう	38
3.12.1.3	G9004A 側 CPU から G9001A へデータを渡す	39
3.12.2	CPU エミュレーションモード時のコマンドについて	40
3.12.3	CPU エミュレーションモード	41
3.12.3.1	連続データの書き込み	41
3.12.4	連続データの読み出し	42
3.12.5	組み合わせ処理	43
3.12.5.1	読み出しコマンド使用時に G9004A から返信されるデータ形式	44
3.12.5.2	読み出しコマンドでの返送データが 256byte を越える場合	44

3.13 I/O 制御 LSI に対してデータ通信を行うとどうなるか？	45
3.14 I/O 制御 LSI と G9001A で異なる PMD の設定は可能か？	45
3.15 G9103C の受信 FIFO 容量以上のデータを送信できるか？	46
3.16 G9004A の受信用 FIFO のデータを読み出さない場合は？	46
3.17 通信が不可能な状態になると各 LSI はどうなるか？	47
3.18 ローカル制御 LSI が機能を停止したことを認識するには？	48
3.19 稼働中に新たなローカル制御 LSI を参入させるには？	49
3.20 通信状態を外部から監視するには？	50
3.21 ウェイト制御用端子の無い CPU を使用するには？	51
3.22 アドレスバスを 2 本しか使わないで G9001A を制御する方法	52
3.23 40MHz クロック使用時の注意点は？	52
3.24 関連部品の選定基準は？	53
3.24.1 ケーブルの選定	53
3.24.2 RS485 製品の選定	53
3.24.3 パルストランスの選定	53
3.24.4 終端抵抗の抵抗値の決定	53
3.25 レベルシフトは必要ありません	53
4. 付録：用語解説	54
4.1 Motionnet 全般に関連する用語	54
4.2 G9103C に関連する用語	62

# 1. はじめに

このたびは Motionnet にご興味を持っていただき、誠にありがとうございます。  
本書は Motionnet の機能と性能をご理解いただくことを目的として記載しています。  
本書で Motionnet についての基本部分をご理解いただければ幸いです。

## 1.1 本書の取扱い

- ① 本書の全部または一部を無断で転載することは、著作権法によって禁止されています。
- ② 本書の内容については、性能や品質の向上に伴い、将来予告なく変更することがあります。
- ③ 本書の内容については、万全を期しておりますが、万一不可解な点や誤り、ならびに記載もれ等お気付きの点がありましたら、弊社営業担当へご連絡をお願いいたします。

### 1.1.1 記号説明

#### 1.1.1.1 負傷レベル

本書では、次のように負傷レベルを定義します。

- 重傷  
失明、けが、火傷、感電、骨折、中毒等後遺症が残るもの、および治療に入院や長期の通院を要するもの。
- 軽傷  
治療に入院や長期の通院が必要ないもの。(上記「重傷」以外)

#### 1.1.1.2 危険レベル

本製品は、運用者の安全を第一に考え、設計されています。しかし、製品の性質上、どうしても取除けないリスクが存在します。本書では、それらのリスクの重大性および危険性のレベルを、「危険」、「警告」および「注意」事項の3段階に分けて表示しています。表示項目をよく読み十分に理解してから、本製品の操作および保守作業を行ってください。

「危険」、「警告」および「注意」事項の表示は、危険性に関する重大性の順(危険>警告>注意)で、その内容を下記に説明します。



**危険**

「危険」項目は、本製品の運用中に、作業者が死亡または重傷に至る切迫した危険性のある場合について記述しています。



**警告**

「警告」項目は、本製品の運用中に、作業者が死亡または重傷を負う可能性のある場合について記述しています。



**注意**

「注意」項目は、本製品の運用中に、作業者が軽傷を負う可能性のある場合について記述しています。

**注意**

 (警告記号)のない「注意」項目は、作業者が負傷する恐れはないが、本製品、設備、機器等に損害や故障を引き起こすことが予想される場合について記述しています。

本書では前述の危険レベル分けのほかに、下記の表記も使用しています。

## 重 要

「重要」項目は、本製品の操作および保守作業上、特に知っておかなければならない情報や内容がある場合に記述します。

## 備 考

「備考」項目は、本製品の操作および保守作業上、役立つ情報や内容がある場合に記述します。

### 1.1.1.3 警告図記号

本書では、「危険」、「警告」、「注意」、「重要」の表記に併せて次のようなシンボル記号を付加し、その警告内容をわかりやすく表現しています。



高電圧が印可される場合があることを表します。  
安全確認を怠ったり、取扱いを誤ると感電によるショック、火傷、および死に至る危険を警告します。



表面温度が高くなる部品等があることを表します。  
取扱いを誤ると、火傷の危険があることを意味します。



取扱いを誤ると、火災を起こす可能性があることを表します。



本製品の操作およびメンテナンス作業において、行ってはいけない「禁止」事項を示します。



本製品の操作およびメンテナンス作業において、必ず行っていただく「強制」事項を示します。

## 1.1.2 専門用語

本書で使用している専門用語は、後半の「4. 付録：用語解説」を参照ください。  
本項に説明がない専門用語については、以下のウェブページをご覧ください。

<<https://www.pulsemotor.com/technology/terms/>>

(日本パルスモーター株式会社 | TOP > 技術・サポート > 用語集 日本語のみ)

## 2. Motionnet の基本

### 2.1 G9000 シリーズには、どのような製品があり、その機能は？

G9000 シリーズは、2023 年 12 月現在で、次のような 7 つのシリーズ製品があります。

型名	機能 / 用途	分類
G9001A	センター制御 LSI	センター制御 LSI
G9002A	I/O 制御 LSI (32 点)	ローカル制御 LSI
G9205A	I/O 制御 LSI (16 点)	ローカル制御 LSI
G9103C	モーター制御 LSI	ローカル制御 LSI (データ制御 LSI)
G9004A	CPU エミュレーション LSI	ローカル制御 LSI (データ制御 LSI)
G9006	機能拡張 LSI	ローカル制御 LSI
G9H50A	HUB 機能 LSI	HUB(中継)用 LSI

#### 2.1.1 G9001A : センター制御 LSI

シリアル通信に関するすべての制御を行います。Motionnet の構築にはひとつのセンター制御 LSI が必要です (注)。このセンター制御 LSI が、Motionnet に接続されているすべてのローカル制御 LSI に対して通信制御を行います。通信速度は最大で 20Mbps で、10Mbps、5Mbps、2.5Mbps の選択が可能です。CPU に接続して使用します。

注：複数のセンター制御 LSI を、同じ通信ラインに接続することはできません。  
G9205A のセンターレスモードを使用する場合、センター制御 LSI は必要ありません。

#### 2.1.2 G9002A : I/O 制御 LSI

32 ビットの入出力端子を持っています。これらの入出力端子は、8 ビット単位でひとつのポートとしてグループ分けします (つまり 4 ポートです)。出力モードに設定されたポートは、リセット時の初期状態 (High 出力 または Low 出力) の設定が、外部端子により行えます。

##### 2.1.2.1 入出力ポートの設定方法

G9002A の持つ 4 つのポートは、外部端子 (PMD2、PMD1、PMD0) の設定により、次のように入力ポート、出力ポートの設定が可能です。

PMD2	PMD1	PMD0	ポート 0	ポート 1	ポート 2	ポート 3
Low	Low	Low	出力	出力	出力	出力
Low	Low	High	入力	出力	出力	出力
Low	High	Low	入力	入力	出力	出力
Low	High	High	入力	入力	入力	出力
High	Low	Low	入力	入力	入力	入力

## 2.1.3 G9205A : I/O 制御 LSI

16 ビットの入出力端子を持っています。

これらの入出力端子は、8 ビット単位でひとつのポートとしてグループ分けします（つまり 2 ポートです）。

さらに、G9205A は次の 3 つの動作モードを持っています。

- ・センター制御ノーマルモード
- ・センター制御ポート間モード
- ・センターレスポート間モード

このなかからひとつの動作モードを選んで使用します（複数のモードを同時に使用することはできません）。

センター制御ポート間モード、または、センターレスポート間モードで使用する場合、必ず 2 つひと組のペアで使用する必要があります。

ペアとしての指定はデバイス番号で行います。デバイス番号は外部端子（DN5～DN0）の設定で行いますが、DN5～DN1 まで同じ状態に設定された G9205A が ペアとして扱われます。

ペア指定された G9205A のうち、DN0=High と設定されたほうを偶数アドレス LSI、DN0=Low と設定されたほうを奇数アドレス LSI と呼びます。

（注：ここでいう「アドレス」は、デバイス番号を指します）

### 2.1.3.1 センター制御ノーマルモード

G9002A の 2 ポート分の機能として動作します。

### 2.1.3.2 センター制御ポート間モード

ペアとなるふたつの G9205A の持つポート情報を、互いにやり取りします。

これにより、一方の G9205A の入力ポートからの入力状態を、もう一方の G9205A の出力ポートへ出力させることができます。

このときの通信ライン上の通信制御は、センター制御 LSI（G9001A）が行います。

### 2.1.3.3 センターレスポート間モード

G9001A を使用せずに、ペアとなるふたつの G9205A の持つポート情報を、互いにやり取りします。

これにより、一方の G9205A の入力ポートからの入力状態を、もう一方の G9205A の出力ポートへ出力させることができます。

このときの通信ライン上の通信制御は、各 G9205A が順番に行います。

通信ライン上には、G9205A 以外の G9000 シリーズ LSI を接続しないでください。

### 2.1.3.4 入出力ポートの設定方法 (センター制御ノーマルモード、偶数アドレス LSI)

G9205A を、センター制御ノーマルモード、もしくは、センター制御ポート間モードの偶数アドレス LSI、センターレスポート間モードの偶数アドレス LSI として使用する場合は、入出力ポートの設定方法を示します。

G9205A の持つ 2 つのポートは、外部端子 (BMD2、BMD1、BMD0) の設定により、次のように入力ポート、出力ポートの設定が可能です。

みてわかるとおり、BMD2~0 端子に対する各ポートの入出力の状態は、奇数アドレス LSI と完全に逆になります。

BMD2	BMD1	BMD0	ポート状態														
			ポート 1							ポート 0							
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1
Low	Low	Low	入力	入力	入力	入力	入力	入力	入力	入力	入力	入力	入力	入力	入力	入力	入力
Low	Low	High	入力	入力	入力	入力	入力	入力	入力	入力	入力	入力	入力	入力	入力	出力	出力
Low	High	Low	入力	入力	入力	入力	入力	入力	入力	入力	入力	入力	出力	出力	出力	出力	出力
Low	High	High	入力	入力	入力	入力	入力	入力	入力	入力	出力						
High	Low	Low	入力	入力	入力	入力	入力	入力	出力	出力	出力	出力	出力	出力	出力	出力	出力
High	Low	High	入力	入力	入力	入力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力
High	High	Low	入力	入力	入力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力
High	High	High	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力

注：偶数アドレス LSI の“アドレス”はデバイス番号を指しています。

### 2.1.3.5 入出力ポートの設定方法 (奇数アドレス LSI)

G9205A を、センター制御ポート間モードの奇数アドレス LSI、もしくは、センターレスポート間モードの奇数アドレス LSI として使用する場合は、入出力ポートの設定方法を示します。

G9002A の持つ 2 つのポートは、外部端子 (BMD2、BMD1、BMD0) の設定により、次のように入力ポート、出力ポートの設定が可能です。

みてわかるとおり、BMD2~0 端子に対する各ポートの入出力の状態は、偶数アドレス LSI と完全に逆になります。

BMD2	BMD1	BMD0	ポート状態														
			ポート 1							ポート 0							
			7	6	5	4	3	2	1	0	7	6	5	4	3	2	1
Low	Low	Low	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力
Low	Low	High	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	入力	入力
Low	High	Low	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力
Low	High	High	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力
High	Low	Low	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力
High	Low	High	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力
High	High	Low	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力
High	High	High	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力	出力

注：奇数アドレス LSI の“アドレス”はデバイス番号を指しています。

## 2.1.4 G9103C : モーター制御 LSI

1 軸分のモーションコントロールが行えます。

弊社 LSI 製品である、PCL6045BL をベースとした機能を備えており、容易に複雑なモーション制御を実現できます。

- モーター制御用クロック同期化機能  
通信ラインに接続されている複数の G9103C が同期することで、高精度な同時スタートや補間動作が行えます。
- 割り込みリクエスト信号出力  
様々な要因で、割り込みリクエスト信号をセンター制御 LSI から出力できます。
- 加減速制御  
直線加減速と S 字加減速が行えます。
- オーバーライド  
動作中の速度の変更、目標位置(移動量)の変更等ができます。
- プリレジスタ  
動作中に次の動作を設定するためのデータを保持する機能です。このプリレジスタを 1 段内蔵しています。
- 同報通信機能  
センター制御 LSI から同時にスタート、ストップ、加速・減速のコマンドを送信できます。
- FH 補正機能(三角駆動回避機能)  
位置決めモードで、加速するための指令パルスが足りない場合に、最高速度(FH 速度)を自動的に低下させて三角駆動を回避します。
- 豊富なカウンタ・コンパレータ回路
- ソフトウェアリミット機能  
ソフトウェアリミット範囲に入ったときに即停止、または減速停止ができます。
- 手動パルサー信号の入力機能  
手動パルサー(PA, PB)信号を入力して直接モーターを動作できます。
- 出力パルスモード  
共通パルスモード、2パルスモード、90 度位相差モード(4 週倍)から選択でき、出力論理も選択できます。
- カレントダウン制御  
停止時の励磁電流を自動的に低下させるためのカレントダウン信号が出力できます(ステッピングモーター用)。
- 動作モード  
連続動作、位置決め動作、原点復帰動作、補間動作等、多数の動作モードがあります。
- 機械系入力信号  
エンドリミット、スローダウン、原点などを示す信号入力端子があります。
- サーボモーター I/F  
位置決め完了信号、偏差カウンタクリア信号 の出力と、サーボドライバの異常を示す信号の入力端子があります。
- 非常停止信号  
この信号が ON すると非常停止します。

## 2.1.5 G9004A : CPU エミュレーション LSI

G9004A は次のふたつの動作モードを持ちます。

- ・ CPU エミュレーションモード
- ・ メッセージ通信モード

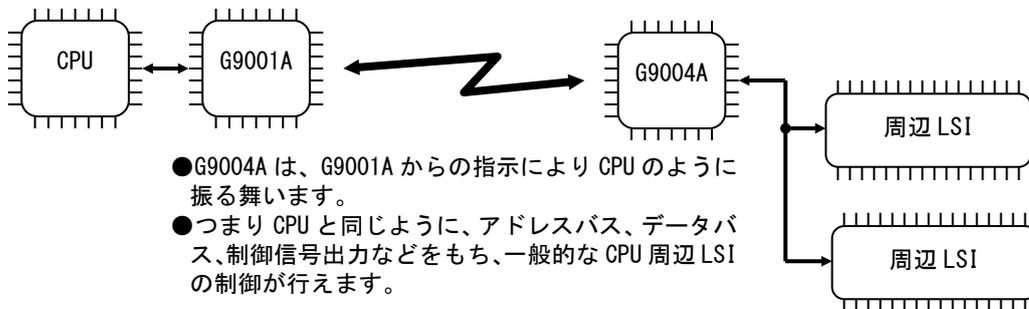
どちらかの動作モードを選んで使用します（ふたつのモードを同時に使用することはできません）。

### 2.1.5.1 CPU エミュレーションモード

G9004A に一般的な CPU 周辺 LSI を接続することで、センター制御 LSI (G9001A) を制御する CPU が、これらの周辺 LSI を遠隔操作できます。

例えば、G9004A に対して、弊社製 LSI である PCL6045BL を 2 個まで接続できます（G9004A の持つ周辺 LSI へのアドレスバスは 6 本です。PCL6045BL は 5 本のアドレスバスで制御します）。

- G9001A を制御する CPU は、通信ラインを介して、G9004A に接続されている周辺 LSI を制御することができます。これにより、周辺 LSI へのデータの書き込みや、読み出しを行います。



### 2.1.5.2 メッセージ通信モード

G9004A に CPU を接続することで、センター制御 LSI を制御する CPU との間で最大 254byte のデータを相互に授受できます。データ形式に規定はありませんので、自由度の高いデータのやりとりが可能です。



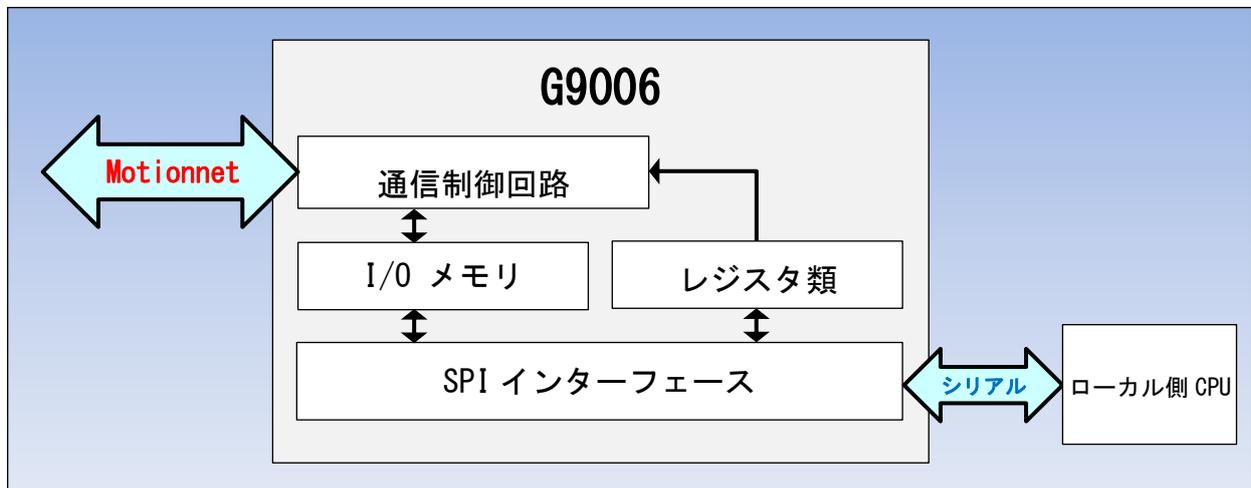
- センター制御 LSI を制御する CPU から、G9004A を制御する CPU へ最大で 254byte のデータの転送が可能です。
- 逆に G9004A を制御する CPU から、センター制御 LSI を制御する CPU へは、最大で 256byte のデータ転送が可能です。ただし転送タイミングはセンター制御 LSI 側の CPU 動作に依存します

## 2.1.6 G9006 : 機能拡張 LSI

入出力ポートを制御する G9002A のような LSI ですが、ポート情報を入出力するための汎用ポート端子がありません。また G9002A と同様で、データ通信には対応していません。

汎用ポート端子が無いかわりに 4 線式シリアルバスで接続されるローカル側 CPU でポート情報を管理します。

また、サイクリック通信でやり取りされている全てのローカル制御 LSI のポート情報を I/O メモリに記憶します。



G9006 には次のふたつの動作モードがあります。

- ・ローカル動作モード
- ・モニター動作モード

これらは電源投入後に、ローカル側 CPU により どちらかを選択します。

さらに G9006 には、「I/O の拡張」、「状態モニター」、「CPU 間通信」の、3 つの機能があります。これらは「ローカル動作モード」の場合は混在した状態で使用できます。「モニター動作モード」の場合は「状態モニター」だけが利用できます。

### 2.1.6.1 ローカル動作モード

G9006 は通常のローカル制御 LSI として動作します。

センター制御 LSI (G9001A) から見ると、G9002A のようにふるまいます。

### 2.1.6.2 モニター動作モード

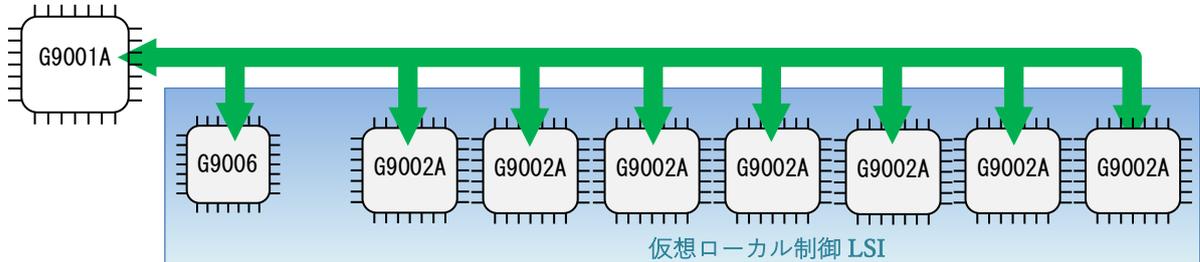
通信ラインを行き来する I/O 通信の内容を、ローカル側 CPU が一方的に監視できます。

G9006 にはデバイス番号を割り当てません。このため G9001A はモニター動作モードの G9006 を認識することはできません。

### 2.1.6.3 I/O の拡張

G9006 は、最大で 7 個までの「仮想ローカル制御 LSI」を追加できます。

G9006 及び追加された「仮想ローカル制御 LSI」は、G9001A からは通常の G9002A のように見えます。つまり、7 個の「仮想ローカル制御 LSI」を追加した場合、G9001A からは 8 個の G9002A が接続しているように見えます。

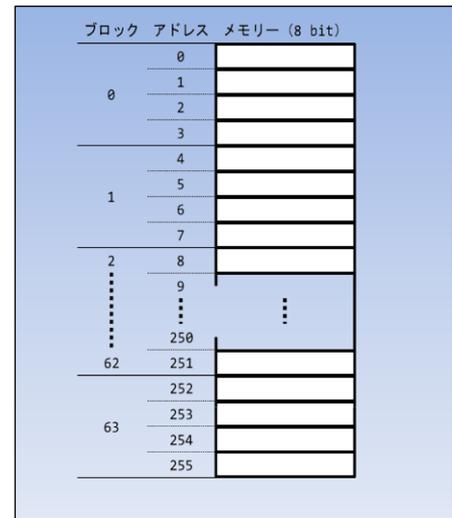


7 個の仮想ローカル制御 LSI を追加した場合、1 個の G9006 で、最大 32byte のデータを扱えます。

### 2.1.6.4 状態モニター

G9006 内の I/O メモリは 256byte あり、デバイス番号ごと (4byte 単位) に区切られています。G9001A の「ポートデータ」エリアと同じ構造です。

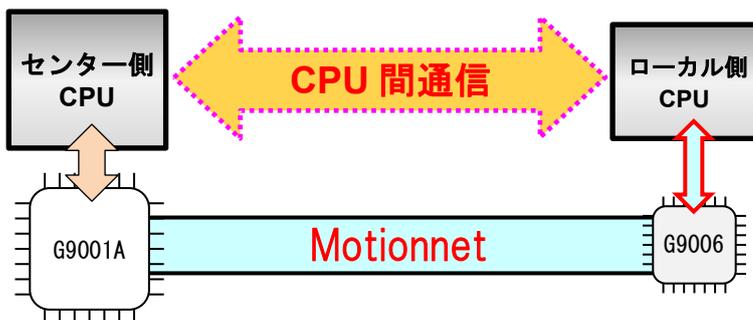
ローカル側 CPU は、このメモリを参照することで、伝送ライン上の全ての「I/O 通信」の状態をモニターできます。



### 2.1.6.5 CPU 間通信

G9006 は I/O 制御 LSI なので、4byte のポートデータエリアを占有しています。

ローカル側 CPU は、この 4byte を利用して自身のデータを送信できます。「仮想ローカル制御 LSI」を追加することで、最大で 32byte の CPU 間通信を、データ通信を使用しないで行えます。



## 2.1.7 G9H50A : HUB 機能 LSI

通信ラインの途中に挿入して使用します。これにより、「通信ラインの分岐」、「通信ライン上のノイズ除去」、「通信距離の延長」等の機能を利用できます。

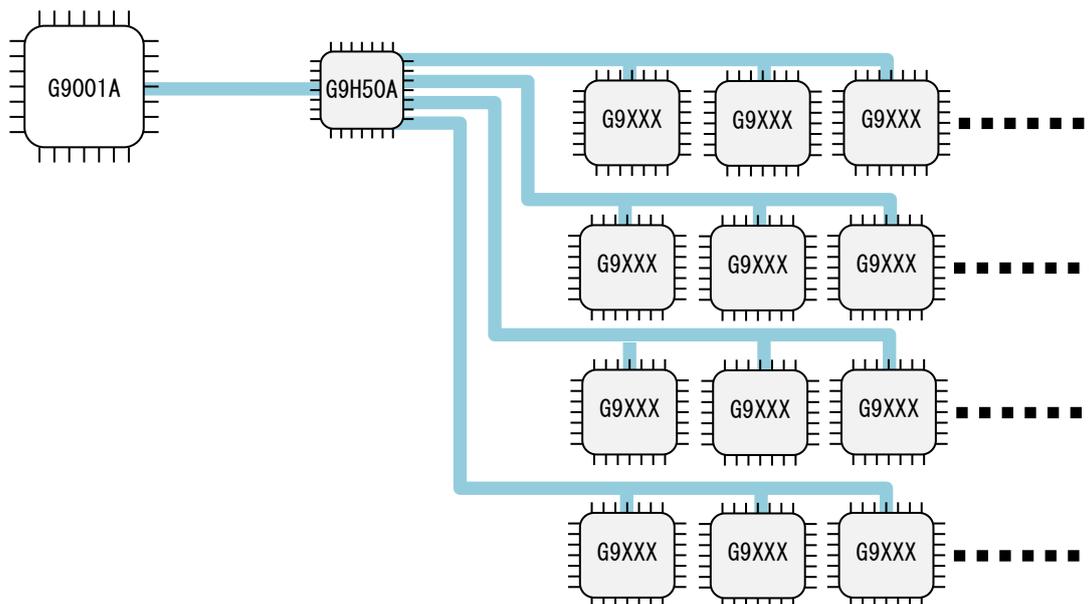
G9H50A にはデバイス番号を割り当てません。このため G9001A は G9H50A を認識することはできません。

このため G9H50A を通信ラインに挿入する際に、G9001A 側 CPU のソフトウェアを変更する必要はありません。

### 2.1.7.1 HUB 機能

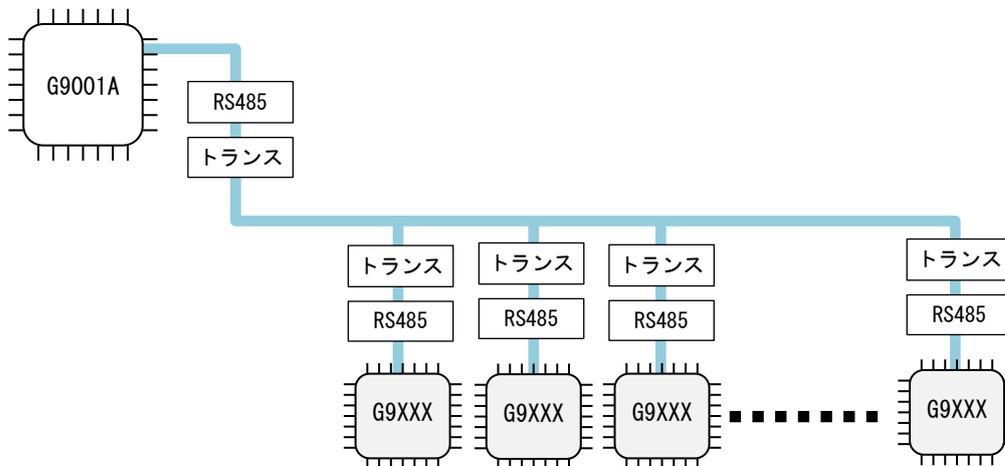
G9H50A は 5 本の通信ラインを制御することができ、ある通信ラインから他の 4 本の通信ラインへ フレームを中継することができます。制御できる 5 本の通信ラインは見かけ上同等で、ケーブルをどこに接続しても同じ結果を得ることができます。

これにより、G9H50A は 1 本の通信ラインを 4 本の通信ラインに分岐させることが可能です。

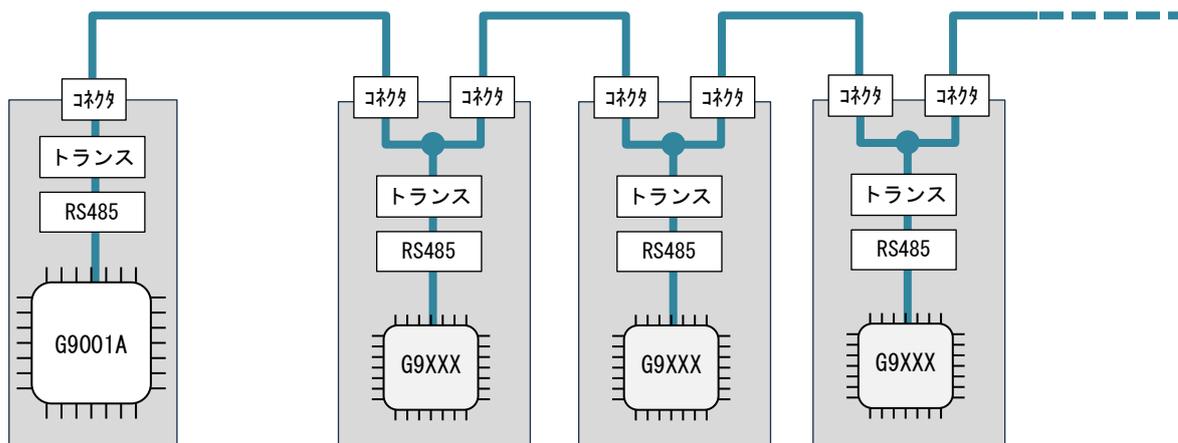


## 2.2 どのように構成すればよいのか？

Motionnet は、ひとつのセンター制御 LSI (G9001A) に接続される通信ラインに、以下のような構成のマルチドロップ形式で、複数のローカル制御 LSI を接続できます。



マルチドロップとは、1本のバスに複数のシステムがぶら下がっているという状態を指し、図で表現すると上図のようになりますが、実際は以下のような“いもづる”形式です。



## 2.3 接続できるローカル制御 LSI はいくつまでか？

ひとつのセンター制御 LSI (G9001A) が制御できるローカル制御 LSI の数は最大で 64 個です。

接続されるローカル制御 LSI は、他と重複しないようなデバイス番号を設定しなければなりません。他と重複している場合は正常な通信は行えません。

センター制御 LSI、モニター動作モードの G9006、G9H50A、はデバイス番号を持っていません。このうち、モニター動作モードの G9006、G9H50A などのようにセンター制御 LSI からの制御を受けない製品は、最大接続数である 64 に含まれません。

## 2.4 通信はどのように行われるのか？

全ての通信はセンター制御 LSI (G9001A) が主導権を持って行われます。

伝送方式は、サイクリック伝送と、トランジェント伝送のふたつに分けられます。

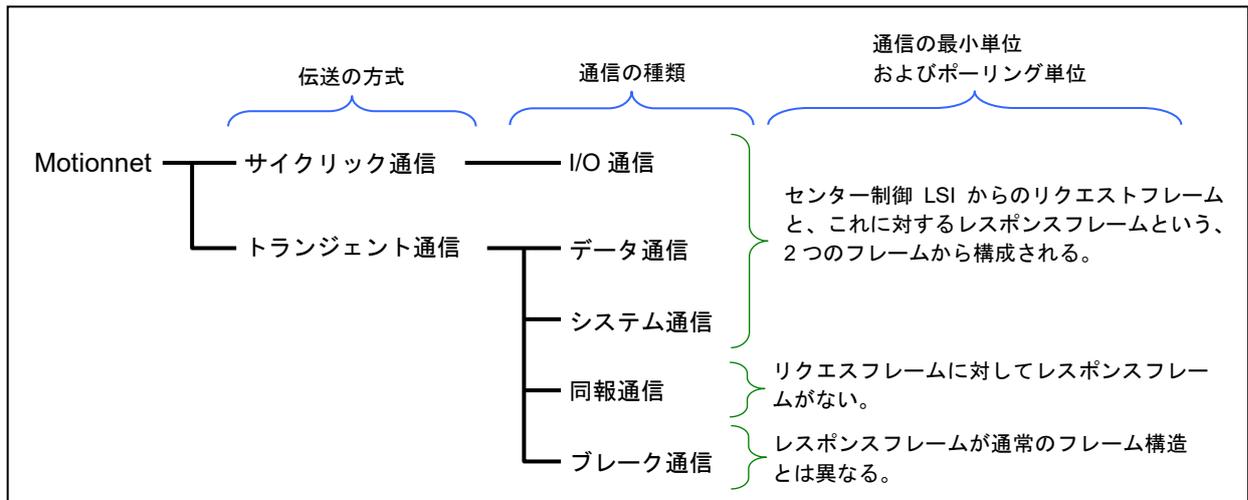
Motionnet では、サイクリック伝送を“サイクリック通信”、トランジェント伝送を“トランジェント通信”と呼称します。

### 2.4.1 Motionnet の通信プロトコル概要

Motionnet の通信プロトコルは以下のような通信体系となっています。

基本的には、I/O 通信によって常に最新の I/O 情報 もしくはステータス情報を得ます。

さらに、データ制御 LSI (G9103C、G9004A) の制御が必要な場合は、トランジェント通信によって行います。



#### 2.4.1.1 通信の最小単位とポーリング単位

##### 2.4.1.1.1 フレーム：通信の最小単位

「通信に使用されるデータ」を送る単位であり、以下のような構成になっています。

スタートコード	フレーム種類コード	デバイス番号	データ	CRC コード
---------	-----------	--------	-----	---------

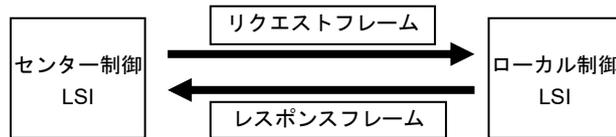
「フレーム種類コード」の値によって「データ」の量が変化します。また後述するデータ通信を行うためのフレームはデータ量を自由に変更できるため、フレームの大きさは一定とは限りません。

##### 2.4.1.1.2 ポーリング単位

Motionnet は、マルチドロップ構成で半二重方式を採用しています。

1本のシリアル通信ライン上に流れるフレームを制御するのはセンター制御 LSI (G9001A) です (G9205A のセンターレスポート間通信を除く)。ここで、センター制御 LSI がローカル制御 LSI との通信を確立させるために「ポーリング・セレクトィング方式」を採用しています。

この方式での Motionnet の基本動作としては、センター制御 LSI が特定のローカル制御 LSI に対して リクエストフレームを送信し、これを受取ったローカル制御 LSI だけが レスポンスフレーム の返送を行います。特定のローカル制御 LSI を指定するためには「デバイス番号」が使われます。



Motionnet は、リクエストフレーム と レスポンスフレーム のペアが基本的な通信単位となっており（ブレイク通信、同報通信は除く）、これを“ポーリング単位”と呼びます。

## 2.4.1.2 ポーリング単位の種類

### 2.4.1.2.1 I/O 通信

送信対象が I/O 制御 LSI であれば、リクエストフレームには出力ポートデータが含まれ、レスポンスフレームには入力ポートデータが含まれます。

送信対象がデータ制御 LSI であれば、リクエストフレームには出力ポートデータやローカル制御 LSI に対する簡素な動作指示が含まれ、レスポンスフレームには主にローカル制御 LSI のステータス情報が含まれます。

また、ポーリング単位で送信されるデータは、リクエストフレーム と レスポンスフレーム あわせて 常に 4byte です。

### 2.4.1.2.2 データ通信

データ制御 LSI に対するデータ量の大きな通信を行います。

ひとつのフレームで 1~256byte のデータを送信できます。

### 2.4.1.2.3 システム通信

シリアルラインに接続されているローカル制御 LSI の情報を取得するために使用される通信です。

取得した情報は、センター制御 LSI (G9001A) 内に、適切な設定値として格納します。

### 2.4.1.2.4 同報通信

センター制御 LSI (G9001A) が、複数のローカル制御 LSI に対して同時にフレームを送り付けるための通信です。

同報通信を受信可能なローカル制御 LSI は、G9103C と、G9006 だけです。

通信ライン上に点在する G9103C (モーター制御 LSI) の同時スタートなどで活用します。

### 2.4.1.2.5 ブレイク通信

通信ラインに対するローカル制御 LSI の新規参入をシームレスに行う際に使用します。

ブレイク応答待ち状態になっているローカル制御 LSI がこのリクエスト通信を受信すると、ブレイク応答を返します。ブレイク応答は単純な High、Low の連続信号で、先に説明したフレームの構造を持っていません。

また、ローカル制御 LSI にはブレイク入力と呼ばれる外部端子があり、これを一定時間以上の High 状態にすることでブレイク応答待ち状態になります。ブレイク応答を返したあと、ブレイク応答待ち状態は解消されます。

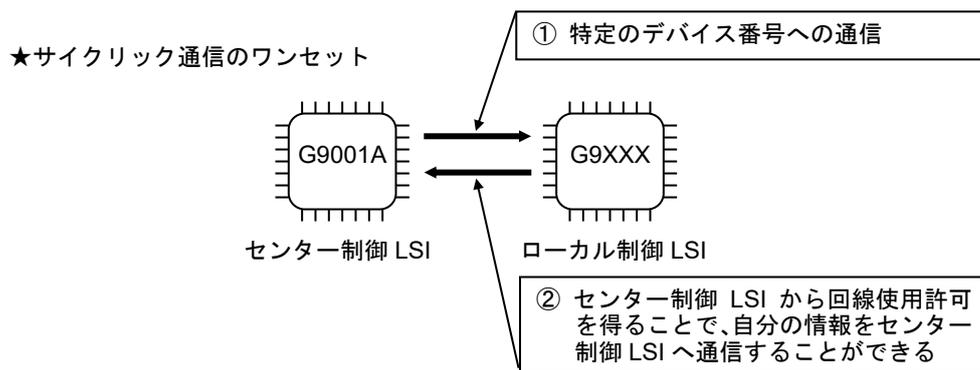
### 2.4.1.3 サイクリック通信

Motionnet の基本的なプロトコルは ポーリング／セレクトイング方式です。

通信ラインの制御権は、通常 センター制御 LSI (G9001A) が持っています。これは通信ライン上でのデータ衝突を避けるため、各ローカル制御 LSI はセンター制御 LSI の許可がなければデータの送出不可能です。

各ローカル制御 LSI と効率よく通信を行うために、センター制御 LSI は、接続されているローカル制御 LSI のうち 一番若いデバイス番号を持つものから順番に通信を行います。

まずセンター制御 LSI から、最初のローカル制御 LSI へリクエストフレームを送信します。このフレームを受け取るローカル制御 LSI は通信ライン上にひとつしか存在しません (デバイス番号は重複してはいけないため)。そして通信を受け取ったローカル制御 LSI が、次に通信ラインを使用する権利を得ることとなり、ローカル制御 LSI からセンター制御 LSI に向けて通信が行われます。これがワンセットの通信です。



その後センター制御 LSI は、次に若いデバイス番号を持つローカル制御 LSI に対してリクエストフレームを送信し、レスポンスフレームを受信します。

このような単純な処理を、デバイス番号順にすべてのローカル制御 LSI に対して行います。

そして、最後のデバイス番号との応答が終了すると、再度一番若いデバイス番号のローカル制御 LSI から順番に同じ処理を繰り返します。

このように順番にすべてのローカル制御 LSI との応答を繰り返す通信をサイクリック通信と呼びます。

### 2.4.1.4 トランジェント通信

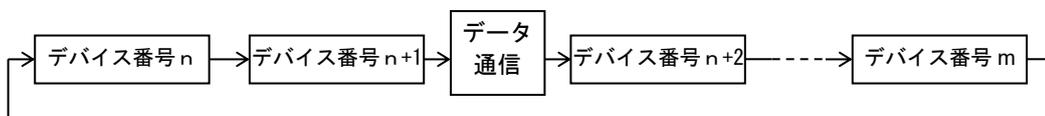
Motionnet は、常にサイクリック通信が継続されています。

これに対してトランジェント通信は、任意のタイミング（GPU からの送信指示のタイミング）で、サイクリック通信に割り込む形で行われます。

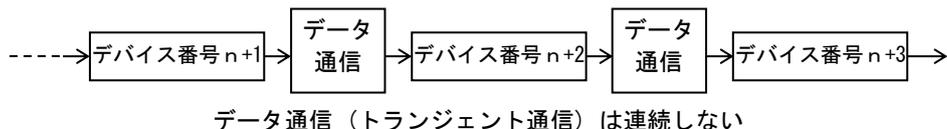
例えば以下のような順序でサイクリック通信が行われていると仮定します。



「A」のタイミングで CPU からデータ通信コマンドが発行されると、以下のようにサイクリック通信の合間に割り込むような形でデータ通信が行われます。



「A」のタイミングで複数回のデータ通信コマンドが発行できてしまった場合は、次のようにデータ通信が連続しないようになり、サイクリック通信の長時間の停止を防ぎます。



データ通信は、最大で 256byte のデータ送信が可能です。

データ通信もサイクリック通信と同様で、特定のローカル制御 LSI に対して送信データを含むリクエストフレームが送られた後、そのローカル制御 LSI からのレスポンスフレームがセンター制御 LSI（G9001A）に対して送られます。このレスポンスフレームにも最大で 256byte のデータを乗せることもできます（現状では G9004A のみ可能です）。

ちなみに、I/O 制御 LSI や機能拡張 LSI はデータ通信を受信することはできません。これらの LSI に対してデータ通信を行った場合、これを受信したローカル制御 LSI からはエラー応答が返ってきます。

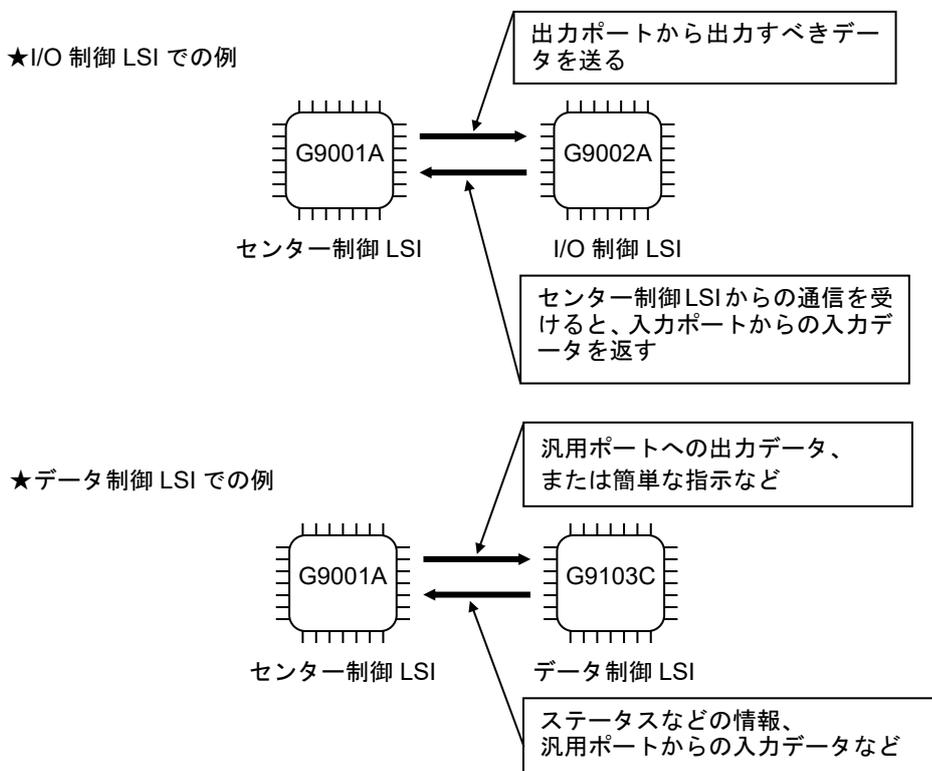
## 2.5 サイクリック通信で送られるのはどのようなデータか？

サイクリック通信は CPU 側からの停止コマンドがない限り、常に行われます。

この通信でやりとりされるデータには、サイズに上限があります。これは常に一定の周期でサイクリック通信が行われるためのルールです（このような条件があるため、大きなサイズのデータ転送のためのデータ通信が存在します）。

このサイズの上限値は 4byte です。このサイズ内で通信されるデータは、ローカル制御 LSI ごとに内容が異なります。

種類	センター制御 LSI からの通信内容	ローカル制御 LSI からの返送内容
G9002A	出力ポートへ出力するためのデータ	入力ポートから入力されたデータ
G9205A	出力ポートへ出力するためのデータ	入力ポートから入力されたデータ
G9103C	汎用出力ポートへの出力データ	汎用入力ポートからの入力データと、モーター制御 LSI のステータス情報
G9004A	動作指示など	CPU エミュレーション LSI のステータス情報
G9006	出力ポートとして設定された RAM エリアへ格納するためのデータ	入力ポートとして設定された RAM エリアに格納されているデータ



サイクリック通信は、I/O 制御 LSI とのデータの受け渡しの効率化（一定時間内でのデータ更新の保証など）を考慮して設計されています。よってサイクリック通信の最小単位である I/O 通信のデータサイズは、リクエストフレームとレスポンスフレーム あわせて常に 4byte となっています。これは G9002A (I/O 制御 LSI) のポート数と一致します。

## 2.6 ステータスとは何か？

G9103C（モーター制御 LSI）や、G9004A（CPU エミュレーション LSI）がどのような状態になっているかを監視するために、各データ制御 LSI にはステータスという情報があります（G9002A、G9205A、G9006 には ステータスはありません）。

ステータスは、モーター制御 LSI の動作停止や、CPU エミュレーション LSI 側からのデータ転送要求など、各データ制御 LSI の状況を知るために使用します。

ステータス情報はサイクリック通信で各データ制御 LSI からセンター制御 LSI（G9001A）へ送られ、センター制御 LSI の内部 RAM へ格納されます。

CPU から この情報を参照することで、各データ制御 LSI の状態の把握が可能です。

ちなみにセンター制御 LSI 自身もステータス情報を持っていて、CPU から参照できます。

G9004A に関しては、センター制御 LSI へ送られて センター制御 LSI 側の CPU で参照されるステータスと、G9004A に接続される CPU が参照するステータスが別々に存在します（詳しくは、「G9004A のステータスに関する注意」の項目を参照してください）。

## 2.7 通信周期はどれくらいの時間か？

サイクリック通信のみで考えた場合以下のようになります（20Mbps の場合）。

ローカル制御 LSI 数	通信周期	備 考
8 個	0.12 ms	全てが I/O 制御 LSI の場合は 256 点の I/O が利用できます
16 個	0.24 ms	全てが I/O 制御 LSI の場合は 512 点の I/O が利用できます
32 個	0.49 ms	全てが I/O 制御 LSI の場合は 1024 点の I/O が利用できます
64 個	0.97 ms	全てが I/O 制御 LSI の場合は 2048 点の I/O が利用できます

ローカル制御 LSI の使用数が上記以外の場合や、データ通信を伴う場合の通信周期に関しては、ユーザーズマニュアルでご確認ください（計算式が載っております）。

## 2.8 通信できる距離は？

弊社にて実施した通信実験の結果と、実験時の環境を示します。

お客様にてシステムの構築をする場合の参考となりますが、これら以外の構成でもシステム構築は可能です。あくまでも参考として見て下さい。

通信レート	条 件					結 果
	ローカル制御 LSI 数	ケーブル規格	終端抵抗	パルストランス	インターフェース	総延長
20Mbps	32 個	Cat5 以降	100Ω	1000μH	RS485	100m
20Mbps	64 個	Cat5 以降	100Ω	1000μH	RS485	50m
10Mbps	64 個	Cat5 以降	100Ω	1000μH	RS485	100m

Cat5 : LAN ケーブルの規格のひとつ。カテゴリー5 を示す。

注 : 上記の内、総延長の数値は実験室内での理想的な条件における結果です。使用する RS485 の品種やケーブルの引き回し環境など、実際の使用条件により上記と同等の結果がでない場合もありますので、ご注意ください。

センター制御 LSI (G9001A) は、シリアルデータの受信端子を 2 系統持っています。1 系統のみの使用時に、思ったように通信距離が伸びない場合などに活用できます。

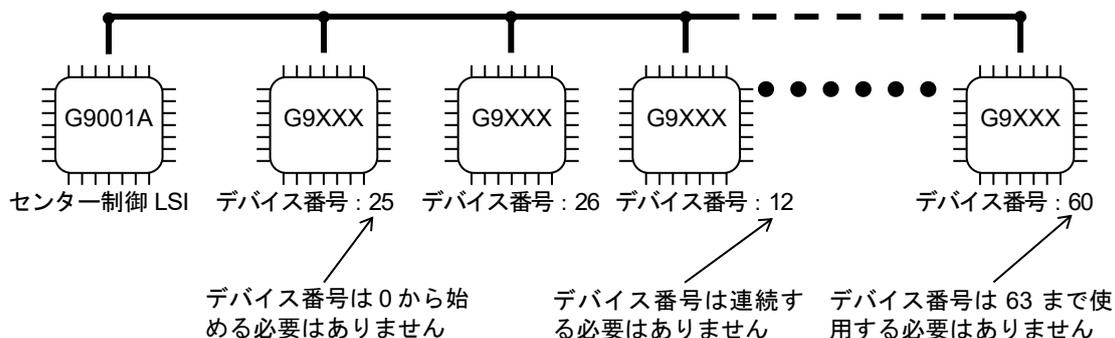
## 2.9 デバイス番号は連続しなくてもよいか？

デバイス番号は、0~63 までの 64 種類の使用が可能です。必ずしも 0 から使用するという規則はありません (ただし、重複してはいけません)。

また途中のデバイス番号が歯抜けになってもかまいません。

通常、センター制御 LSI (G9001A) は通信ラインに接続されているローカル制御 LSI のデバイス番号を把握しており、接続されているデバイス番号にのみ通信を行います。

このデバイス番号の把握は、自動的に行うことが可能です (お客様が手動で行うこともできます)。



## 2.10 RS485 とはなにか？

RS485 とは、シリアル通信を行うために定められた規格です。EIA-485 規格などとも呼びます。

Motionnet は、この規格を使用したシステム構築を推奨しています。それは、高速・遠距離のシリアル通信の規格として現時点で比較的優れていることと、RS485 の規格に準じたドライバ IC が容易に入手できることによります。ここでいうドライバ IC とは、トランシーバ機能を備えたものを指しています。要するに、通信ラインへの信号送出機能と、通信ラインからの信号受信機能の両方を備えたものです。

この RS485 という規格は、通信に必要な電気的特性のみを規定したものです。RS232C などは、回線を行き来する信号の状態まで規定したものとなっており、この点で明らかに異なります。

## 2.11 パルストランス (トランス) が必要な理由は？

パルストランスとは、通信用途向けのトランスという位置づけの製品のようなのですが、構造自体は通常のトランスと同じです。

交流信号のみ通過できるという特性を利用し、通信システムごとの電源分離を行います。Motionnet は、パルストランスを使用した通信システムの構築を推奨します。

RS485 規格の IC のみで、パルストランスなしの通信はもちろん可能です。しかしこの場合、送信側と受信側で GND を共通にする必要があります。GND レベルを共通にしなければ、RS485 規格の IC 間でやりとりされる信号の電圧レベルが、RS485 規格の IC の許容範囲を超えて破壊される可能性があります。GND を共通にするためには、通信ラインと平行して GND 線を追加するようになります。

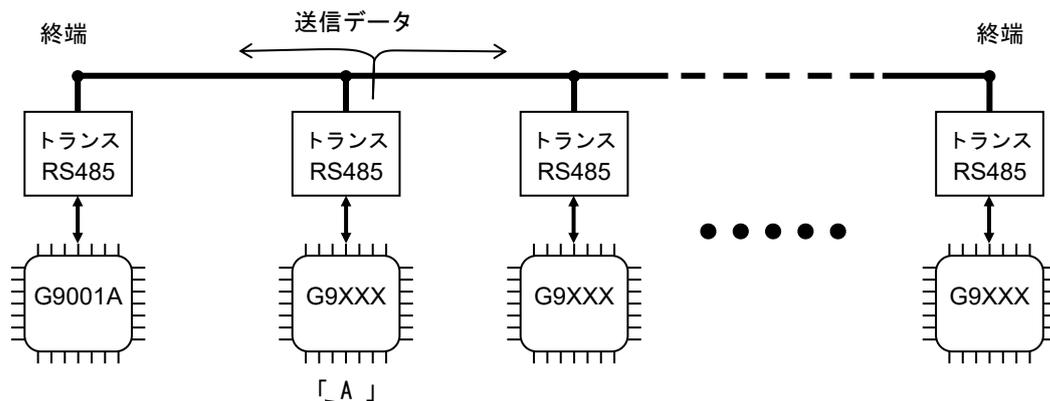
このようにパルストランス抜きでの通信は、技術的には可能ですが、GND 信号線を引き回すことで、大地を含んだ大きな GND ループが作られることなどの影響で、ノイズに対しては比較的弱いものとなります。

またパルストランスを使用しないで活線挿抜を行った場合、RS485 が壊れる可能性もあります。



通信ラインはパルストランスにより絶縁されます。  
よって上図の左右のシステムは電氣的に分離されます。

## 2.12 終端抵抗の役割は？

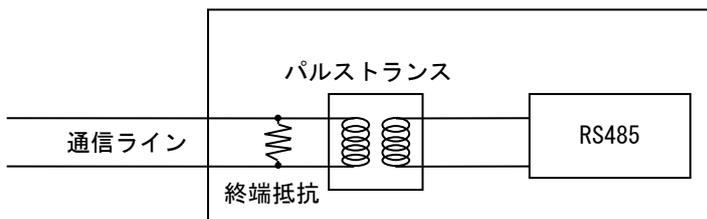


例えば上図の「A」からシリアルデータが送信されたと仮定します。

この信号は通信ラインの両側に向かって伝わり、それぞれの通信ラインの端（終端）まで到達します。このデータは、通信ラインの端で跳ね返り、通信ラインを逆送します。これは反射と呼ばれる現象で、通信障害の原因のひとつです。

終端抵抗は、この反射の発生を抑止するために使用します。通常は上図のように、通信ラインの末端側（両方の終端）に  $100\Omega$  前後の抵抗を設置します。通信ラインの途中には必要ありません。

下図は、パルストランスからみて、通信ライン側へ終端抵抗を挿入していますが、逆側（パルストランスの RS485 規格の IC 側）に終端抵抗を設置しても同様な効果が得られます。



## 2.13 G9000 シリーズを制御できる CPU の種類は？

G9000 シリーズで CPU を必要とするのは、G9001A、G9004A（メッセージ通信モードで使用する場合）、G9006 です。さらに、G9103C も CPU を接続することはできます。ただし、その CPU から G9103C を制御することはできません。ローカル側 CPU で制御可能なもの以外のローカル制御 LSI は、センター制御 LSI（G9001A）により制御されます。G9001A と G9004A に接続できる CPU のタイプは、パラレルバスを持つタイプです。G9103C と G9006 に接続できる CPU のタイプは、SPI バスを持つタイプです。以下に、それぞれに接続可能な CPU を示します。

### 2.13.1 G9001A に接続可能な CPU タイプ

外部端子である“IF0”と“IF1”を使用して CPU タイプを選択できます。

設定		インターフェースモード	CPU タイプ	例
IF1	IF0			
Low	Low	モード 1	旧モトローラ系 16 ビット CPU アドレスがビッグエンディアン	68000
Low	High	モード 2	旧日立製作所系 16 ビット CPU アドレスがビッグエンディアン	H8
High	Low	モード 3	インテル系 16 ビット CPU アドレスがリトルエンディアン	8086
High	High	モード 4	8 ビット CPU アドレスがリトルエンディアン	Z80

### 2.13.2 G9004A に接続可能な CPU タイプ

外部端子である“IF0”と“IF1”を使用して CPU タイプを選択できます。

設定		インターフェースモード	CPU タイプ	例
IF1	IF0			
Low	Low	モード 1	旧モトローラ系 16 ビット CPU	68000
Low	High	モード 2	旧モトローラ系以外の 16 ビット CPU	H8、8086
High	Low	モード 3	旧モトローラ系 8 ビット CPU	6809
High	High	モード 4	旧モトローラ系以外の 8 ビット CPU	Z80

### 2.13.3 G9103C、G9006 に接続可能な CPU タイプ

SPI バス制御が可能な CPU であれば問題なく接続できます。

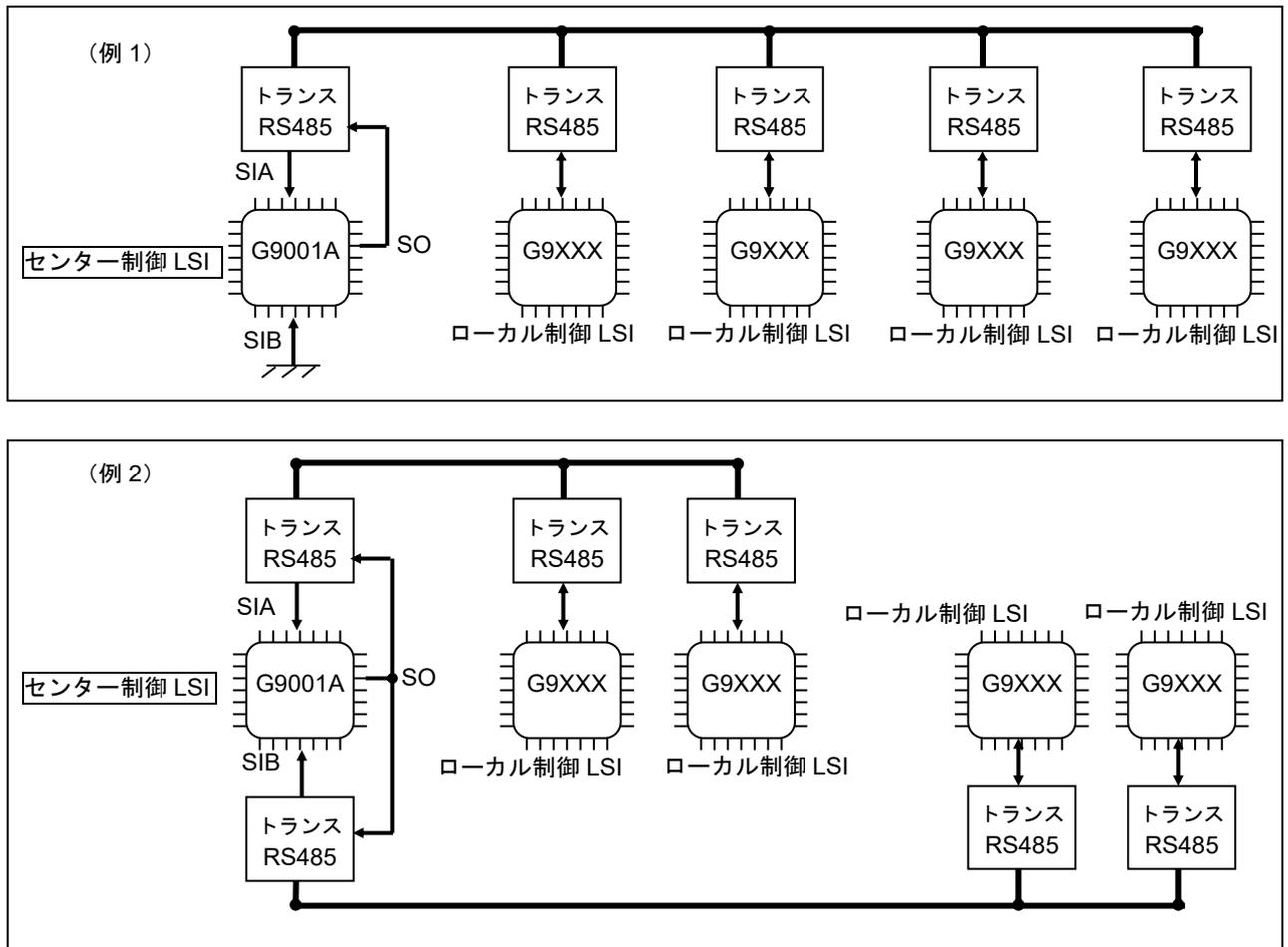
## 2.14 センター制御 LSI にシリアル入力端子が 2 本ある理由は？

“SIA” と “SIB” という名前の 2 本の外部端子は、RS485 などの IC で受けた通信ラインからのシリアル入力信号を受け取るための端子です。

これに対して、通信ラインへデータを送り出すための端子としては シリアル出力端子 (SO) があり、こちらは 1 本です。

通常は、2 本あるシリアル入力端子のうちの片方 (“SIA” という名称の端子) を使用し、もう片方 (“SIB” という名称の端子) はプルアップかプルダウンの処置をしておけば通信動作は行えます。

センター制御 LSI (G9001A) にシリアル入力端子が 2 本ある理由は次の理由によります。



上図の 2 つの例のように異なる構成としても、CPU からの操作方法は同じです。どちらの構成であっても、センター制御 LSI 側の CPU からは区別が付きません。

ここで、通信ラインにかかる負荷を考えた場合、例 1 ではひとつの通信ラインに 4 個のローカル制御 LSI が接続された構成で、例 2 ではひとつの通信ラインには 2 個のローカル制御 LSI が接続されており、これが 2 ラインある構成となっています。このため、あきらかに例 2 の方がひとつの通信ライン当たりの負荷は軽くなっています。

通信ラインにかかる負荷を減らせることで、長距離のケーブル構成が実現しやすくなります。例えば総延長が 100m の通信ラインに 32 個のローカルをぶら下げる場合と、総延長が 50m 通信ラインに 16 個の構成をふたつ使用するのでは、後者の方が断然有利で、ノイズに対しても強くなります。良好な通信品質を確保するには後者の構成が望ましいといえます。

## 3. Motionnet の操作

### 3.1 Motionnet はどれだけ簡単に使えるか？

適切に接続された Motionnet で通信を行う場合、CPU が G9001A に指示すべきことは非常に単純です。  
以下の 2 つの指示を行うだけでサイクリック通信が開始されます。

① “システム通信” コマンドの発行

このコマンドを G9001A に対して行うと、G9001A は通信ライン上にどのようなローカル制御 LSI が接続されているのかのサーチを行います。

サーチの結果、ローカル制御 LSI の種類 (I/O 制御 LSI か、それ以外か) の判定を行い、各 LSI のデバイス番号を取得します。また I/O 制御 LSI の場合は、入出力ポートの設定状態の取得も行います。

これにより通信ライン上のローカル制御 LSI の情報を自動的に取得し、設定できます。この設定結果は CPU から参照すること (もしくは変更すること) が可能です。

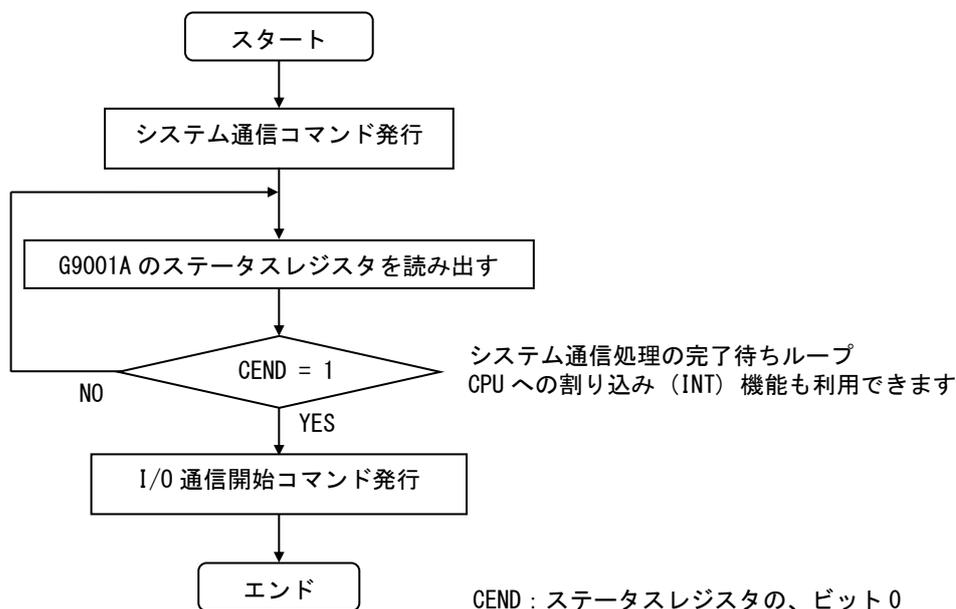
この確認動作が完了すると、G9001A のステータスレジスタの “CEND” (ビット 0) が変化します。これを確認後、次の指示を行ってください。

② “I/O 通信の開始” コマンドの発行

①で取得したデバイス番号に対して、サイクリック通信が開始されます。

これだけで、例えば入力ポートに設定されている I/O 制御 LSI からは、その入力データが G9001A の内部 RAM へ格納され、逆に出力ポートに設定されている I/O 制御 LSI へは、内部 RAM へ出力したいデータを書き込むだけでデータが転送され、出力されます。

内部 RAM のアドレスは、デバイス番号に対応した値が割り振られております。さらに詳細な情報は、はユーザーズマニュアルでご確認ください (アドレスマップが記載されております)。



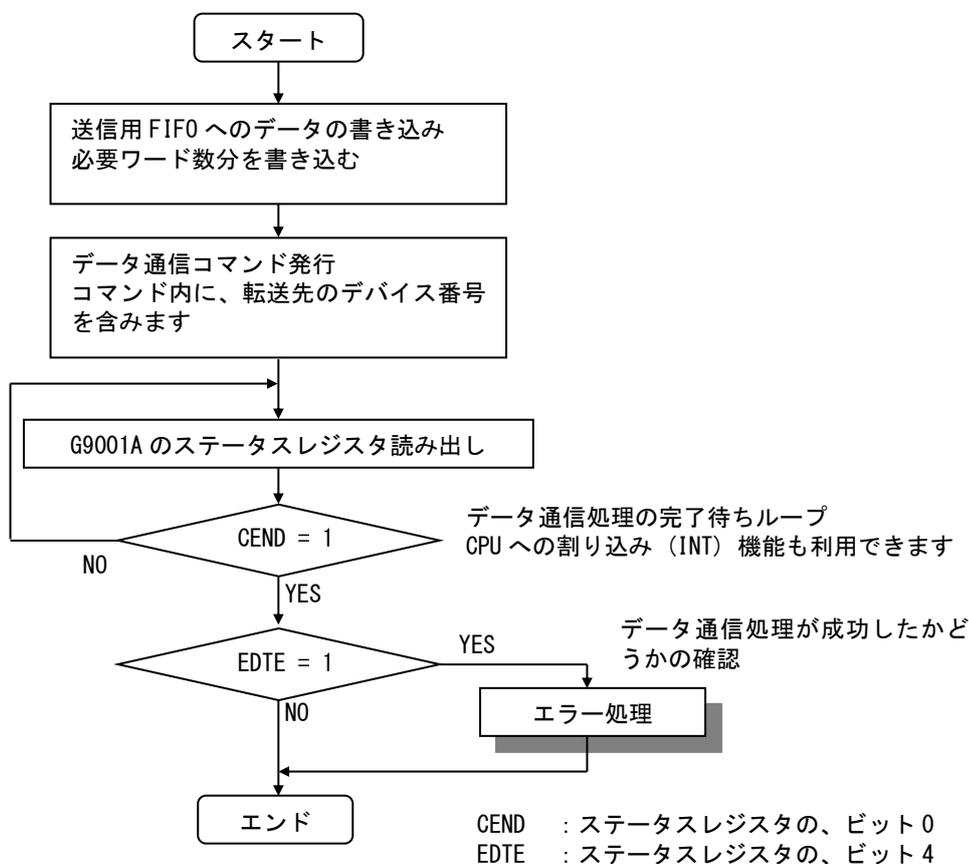
## 3.2 データ通信のやり方は？

データ通信を行う上での前提条件として、サイクリック通信がスタートしていなければなりません。

サイクリック通信がスタートされてさえいれば、いつでもデータ通信が可能です。

データ通信の手順は以下の通りです。

- ① 送信データを G9001A の送信用 FIFO へ設定する  
G9001A は、送信用、受信用それぞれで 256byte ずつの FIFO を持っています。  
データ通信で送り出したいデータは送信用 FIFO へ 1word (16bit または 8bit) ずつ書き込んでゆきます。
- ② “指定のデバイス番号に対するデータ通信コマンド” を発行する  
このコマンドを G9001A に対して発行することで、先に送信用 FIFO へ書き込んでおいたデータが、指定したデバイス番号を持つローカル制御 LSI に対して送信されます。  
G9001A のステータス状態により、送信が正常に完了したことを確認してください。



### 3.3 送信用 FIFO へのデータの書き込み方法は？

FIFO とは、容量分のアドレスを持たないメモリです。入力用の入り口と、出力用の出口があり、入り口から書き込んだデータがメモリ内部に順番にたまってゆきます。たまったデータは出口から取り出せますが、このとき書き込んだ時と同じ順番でしか取り出せません。先入れ、先出しの RAM なので、First In First Out の頭文字を取って FIFO と呼びます。

送信用 FIFO へのデータの入り口は、G9001A のアドレスマップ上で次のようになっています。

アドレス	内 容 (16 ビット)	
000h	他の目的で使用	
002h		
004h		
006h	送信用 FIFO へのデータ設定用の入り口	
008h	他の目的で使用	
S		
		1FFh

注：インターフェースモード 3 (IF1=High、IF0=Low) のアドレスマップです

16 ビット CPU で “006h” 番地のアドレスヘデータを書き込むと、そのデータは送信用 FIFO へ格納されます。

### 3.4 受信用 FIFO からのデータの読み出し方法は？

受信用 FIFO からのデータの出口は、G9001A のアドレスマップ上で次のようになっています。

見てわかるとおり、送信用 FIFO と同じアドレスです。要するに書き込み操作を行うと、送信用 FIFO への書き込みとなり、読み出し操作を行うと受信用 FIFO からの読み出しとなります。

アドレス	内 容 (16 ビット)	
000h	他の目的で使用	
002h		
004h		
006h	送信用 FIFO へのデータ設定用の入り口	
008h	他の目的で使用	
S		
		1FFh

注：インターフェースモード 3 (IF1=High、IF0=Low) のアドレスマップです

16 ビット CPU で “006h” 番地のアドレスヘデータを読み出すと、受信用 FIFO からのデータを取り出せます。

### 3.5 G9002A の入力ポートのデータ取得方法は？

サイクリック通信が開始されると、I/O 制御 LSI（G9002A など）の入力ポートの状態は サイクリック周期ごとに自動的に更新されます。サイクリック通信で受信した値は、G9001A 内の RAM に格納されます（ポートデータエリアと呼びます）。

アドレス	内 容 (16 ビット)
000h	他の目的で使用
0FEh	
100h	
102h	デバイス番号=0 の、ポート 3 とポート 2
104h	デバイス番号=1 の、ポート 1 とポート 0
106h	デバイス番号=1 の、ポート 3 とポート 2
1F8h	デバイス番号=62 の、ポート 1 とポート 0
1FAh	デバイス番号=62 の、ポート 3 とポート 2
1FCh	デバイス番号=63 の、ポート 1 とポート 0
1FFh	デバイス番号=63 の、ポート 3 とポート 2

} 16bit × 128word

注：インターフェースモード 3（IF1=High、IF0=Low）のアドレスマップです

G9002A は、8 ビットの入出力ポートを 4 つ持っています（ポート 0～3）。これらのポートと、G9001A 内の RAM は上図のように対応しています。

よって、任意の G9002A のポート状態を取得するには、その G9002A のデバイス番号に対応する RAM アドレスからデータを読み出せば良いことになります。

例えばデバイス番号=62 のポート 1 の状態を取得するには、G9001A 内の RAM の“1F8h”番地を読み出し、そのデータの上位 8 ビットを参照することとなります。

また G9002A の持つ 4 つのポートは、外部端子（PMD2、PMD1、PMD0）により以下のように入力ポート、出力ポートの設定が可能です。

PMD2	PMD1	PMD0	ポート 0	ポート 1	ポート 2	ポート 3
Low	Low	Low	出力	出力	出力	出力
Low	Low	High	入力	出力	出力	出力
Low	High	Low	入力	入力	出力	出力
Low	High	High	入力	入力	入力	出力
High	Low	Low	入力	入力	入力	入力

ここで出力モードに設定されたポートに関しては、次項で説明するように G9001A 内の RAM へ出力データを書き込みます。このデータを読み出すと、出力データとして設定したデータが得られます。

### 3.6 G9002A の出力ポートへのデータ出力方法は？

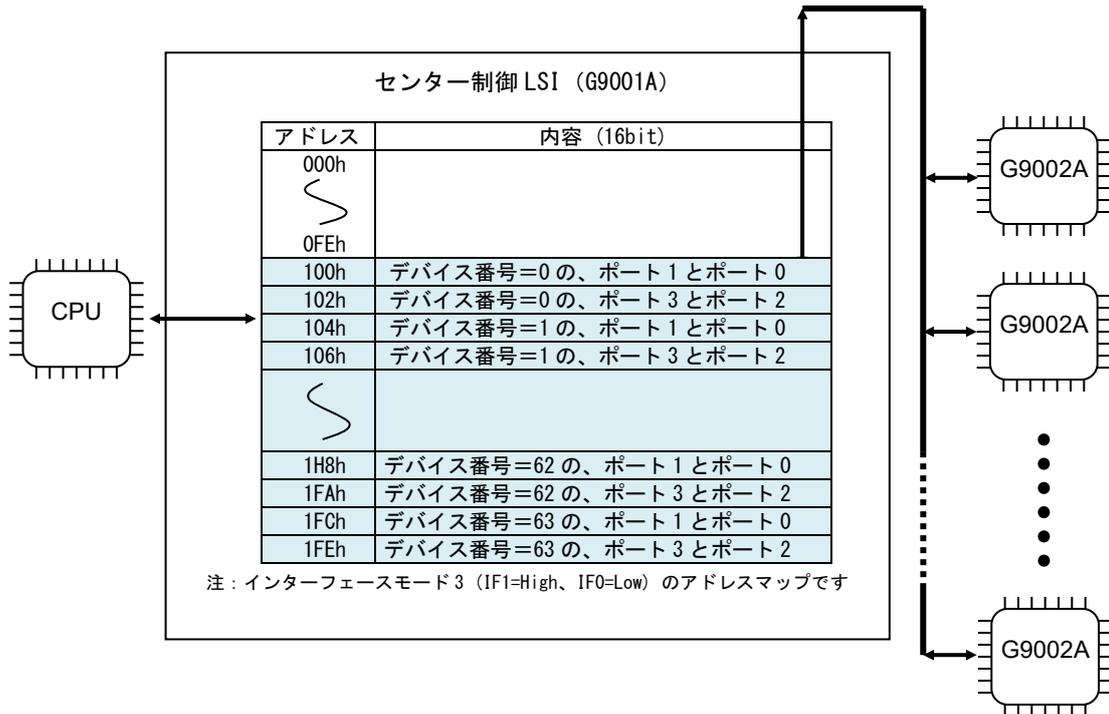
サイクリック通信が開始されると、G9001A 内の RAM に格納されたデータがサイクリック周期ごとに自動的に、G9002A の出力ポートへ送信されます。

アドレスマップに関しては前項のものと同じです。指定のデバイス番号に対応する RAM アドレスに対して、出力したいデータを書き込んでください。

ただし、G9002A 側で入力モードに設定されたポートに対しては、G9001A 内の RAM への書き込みはできません。

以上をまとめると、G9001A に接続されているローカル制御 LSI のポート情報は、内部 RAM を通じて周期的に更新されます。

CPU からみると、単に RAM をアクセスする時と同じイメージで扱うことになります。





### 3.8 通信ラインに接続されているローカル制御 LSI 種類の確認方法

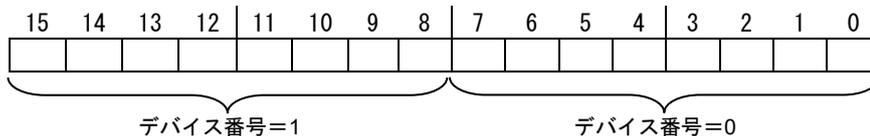
通信ラインに接続するローカル制御 LSI は、G9002A、G9205A、G9103C、G9004A、G9006 など自由に組み合わせて配置できます。

そして通信ライン上に接続されているローカル制御 LSI に関する情報は、“システム通信コマンド”によって自動的に収集が可能です。ここで収集したデータは、G9001A 内の RAM へ以下のように格納されます（デバイス情報エリアと呼びます）。これは CPU からの参照や、変更が可能です。

アドレス	内容 (16 ビット)
000h ⋮ 076h	他の目的で使用
078h	デバイス番号=1、0 のデバイス情報
07Ah	デバイス番号=3、2 のデバイス情報
⋮	⋮
0B4h	デバイス番号=61、60 のデバイス情報
0B6h	デバイス番号=63、62 のデバイス情報
0B8h ⋮ 1FFh	他の目的で使用

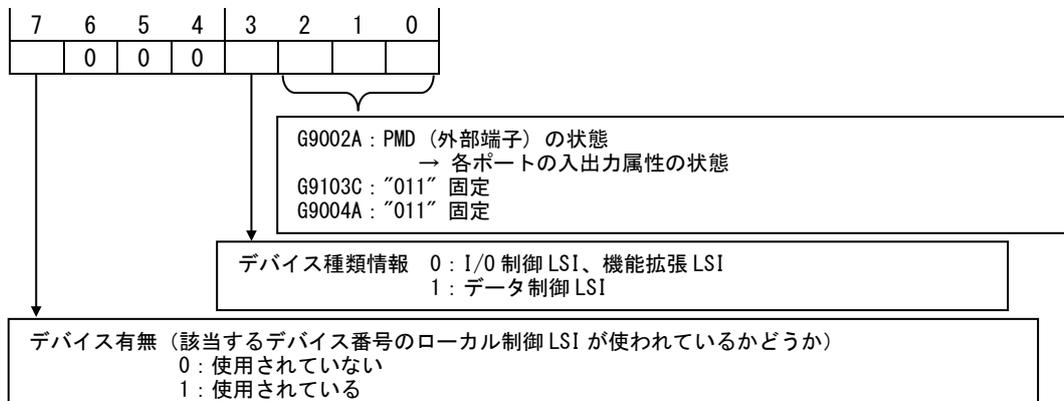
注：インターフェースモード 3 (IF1=High、IF0=Low) のアドレスマップです

アドレスマップ中の“078h”番地の詳細を示します。



16 ビット CPU が“078h”番地をみる場合、上図のようになります。

8 ビットごとにひとつのデバイス番号に対応し、各デバイス番地単位では以下ようになります。



## 3.9 I/O 制御 LSI の入力ポートの変化を割り込み機能で認識するには？

G9002A や G9205A の入力ポートの状態は、サイクリック通信にて常に最新のデータに更新されますが、常に値が変化することも限りません。いつ変化するかわからないポートを監視するために、“入力変化割り込み”と呼ばれる入力状態の変化があったことを CPU へ通知する機能があります。

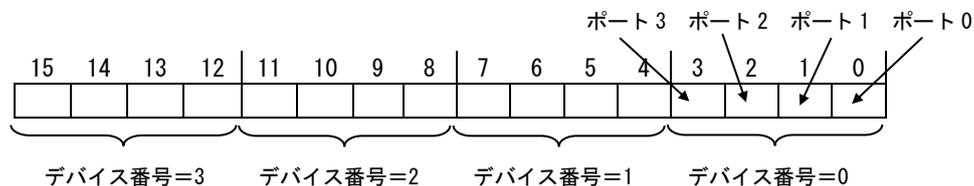
### 3.9.1 入力値変化を監視するローカル制御 LSI の指定

“入力変化割り込み”を使うには、I/O 制御 LSI の「どのポートの入力状態の変化を監視するか」の設定が必要となります。これには G9001A 内の RAM の“入力変化割り込み設定エリア”で行います。

アドレス	内 容 (16 ビット)
000h ∑ 0BEh	他の目的で使用
0C0h	デバイス番号=3、2、1、0 の入力変化の監視設定
0C2h	デバイス番号=7、6、5、4 の入力変化の監視設定
∑	∑
0DCh	デバイス番号=59、58、57、56 の入力変化の監視設定
0DEh	デバイス番号=63、62、61、60 の入力変化の監視設定
0E0h ∑ 1FFh	他の目的で使用

注：インターフェースモード 3 (IF1=High、IF0=Low) のアドレスマップです

アドレスマップ中の“0C0h”番地の詳細を示します。



16 ビット CPU が“0C0h”番地をみる場合、上図のようになります。

4 ビットごとにひとつのデバイス番号に対応します。

4 ビットの内訳は、下位側からポート 0、ポート 1、ポート 2、ポート 3 に対応します。

この構成で、監視を行いたいデバイス番号のポート番号に対応するビットを“1”にすると、ポートの入力値状態の監視が開始されます。

たとえば、デバイス番号=5 の、ポート 2 を監視したい場合以下ようになります。

G9001A : RAM アドレス=0C2h  
設定データ=0040h

ポート状態の監視を行う場合、該当するデバイス番号のローカル制御 LSI のポートは、入力モードに設定してください。以上の設定が行われていると、監視しているポートの変化により、CPU に対して割り込み (INT) が発生します。

### 3.9.2 入力値変化のあったローカル制御 LSI の特定

前項で入力変化の監視と、それによる CPU への割り込み信号が出力できるようになります。

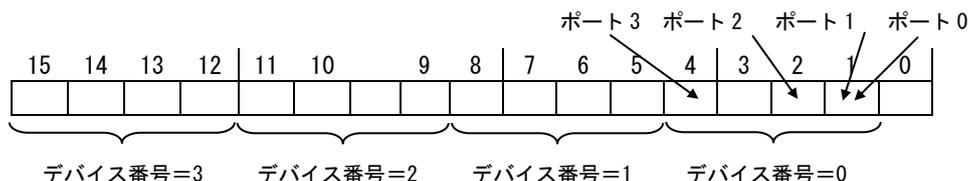
しかしこのローカル制御 LSI のポートが変化したかまでわからなければなりません（特に多数のポートを監視している場合など）。

この場合は以下の RAM の“入力変化割り込みフラグエリア”の解析を行うことで確認できます。

アドレス	内 容 (16 ビット)
000h Σ 0DEh	他の目的で使用
0E0h	デバイス番号=3、2、1、0の入力変化割り込みフラグ
0E2h	デバイス番号=7、6、5、4の入力変化割り込みフラグ
Σ	Σ
0FCh	デバイス番号=59、58、57、56の入力変化割り込みフラグ
0FEh	デバイス番号=63、62、61、60の入力変化割り込みフラグ
100h Σ 1FFh	他の目的で使用

注：インターフェースモード3（IF1=High、IF0=Low）のアドレスマップです

アドレスマップ中の“0E0h”番地の詳細を示します。入力変化の監視設定と同じイメージです。



16 ビット CPU が“0E0h”番地をみる場合、上図のようになります。

4 ビットごとにひとつのデバイス番号に対応します。

4 ビットの内訳は、下位側からポート0、ポート1、ポート2、ポート3に対応します。

このなかで、“1”となっているビットがあれば、それに対応するデバイス番号のポート番号の入力値が変化したことを表します。

例えば以下のアドレスの値である場合、デバイス番号=58の、ポート3とポート0の入力状態が変化したと判断できます。

G9001A : RAM アドレス=0FCh (デバイス番号=59、58、57、56の部分)  
読み出しデータ=0900h

“1”になったビットは自分で“0”にしなければなりません（戻さなくても割り込みは発生するのですが、ローカル制御 LSI の特定が困難になります）。

“0”に戻すには、該当するビットへ“1”を書き込んでください。単純に読み出したデータをそのまま書き込む事で“0”に戻せません。

## 3.10 G9103C、G9004A のステータス状態の変化を認識するには？

基本的には I/O ポートの入力変化割り込み機能を使用しますが、注意すべき点がありますので、よく理解していただけるよう、お願いします。

### 3.10.1 ステータス変化を監視するローカル制御 LSI の指定

設定方法自体は、I/O 制御 LSI の“入力値変化を監視するローカル制御 LSI の指定”の項目と全く同じです。

先に述べたとおり、I/O 制御 LSI 以外のローカル制御 LSI は、サイクリック通信で自分のステータス情報をセンター制御 LSI (G9001A) へ送っています。これらは I/O 制御 LSI の I/O データと同じエリアへ格納されることも説明しました。

それでは、I/O 制御 LSI の各ポートに相当する部分にはどのような情報が割り付けられているのでしょうか。それを以下に示します。

#### G9103C

ポート 3	ポート 2	ポート 1	ポート 0
汎用 I/O 出力データ	汎用 I/O 入力データ	メインステータス (上位)	メインステータス (下位)

#### G9004A

ポート 3	ポート 2	ポート 1	ポート 0
動作指示等	ステータス 2	ステータス 1	ステータス 0

色がついた部分はローカル制御 LSI からデータが送られてくる部分で、それ以外はセンター制御 LSI からローカル制御 LSI へ送られるデータの部分です。つまりポート 0 からポート 2 まではローカル制御 LSI からステータスなどが送られてきて、センター制御 LSI 内の RAM へセットされます。

デバイス番号と RAM アドレスの関係は“G9002A の入力ポートのデータ取得方法は？”の項目での説明と同じです。

つまりポート 0 からポート 2 までの部分が入力データであると考えれば、この部分を監視するように設定 (“入力値変化を監視するローカル制御 LSI の指定”の項目と同じです) すれば良いということとなります。

### 3.10.2 ポート 0 の監視上の注意 (変化割り込みの使用時)

G9103C (モーター制御 LSI) と G9004A (CPU エミュレーション LSI) の、ポート 0 に相当する部分 (G9103C : メインステータスの下位 8 ビット、G9004A : ステータス 0) の状態監視は、通常とは若干異なります。

他のポートは、8 ビットデータのどこかが変化すれば割り込みが発生しますが、ポート 0 に相当する部分はビット 0 のみしか監視しません。しかも “0” → “1” の状態遷移のみの監視となります (I/O 制御 LSI が接続されている場合は、全ビットを監視します)。

各データ制御 LSI のユーザーズマニュアルを見ていただければわかるのですが、ポート 0 に相当するビット 0 は、他のビット (各ローカル制御 LSI 内の割り込み要因を示す部分) の論理和で構成されています。つまりビット 0 の監視は、各データ制御 LSI の割り込み要求を監視することとなります。

そこで、お客様がどこに注意すべきかですが、一度発生した状態変化割り込みは、その要因となったデータ制御 LSI 側の割り込み要因を解決しなければ、次に発生するかもしれない状態変化割り込みをとらえられないということです。さっぱりわからないので図解しましょう。

G9103C (モーター制御 LSI) で話を進めます。

- ① モーター制御 LSI の割り込み発生前では、モーター制御 LSI のメインステータス下位側と、対応するポート 0 の入力変化フラグは以下の状態です。

	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0
メインステータス	0	0	0	0	0	0	0	0	入力変化フラグ	0	0	0	0	0	0	0	0

- ② モーター制御 LSI の動作停止により、モーター制御 LSI のメインステータスが変化したとします。

	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0
メインステータス	0	0	0	0	0	0	1	1	入力変化フラグ	0	0	0	0	0	0	0	1

モーター制御 LSI のメインステータスのビット 1 が動作停止割り込みを示すので、ここが “1” になります。さらにビット 0 はビット 3~1 の論理和なので、ここも “1” となります。

この状態がサイクリック通信でポート 0 の部分 (G9001A 内の RAM) へ反映されます。

このとき変化割り込みが発生し、同時に入力変化フラグが “1” になります。

- ③ CPU から、G9001A 内の RAM の入力変化フラグをクリアします。

これは入力変化フラグを読み出したデータを書くことで行えます。

	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0
メインステータス	0	0	0	0	0	0	1	1	入力変化フラグ	0	0	0	0	0	0	0	0

クリア

- ④ 次にエラー割り込みが発生したとしましょう。

	7	6	5	4	3	2	1	0		7	6	5	4	3	2	1	0
メインステータス	0	0	0	0	0	1	1	1	入力変化フラグ	0	0	0	0	0	0	0	0

G9103C のメインステータスのビット 2 が “1” になります。

ビット 0 も “1” になるはずなのですが、すでに “1” なので変化なしと見なします (データ制御 LSI が割り当てられたポートデータエリアのポート 0 は、ビット 0 しか監視しません)。

つまりここで発生した割り込みは、GPU まで伝わらないこととなります。

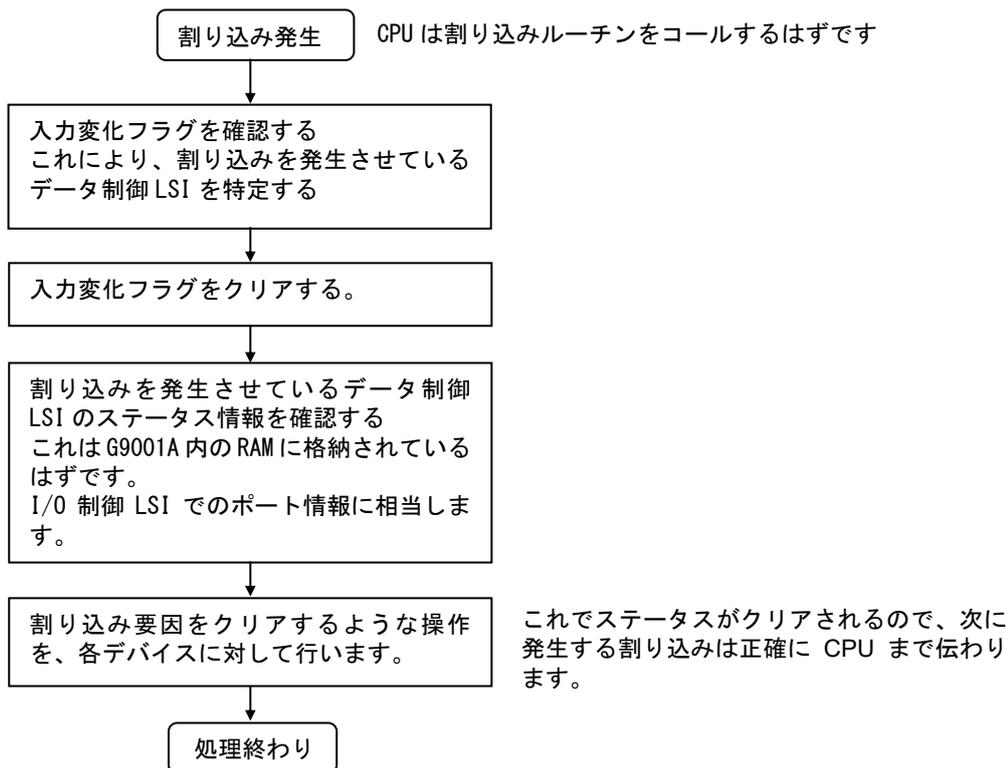
ここで重要な操作が行われていませんでした。

GPU により入力変化フラグをクリアしただけで、割り込み要因が解決されたと勘違いするケースが多いようです。

しかし実際には各データ制御 LSI 側の割り込み要因もクリアしなければなりません。今回の例では、G9103C のメインステータスに現れている “1” をクリアしなければなりません。

入力変化フラグのクリアは、割り込み要因のクリアではなく、割り込みを発生させたデータ制御 LSI を特定する情報のクリアでしかないということです。

では具体的に何をすべきなのでしょう。要因チェックの手順を示します。



次に要因クリアの操作をデータ制御 LSI 別に表で示します。

データ制御 LSI	ビット位置	割り込み要因	クリア操作概要
G9103C	1	動作停止割り込み	割り込みリセットコマンド "0008h" の送信
	2	エラー割り込み	G9103C の REST レジスタ読み出し
	3	イベント割り込み	G9103C の RIST レジスタ読み出し
G9004A	1	ローカル側受信処理完了割り込み	受信処理完了割り込みリセットコマンド (0400h) の送信
	2	ローカル側 LIRQ 入力割り込み	LIRQ 入力割り込みリセットコマンド (0500h) の送信 さらに G9004 が制御している周辺 LSI への対応処理など
	3	ローカル側送信要求割り込み	メッセージ通信コマンド (0001h) の送信

コマンドの送信やレジスタからの読み出し操作は、全てデータ通信経由での指示にて行われます。  
詳しくは各データ制御 LSI のユーザーズマニュアルを参照ください。

### 3.11 G9103C を動作させるには？

G9103C は弊社製 LSI 製品である PCL シリーズと同様で、モーターを制御するためのパルス列を出力できる LSI 製品です。基本的なコマンド体系や操作は従来の PCL に類似しています。従来の PCL と異なるのは、直接 LSI を制御する CPU が無く、代わりに G9001A からコマンドを受け取って動作するという点です。

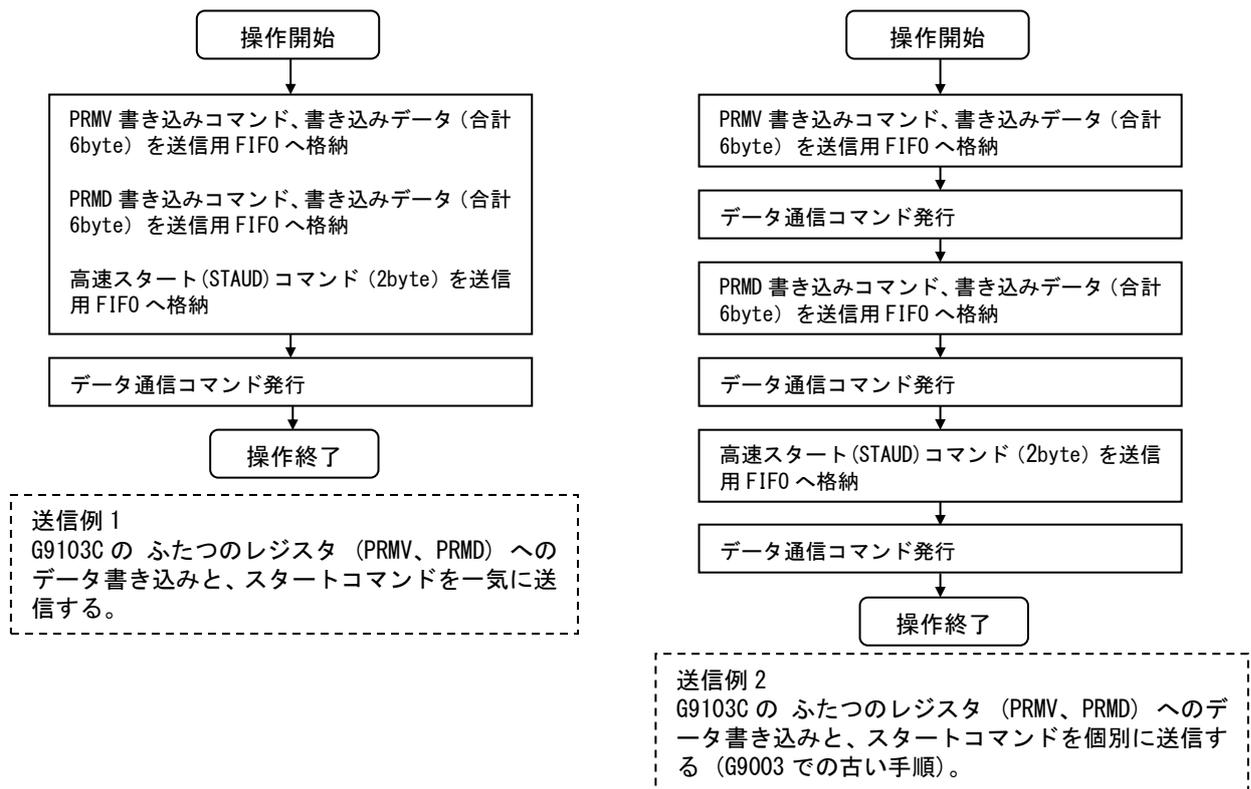
では G9001A はどのようなコマンドを、どのように G9103C へ渡すのでしょうか。

まずコマンドは、先に述べたとおり従来の PCL に類似しています。このコマンドをデータ通信で G9103C へ渡します。

具体的には、コマンドを G9001A の送信用 FIFO へ格納し、データ通信コマンドを発行することで、送信用 FIFO へ格納させたコマンドを G9103C へ渡せます。

一度のデータ通信で、G9103C の持つ複数のレジスタに対するデータの書き込みが可能です。

また、旧タイプ製品 (G9003) に対する操作のように、レジスタ単位で渡すことも、当然可能です。



より詳細なフローがユーザーズマニュアル (G9001A または G9103C) にあるので、参照ください。

## 3.12 G9004A とのデータのやりとりの方法は？

基本的な操作は、「データ通信のやり方は？」の項目と同じです。

また、以降で示す手順では、入力変化割り込みの発生などが適切に設定されていることを前提とします。

### 3.12.1 G9004A のステータスに関する注意

G9004A のマニュアルを見ると、2 種類のステータス情報が記載されており、混乱するかもしれません。

2 つの違いを整理しましょう。

ひとつ目のステータスは、サイクリック通信で G9001A へ送られる情報です。G9004A ユーザーズマニュアルの、次の項目を参照ください。

ステータス情報と G9004A への動作情報の設定

ふたつ目のステータスは、G9004A をメッセージ通信モードで使用する場合のものです。

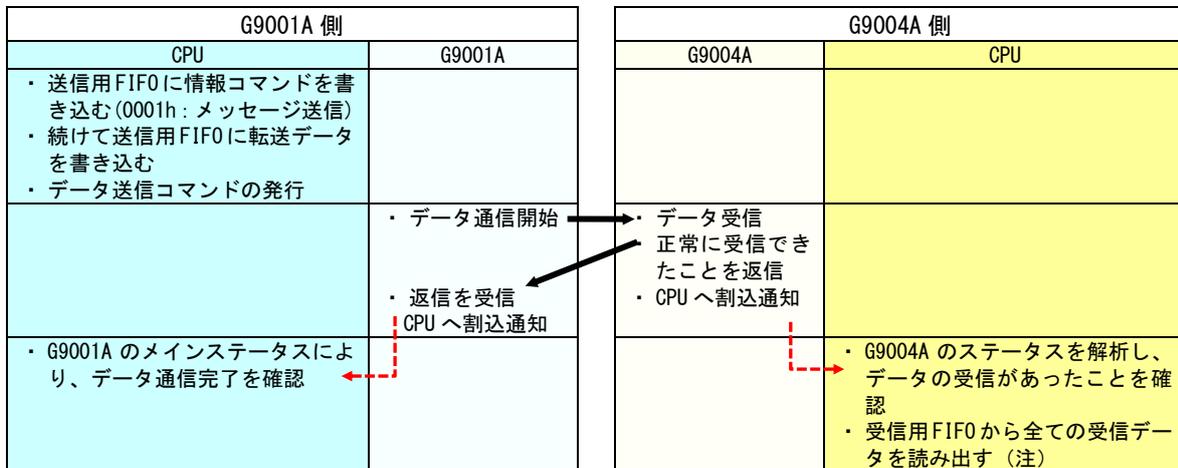
G9004A のユーザーズマニュアルの、次の項目を参照ください。

ローカル側 CPU が操作できるコマンド・ステータス情報

このステータス情報は、G9004A 側の CPU が利用するものです。

### 3.12.1 メッセージ通信

#### 3.12.1.1 G9001A から G9004A 側 CPU へデータを渡す

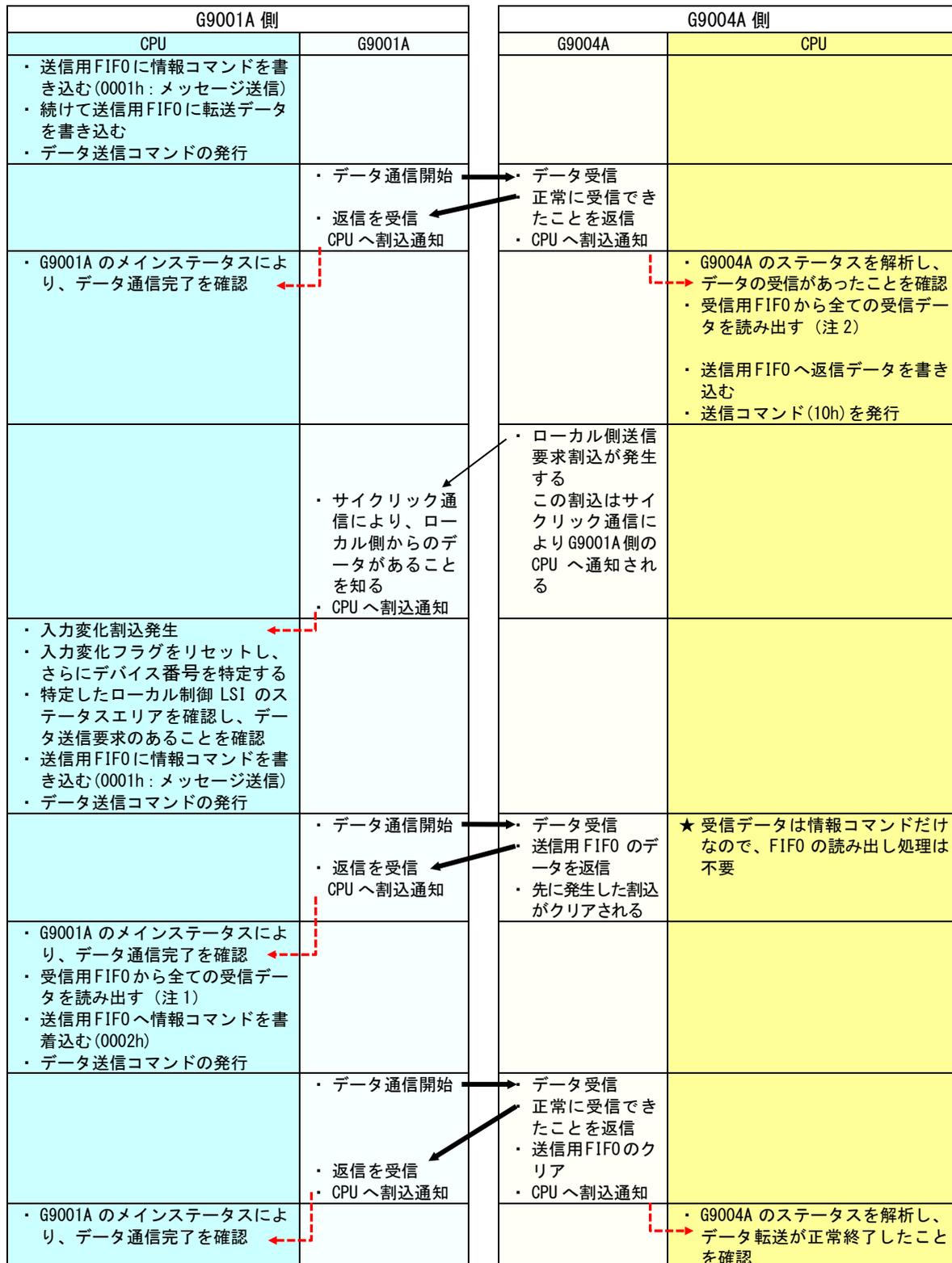


太線の矢印はデータ通信を表し、矢印の向きは方向を表します。

点線は CPU に対する割り込みの発生を表します。

注：FIFO から受信データを読み出す場合、G9004A のステータスのビット 5 が “0” になるまで読み出し処理を繰り返すことで全てのデータを取り出せます。

## 3.12.1.2 G9001A から G9004A 側 CPU ヘデータを渡し、返信データをもろう



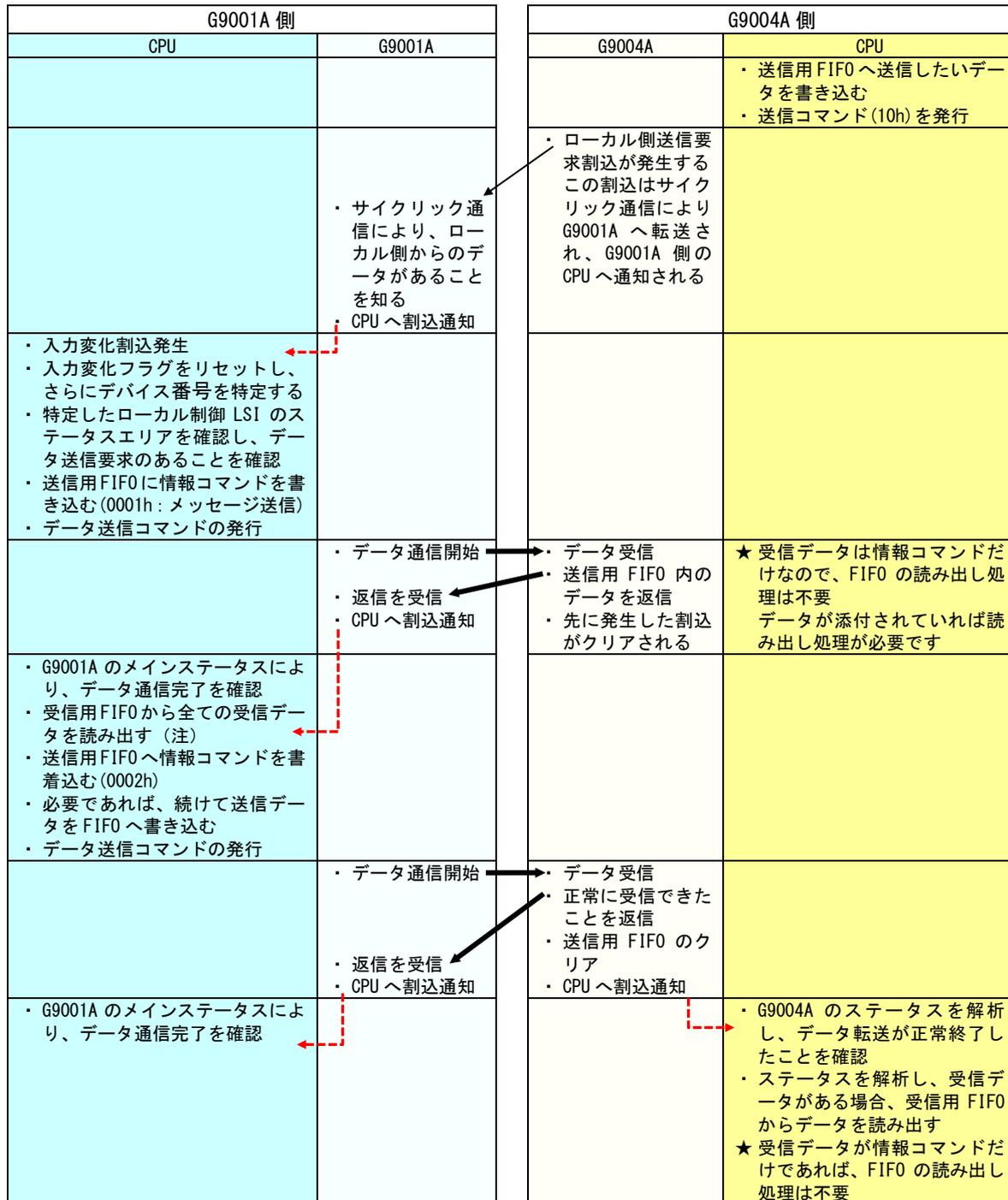
太線の矢印はデータ通信を表し、矢印の向きは方向を表します。

点線は CPU に対する割り込みの発生を表します。

注 1: FIFO から受信データを読み出す場合、G9001A のメインステータスのビット 10 が "0" になるまで読み出し処理を繰り返すことで全てのデータを取り出せます。

注 2: FIFO から受信データを読み出す場合、G9004A のステータスのビット 5 が "0" になるまで読み出し処理を繰り返すことで全てのデータを取り出せます。

## 3.12.1.3 G9004A 側 CPU から G9001A へデータを渡す



太線の矢印はデータ通信を表し、矢印の向きは方向を表します。

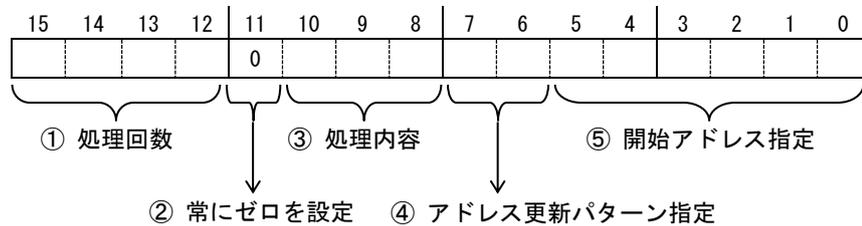
点線は CPU に対する割り込みの発生を表します。

注: FIFO から受信データを読み出す場合、G9001A のメインステータスのビット 10 が "0" になるまで読み出し処理を繰り返すことで全てのデータを取り出せます。

### 3.12.2 CPU エミュレーションモード時のコマンドについて

G9001A 側から、G9004A に対して発行できるコマンドに関して説明します。

メッセージ通信モード時は、使用できるコマンドは3種類しかなく、かつ単純なものなので特に説明しませんでした。これに対して CPU エミュレーションモード時のコマンドは若干複雑です。以下に詳細を示します。

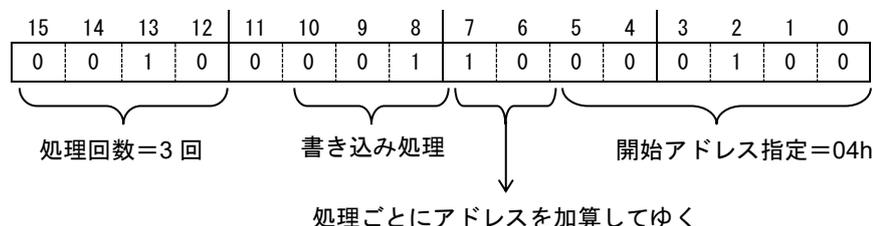


- ① 処理回数  
“開始アドレス指定”で指定したアドレスから、“処理内容”で指定した処理を繰り返し行いたい場合、ここに(希望回数-1)を設定します。ゼロを設定すると、動作は1回です。
- ② 常にゼロを設定  
古いマニュアル類には「通信待機設定」と記載されていますが、使い勝手がよくありません。ここでは常にゼロでの運用を推奨します。
- ③ 処理内容  
以下が利用できます。
 

000b : NOP	.....	無効コマンド
001b : 書き込み処理	.....	G9004A が制御する周辺 LSI ヘデータを書き込む
010b : 読み出し処理	.....	G9004A が制御する周辺 LSI からデータを読み出す
011b : データ送信処理	.....	G9004A の FIFO にたまっているデータの送信要求
100b : 受信処理完了割込リセット	.....	G9004A に発生した“受信処理完了割込”のクリア
101b : LIRQ 入力割込リセット	.....	G9004A に発生した“LIRQ 入力割込”のクリア
- ④ アドレス更新パターン指定  
処理回数を設定した場合に有効です。以下が利用できます。
 

00b : アドレス固定
01b : アドレス固定 (00b と同じです)
10b : 処理ごとにアドレスを加算してゆく (16bit アクセスなら+2)
11b : 処理ごとにアドレスを減算してゆく (16bit アクセスなら-2)
- ⑤ 開始アドレス指定  
処理を行うための先頭アドレスを指定します

例えば以下 (2184h) のように設定した場合の、G9004A の動作を説明します。



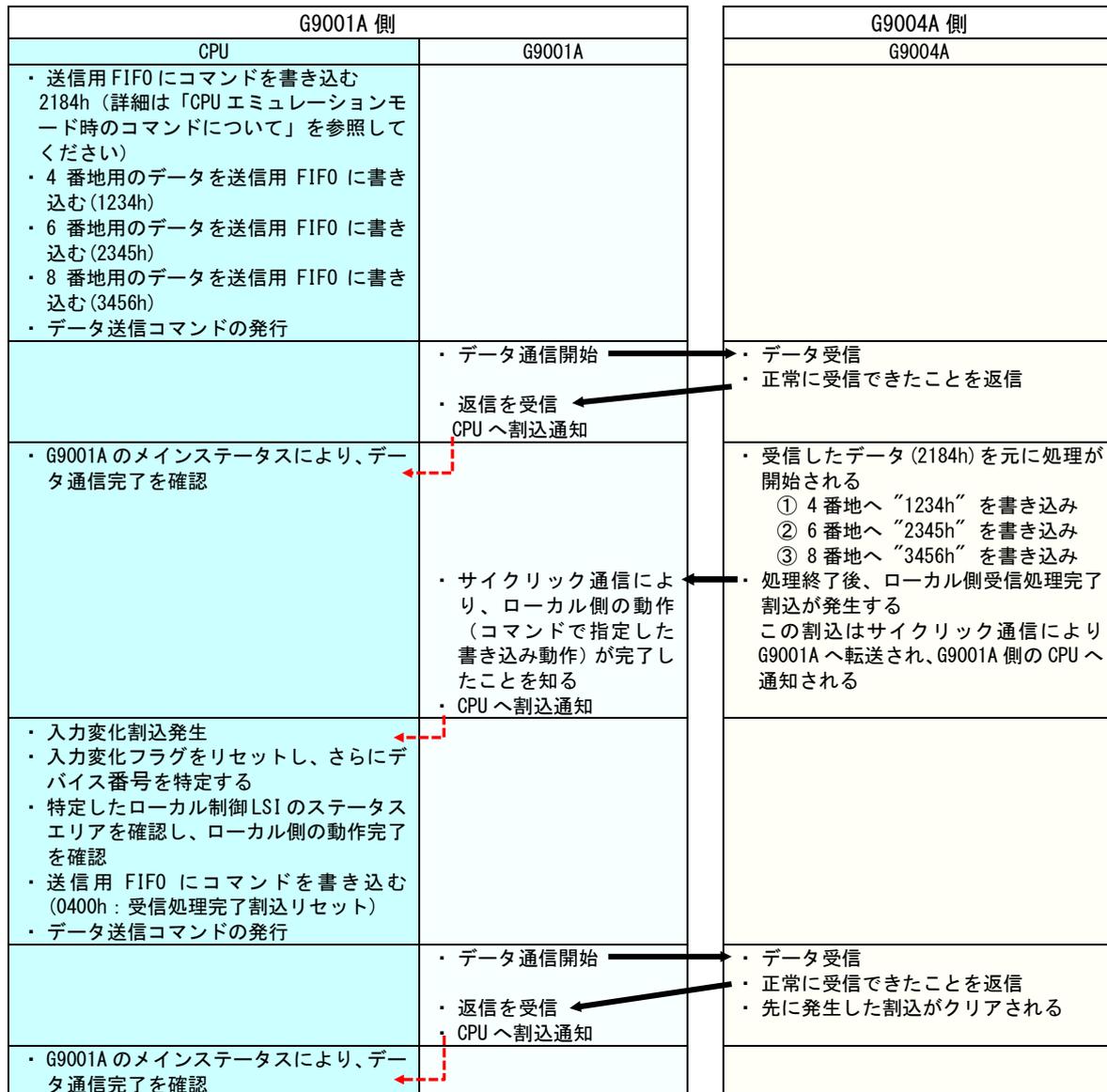
アドレス=4番地から順番に、3回の書き込み処理を行います。

このとき書き込むデータは、このコマンドの後に続けて指定します。詳細は次項を参照ください。

### 3.12.3 CPU エミュレーションモード

#### 3.12.3.1 連続データの書き込み

G9004A が制御する周辺 LSI の、4 番地（アドレスバス=000100b）に “1234h” というデータを書き込み、続けて 6 番地に “2345h”、8 番地に “3456h” という 3word (6byte) 分の処理を行う動作を想定します。

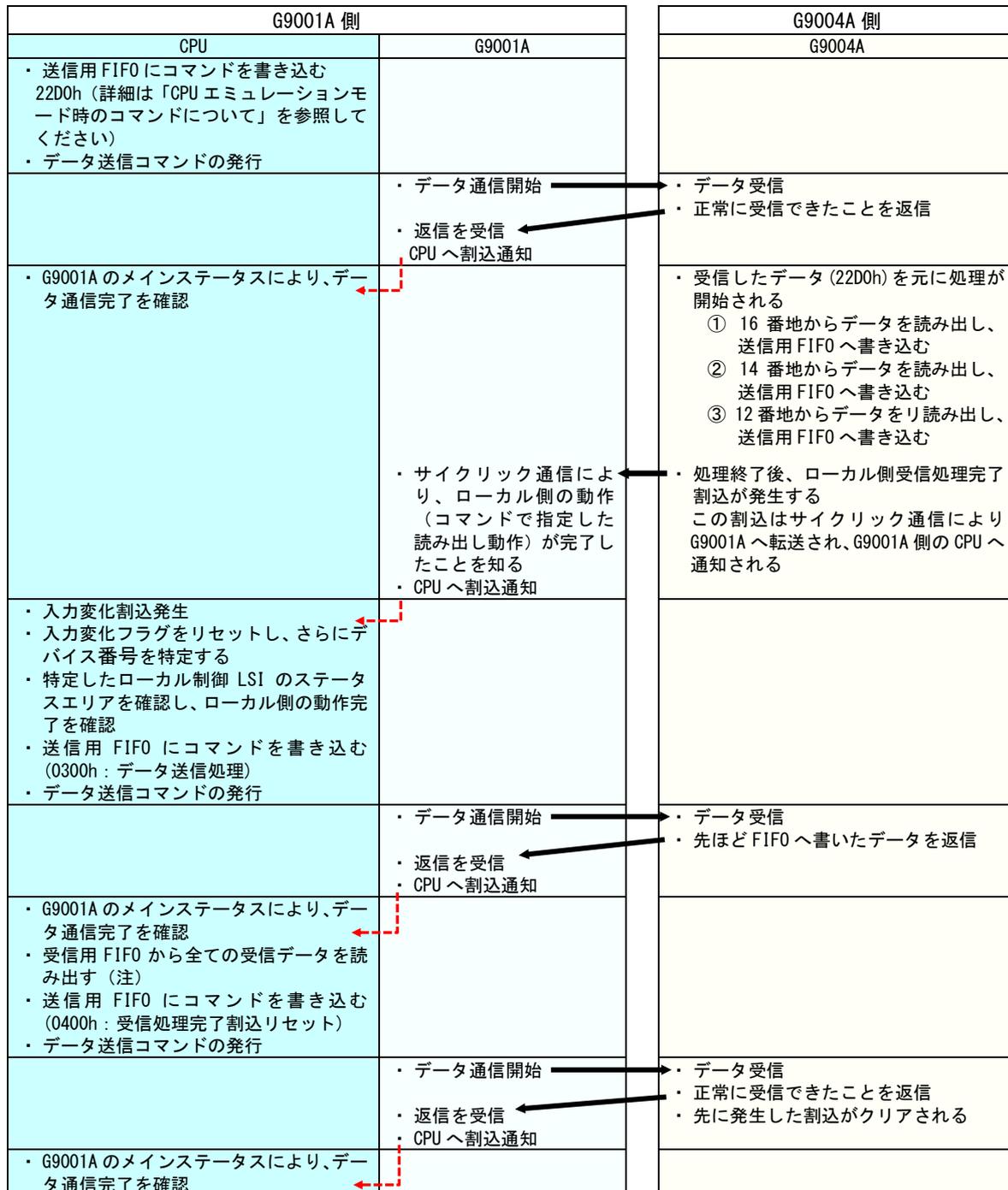


太線の矢印はデータ通信を表し、矢印の向きは方向を表します。

点線は CPU に対する割り込みの発生を表します。

### 3.12.4 連続データの読み出し

G9004A が制御する周辺 LSI の、16 番地（アドレスバス=010000b）、14 地、12 番地から、合計で 3word (6byte) 分のデータを読み出す処理を行う動作を想定します。



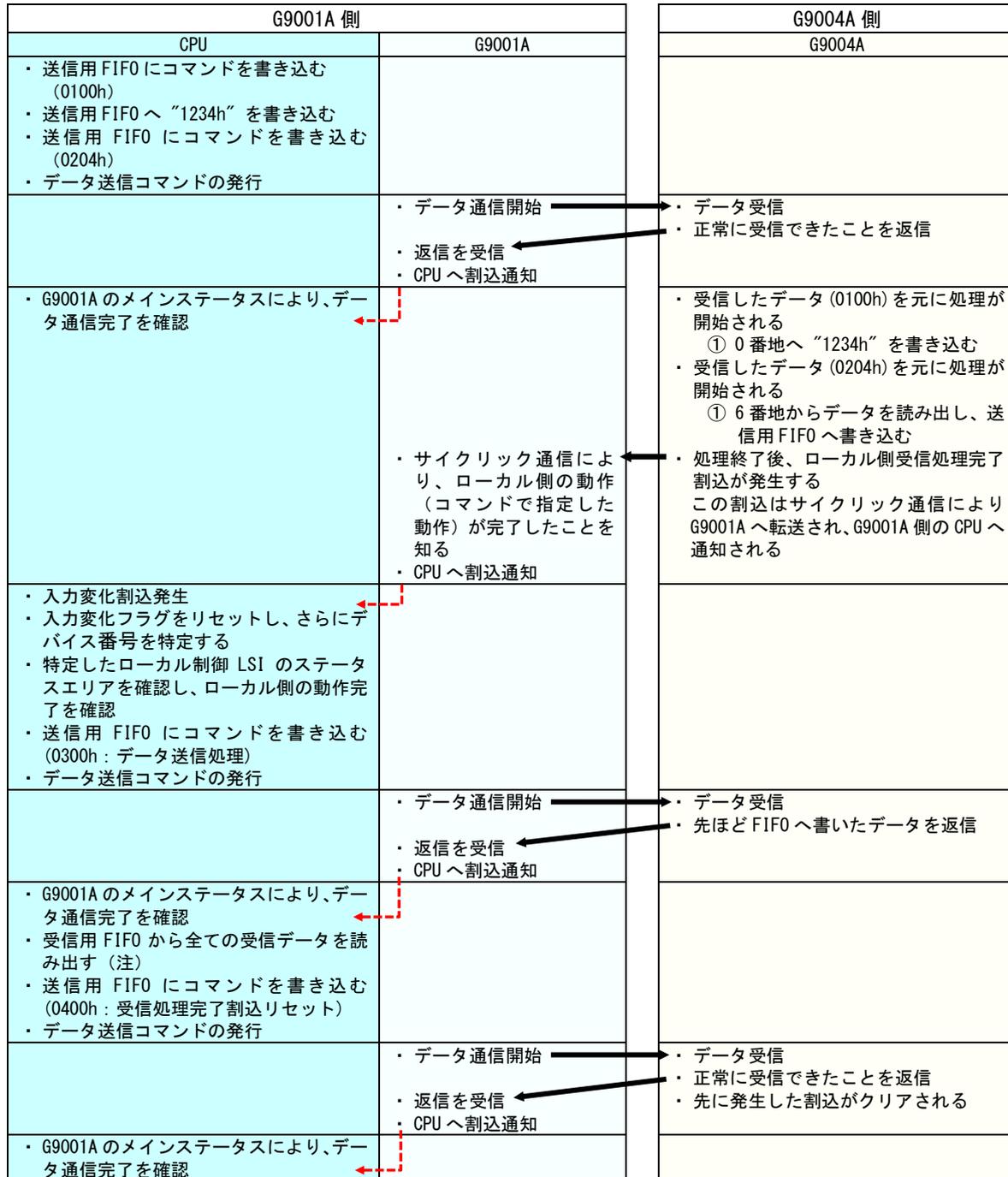
太線の矢印はデータ通信を表し、矢印の向きは方向を表します。

点線は CPU に対する割り込みの発生を表します。

注 : FIFO から受信データを読み出す場合、G9001A のメインステータスのビット 10 が "0" になるまで読み出し処理を繰り返すことで全てのデータを取り出せます。

### 3.12.5 組み合わせ処理

G9004A が制御する周辺 LSI の、0 番地（アドレスバス=000000b）へ “1234h” というデータを書き込み、6 番地から、合計で 1word(2byte) 分のデータを読み出す処理を行う動作を想定します。



太線の矢印はデータ通信を表し、矢印の向きは方向を表します。

点線は CPU に対する割り込みの発生を表します。

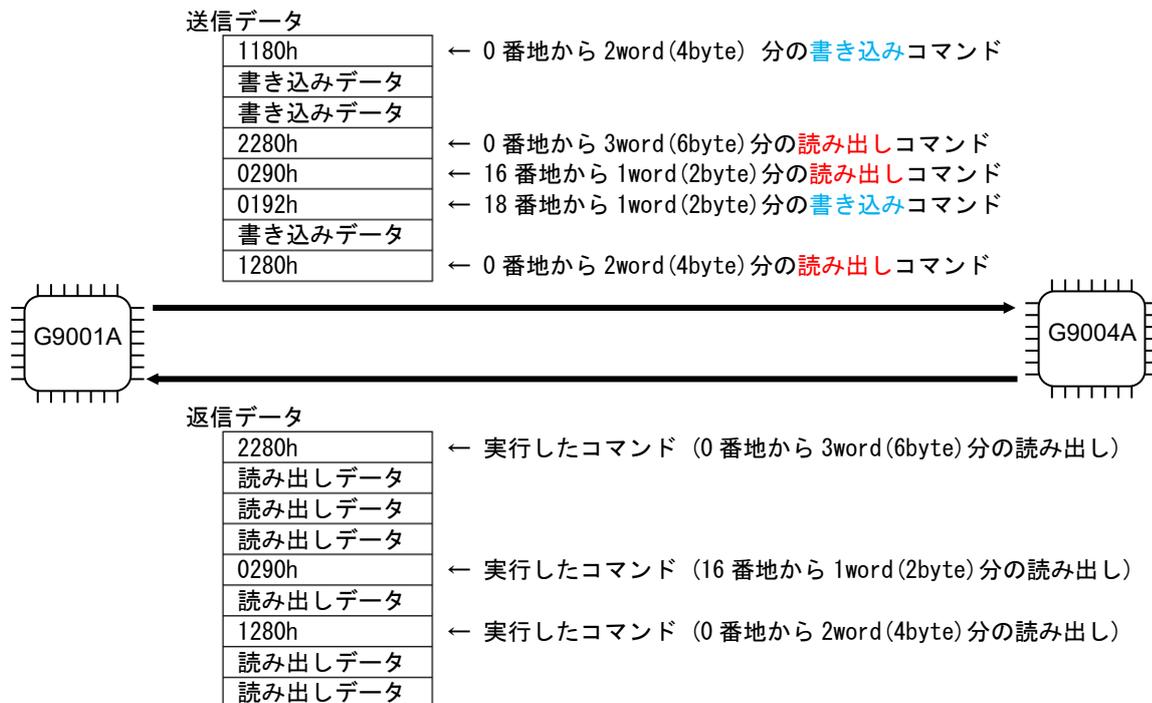
注 : FIFO から受信データを読み出す場合、G9001A のメインステータスのビット 10 が “0” になるまで読み出し処理を繰り返すことで全てのデータを取り出せます。

### 3.12.5.1 読み出しコマンド使用時に G9004A から返信されるデータ形式

G9004A が制御する周辺 LSI からデータを読み出すわけですから、当然読み出したデータが返送されてきます。しかし返信データには読み出したデータ以外にも、そのデータの読み出しに使用されたコマンドが付加された形で戻ってくるので注意してください。

例えば以下のように、書き込みコマンドと読み出しコマンドが混在した場合（読み出しコマンドだけのケースでも同様ですが）、送信されたコマンド群は G9004A 側で実行されます。

実行された結果、読み出されたデータが G9001A へ返信されるのですが、その返信データは次に示すような内容となります。



まず読み出しコマンドがそのまま入り、その後実際に読み出されたデータが続きます。

G9001A の CPU では、発行した読み出しコマンドと、読み出されたデータの対比を行いながら処理を進められます（発行した読み出しコマンドを覚えておく必要がなくなります）。

### 3.12.5.2 読み出しコマンドでの返送データが 256byte を越える場合

読み出しコマンドで G9004A が制御している周辺 LSI からデータを読み込む場合、読み出したデータの前に読み出しコマンドが添付されることは前項で述べました。

このように、読み出しコマンドによって G9004A から返ってくるデータは予想以上に大きなものとなります。

これらのデータが 256byte を越えた場合、越えた分のデータは保証されません。

またエラーの発生もありませんので、十分気をつけてください。

### 3.13 I/O 制御 LSI に対してデータ通信を行うとどうなるか？

I/O 制御 LSI (G9002A 等) に対してデータ通信を行ってはいけません。

I/O 制御 LSI は受信用 FIFO を持っていませんので、データ通信を受信できません。

間違って I/O 制御 LSI へデータ通信を行った場合、I/O 制御 LSI からはエラー応答と呼ばれるフレームが返送されてきます。この場合、G9001A の割り込みステータスが次のようになり (ビット状態の記載がない部分は、本説明とは無関係なので省略されています)、割り込みが発生します。

割り込みステータス (ISTW)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				0	0	1	0								

ERA 3~0

### 3.14 I/O 制御 LSI と G9001A で異なる PMD の設定は可能か？

I/O 制御 LSI (G9002A 等) は、PMD と呼ばれる外部端子があり、この外部端子の設定で 複数ある入出力ポートの I/O 属性を決定します。

一方 G9001A には、制御対象となる I/O 制御 LSI が持つ入出力ポートの I/O 属性がどうなっているかを設定する「デバイス情報エリア」があります。

サイクリック通信時には、このエリアな情報を参考にして ポートデータのやりとりが行われます。このため G9001A に設定した入出力ポートの I/O 属性情報と、実際の I/O 制御 LSI の I/O 属性の状態が一致していなければなりません。

通信処理開始前に、G9001A でシステム通信を行い、通信ラインに接続されているローカル制御 LSI の情報を自動収集した場合、通信ライン上の I/O 制御 LSI の I/O 属性情報も収集されて G9001A の「デバイス情報エリア」に設定されます。このためデバイス情報エリアの情報と実際の I/O 制御 LSI の I/O 属性の状態が食い違うことはありません。

しかし、CPU 側からこのデバイス情報エリアを手動で設定した場合などは食い違う可能性があります。食い違った状態でサイクリック通信が行われると、サイクリック通信を受け取った I/O 制御 LSI はエラー応答を返します。

この場合、G9001A の割り込みステータスが次のようになり (ビット状態の記載がない部分は、本説明とは無関係なので省略されています)、割り込みが発生します。

割り込みステータス (ISTW)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				0	0	0	1								

ERA 3~0

### 3.15 G9103C の受信 FIFO 容量以上のデータを送信できるか？

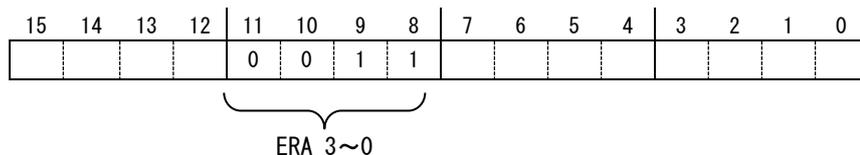
G9001A の送信用 FIFO は 256byte あります。

これに対し、G9103C の受信用 FIFO は 128byte しかありません。

仮に G9001A の送信用 FIFO に 128byte を超えるのデータを書き込んで、これを G9103C へ送信した場合、G9103C はエラー応答を返します。

この場合、G9001A の割り込みステータスが次のようになり（ビット状態の記載がない部分は、本説明とは無関係なので省略されています）、割り込みが発生します。

割り込みステータス (ISTW)



### 3.16 G9004A の受信用 FIFO のデータを読み出さない場合は？

G9004A をメッセージ通信モードで使用する場合、G9004A 自体はローカル側 CPU により制御されます。

そして G9001A からのデータ通信を受け、受信用 FIFO に受信データが格納された場合は、G9004A 側の CPU が、受信用 FIFO 内のデータを読み出さなければなりません。

ここで、G9004A 側の CPU が別の処理に追われていて、受信用 FIFO の読み出し処理を保留にするとどうなるでしょうか？

実は G9004A は、受信用 FIFO 内にデータが残っていると、次のデータ通信を受信しません（無視します）。G9001A 側から一斉懸命データ通信を行っても 無応答となります。

G9001A は 3 回のリトライで処理をあきらめるので、目的のデータ通信を行えません。

ですので、G9004A がデータを受信した場合、速やかに受信用 FIFO からデータを読み出してください（受信用 FIFO のリセットでも良い）。

ただし、G9004A の受信用 FIFO にデータが残っていてもサイクリック通信には正常に応答します。

### 3.17 通信が不可能な状態になると各 LSI はどうなるか？

通信ラインに何らかの障害が発生し、G9001A から、各ローカル制御 LSI 側へ通信ができなくなるとどうなるでしょうか？

各ローカル制御 LSI は G9001A からの自分宛の通信が来なくなるので、身動きがとれなくなります。

このような場合、各ローカル制御 LSI に内蔵されているウォッチドックタイマーが働きます。

このタイマーは各ローカル制御 LSI の外部端子により指定された待ち時間をカウントしています。カウンタが待ち時間をカウントし終わる前に自分宛の通信を受信すると、カウンタがクリアされます（実際のクリアタイミングは、サイクリック通信に対する応答を行った後です）。

そして、カウンタが待ち時間をカウントし終わっても自分宛の通信が来ないときはタイムアップとなり、次のような処置が行われます（外部端子 TUD で動作を選択します）。

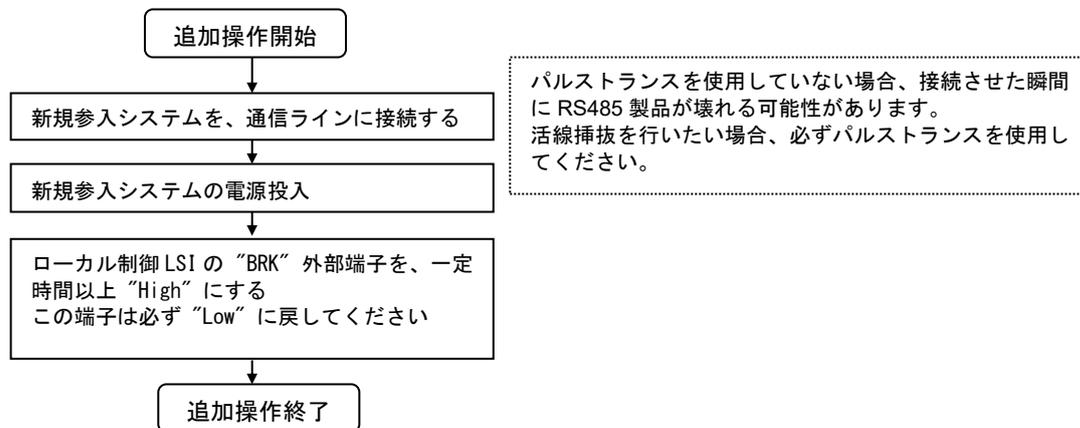
TUD 入力端子	処 理
High	タイムアップ時のローカル制御 LSI の状態を保持したままで、自分宛の通信の再開を待ちます。 この場合、ローカル制御 LSI の出力端子の出力状態はタイムアップ時の状態を保持します。 G9103C がパルス列を出力している場合、その動作が継続されます。
Low	LSI をリセットし、自分宛の通信の再開を待ちます。 ローカル制御 LSI の出力端子の状態も初期化されます。



### 3.19 稼働中に新たなローカル制御 LSI を参入させるには？

何らかの原因で通信ラインから脱落してしまったローカル制御 LSI や、全く新しいローカル制御 LSI を、サイクリック通信へ新規に参入させる場合、サイクリック通信を中断しなくとも可能です（注）。

このときの手順を示します。



G9001A は、一定周期で新規参入したローカル制御 LSI がないかどうかを監視しています。

ここで、ローカル制御 LSI の BRK 外部端子に一定時間（3.2ms 以上）の “High” 信号を入力すると、G9001A に対して新規参入したいローカル制御 LSI があることを知らせることができます。

新規参入したいローカル制御 LSI があることを察知した G9001A は、メインステータスのビット 1 (BRKF) を “1” とし、CPU へ割り込みを発生させます。

これを受けた CPU は、G9001A に対して「システム通信」コマンドを発行してください。

これで新規参入したローカル制御 LSI の情報を取得できます。取得後は、この新規したローカル制御 LSI に対してもサイクリック通信が行われるようになります。

注：通信ラインとローカル制御 LSI のシステムが、パルストランスなどにより絶縁されていない場合、回線接続時のノイズなどでラインドライバが破壊される可能性があります。

このような場合はお手数ですが、通信ラインに接続されているシステムの全ての電源を落としてから接続操作を行ってください。

### 3.20 通信状態を外部から監視するには？

通信に異常がないかどうかを判断できる外部端子が用意されています。

G9001A と、それ以外のローカル制御 LSI で端子状況が異なります。以下の表を参照ください。

種 類	端子名	内 容
センター制御 LSI (G9001A)	MCRY	通信ライン上の状態を示します。 通信ライン上にフレームが流れている場合に “Low” となります。
	MERR	以下のケースで 3.2 $\mu$ s だけ “Low” となります。 ① ローカル制御 LSI からのフレームを正常に受け取れなかった場合 (CRC コードでエラーがあった場合)。 ② ローカル側への通信完了後、20 $\mu$ s 待っても応答がなかった場合 (無応答と判断)。
	MERF	エラー応答を受信した場合に 3.2 $\mu$ s だけ “Low” となります。
	MSYN	サイクリック周期毎に出力信号のレベルが反転します。
ローカル制御 LSI	MSEL	受信成功 センター制御 LSI (G9001A) からの通信を正常に受信できた場合、一定の時間 “Low” となります。
	MRER	受信失敗 センター制御 LSI からの通信を正常に受け取れなかった場合 (CRC コードでエラーがあった場合など)、一定の時間だけ “Low” となります。
	TOUT	通信停止 一定時間内に自分宛の通信が来なければ、ウォッチドックタイマーがタイムアップし、それと共に本端子が “Low” となります。 サイクリック通信が再開されると “High” になります。

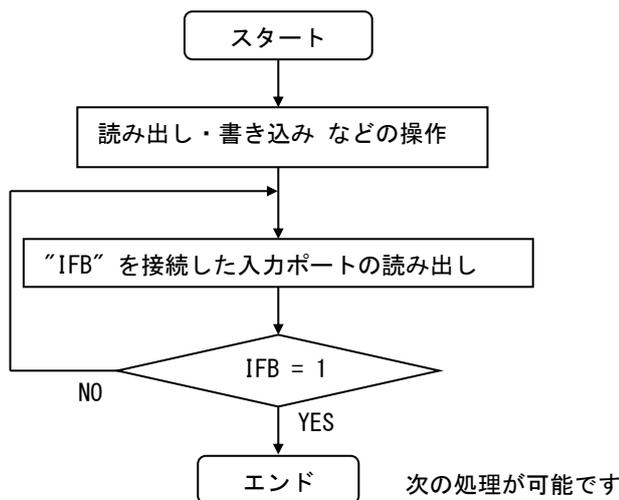
### 3.21 ウェイト制御用端子の無い CPU を使用するには？

G9001A と G9004A はパラレルバスを持つ CPU により制御されます。このとき CPU からの読み出し操作や書き込み操作に対して、内部処理時間が必要となります。

この内部処理中は、CPU が次の操作を行えないように待たせるための“WRQ”（G9004A では“LWRQ”）という外部端子を持っています。通常はこの信号を、CPU のウェイト端子へ接続する事で自動的に CPU の処理を待たせることができます。

しかし、世の中にはウェイト端子を持たない CPU もあります。そのような CPU を使用する場合、以下の手順で CPU を待たせるようにすると効率的です。

前提：“IFB”（G9004A の場合“LIFB”）を、使用する CPU の入力ポート端子（場所は任意）へ接続してください



G9001A のユーザーズマニュアルに詳細な説明があるので、そちらも参照ください。

## 3.22 アドレスバスを 2 本しか使わないで G9001A を制御する方法

G9001A はアドレスバスを 9 本持っています。

これは 512byte ある内蔵メモリへ直接アクセスできるようにしたため、通常は CPU から 9 本(16 ビット CPU なら 8 本)のアドレス信号を接続して使用します。

しかしアドレス信号を 2 本 (8 ビット CPU は 3 本) しか接続しなくても G9001A の制御は可能です。

これは G9001A へのコマンド操作によって内蔵メモリへアクセスする機能を利用するもので、従来の弊社製 LSI 製品である PCL シリーズの操作と同じものです。

G9001A のユーザーズマニュアルに詳細な説明があるので、そちらも参照ください。

## 3.23 40MHz クロック使用時の注意点は？

G9000 シリーズはシステムクロックとして 40MHz もしくは 80MHz の接続を選択できます。どちらのクロックを使用しても機能的には変わりません。その選択基準は何でしょうか？

詳しい説明は各ユーザーズマニュアルに譲るとし、ここでは単純に、以下のように考えてかまいません。

- G9000 シリーズの LSI に接続されているクロックは専用クロックで、1 個の G9000 シリーズの LSI しか駆動していない  
対処→40MHz のオシレータで問題ないでしょう
- G9000 シリーズの LSI に接続されているクロックは兼用クロックですが、2~3 個程度の G9000 シリーズの LSI しか駆動していない  
対処→40MHz のオシレータで問題ないでしょう
- G9000 シリーズの LSI に接続されているクロックは兼用クロックで、3 個を超える複数個のローカル制御 LSI を駆動している  
対処 1→新たに 40MHz の専用クロックをもうけてください  
対処 2→兼用クロックを 80MHz にしてください

## 3.24 関連部品の選定基準は？

通信に関連する部品は、選定する品種により通信品質が左右されます。よりよい品質の部品を選定しましょう。ただ高品質の定義が不明瞭では選定できないので、おおまかな基準を述べてゆきます。

### 3.24.1 ケーブルの選定

LAN で使用されるタイプで、シールド付きのものを選定すれば、ほぼ問題ありません（内部のツイストペアが確実に余ってしまう問題はあります。この場合、ツイストが一番ゆるいペアを使用してください）。

当社ではカテゴリ5 以上の製品を推奨します。

また CC-Link などのフィールドバス専用のケーブルも高品質です（高価です）。

品質保証のあるケーブルは必ず特性表の入手ができます。特性表にケーブルの品質を保証する値が記載されています（単に材質の記載だけでは特性表とはいえません。特性インピーダンスや減衰量の保証は必須です）。

特性表の入手できないケーブルは使用すべきではありません。

### 3.24.2 RS485 製品の選定

20Mbps 以上のレートで送受信できるものの選定が大前提です。

それ以外に、RS485 製品のデータシートに記載されている“入力抵抗”（Input resistance）の値が大きなもの（24k $\Omega$ 以上）を選んでください（RS485 の標準は 12k $\Omega$ です）。

5V と、3.3V の製品が選択できますが、5V の製品が理想です。G9000 シリーズは 3.3V ですが、5V 製品の方が安定した通信を期待できます。

またフェイルセーフ機能付きのものを選択してください。

### 3.24.3 パルストランスの選定

20Mbps に応答できるものが大前提です。

20Mbps に応答し、かつ 4Mbps の信号通過時も飽和しないものが必須です（20Mbps で通信時）。

インダクタンスは 1000 $\mu\text{H}$  前後のものを使用してください（これより大きい分には問題ないと思われませんが、数百  $\mu\text{H}$  程度だと 64 個ものローカル制御 LSI の接続は不可能です。また 5V の RS485 製品の使用を前提としています）。

### 3.24.4 終端抵抗の抵抗値の決定

通信ラインの両端には終端抵抗が不可欠です。

終端抵抗の抵抗値は、使用するケーブルにより決定されます。

使用するケーブルの特性表に記載されている、“特性インピーダンス” の値が終端抵抗の値となります。

## 3.25 レベルシフタは必要ありません

G9000 シリーズは 3.3V 単一電源で動作する LSI です。

これに対して RS485 製品は 5V の LSI を推奨しています。

これらを接続する場合、レベルシフタなどの電圧変換が必要ではと考えますが、G9000 シリーズは全て 5V 耐圧の端子を採用しているため、5V 製品と直接接続しても問題ありません。

## 4. 付録：用語解説

### 4.1 Motionnet 全般に関連する用語

**【bps】****【通信レート】****【転送レート】****【転送速度】****【伝送レート】****【伝送速度】****【通信速度】****【レート】**

シリアル通信の伝送速度です。1 秒間に送信されるデータのビット数を示します。

Motionnet では、数値の単位としては「bps」を使用し、文書内の表現としては「通信レート」を使用します。

**【CPU 間通信】**

センター側 CPU とローカル側 CPU で、相互にデータの送受信が行える機能を提供します。

**【CPU 周辺 LSI】**

CPU により制御される LSI などの半導体製品です。

RAM、ROM などのメモリや、AD/DA コンバータなどは勿論、弊社の PCL や PCD など CPU 周辺 LSI です。

**【CPU エミュレーション LSI】**

G9004A を指します。

**【CRC コード】****【サイクリックコード】**

巡回冗長検査と呼ばれ、「Cyclic Redundancy Check」の略です。

データ量の多いシリアル通信の誤り検出によく使用されています。

**【FIFO】**

先入れ・先出し構造のメモリの呼称です。

RAM と同じで、大量のデータを記憶することができますが、RAM のようにデータ数に対応したアドレスは持っていません。

FIFO にデータを記憶させる場合、書き込み用アドレス（常に同じ値）に対して複数のデータを順番に書き込んでゆきます。

FIFO からデータを取り出す場合、読み出しアドレス（常に同じアドレスで、書き込みアドレスと同じ場合もある）から欲しいだけのデータを順番に読み出します。このとき、最初に書き込んだデータから順番に読み出されます。

**【I/O 制御 LSI】****【I/O デバイス】****【I/O チップ】**

G9002A、G9205A を指します。

汎用入出力ポートを制御するローカル制御 LSI を指します。

データ通信を受信できません。

Motionnet では「I/O 制御 LSI」の表現を使用します。

**【I/O 通信】**

サイクリック伝送を構成する最小単位です。

センター制御 LSI からローカル制御 LSI に向かうリクエストフレーム（“下り” フレーム）と、その逆のレスポンスフレーム（“上り” フレーム）のふたつ一組で構成します。

**【Motionnet】****【モーションネット】**

モーションネットは、弊社が提唱するシリアル通信システムです。

このシステムは、弊社が独自に開発した数種類の LSI “G9000 シリーズ” をコアにして、15.1 $\mu$ s/局(4byte) のシリアル通信により、入出力(I/O)制御、モーション制御、CPU エミュレーション、メッセージ通信まで行えるトータル省配線システムです。

**【PCL 制御 LSI】****【モーター制御 LSI】**

G9103C を指します。

Motionnet では「モーター制御 LSI」の表現を使用します。

**【RS485 規格】**

シリアル通信のための規格で、EIA-485 と呼びます。

一昔前でシリアル通信といえば “RS232C” でしたが、この規格をバージョンアップしたものとお考えください (実際はもっと複雑ですが)。

**【アクセス】**

メモリや I/O ポートなどからデータの書き込みまたは読み出しをすることを「アクセスする」などと表現します。

**【アドレス】**

CPU が管理するメモリ (RAM、ROM など) や I/O ポートは、ソフトウェアから見ると連続した番号が割り振られています。ソフトウェアが、とある I/O ポートにデータを出力したいときなどは、その I/O ポートに割り付けられている番号を使って指定します。この番号がアドレスです (住所にたとえられる場合が多いようです)。

アドレスのどこに、誰 (メモリなのか、I/O なのか、など) がいるのかを表にしたものは「アドレスマップ」と呼びます。

G9205A に限ってデバイス番号を指す場合があります。この場合「偶数アドレス LSI」もしくは「奇数アドレス LSI」という表現で使用されます。

**【エラー応答】**

センター制御 LSI からのリクエストフレームを、ローカル制御 LSI が拒否した場合に返送されるレスポンスフレームです。

I/O 制御 LSI のような データ通信を受信できないローカル制御 LSI に対してデータ通信を行った場合や、入出力ポートの入出力属性が異なる I/O 通信を受信した場合に、返送されます。

**【エンディアン】**

メモリや通信において、複数のバイトで表現される数値をどういう順番で扱うかの定義です。

ビッグエンディアンと、リトルエンディアンの ふたつがあります。

バイト順 (Byte Order、バイトオーダー) と呼びます。

たまに ビット順の CPU も見かけます。

**【活線挿抜】**

装置の電源を入れたまま、通信ラインのケーブルの抜き差しが可能なことです。

**【奇数アドレス LSI】**

G9205A を “センター制御ポート間モード”、または、“センターレスポート間モード” で使用する場合、必ずふたつひと組のペアで使用する必要があります。

ペアとしての指定はデバイス番号で行います。デバイス番号は外部端子 (DN5~DN0) の設定で行いますが、DN5~DN1 までが同じ状態に設定された G9205A が ペアとして扱われます。

ペア指定された G9205A のうち、DN0=Low と設定されたほうを奇数アドレス LSI と呼びます。

## 【偶数アドレス LSI】

G9205A を “センター制御ポート間モード”、または、“センターレスポート間モード” で使用する場合、必ずふたつひと組のペアで使用する必要があります。

ペアとしての指定はデバイス番号で行います。デバイス番号は外部端子（DN5～DN0）の設定で行いますが、DN5～DN1 までが同じ状態に設定された G9205A が ペアとして扱われます。

ペア指定された G9205A のうち、DN0=High と設定されたほうを偶数アドレス LSI と呼びます。

## 【サイクリック通信】

## 【サイクリック伝送】

## 【サイクリック】

Motionnet の通信プロトコルのひとつで、次のようなものです。

センター制御 LSI は、通信ラインに接続されている多数のローカル制御 LSI に対し、デバイス番号の小さい方から順に通信を開始し、最も大きいデバイス番号に達したら、再度 最も小さいデバイス番号へ戻って通信を続けます。この最も小さいデバイス番号から 最も大きなデバイス番号までの一連の通信を 1 サイクルとし、このサイクルを自動的に繰り返します。

Motionnet では「サイクリック通信」の表現を使用します。

## 【システム通信】

トランジェント通信のひとつです。

通信ラインをポーリングし、ローカル制御 LSI の接続個数、デバイス番号、種類および入出力ポートの割当て情報の収集を行う通信です。

## 【シリアル通信】

データを送受信するための伝送路を 1 本、または 2 本使用して、データを 1 ビットずつ連続的に送受信する通信方式です。

Motionnet は、1 組（注）の信号線でデータの送受信を行います。

注：シリアル通信では、1 本の信号線でも信号を送信できますが、Motionnet のようなケーブルを用いた長距離通信では 2 本の信号線をついストさせた 2 本 1 組のペアで信号送信を行っています。

## 【セレクトィング】

## 【ポーリング】

ポーリングとは、センター制御 LSI が特定のローカル制御 LSI に向けてフレームを送信する行為です。フレームにはアドレス番号が含まれており、これによりローカル制御 LSI を特定します。つまり、フレームに含まれるアドレス番号と同じ番号が設定されたローカル制御 LSI だけがフレームを受信できます。

フレームを受信できたローカル制御 LSI は、センター制御 LSI から選ばれたことになり、“セレクトィング”された状態となります。センター制御 LSI から選ばれたローカル制御 LSI は、センター制御 LSI に対してレスポンスデータを含むフレームを返送する許可を得たこととなります。

フレームに含まれるアドレス番号と同じ番号が設定されたローカル制御 LSI が存在しない場合、応答フレームが返ってこないのが、未接続であることがわかります。

## 【センター】

## 【センターLSI】

## 【センター制御 LSI】

## 【センターデバイス】

## 【センターチップ】

## 【センターIC】

G9001A を指します。

Motionnet の全ての通信制御を行う機能を持った LSI です。

Motionnet では主に「センター制御 LSI」の表現を使用します。

**【センターCPU】****【センター側 CPU】****【CPU】**

センター制御 LSI を制御する CPU です。

Motionnet で「CPU」と表現した場合はセンター側 CPU を指すことが多く、ローカル側 CPU との区別を明確にする場合は「センター側 CPU」の表現を使用します。

**【端末装置】****【ノード】**

センター制御 LSI からのフレームを受信し、処理や作業を行う、通信ラインの各所に接続される装置を指します。G9002A、G9205A、G9103C、G9004A、G9006 等のローカル制御 LSI で構成されます。

Motionnet では主に「端末装置」の表現を使用します。

**【通信周期】****【サイクリック通信周期】****【サイクル時間】****【サイクル周期】****【サイクリック伝送周期】**

Motionne はセンター制御 LSI が主導権を持った通信プロトコルを採用しています。センター制御 LSI は、通信ラインに接続されている多数のローカル制御 LSI に対し、自動的にポーリング/セレクトング処理を行い、全てのローカル制御 LSI との通信完了後、再度はじめのローカル制御 LSI から順番に同じような処理を行います。これを延々と繰り返し、常に（順番に）ローカル制御 LSI との通信が行われます。

つまり特定のローカル制御 LSI から見ると、一定の間隔で（ローカル制御 LSI の接続数で変わりますが）、センター制御 LSI から自分宛のフレームを受信しています。

この一定の間隔が通信周期です。

LAN などの通信プロトコルは、明確な主導権を持った機器が存在しないため、通信周期というものの自体がありません。

Motionnet では「通信周期」の表現を使用します。

**【通信プロトコル】**

通信を行うために決められた手順のことです。

1 本しかないケーブルに多数の機器が接続された状態で、正確なデータの送受信を保証しなければなりません。そのための仕組み（手順）をプロトコルといいます。通信する機器の全てが同じプロトコルを使用していなければ正常な通信は行えません。

例えば Motionnet は弊社独自のプロトコルで、これと LAN を接続するなどと言う行為は成立しません。

**【通信ライン】****【伝送ライン】****【信号ライン】****【ライン】**

Motionnet 通信が行われる信号線、もしくは通信ケーブルです。

Motionnet では「通信ライン」の表現を使用します。

**【データ制御 LSI】****【データデバイス】**

G9103C や G9004A のような、データ通信を受信できるローカル制御 LSI を指します。

Motionnet では「データ制御 LSI」の表現を使用します。

**【データ通信】**

トランジェント通信のひとつです。

“データ制御 LSI” との間のデータの読み書きに使用します。センター制御 LSI の送信 FIFO にデータを書込み、送信コマンドを発行すると、サイクリック通信の合間に割り込む形で送信されます。ローカル側から返信されたデータは、自動的にセンター制御 LSI の受信 FIFO に格納されます。

G9103C では、コマンド書き込み/レジスタ読み書き、また、G9004A では、接続されている LSI のデータの読み書き等で使用されています。

**【データ通信時間】**

データ通信での、センター制御 LSI とローカル制御 LSI との間で 1 組のデータ通信を送受信するのにかかる時間です。

**【デバイス】**

装置や機器をデバイスと表現する場合があります。また LSI 単体を指してデバイスと呼ぶこともあります。要するに特定の機能を持ったハードウェア製品を総称してデバイスと呼びます。

Motionnet の場合、G9000 シリーズを構成する個々の LSI をさす場合と、これらの LSI が搭載された基板やシステムをさす場合があります。

端末装置、ノード、と同義に扱う場合もあります。

**【デバイスアドレス】****【デバイス番号】****【LSI アドレス】**

通信ラインでの競合が発生しないよう、ローカル制御 LSI に割当てたユニークな番号です。

Motionnet では「デバイス番号」の表現を使用します。

**【デバイス情報】**

ローカル制御 LSI の接続状況、接続されているローカル制御 LSI の種類や I/O の設定状態、等の情報です。

CPU からセンター制御 LSI に設定します。

また、システム通信によって自動的に得ることもできます。

**【同報通信】**

トランジェント通信のひとつです。

センター制御 LSI が、複数のローカル制御 LSI に対して同時にフレームを送り付けるための通信です。

同報通信を受信可能なローカル制御 LSI (G9103C、G9006) だけが受信できます。

通信ライン上に点在する G9103C (モーター制御 LSI) の同時スタートなどで活用します。

**【トランジェント通信】****【トランジェント伝送】**

任意のタイミングで、指定したローカル制御 LSI に対して行う通信形態です。

常に継続されているサイクリック通信とは異なり、センター側 CPU からの指示が無い限り行われません (ブレーク通信だけは自動的に送信できます。)

Motionnet では「トランジェント通信」の表現を使用します。

**【バス】**

フィールドバスの説明文に出てくる“バス”とは、シリアル通信に用いられるケーブルや信号線を指します。

Motionnet での通信ラインと同義です。

CPU や、周辺回路などの話の中に出てくる“バス”は、データやアドレスなどの信号をひとまとめとした、いわゆるパラレル信号線を指します。

## 【パルストランス】

一般的なトランスと全く同じ構造です。高速信号用のトランスとして扱われます。  
直流信号は通過させないという性質を利用し、システム間の絶縁を行えます。

## 【半二重方式】

通信用ケーブルが1本しかないときに、通信方向が異なる次のような通信は同時に行えません。

- ・ センター制御 LSI から ローカル制御 LSI への通信
- ・ ローカル制御 LSI から センター制御 LSI への通信

この場合、交互に信号の流れる方向を切り替えて通信する必要があり、このような通信のやり方を半二重通信と呼びます。

2組の通信ケーブルを使用し、方向別にそれぞれ別ケーブルで通信する方式は全二重通信と呼びます。

Motionnet は半二重方式です。

## 【ビッグエンディアン】

一般的には、メモリアドレスの下位側から順番に 上位側バイトから並べるやり方です。

64 ビット CPU がメモリにアクセスする際に、“12345678h” というデータを書き込む場合、次のようになります。

アドレス	アドレス(下位 2bit)	11	10	01	00
0003h~0000h		78h	56h	34h	12h
0007h~0004h					
000Bh~0008h					
⋮					
⋮					

ルネサスエレクトロニクス株式会社が販売している旧日立 CPU(H8 系)は、こちらになります。

## 【フレーム】

## 【通信フレーム】

## 【パケット】

通信ライン上を流れる送信データの1単位です。

Motionnet では「フレーム」の表現を使用します。

## 【ポート】

ポート (port) を直訳すると「港」です。国内と国外の境界であり、ここから輸出や輸入が行われます。

CPU が持つポートも意味合いは同じです。CPU 内 (国内) と、CPU 周辺 LSI (国外) とでデータ (デジタル信号) のやりとりを行う部分です。

CPU は、外部端子として何本かのポートを持っているのが普通ですが、これでは足りない場合が多々あります。この場合、CPU 外部にポート専用の LSI を設けることがあります。

I/O 制御 LSI は CPU 外部に設けられる増設ポートで、なおかつ通信ライン経由で CPU から遠く離れた場所でポートとして働くことができます。

## 【ポーリング/セレクティング方式】

通信ケーブルに多数の機器が接続されているケースで、ケーブルの管理をひとつの機器のみで行う場合によく採用されます。

Motionnet では、ケーブルを管理する機器を“センター制御 LSI”、それ以外を“ローカル制御 LSI”と呼びます。

通信はセンター制御 LSI が、ローカル制御 LSI ひとつひとつに対して順番に問いかけ (ポーリング) を行い、これに回答する形でローカル制御 LSI がデータを送信します。

このようにローカル制御 LSI は、センター制御 LSI からの要求 (呼びかけ) があって初めてデータの送信が可能となります。

## 【メッセージ通信】

センター側 CPU とローカル側 CPU で、相互にデータの送受信が行える通信です。

CPU 間通信と同等の機能ですが、Motionnet においては データ通信によって行う CPU 間通信を「メッセージ通信」と表現します。

## 【無応答】

センター制御 LSI からのリクエストフレームに対する レスポンスフレームが無い状態を、無応答と呼びます。ローカル制御 LSI が無い、壊れた、等の場合に発生します。

## 【ラインドライバ】

通信ラインにシリアルデータを送出するための部品です。  
Motionnet では、RS485 製品を使用します。

## 【リクエストフレーム】

## 【下りフレーム】

## 【下りパケット】

“センター制御 LSI” から “ローカル制御 LSI” への要求を含むフレームです。

これには、送信先（ローカル制御 LSI）の「デバイス番号」や、送信先に渡すデータを含みます。

Motionnet では「リクエストフレーム」の表現を使用します。

## 【リトルエンディアン】

一般的には、メモリアドレスの下位側から順番に 下位側バイトから並べるやり方です。

64 ビット CPU がメモリにアクセスする際に、“12345678h” というデータを書き込む場合、次のようになります。

アドレス	11	10	01	00
0003h~0000h	12h	34h	56h	78h
0007h~0004h				
000Bh~0008h				
⋮				
⋮				

Windows などのパソコンで使用されること多いインテル系の CPU (x86 系) は、こちらになります。

## 【レスポンスフレーム】

## 【上りフレーム】

## 【上りパケット】

“ローカル制御 LSI” から “センター制御 LSI” への応答を含むフレームです。

これには、送信元（ローカル制御 LSI）の「デバイス番号」や、センター制御 LSI へ渡すデータを含みます。

Motionnet では「レスポンスフレーム」の表現を使用します。

## 【ローカル】

## 【ローカル LSI】

## 【ローカル制御 LSI】

## 【ローカルデバイス】

## 【ローカルチップ】

## 【ローカル IC】

センター制御 LSI からのフレームを受信し、処理や作業を行う LSI を指します。  
次のような製品から構成します。

G9002A	汎用入出力ポート制御 (4 ポート)	(I/O 制御 LSI)
G9103C	1 軸のモーター制御	(データ制御 LSI)
G9004A	CPU エミュレーション メッセージ通信	(データ制御 LSI)
G9205A	汎用入出力ポート制御 (2 ポート) センターレス通信	(I/O 制御 LSI)
G9006	機能拡張	(機能拡張 LSI)

Motionnet では「ローカル制御 LSI」の表現を使用します。

## 【ローカル CPU】

## 【ローカル側 CPU】

ローカル制御 LSI を制御する CPU です。

Motionnet では主に「ローカル側 CPU」の表現を使用します。

## 【割り込み】

ある状態になるとそれに応じた処理を行う、といった機能をソフトウェアで作成すると仮定しましょう。例えば I/O 制御 LSI の特定のポート入力に変化したら、特定の動作を行う等です。

CPU (ソフトウェア) は、その特定のポートを一定の間隔で監視し、状態が変化したときに必要なソフトウェアが動作するようにすれば良いのですが、監視すべき入力ポートがたくさん存在し、かつ監視以外の処理がたくさんあると、一定の間隔での処理が困難になります。このときの監視動作を別のデバイスに任せてしまい、必要なときだけそのデバイスから呼びかけてもらうという仕組みが割り込みです。

例えばセンター制御 LSI は、I/O 制御 LSI からの入力ポート変化を監視する機能があります。これを利用すれば、ポート変化があったときに「INT」という外部端子を通じて CPU に変化があったことを知らせてくれます。

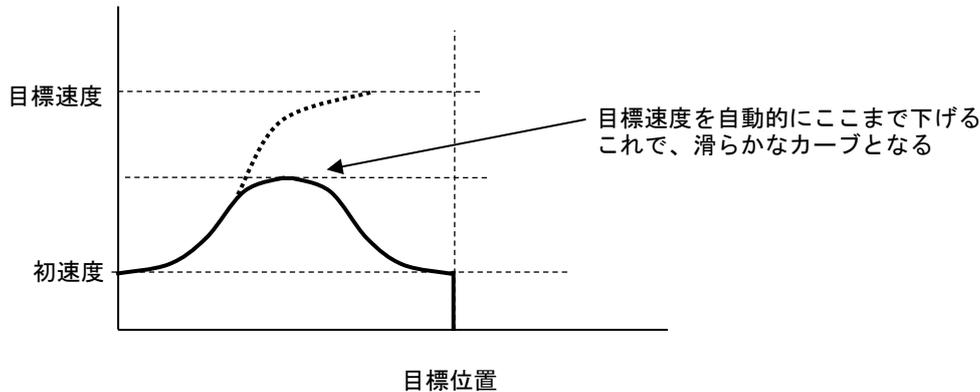
CPU 側は「INT」が変化した (有効になった) ときに、必要な動作を行うようにソフトウェアを構成すれば良いわけです。これで常に監視するという動作が無くなり、CPU の負荷が軽くなります。

## 4.2 G9103C に関連する用語

### 【FH 補正機能】

目標速度へ到達しないような位置決め動作を行うと三角駆動になる可能性があります。この三角駆動を防止するために FH 補正機能があります。

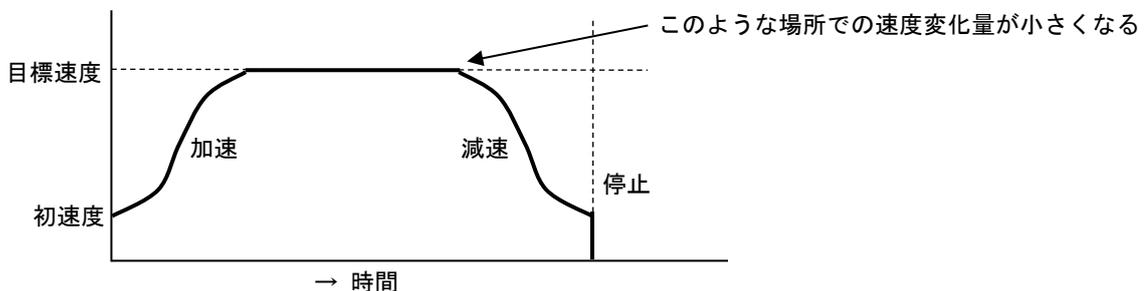
自動的に目標速度を下げることで滑らかな動作を実現します。



### 【S 字加減速】

加速や減速動作を、S 字を描くような滑らかな動作で行うことを S 字加減速と呼びます。

加減速が直線的だと、加速と定速の切り替えタイミングでの速度差が大きくなり、振動の発生や、モーターがついてこれない可能性があるため、このような制御があります。



### 【アイドルパルス】

加速動作を行うようにしてステップモーターをスタートさせるケースを考えます。初速が脱調ぎりぎりの速度の場合、正常にスタートしないで脱調する可能性があります。この原因は加速動作にあります。たとえ初速度をぎりぎりの値にしておいても、最初の動作パルスはほんの少し加速した結果である可能性があるためです。

このようなことがないように、加速開始前に脱調しない速度のパルスが確実に出力されるようにするのが「アイドルパルス」出力機能です。加速を始める前に、予備的に数パルス分の脱調しない速度で動作させられます。

### 【エンドリミット】

モーターの回転により、ある装置が直線状の動作を行うケースを考えてください。

モーター自体は無限に回転しても問題ありませんが、装置は有限の距離しか移動できません。つまり行き過ぎてはいけな場所までモーターが回転することは許されません。

これを防ぐため、直線移動をする装置の端にはセンサーがつけられ、そのセンサーが反応するとモーターの回転を停止させるなどの動作を行うシステムが一般的です。

このような、「装置の移動はここで終わりです」という目印がエンドリミット (EL) と呼ばれます。

### 【オーバーライド】

速度や目標位置は、LSI 内部の特定のレジスタへ値を設定することで行います。

オーバーライドとは、モーター動作中に、これらのレジスタの値を上書きして書き換えてしまうことです。上書きすることで、即座にその設定値が動作に反映されます。

### 【減速停止】

モーター制御 LSI が停止すると判断したとき、安全に停止できる速度まで減速してから停止します。

即停止と違って、停止するまでの時間がかかります。このため、停止すると判断した位置を通り過ぎる可能性があります。

### 【原点】

モーターで装置を移動させるときには、「どこから」「どのくらい」移動するのかが情報が必要です。「どのくらい」は移動量として設定しますが、「どこから」は現在の自分にいる位置になります。連続して動作する場合は、現在位置からの動作を行うことは、相対位置の管理となり 問題はないはずですが、最初の位置は誰かが教えてやらなければなりません。

そのため、制御の基点となる位置を示すためのセンサーがもうけられています。これが原点となり、動作の（最初の）基点として扱われます。

### 【サーボモーター】

サーボモーターというモーターがあるわけではありません。

自分が使用するモーターに対して、任意の位置決めができるようなシステムを組み込んだ状態になって、始めてサーボモーターになります。

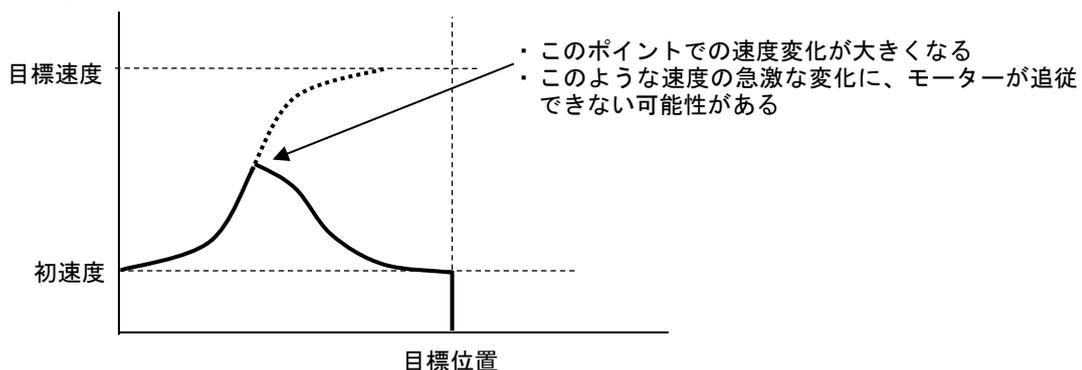
この業界では、位置決めすることを「サーボをかける」と呼ぶこともあるようです。

したがって、高価なモーターでなくとも、位置決めシステムさえ付け加えればサーボシステムです。

### 【三角駆動】

位置決め動作を行うときにも、加減速動作を伴った制御が可能です。

このとき、移動量が小さく、かつ目標速度が高い場合が問題となります。モーターをスタートさせると、指定された加速度で加速して行くのですが、目標速度に到達する前に目標位置に着いてしまうと困ったこととなります。通常は目標位置で停止する前に減速し、初速度まで戻りますが、この減速開始が下図のように加速中に開始されるケースがあります。



このような頂点のとがった速度パターンが、三角駆動と呼ばれます。

### 【指令パルス】

#### 【パルス列】

日本では一般的に、モーターを回転させる場合はパルスを使用します。

「特定の角度だけ回したい」とか、「モーターで制御されるワークを目標の位置まで移動させたい」などの制御は、与えるパルス数で管理しています。

このような連続したパルスをパルス列、もしくは指令パルスと呼びます。

**【ステッピングモーター】**

パルス信号を与えることで、決められた角度だけ回転するモーターです。パルスモーターとも呼ばれます。

回転角度と回転速度は、与えるパルスにより簡単に制御できるというメリットがある反面、高速回転が困難であったり、大きなトルクが出せなかったりという欠点もあります。

**【スローダウン】**

エンドリミットにより装置が停止するケースを考えてみてください。

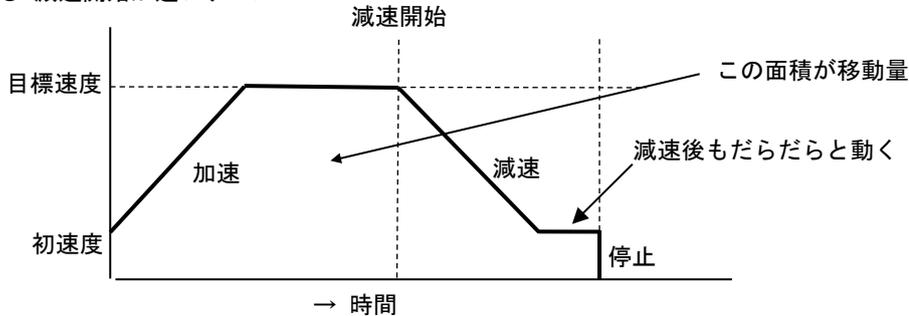
それ以上移動できないからといって、いきなりモーターが停止するとどうなるでしょうか。大きな装置であれば停止時の反動で壊れるかもしれません。壊れては困るので、通常はエンドリミットの手前にスローダウン開始ポイントと呼ばれるセンサーがもうけられます。これが反応したときに、装置が停止しても問題ない速度まで減速を行うようにするのが一般的です。

このセンサーからの信号をスローダウン（SD）信号と呼びます。

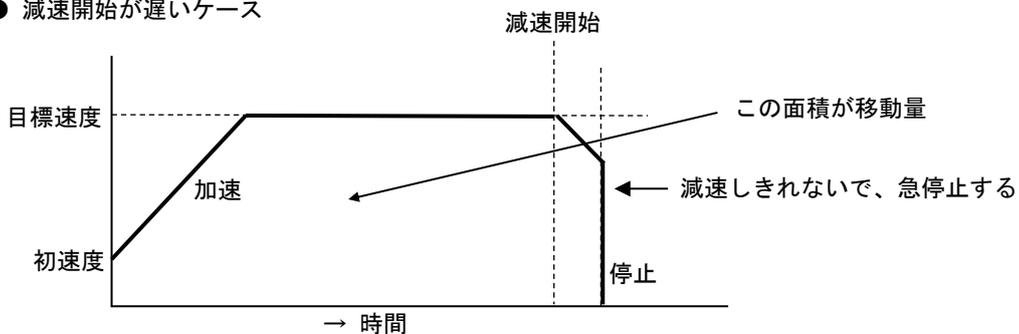
## 【スローダウンポイント】

加減速を伴う位置決め動作では、停止する前に減速を行うのが普通です。この減速により初速度までスピードが落ちるのですが、最後にこの初速度で動作してはいけなし、初速度になる前に停止してもいけません。つまり適切な場所で減速が開始されなければなりません。

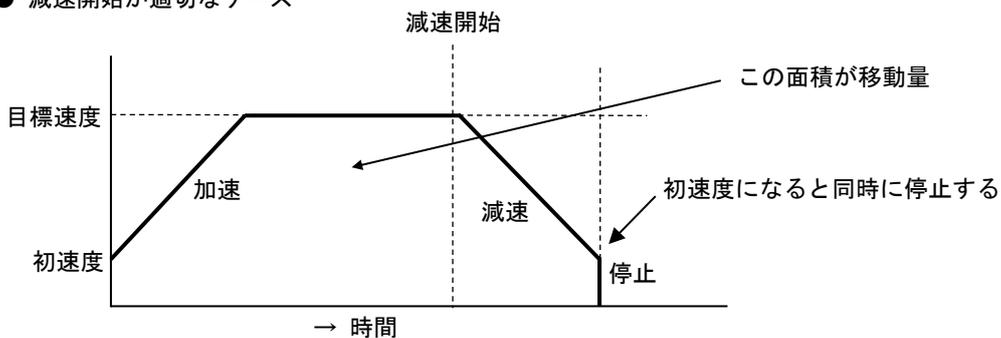
## ● 減速開始が早いケース



## ● 減速開始が遅いケース



## ● 減速開始が適切なケース



このような適切な減速開始位置は計算で簡単に求められますが、LSI 内部で自動的に設定できる PCL 製品もあります。そして、この減速開始位置をスローダウンポイントと呼びます。

## 【即停止】

モーター制御 LSI が停止すると判断したとき、現在出力中のパルスが完了した後で停止します。  
モーターを制御するためのパルスは特定の周期をもっています。この周期にかかる時間を待ってから停止することになります。

## 【ソフトリミット】

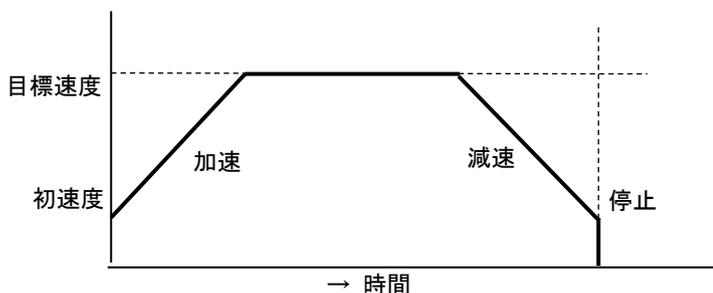
モーターの回転により、ある装置が直線状の動作を行うケースを考えてください。  
モーター自体は無限に回転しても問題ありませんが、装置は有限の距離しか移動できません。つまり行き過ぎてはいけな場所までモーターが回転することは許されません。  
これを防ぐため、直線移動をする装置の端にはセンサーがつけられ、そのセンサーが反応するとモーターの回転を停止させるなどの動作を行うシステムが一般的です。これはハードウェア的な制限（リミット）で、通常エンドリミット（EL）と呼ばれます。  
これに対し、移動量を数値で管理し、ある数値以上になるとモーターを停止させるようなソフトウェア的構成にすると、センサーを省略できます。このようなソフトウェア処理による制限をソフトリミットと呼びます。

## 【脱調】

ステッピングモーターは、パルスを与えるとそのパルス数だけ回転する礼儀正しいモーターです。しかし、与えるパルスの周波数が高いと、モーターの回転が追いつかなくなってきます。追いつかなくなると、当然与えたパルス数の回転は行えません。これが脱調と呼ばれる現象です。

## 【直線加減速】

ステッピングモーターをいきなり高速で回転させると、ステッピングモーター自身はその動作について行けなくなり、思ったような動作を行えない場合があります（脱調など）。  
このためステッピングモーターに対し、はじめは初速度などと呼ばれる、比較的ゆっくりとした回転数で動作させ、徐々に目的の高速へ加速させるようにします。このときの加速動作を直線的に行うのが直線加速動作です（加速度／減速度は一定の値です）。  
逆に高速から一気に停止しようとする時、ステッピングモーター自身の慣性により停止指示後もしばらく回転してしまう可能性があります。  
このときも、高速状態から徐々に減速し、元の初速度までスピードダウンしてから停止させます。



## 【定速動作】

モーターを一定の速度で回転させるような動作を指します。

## 【通倍機能】

パルサーなどの周期性のある信号を、数倍の周期に上げることを指します。  
2 通倍ならもとの周期の 2 倍に、4 通倍なら 4 倍です。

## 【非常停止】

モーター制御 LSI が停止すると判断したとき、パルス出力の完了を待たずに停止します。  
人命にかかわるようなケースで、モーター動作を急停止させる場合などで利用されます。  
パルス周期の完了を待たないで停止するので、そのパルスでモーターが回ったか否かの判断がつかなくなり、モーターの位置管理に支障をきたします。

**【分周機能】**

パルサーなどの周期性のある信号を、数分の1の周期に下げることが指します。  
2分周ならもとの周期の1/2に、4分周なら1/4です。

**【励磁シーケンス】**

弊社のPM型ステッピングモーターは、円筒形の永久磁石で作られたロータと、その周りを囲むように配置された2つのコイルから成ります（実際にはヨークなるものもありますが）。

この2つのコイルに対して電流を流します。このときの流れる電流の方向を切り替えることにより、発生する磁界が変化します。この変化に追従しながらロータが回転します。

ただし、電流の向きを適切に制御しなければ（順序よく磁界が変化しなければ）ロータは回転しません。この順序は一般的に決まっており、それを励磁シーケンスなどと呼びます。

## 索引

<b>B</b>			
BMD	6		
bps	54		
<b>C</b>			
CPU	57		
CPU エミュレーション LSI	8, 54		
CPU エミュレーションモード	8		
CPU 間通信	10, 54		
CPU 周辺 LSI	54		
CRC コード	54		
<b>F</b>			
FH 補正機能	62		
FIFO	26, 54		
<b>G</b>			
G9000 シリーズ	4		
G9001A	4		
G9002A	4		
G9004A	8		
G9006	9		
G9103C	7		
G9205A	5		
G9H50A	11		
<b>H</b>			
HUB 機能 LSI	11		
<b>I</b>			
I/O 制御 LSI	4, 54		
I/O チップ	54		
I/O 通信	14, 54		
I/O 通信エラーフラグエリア	48		
I/O デバイス	54		
<b>L</b>			
LSI アドレス	58		
LWRQ	51		
<b>M</b>			
MCRY	50		
MERR	50		
Motionnet	55		
		MRER	50
		MSEL	50
		<b>P</b>	
		PCL 制御 LSI	55
		PMD	27, 45
		<b>R</b>	
		RS485	20
		RS485 規格	55
		<b>T</b>	
		TOUT	50
		<b>W</b>	
		WRQ	51
		<b>あ</b>	
		アイドルパルス	62
		アドレス	55
		<b>え</b>	
		S 字加減速	62
		エラー応答	45, 46, 55
		エンディアン	55
		エンドリミット	62
		<b>お</b>	
		オーバーライド	63
		<b>か</b>	
		仮想ローカル制御 LSI	10
		活線挿抜	55
		<b>き</b>	
		奇数アドレス LSI	5, 55
		機能拡張 LSI	9
		<b>く</b>	
		偶数アドレス LSI	5, 56
		下りパケット	60
		下りフレーム	60

**け**

減速停止 .....	63
原点 .....	63

**さ**

サーボモーター .....	63
サイクリック .....	56
サイクリックコード .....	54
サイクリック通信 .....	15, 56
サイクリック通信周期 .....	57
サイクリック伝送 .....	56
サイクリック伝送周期 .....	57
サイクル時間 .....	57
サイクル周期 .....	57
三角駆動 .....	63

**し**

システム通信 .....	14, 56
終端抵抗 .....	21
受信用 FIFO .....	26
状態モニター .....	10
シリアル通信 .....	56
指令パルス .....	63
信号ライン .....	57

**す**

ステータス .....	18, 29
ステップモーター .....	64
スローダウン .....	64
スローダウンポイント .....	65

**せ**

セレクティング .....	56
センター .....	56
センターCPU .....	57
センターIC .....	56
センターLSI .....	56
センター側 CPU .....	57
センター制御 LSI .....	4, 56
センター制御ノーマルモード .....	5
センター制御ポート間モード .....	5
センターチップ .....	56
センターデバイス .....	56
センターレスポート間モード .....	5
全二重通信 .....	59

**そ**

送信用 FIFO .....	26
即停止 .....	66
ソフトリミット .....	66

**た**

脱調 .....	66
端末装置 .....	57

**ち**

直線加減速 .....	66
-------------	----

**つ**

通信周期 .....	18, 57
通信速度 .....	54
通信待機設定 .....	40
通信フレーム .....	59
通信プロトコル .....	57
通信ライン .....	57
通信レート .....	54

**て**

定速動作 .....	66
通倍機能 .....	66
データ制御 LSI .....	57
データ通信 .....	14, 58
データ通信時間 .....	58
データデバイス .....	57
デバイス .....	58
デバイスアドレス .....	58
デバイス情報 .....	58
デバイス情報エリア .....	30
デバイス番号 .....	19, 58
転送速度 .....	54
伝送速度 .....	54
伝送ライン .....	57
転送レート .....	54
伝送レート .....	54

**と**

同報通信 .....	14, 58
トランジェント通信 .....	16, 58
トランジェント伝送 .....	58
トランス .....	20

**に**

入力変化割り込み設定エリア .....	31
入力変化割り込みフラグエリア .....	32

**の**

ノード .....	57
上りパケット .....	60
上りフレーム .....	60

**は**

パケット	59
バス	58
パルストランス	20, 59
パルス列	63
反射	21
半二重方式	59

**ひ**

非常停止	66
ビッグエンディアン	59

**ふ**

ブレイク通信	14
ブレイク入力	14
フレーム	13, 59
プロトコル	57
分周機能	67

**へ**

ペア	5
----	---

**ほ**

ポート	59
ポートデータエリア	27
ポーリング	56
ポーリング/セレクトイング方式	59
ポーリング単位	13, 14

**ま**

マルチドロップ形式	12
-----------	----

**む**

無応答	46, 48, 60
-----	------------

**め**

メッセージ通信	60
メッセージ通信モード	8

**も**

モーションネット	55
モーター制御 LSI	7, 55
モニター動作モード	9

**ら**

ライン	57
ラインドライバ	60

**り**

リクエストフレーム	60
リトルエンディアン	60

**れ**

励磁シーケンス	67
レート	54
レスポンスフレーム	60
レベルシフタ	53, 54, 62

**ろ**

ローカル	61
ローカル CPU	61
ローカル IC	61
ローカル LSI	61
ローカル制御 LSI	61
ローカルチップ	61
ローカルデバイス	61
ローカル動作モード	9

**わ**

割り込み	61
------	----

弊社は、弊社ソフトウェアについて著作権を含む一切の知的所有権を保持します。弊社は、弊社ソフトウェアに関するいかなる権利もお客様に譲渡しません。お客様は、弊社の製品を使用する目的でのみ、現状有姿の弊社ソフトウェアを使用することができます。弊社は、弊社ソフトウェアの完全性、正確性、適用性、有用性、第三者知財の非侵害性を含め、明示たと黙示たとを問わず何らの保証をいたしません。また、弊社ソフトウェアを使用したことで生じる損害（収入または利益の逸失を含む）について、一切の責任を負いません。お客様が、購入国以外で弊社ソフトウェアを使用する場合は、購入国と使用国の輸出管理法や規制を遵守する必要があります。

## 承認履歴

版数	日付	内容
初版	2004年4月1日	DZ10001-0 新規作成。
第2版	2009年4月8日	DZ10001-0/1 改版。
第3版	2023年12月13日	TA600170-JP0/0 DZ10001-0/1の全面的な見直し。 ・内容を最新のG9000シリーズの内容と差し替え。

**NPM** 顧客「満足」から「感動」へ。  
日本パルスモーター株式会社

[www.pulsemotor.com](http://www.pulsemotor.com)

お問い合わせ

[www.pulsemotor.com/support](http://www.pulsemotor.com/support)

東京 電話 03(3813)8841 FAX 03(3813)8550  
大阪 電話 06(6576)8330 FAX 06(6576)8335  
お電話受付時間 平日 9:00～17:00