

G9000シリーズ導入にあたって

Motionnet にできること

NPM 日本パルスモーター株式会社

<2009年4月8日>

はじめに

G9000シリーズは高速シリアル通信を実現するためのASIC製品群です。

Motionnet（モーションネット）と呼ばれる弊社独自のプロトコルを採用し、RS485をベースとしたフィールドバスの構築が容易に可能となります。

この技術の目指すところ、果たすべき役割は次のような用途・市場となることでしょう。

- モータなどの軸制御における省配線

軸制御は弊社の最も得意とする分野です。PCLシリーズに代表される軸制御デバイスで培われた技術を省配線で利用できます。

- CPUの周辺デバイスにおける省配線

CPUの制御信号（動作）を擬似的に生成するデバイスがあります。

このデバイスの使用で、遠方にある周辺デバイスを省配線でコントロールできます。

- CPU同士のデータのやりとりを可能とする省配線

別の場所にあるCPUの持っているデータが欲しい。そんな処理も実現できます。

- 一般的なフィールドバスとして

もちろん、一般的なI/O制御も可能です。

Motionnetというプロトコルは、モータ動作の制御、そしてCPU動作の再現など、フィールドバス上に点在する各種機能の“動き”に注目したものとなっています。ここから“Motion”という単語を冠とした名前が付けられました。

この機会にぜひ一度、Motionnetをお試しいただくことを、お勧めいたします。

Motionnet は、日本パルスモーター株式会社の登録商標です。

目次

基本編.....	1
1. G9000シリーズには、どのような製品があり、その機能は？.....	2
1-1. G9001A : センターデバイス.....	2
1-2. G9002 : I/Oデバイス.....	2
1-3. G9003 : PCLデバイス.....	2
1-4. G9004A : CPUエミュレーションデバイス.....	3
2. どのように構成すればよいのか？.....	4
3. 接続できるローカルデバイスはいくつまでか？.....	4
4. 通信はどのように行われるのか？.....	5
4-1. サイクリック通信.....	5
4-2. データ通信.....	6
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？.....	7
6. ステータスとは何か？.....	8
7. 通信周期はどれくらい速いか？.....	8
8. 通信できる距離は？.....	9
9. デバイスアドレスは連続しなくてもよいか？.....	9
10. RS485とは何か？.....	10
11. パルストランス（トランス）が必要な理由は？.....	10
12. 終端抵抗の役割は？.....	11
13. センターデバイスを制御できるCPUの種類は？.....	12
14. センターデバイスにシリアル入力が2本ある（SIA、SIB）理由は？.....	13
操作編.....	14
1. Motionnetはどれだけ簡単に使えるか？.....	15
2. データ通信のやり方は？.....	16
3. 送信用FIFOへのデータの書き込み方法は？.....	17
4. 受信用FIFOからのデータの読み出し方法は？.....	17
5. G9002（I/Oデバイス）の入力ポートのデータ取得方法は？.....	18
6. G9002（I/Oデバイス）の出力ポートへのデータ出力方法は？.....	19
7. G9003、G9004Aのステータス情報はどのように参照するか？.....	20
8. 通信回線に接続されているデバイス種類の確認方法.....	21
9. I/Oデバイスの入力ポートの変化を割り込み機能で認識するには？.....	21
9-1. 入力値変化を監視するデバイスの指定.....	22
9-2. 入力値変化のあったデバイスの特定.....	22
10. G9003、G9004Aのステータス状態の変化を認識するには？.....	24
10-1. ステータス変化を監視するデバイスの指定.....	24
10-2. ポート0の監視上の注意（変化割り込みの使用時）.....	25
11. G9003（PCLデバイス）を動作させるには？.....	28

12. G9004A (CPUエミュレーションデバイス) とのデータのやりとりの方法は？	29
12-1. G9004Aのステータスに関する注意	29
12-2. メッセージ通信モード：1 (センター側CPUからG9004A側CPUへデータを渡す)	29
12-3. メッセージ通信モード：2 (センター側CPUからG9004A側CPUへデータを渡し、返信データもらう)	30
12-4. メッセージ通信モード：3 (G9004A側CPUからセンター側CPUへデータを渡す)	32
12-5. CPUエミュレーションモード時のコマンドについて	33
12-6. CPUエミュレーションモード：1 (連続データの書き込み)	34
12-7. CPUエミュレーションモード：2 (連続データの読み出し)	35
12-8. CPUエミュレーションモード：3 (組み合わせ処理)	36
12-9. CPUエミュレーションモード：通信待機設定とは？	37
12-10. CPUエミュレーションモード：リードコマンド使用時にG9004Aから返信されるデータ形式	38
12-11. CPUエミュレーションモード：リードコマンドでの返送データが128ワードを越える場合	38
13. I/Oデバイスに対してデータ通信を行うとどうなるか？	39
14. I/Oデバイスとセンターデバイスで、異なるPMD設定をするとどうなるか？	39
15. PCLデバイスが、受信FIFO容量以上のデータを送信するとどうなるか？	40
16. G9004Aで、受信FIFOのデータをリードしないっているとどうなるか？	40
17. 通信が不可能な状態になると各デバイスはどうなるか？	41
18. ローカルデバイスが機能を停止したことを認識するには？	42
19. 稼働中に新たなデバイスを参入させるには？	43
20. 通信状態を外部から監視するには？	44
21. ウェイト制御用端子の無いCPUを使用するには？	45
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	46
23. 40MHzクロック使用時の注意点は？	46
24. 関連部品の選定基準は？	47
24-1. ケーブルの選定	47
24-2. RS485デバイスの選定	47
24-3. パルストランスの選定	47
24-4. 終端抵抗の抵抗値の決定	47
25. レベルシフタが必要なケースとは？	48

索引	50
----------	----

基本編

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、2009年3月現在で、次のような4つのシリーズ製品があります。

型名	製品名	分類
G9001A	センターデバイス	センターデバイス（マスターデバイス）
G9002	I/Oデバイス	ローカルデバイス（スレーブデバイス）
G9003	PCLデバイス	ローカルデバイス（スレーブデバイス）
G9004A	CPUエミュレーションデバイス	ローカルデバイス（スレーブデバイス）

1-1. G9001A : センターデバイス

シリアル通信に関するすべての制御を行います。Motionnetの構築には一つのセンターデバイスが必要です（複数個は不可）。

このセンターデバイスが、Motionnetに接続されているすべてのデバイス（G9002、3、4などで、ローカルデバイスとも呼ばれます）に対して通信制御を行います。

通信速度は最大で20Mbps（10M、5M、2.5Mの選択が可能）と高速です。

CPUに接続して使用します。

1-2. G9002 : I/Oデバイス

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ビット単位で一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）。

G9002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
L	L	L	出力	出力	出力	出力
L	L	H	入力	出力	出力	出力
L	H	L	入力	入力	出力	出力
L	H	H	入力	入力	入力	出力
H	L	L	入力	入力	入力	入力

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（H出力 or L出力）の設定が外部端子により行えます。

1-3. G9003 : PCLデバイス

1軸分のモーションコントロールが行えます。

弊社デバイス製品である、PCL6045B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。

1-4. G9004A : CPUエミュレーションデバイス

G9004Aは以下のような2つの機能を持ちます（どちらかの機能を選択して使用します）。

① メッセージ通信モード

G9004AにCPUを接続することで、センターデバイスを制御するCPUとの間で最大127ワード（254バイト）のデータ通信が行えます。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となることでしょう。



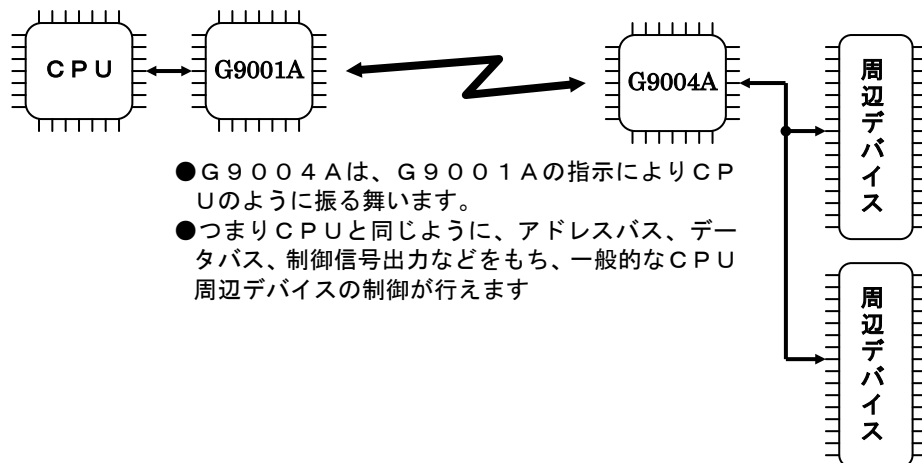
- センターデバイスを制御するCPUから、G9004Aを制御するCPUへ最大で127ワードのデータの転送が可能です。
- 逆にG9004Aを制御するCPUから、センターデバイスを制御するCPUへは、最大で128ワードのデータ転送が可能です。ただし、転送タイミングはセンターデバイス側のCPU動作に依存します

② CPUエミュレーションモード

G9004Aに一般的なCPU周辺デバイスを接続することで、センターデバイスを制御するCPUが、これらの周辺デバイスを遠隔操作できます。

例えば、G9004A一つに対して、弊社製ASICであるPCL6045Bを2個まで接続できます（G9004Aの持つ周辺デバイスへのアドレスバスは6本です。PCL6045Bは5本のアドレスバスで制御します）。

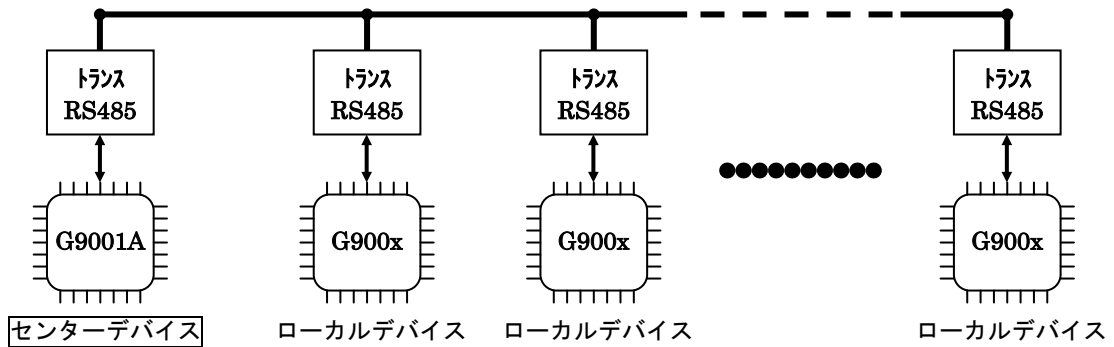
- G9001Aを制御するCPUは、シリアル通信を介して、G9004Aに接続されている周辺デバイスを制御することができます。これにより、周辺デバイスへのデータライトや、リードを行います。



- G9004Aは、G9001Aの指示によりCPUのように振る舞います。
- つまりCPUと同じように、アドレスバス、データバス、制御信号出力などをもち、一般的なCPU周辺デバイスの制御が行えます

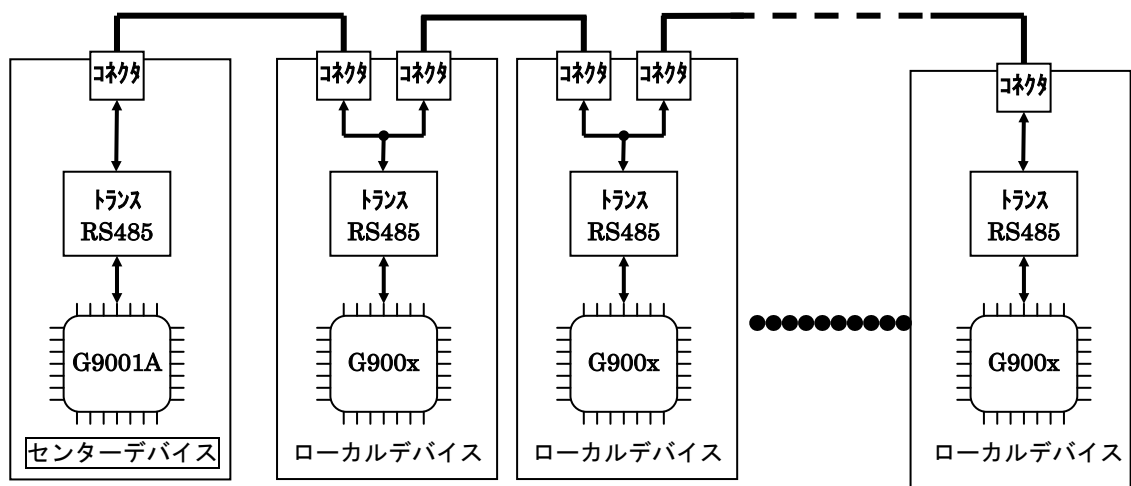
2. どのように構成すればよいのか？

Motionnetは、一つのセンターデバイスに接続されるシリアルバスに、以下のような構成のマルチドロップ形式で、複数のローカルデバイスを接続できます。



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ます。

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”形式が一般的でしょう。



3. 接続できるローカルデバイスはいくつまでか？

一つのセンターデバイスが制御できるローカルデバイスの数は最大で64個です。

接続されるローカルデバイスは、他と重複しないようなデバイスアドレスを設定しなければなりません（各デバイスの外部端子で決定します）。他と重複している場合は正常な通信は行えません。

センターデバイスはアドレスを持っていません。

4. 通信はどのように行われるのか？

通信はセンターデバイスが主導権を持って行われ、サイクリック通信と、データ通信の二つに分けられます（システム通信というものもありますが、ここでは割愛します）。

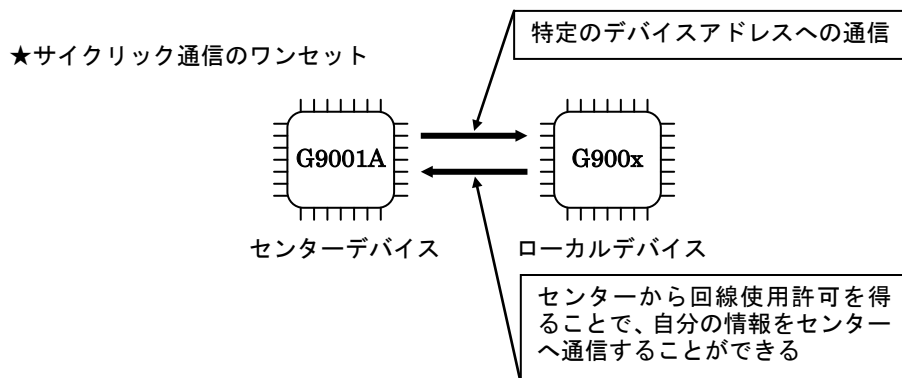
4-1. サイクリック通信

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に似ています。

通信回線はいかなる状況においても、その制御権はセンターデバイスが持っています。これは回線上でのデータ衝突を避けるため、各ローカルデバイスはセンターデバイスの許可がなければデータの送出ができません。

各ローカルデバイスと効率よく通信を行うために、センターデバイスは、接続されているローカルデバイスのうち、一番若いデバイスアドレスを持つものから順番に通信を行います。

まずセンターデバイスから、最初のローカルデバイスへ通信を行います。この通信を受け取るデバイスは通信回線上に一つしか存在しません（デバイスアドレス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デバイスが、次に通信回線を使用する権利を得ることとなり、ローカルデバイスからセンターデバイス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。



このような単純な応答を、デバイス番号順にすべてのローカルデバイスに対して行います。

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デバイス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デバイス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。

4-2. データ通信

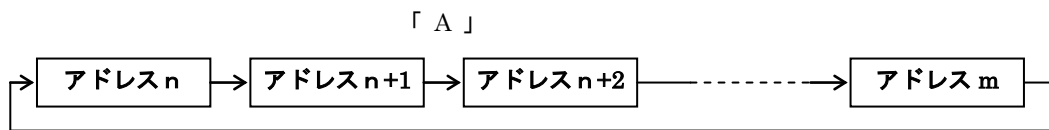
サイクリック通信でやりとりされるデータは最大で4バイトまでです（理由は後述します）。

ここでは、PCLデバイスに対する動作指示や、CPUエミュレーションデバイスの制御にはデータ量が少なすぎます。またサイクリック通信の場合は全てのデバイスとのやりとりが一巡しなければ自分の番が来ないなどの時間的制約がつきまといまいます。

これを解決するのがデータ通信です。

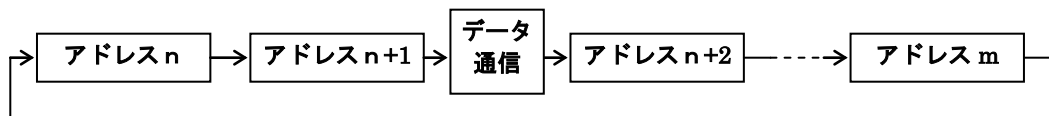
データ通信は、任意のタイミング（CPUからの通信指示のタイミング）で、サイクリック通信に割り込む形で行われます。

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。



ここで“アドレス n”とは、デバイスアドレス = n のローカルデバイスを表しています。図中の矢印は、サイクリック通信が行われる順番を表します。

ここで「A」のタイミングでデータ通信コマンドが発行されると、以下のようになります。



サイクリック通信の1周期に要する時間が狂いますが、目的のデバイスやりとりは迅速に行えます。

任意のタイミングで、と説明しましたが、若干の制約があります。

データ通信を連続して行いたい場合や、通信エラーによりデータ通信をリトライする場合などは、必ず1回のサイクリック通信を挟んで行います。



データ通信は連続しない

これでサイクリック通信が、データ通信により長時間停止させられることを防いでいます。

データ通信は、最大で128ワード（256バイト）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、特定のローカルデバイスに対してデータが送られた後、そのローカルデバイスからの応答通信がセンターデバイスに対して送られます。この応答通信に最大で256バイト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デバイスはデータ通信に応答できません（必要ないためです）。

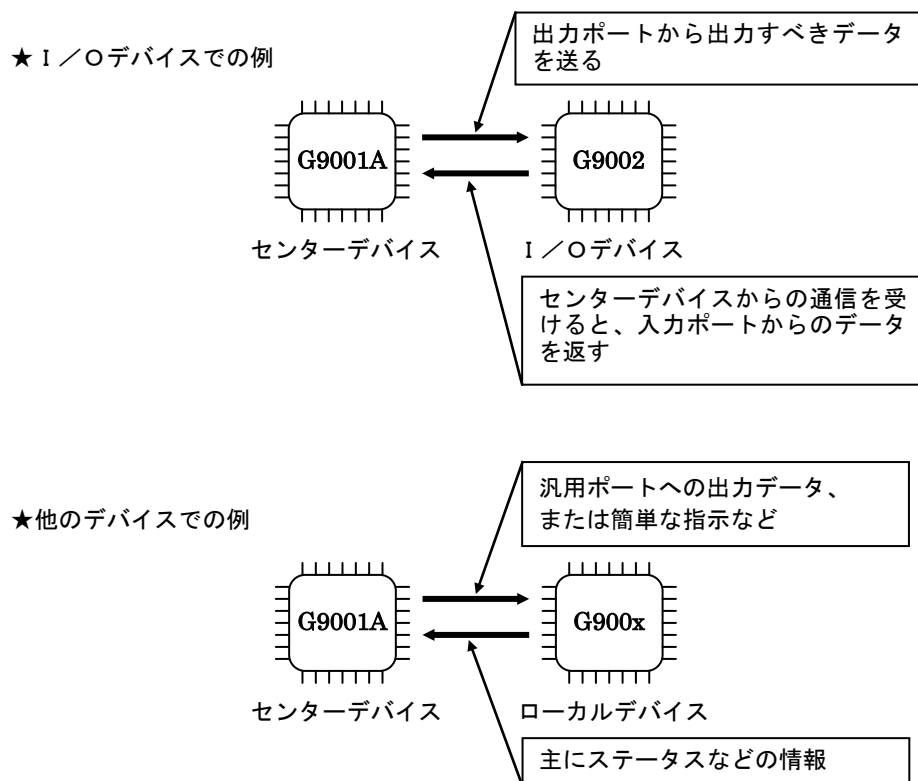
5. サイクリック通信で送られるのはどのようなデータか？

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4バイトです。このサイズ内で通信されるデータは、ローカルデバイスごとに内容が異なります。

デバイス名称	センターからローカルへの通信内容	ローカルからセンターへの返信内容
G9002	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
G9003	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、PCLデバイスのステータス情報
G9004A	動作指示など	CPUエミュレーションデバイスのステータス情報



サイクリック通信は、I/Oデバイス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）のために設計されています。よってデータサイズの上限は、I/Oデバイスのポート数である4バイトとなっています。

6. ステータスとは何か？

G9003 (PCLデバイス) や、G9004A (CPUエミュレーションデバイス) がどのような状態になっているかを監視するために、各デバイスにはステータスという情報があります (G9002: I/Oデバイスには、ステータスはありません)。

例えば、PCLデバイスの動作停止や、CPUエミュレーションデバイス側からのデータ転送要求などを知るために使用できます。

ステータス情報はサイクリック通信で各デバイスからG9001A (センターデバイス) へ送られ、センターデバイスの内部RAMへ格納されます。

CPUからこの情報を参照することで、各デバイスの状態の把握が可能です。

ちなみにG9001A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、センターデバイスへ送られて、センターデバイス側の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、“操作編12-1. G9004Aのステータスに関する注意” の項目をご参照下さい)。

7. 通信周期はどれくらい速いか？

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時)。

ローカルデバイス数	通信周期	備 考
8 個	0.12ms	全て I/O デバイスの場合、256 点の I/O が利用できます
16 個	0.24ms	全て I/O デバイスの場合、512 点の I/O が利用できます
32 個	0.49ms	全て I/O デバイスの場合、1024 点の I/O が利用できます
64 個	0.97ms	全て I/O デバイスの場合、2048 点の I/O が利用できます

ローカルデバイス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、ユーザズマニュアルをご参照下さい (計算式が載っております)。

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、実験時の環境を列挙します。

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。

伝送レート	条 件					結 果 最大長
	ローカルデバイス数	ケーブル	終端抵抗	パルストランス	I/Fチップ	
20Mbps	32	CAT5	100Ω	1000μH	RS485	100m
20Mbps	64	CAT5	100Ω	1000μH	RS485	50m
10Mbps	64	CAT6	100Ω	1000μH	RS485	100m

注：上記の内、最大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出ない場合もありますので、ご注意下さい。

G9001A（センターデバイス）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。

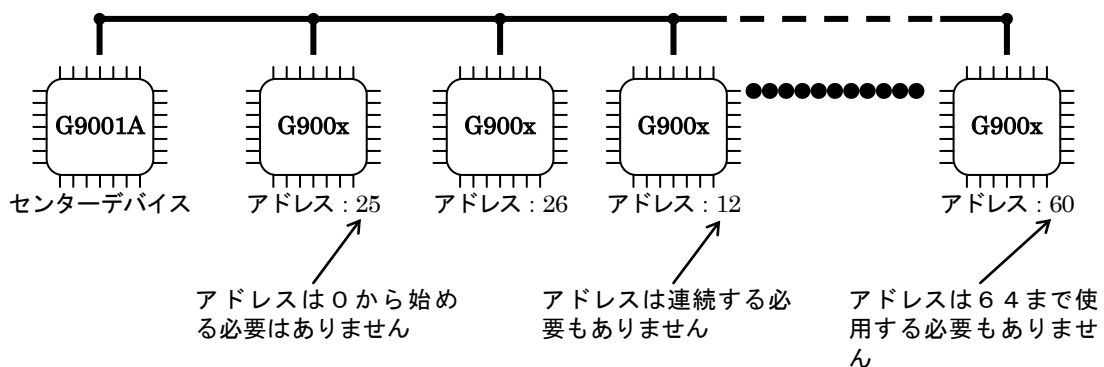
9. デバイスアドレスは連続しなくてもよいか？

デバイスアドレスは、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アドレスが歯抜けになってもかまいません。

通常、センターデバイスは通信ラインに接続されているローカルデバイスのアドレスを把握しており、接続されているアドレスにのみ通信を行います。

このアドレスの把握は、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）。



10. RS485とは何か？

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規格などとも呼ばれます。

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。その理由は、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チップ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チップとは、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、シリアル回線への信号送出機能と、シリアル回線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、回線上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。

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付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する通信システムの構築を推奨します。

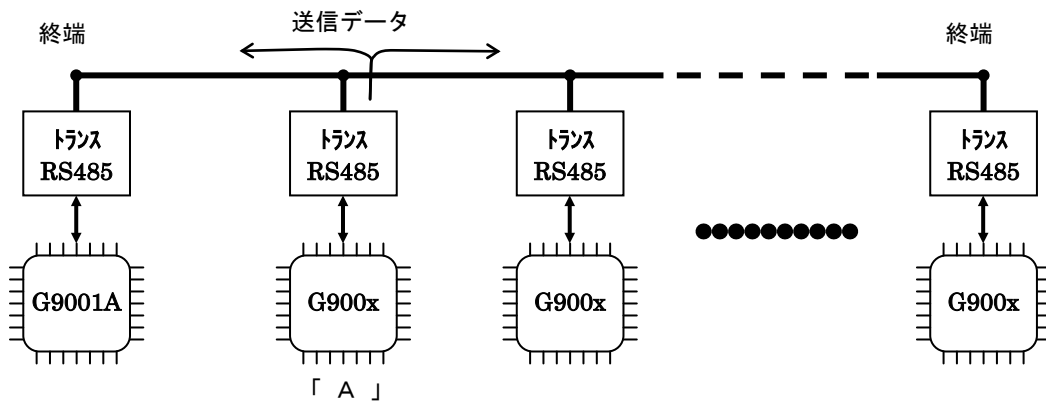
RS485デバイスのみで、パルストランスなしの通信はもちろん可能です。しかしこの場合、送信側と受信側でGNDを共通にする必要があります。GNDレベルを共通にしなければ、RS485デバイス間でやりとりされる信号の電圧レベルが、RS485デバイスの許容範囲を超えて破壊される可能性があります。GNDを共通にするためには、通信回線と平行してGND線を1本追加する必要があります（通信回線に3本必要となります）。

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。



通信回線はパルストランスにより絶縁されます。
よって上図の2つのシステムは電気的に分離されます。

12. 終端抵抗の役割は？

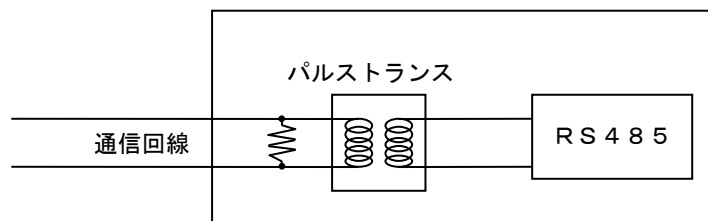


例えば上図の「A」からシリアルデータが送信されたと仮定します。

この信号は通信回線の両側に向かって伝わり、それぞれの通信回線の端（終端）まで到達します。このデータは、回線の端で跳ね返り、通信回線を逆送します。これは反射と呼ばれる現象で、通信障害の原因の一つです。

終端抵抗は、この反射の発生を抑止するために使用されます。通常は以下のように、通信回線の末端側（両方の終端）に100Ω前後の抵抗を設置します。通信回線の途中には必要ありません。

下図は、パルストランスから見て、通信回線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デバイス側）に終端抵抗を設置しても同様な効果が得られるようです。



13. センターデバイスを制御できるCPUの種類は？

G9000シリーズでCPUを必要とするのは、G9001AとG9004A（メッセージ通信モードで使用する場合）です。G9002、G9003、G9004A（CPUエミュレーションモードで使用する場合）は、G9001A（センターデバイス）により制御されます。

接続できるCPUのタイプは、G9001AとG9004Aでは若干異なります。

以下にそれぞれのデバイスで接続可能なCPUを示します（基本的には大抵のCPUが接続可能です）。CPUタイプの選択には、外部端子である“IF0”と“IF1”を使用します。

● G9001Aで選択できるCPUタイプ

設定状態		I/F モード	CPUタイプ	CPU例
IF1	IF0			
L	L	I/F モード 1	モトローラ系の16ビット系CPU アドレスがビッグエンディアン	68000
L	H	I/F モード 2	その他の16ビット系CPU アドレスがビッグエンディアン	H8
H	L	I/F モード 3	インテル系の16ビット系CPU アドレスがリトルエンディアン	8086
H	H	I/F モード 4	8ビット系CPU	Z80

● G9004Aで選択できるCPUタイプ

設定状態		I/F モード	CPUタイプ	CPU例
IF1	IF0			
L	L	I/F モード 1	モトローラ系の16ビット系CPU	68000
L	H	I/F モード 2	その他の16ビット系CPU	H8,8086
H	L	I/F モード 3	8ビット系CPU (液晶コントローラIC制御を意識しています)	6809
H	H	I/F モード 4	8ビット系CPU	Z80

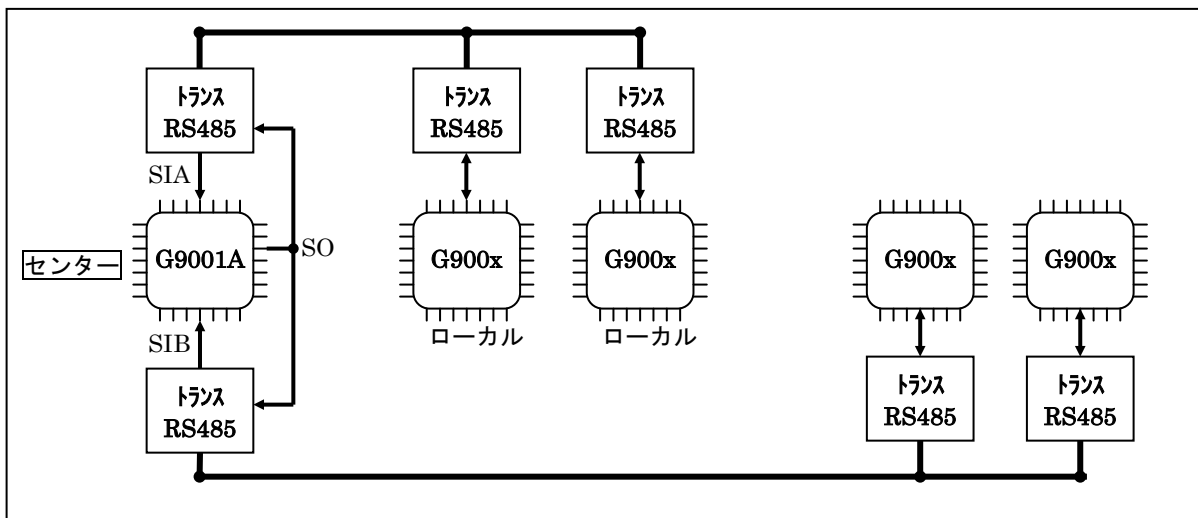
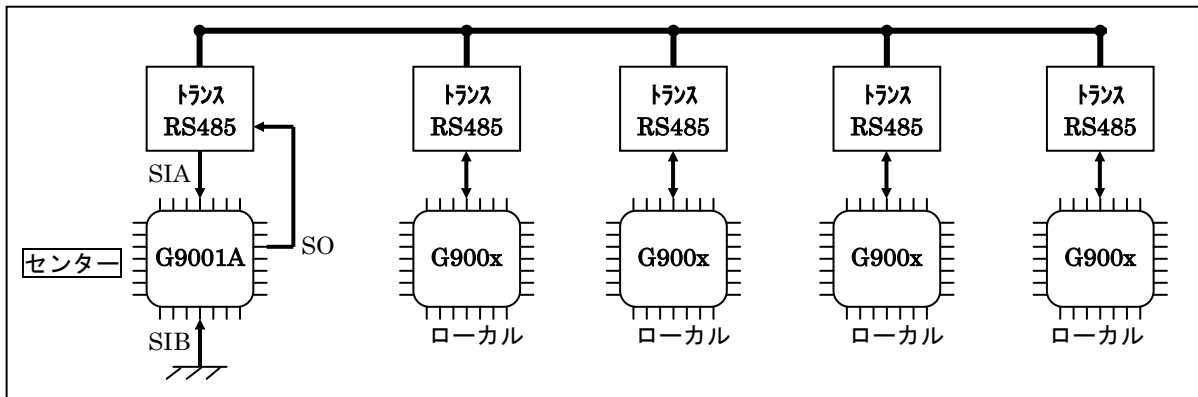
G9000シリーズは全て3.3V単一電源で動作するデバイスです。しかしほとんどの端子が5Vデバイスの接続にも対応します（I/Oデバイスの通信ライン関連などで例外もあります）ので、5VタイプのCPUを直接接続することも可能です。

14. センターデバイスにシリアル入力が2本ある（SIA、SIB）理由は？

通信用のシリアル入力は、RS485などのデバイスで受けた通信回線からのデータを入力するための端子です。

これに対して、通信回線へデータを送り出すための端子として、“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のうちの片方（SIAがメインです）を使用し、もう片方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。

なぜ2本あるかは、以下の図で説明します。



上図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。出力データ（SO）が共通だからです。しかし回線を分けることで、1本にかかる負荷を減らせることができ、長距離のケーブル構成が実現しやすくなります。例えば100mのラインに32個のローカルをぶら下げるのと、50mに16個のラインを2本使用するのでは、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言えます。

操 作 編

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（センターデバイス）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（センターデバイス）に対して行うと、G9001Aは通信回線上にどのようなデバイスが接続されているのかの確認を行います。

確認内容は、デバイスの種類（G9002（I/Oデバイス）か、それ以外かの判定）、I/Oデバイスの場合は入出力ポートの設定状態の確認、各デバイスのデバイスアドレスの取得、などです。

これにより回線上のデバイス情報を自動的に取得、設定できます。この取得結果はCPUから参照すること（もしくは変更すること）が可能です。

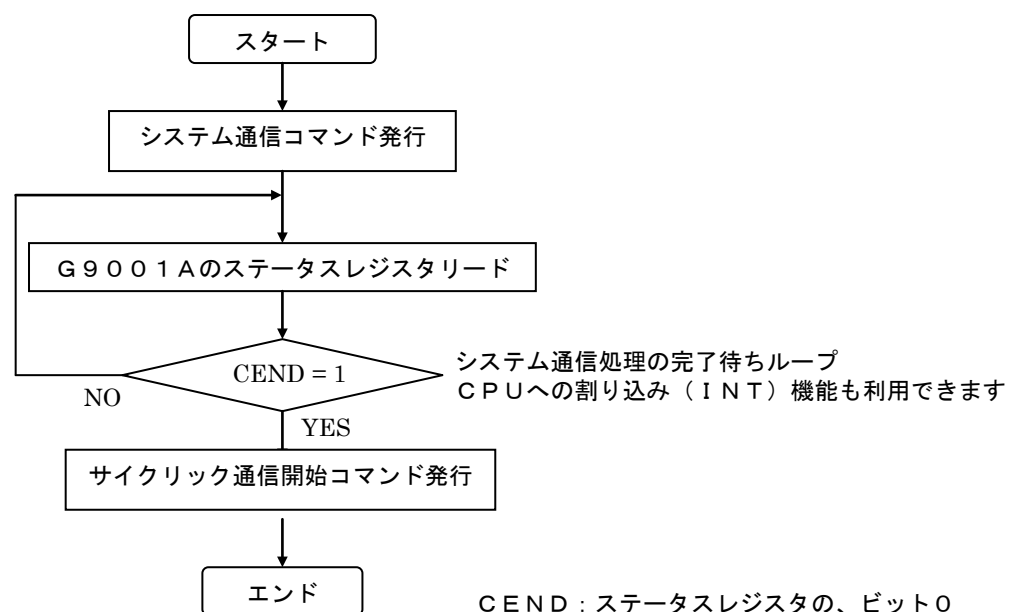
この確認動作が完了すると、G9001Aのステータスレジスタの“CEND”（ビット0）が変化します。これを確認後、次の指示を行って下さい。

② “サイクリック通信の開始” コマンドの発行

①で取得したデバイスアドレスに対して、サイクリック通信が開始されます。

これだけです。例えば入力ポートに設定されているI/Oデバイスからは、その入力データがG9001A（センターデバイス）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デバイスへは、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。これに関しては、以降の項目で説明いたします。また、さらに詳細な情報は、はユーザーズマニュアルでご確認下さい（アドレスマップが記載されております）。



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（センターデバイス）のFIFOへ設定する

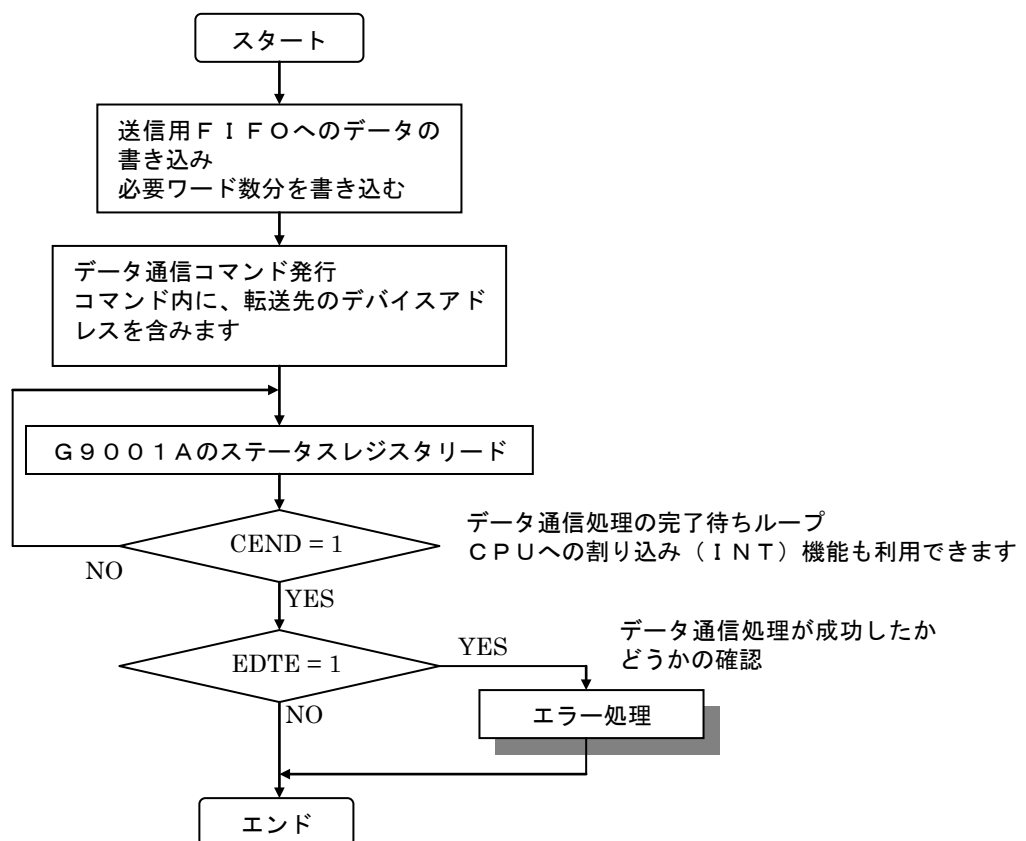
G9001Aは、送信用、受信用それぞれで256バイトずつのFIFOを持っています。

データ通信で送り出したいデータは送信用FIFOへ1ワード（16bitまたは8bit）ずつ書き込んでゆきます。

②指定のデバイスアドレスに対するデータ通信コマンドを発行する

データ通信コマンドは、任意のデバイスアドレスを含むことができます。このコマンドをG9001A（センターデバイス）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、目的のデバイス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下さい。



CEND : ステータスレジスタの、ビット0

EDTE : ステータスレジスタの、ビット4

3. 送信用FIFOへのデータの書き込み方法は？

FIFOとは、特定のアドレスを持たないRAM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RAM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、この時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と呼ばれます。

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は以下のようになっています。

アドレス	内容 (16bit)
000h 002h 004h	他の目的で使用
006h	送信用FIFOへのデータ設定ポート
008h ⋮ 1FEh	他の目的で使用

注：I/Fモード3（IF1=H,IF0=L）のアドレスマップです

16ビットCPUが“006h”番地へ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。

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以下のようになっています。

見ておわかりのとおり、送信用FIFOと同じアドレスです。要するにライト操作を行うと、送信用FIFOへの書き込みとなり、リード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 (16bit)
000h 002h 004h	他の目的で使用
006h	受信用FIFOからのデータ設定ポート
008h ⋮ 1FEh	他の目的で使用

注：I/Fモード3（IF1=H,IF0=L）のアドレスマップです

16ビットCPUが“006h”番地から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。

5. G9002 (I/Oデバイス) 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、G9002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読み込まれます。読み込まれた値は、G9001A (センターデバイス) 内のRAMに格納されます。

アドレス	内 容 (16bit)
000h ⋮ 0FEh	他の目的で使用
100h	デバイスアドレス=00の、ポート1、ポート0
102h	デバイスアドレス=00の、ポート3、ポート2
104h	デバイスアドレス=01の、ポート1、ポート0
106h	デバイスアドレス=01の、ポート3、ポート2
⋮	⋮
1F8h	デバイスアドレス=62の、ポート1、ポート0
1FAh	デバイスアドレス=62の、ポート3、ポート2
1FCh	デバイスアドレス=63の、ポート1、ポート0
1FEh	デバイスアドレス=63の、ポート3、ポート2

} 16bit × 128word

注：I/Fモード3 (IF1=H, IF0=L) のアドレスマップです

G9002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 (センターデバイス) のRAMは上図のように対応しています。

よって、任意のG9002のポート状態を取得するには、そのG9002のデバイスアドレスに対応するRAMからデータをリードすれば良いことになります。

例えば、デバイスアドレス=62のポート1の状態を取得するには、G9001A (センターデバイス) のRAMの“1F8h”番地をリードし、そのデータの上位8ビットを参照することになります。

ただし、G9002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
L	L	L	出力	出力	出力	出力
L	L	H	入力	出力	出力	出力
L	H	L	入力	入力	出力	出力
L	H	H	入力	入力	入力	出力
H	L	L	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RAMへ出力データを書き込みます。このデータをリードすると、出力データとして設定したデータが読み出されます (初期値は不定です)。

6. G9002 (I/Oデバイス) の出力ポートへのデータ出力方法は？

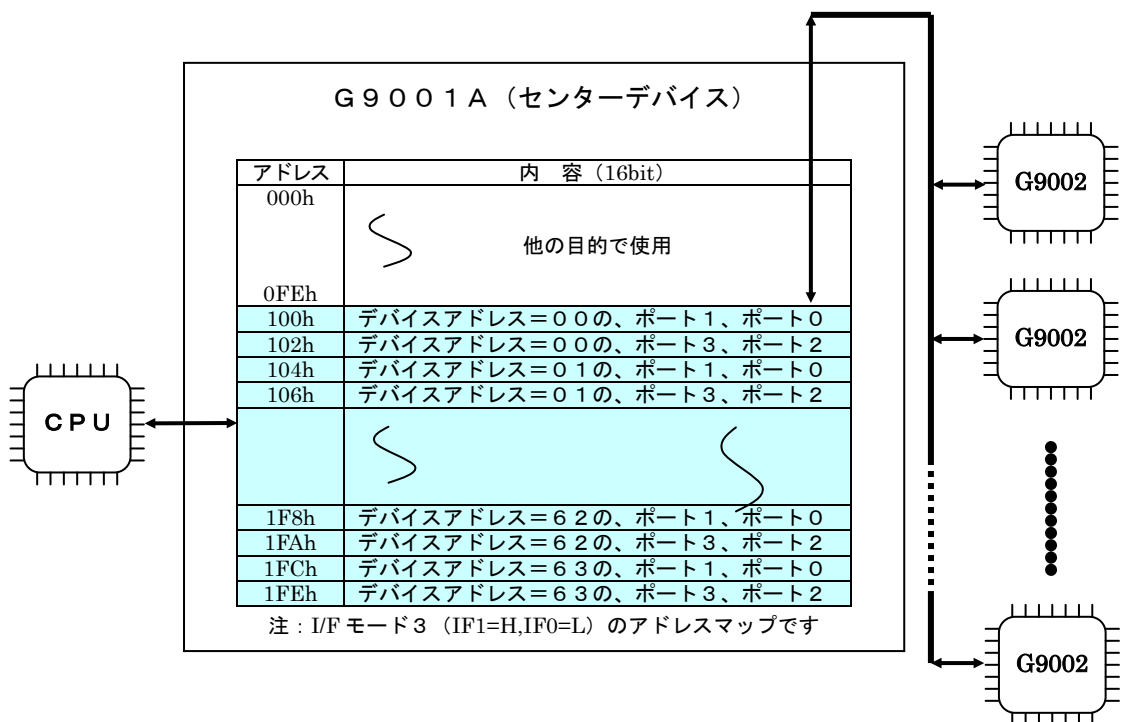
サイクリック通信が開始されると、G9001A (センターデバイス) 内のRAMに格納されたデータがサイクリック周期ごとに自動的に、G9002の出力ポートへ送信されます。

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

ただし、G9002側で入力モードに設定されたポートに対しては、RAMへの書き込みデータは無視されます (G9002からの入力データが常に反映されます)。

以上をまとめると、G9001A (センターデバイス) に接続されているローカルデバイスのポート情報は、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。



7. G9003、G9004Aのステータス情報はどのように参照するか？

G9001A（センターデバイス）が自分の状態をCPUへ示すために、ステータス情報を持っています。

同じようにG9003（PCLデバイス）やG9004A（CPUエミュレーションデバイス）も、それ自体でステータス情報を持っています。

これらのローカルデバイスのステータス情報は、サイクリック通信によりG9001AのRAMへ転送されます。これはG9002（I/Oデバイス）のポート情報が、G9001AのRAMを介してやりとりされるシステムと同じです。入出力ポートを持たないデバイスは、自分のステータス情報をサイクリック通信でG9001Aへ渡しているわけです（G9003は8ビットの汎用ポートを持っているので、これに関してもサイクリック通信でやりとりしています）。

つまりサイクリック通信では、I/OデバイスではI/Oデータがやりとりされ、それ以外のデバイスではステータス情報がやりとりされているのです。従ってI/O情報とステータス情報が格納される部分は同じRAM内となります。

“操作編5. G9002（I/Oデバイス）の入力ポートのデータ取得方法は？”の項目で説明した表は以下のような内容となります。

とあるデバイスアドレスをもつローカルデバイスが、I/Oデバイスであれば、該当するRAMアドレスにはI/O情報が格納され、I/Oデバイスでなければステータス情報が格納されることとなります。

アドレス	内 容 (16bit)
000h ⋮ 0FEh	他の目的で使用
100h	デバイスアドレス=00の、I/Oもしくはステータス
102h	デバイスアドレス=00の、I/Oもしくはステータス
104h	デバイスアドレス=01の、I/Oもしくはステータス
106h	デバイスアドレス=01の、I/Oもしくはステータス
⋮	⋮
1F8h	デバイスアドレス=62の、I/Oもしくはステータス
1FAh	デバイスアドレス=62の、I/Oもしくはステータス
1FCh	デバイスアドレス=63の、I/Oもしくはステータス
1FEh	デバイスアドレス=63の、I/Oもしくはステータス

} 16bit × 128word

注：I/Fモード3（IF1=H,IF0=L）のアドレスマップです

8. 通信回線に接続されているデバイス種類の確認方法

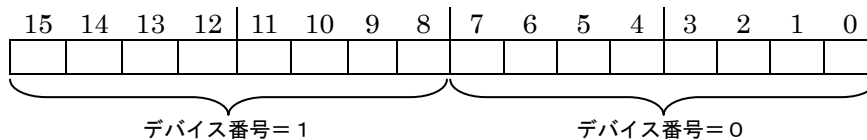
通信回線に接続されているローカルデバイスは、G9002、G9003、G9004Aなどを自由に組み合わせて配置できます。

そして通信回線上に接続されているローカルデバイスに関する情報は、“システム通信コマンド”により自動的に収集が可能です。ここで収集したデータは、G9001A内部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ばれます）。これはCPUからの参照や、変更が可能です。

アドレス	内 容 (16bit)
000h └─┬─┘ 076h	他の目的で使用
078h	デバイスアドレス=01、00のデバイス情報
07Ah	デバイスアドレス=03、02のデバイス情報
└─┬─┘	└─┬─┘
0BCh	デバイスアドレス=61、60のデバイス情報
0BEh	デバイスアドレス=63、62のデバイス情報
0B8h └─┬─┘ 1FEh	他の目的で使用

注：I/Fモード3（IF1=H,IF0=L）のアドレスマップです

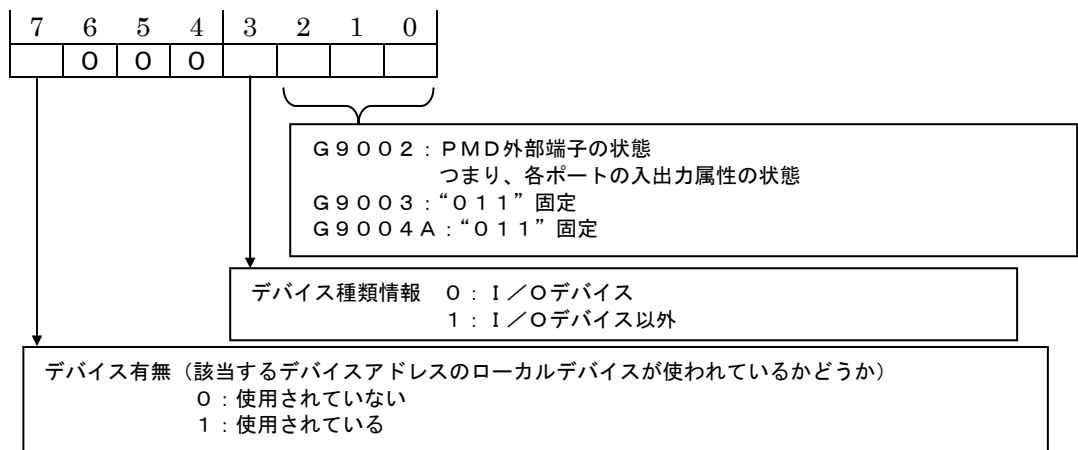
アドレスマップ中の“078h”番地の詳細を示します。



16ビットCPUが“078h”番地を見る場合、上図のようになります。

8ビットごとに一つのデバイスアドレスに対応します。

各デバイスアドレス単位では以下ようになります。



9. I/Oデバイスの入力ポートの変化を割り込み機能で認識するには？

G9002 (I/Oデバイス) 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、常に値が変化するとは限りません。いつ変化するかわからないポートを監視するために、入力変化割り込みによって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。

9-1. 入力値変化を監視するデバイスの指定

どのI/Oデバイスの、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 (センターデバイス) のRAM内の“入力変化割り込み設定エリア”で行います。

アドレス	内 容 (16bit)
000h ∩ 0BEh	他の目的で使用
0C0h	デバイスアドレス=03、02、01、00の入力変化の監視設定
0C2h	デバイスアドレス=07、06、05、04の入力変化の監視設定
∩	∩
0DCh	デバイスアドレス=59、58、57、56の入力変化の監視設定
0DEh	デバイスアドレス=63、62、61、60の入力変化の監視設定
0E0h ∩ 1FEh	他の目的で使用

注：I/Fモード3 (IF1=H,IF0=L) 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”番地を見る場合、上図のようになります。

4ビットごとに一つのデバイスアドレスに対応します。

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アドレス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、デバイスアドレス=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、該当するデバイスアドレスのデバイスのポートは、入力モードに設定して下さい。

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に対して割り込み信号 (INT) が発生します。

9-2. 入力値変化のあったデバイスの特定

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。

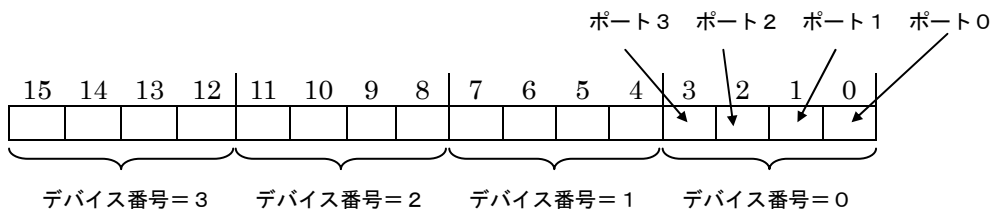
しかし、どこのデバイスのポートが変化したかまでわからなければなりません（特に多数のポートを監視している場合など）。

この場合は以下のRAM内の“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 (16bit)
000h ∩ 0DEh	他の目的で使用
0E0h	デバイスアドレス=03、02、01、00の入力変化割り込みフラグ
0E2h	デバイスアドレス=07、06、05、04の入力変化割り込みフラグ
∩	∩
0FCh	デバイスアドレス=59、58、57、56の入力変化割り込みフラグ
0FEh	デバイスアドレス=63、62、61、60の入力変化割り込みフラグ
100h ∩ 1FEh	他の目的で使用

注：I/Fモード3（IF1=H,IF0=L）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ビットごとに一つのデバイスアドレスに対応します。

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。

この中で、“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アドレス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、デバイスアドレス=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 (デバイスアドレス=59、58、57、56の部分)
リードデータ=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、デバイスの特定ができなくなります）。

“0”に戻すには、該当するビットへ“1”を書き込んで下さい。面倒であれば、リードしたデータをそのままライトする事で“0”に戻せます。

10. G9003、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ますよう、お願いします。

10-1. ステータス変化を監視するデバイスの指定

設定方法自体は、G9002（I/Oデバイス）の“入力値変化を監視するデバイスの指定”の項目と全く同じです。

先に述べたとおり、G9002以外のローカルデバイスは、サイクリック通信で自分のステータス情報をセンターデバイスへ送っています。これらはG9002のI/O情報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、G9002（I/Oデバイス）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。

G9003（PCLデバイス）

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

G9004A（CPUエミュレーションデバイス）

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

網掛け部分はローカルデバイスからデータが送られてくる部分で、それ以外はセンターデバイスからローカルデバイス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まではローカルデバイスからステータスなどが送られてきて、センターデバイスのRAMへセットされます。

デバイスアドレスとRAMアドレスの関係は“G9002（I/Oデバイス）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デバイスの指定”の項目と同じです）すれば良いということになります。

10-2. ポート0の監視上の注意（変化割り込みの使用時）

G9003（PCLデバイス）とG9004A（CPUエミュレーションデバイス）の、ポート0に相当する部分（G9003：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）（I/Oデバイスが接続されている場合は、全ビットを監視します）。

各デバイス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（各ローカルデバイス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、各ローカルデバイスの割り込み要求を監視することになります。

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、その要因となったローカルデバイス側の割り込み要因を解決しなければ、次に発生するであろう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。

G9003（PCLデバイス）で話を進めます。

- ① PCLデバイスの割り込み発生前では、PCLデバイス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 PCLデバイスの動作停止により、PCLデバイスのメインステータスが変化します。

	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

PCLデバイス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のRAM）へ反映されます。

この時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のRAM内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リード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

G9003（PG7デバイス）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されます（ポート0はビット0しか監視しません）。

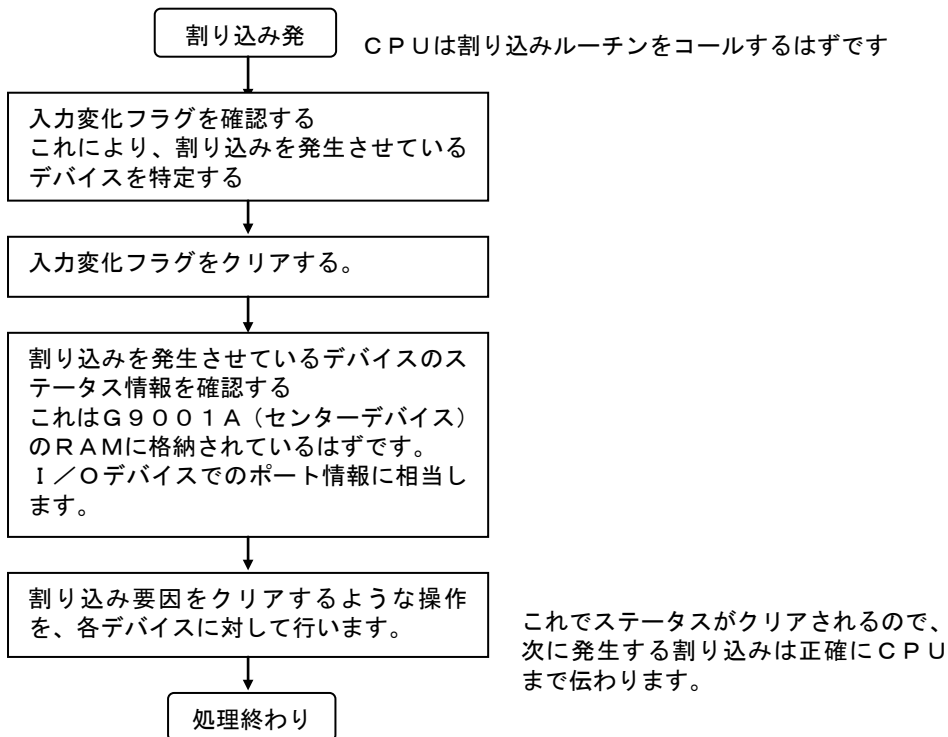
つまりここで発生した割り込みは、CPUまで伝わらないこととなります。

ここで重要な操作が行われていませんでした。

CPU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バイスにおいて割り込み要因をクリアしなければなりません。今回の例では、G9003（PCLデバイス）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、発生デバイスを特定する情報のクリアでしかありません）。

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バイス別に表で示します。

ローカルデバイス	ビット位置	割り込み要因	クリア操作概要
G9003 PCLデバイス	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	G9003のRESTレジスタリード
	3	イベント割り込み	G9003のRISTレジスタリード
G9004A CPU エミュレーションデバイス	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の送信（0400h）
	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の送信（0500h） さらにG9004Aがコントロールしている周辺チップへの対応処理など
	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）を送信

コマンドの送信やレジスタリード操作は、全てデータ通信にて行われます。
詳しくは各デバイスのユーザーズマニュアルをご参照下さい。

11. G9003 (PCLデバイス) を動作させるには？

G9003は弊社製ASIC製品であるPCLシリーズと同様で、モータを制御するためのパルス列を出力できるチップです。

基本的なコマンド体型や操作は従来のPCLに類似しています。従来のPCLと異なるのは、デバイスを制御するCPUがなく、代わりにG9001A (センターデバイス) 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G9003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G9003へ渡します。

G9003のコマンドは、1ワードタイプ (16bit) と3ワードタイプがあります。

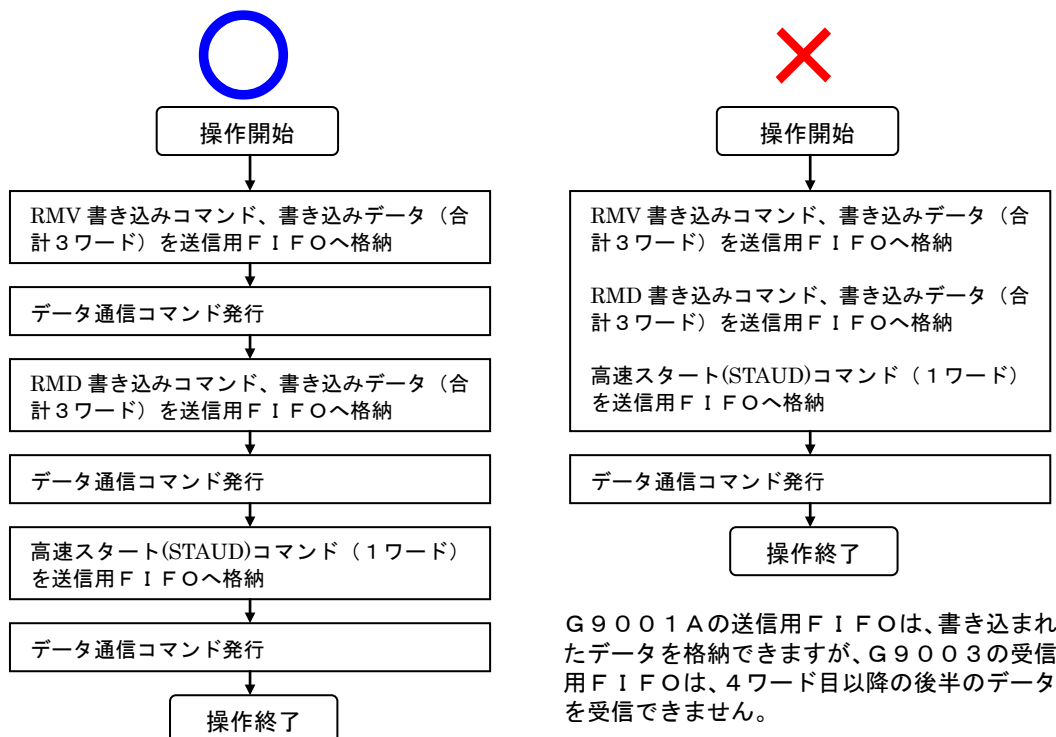
これらコマンドを、送信用FIFOへ格納し、送信コマンドを発行します。“データ通信のやり方は？”の項目を参照下さい。

さらに詳細なフローチャートがユーザーズマニュアル (G9001AまたはG9003) にありますので、ご参照下さい。

ここで注意していただきたいのは、送信用FIFOへ格納できるコマンドは、G9003に対する1アクション分だけということです。いくつかのレジスタへデータをセットしたい場合、コマンドを一気にFIFOへ格納し、送信すると、G9003側でエラーが発生する可能性があります。

G9001Aは、256バイト分の送信用FIFOを持っていますが、G9003は送信用、受信用ともに8バイト分のFIFOしか持っていないためです。

例えばG9003の、RMVレジスタとRMDレジスタへデータを設定し、高速スタート (STAUD) を行おうとしたとします。以下の右の手順ではうまくいかないでしょう。



次の送信データコマンドを発行する前に CPU は前データ通信が完了したかどうかを確認する必要があります。

12. G9004A (CPU エミュレーションデバイス) とのデータのやりとりの方法は？

基本的な操作は、“操作編2. 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。

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。

一つ目のステータスは、サイクリック通信でG9001A (センターデバイス) へ送られてくる情報です。G9004Aのユーザズマニュアルの、“ステータス情報とG9004Aへの動作情報の設定” の項目をご参照下さい。

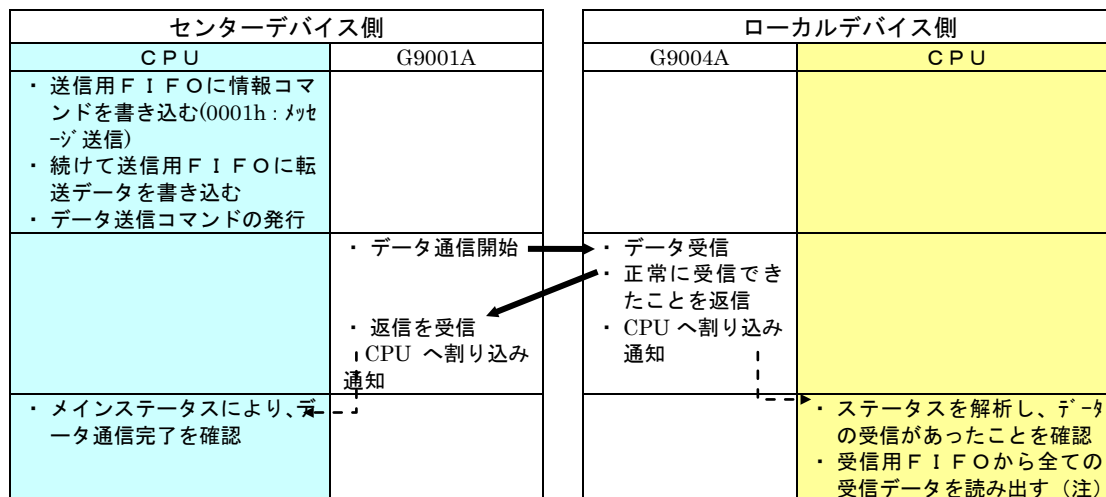
このステータス情報は、G9001A側のCPUでチェックするものです。

二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があります。

G9004Aのユーザズマニュアルの、“ローカル側CPUが操作できるコマンド・ステータス情報” の項目を参照下さい。

このステータス情報は、G9004A側のCPUでチェックするものです。

12-2. メッセージ通信モード : 1 (センター側 CPU から G9004A 側 CPU へデータを渡す)



注：FIFOから受信データを読み出す場合、以下のフラグが“0”になるまでリード処理を繰り返すことで全てのデータを取り出せます。

G9001A：メインステータスのビット10

G9004A：ステータスのビット5

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。

1 2 - 3. メッセージ通信モード : 2 (セタ側CPUからG9004側CPUへデータを渡し、返信データもらう)

センターデバイス側		ローカルデバイス側	
CPU	G9001A	G9004A	CPU
<ul style="list-style-type: none"> ・送信用FIFOに情報コマンドを書き込む(0001h:メッセージ送信) ・続けて送信用FIFOに転送データを書き込む ・データ送信コマンドの発行 			
	<ul style="list-style-type: none"> ・データ通信開始 ・返信を受信 ↓ CPUへ割り込み通知 	<ul style="list-style-type: none"> ・データ受信 ・正常に受信できたことを返信 ・CPUへ割り込み通知 	
<ul style="list-style-type: none"> ・メインステータスにより、データ通信完了を確認 			<ul style="list-style-type: none"> ・ステータスを解析し、データの受信があったことを確認 ・受信用FIFOから全ての受信データを読み出す(注) ・送信用FIFOへ返信データを書き込む ・送信コマンド(10h)を発行
	<ul style="list-style-type: none"> ・サイクリック通信により、ローカル側からのデータがあることを知る ・CPUへ割り込み通知 	<ul style="list-style-type: none"> ・ローカル側送信要求割り込みが発生する この割り込みはサイクリック通信によりG9001Aへ転送され、G9001A側のCPUへ通知される 	
<ul style="list-style-type: none"> ・入力変化割り込み発生 ・入力変化フラグをリセットし、さらにデバイスアドレスを特定する ・特定したデバイスのステータスエリアを確認し、データ送信要求のあることを確認 ・送信用FIFOに情報コマンドを書き込む(0001h:メッセージ送信) ・データ送信コマンドの発行 			
	<ul style="list-style-type: none"> ・データ通信開始 ・返信を受信 ↓ CPUへ割り込み通知 	<ul style="list-style-type: none"> ・データ受信 ・送信用FIFO内のデータを返信 ・先に発生した割り込みがクリアされる 	<ul style="list-style-type: none"> ★受信データは情報コマンドだけなので、FIFOの読み出し処理は不要
<ul style="list-style-type: none"> ・メインステータスにより、データ通信完了を確認 ・受信用FIFOから全ての受信データを読み出す(注) ・送信用FIFOへ情報コマンドを書き込む(0002h) ・データ送信コマンドの発行 			
	<ul style="list-style-type: none"> ・データ通信開始 ・返信を受信 ↓ CPUへ割り込み通知 	<ul style="list-style-type: none"> ・データ受信 ・正常に受信できたことを返信 ・送信用FIFOのクリア ・CPUへ割り込み通知 	
<ul style="list-style-type: none"> ・メインステータスにより、データ通信完了を確認 			<ul style="list-style-type: none"> ・ステータスを解析し、データ転送が正常終了したことを確認

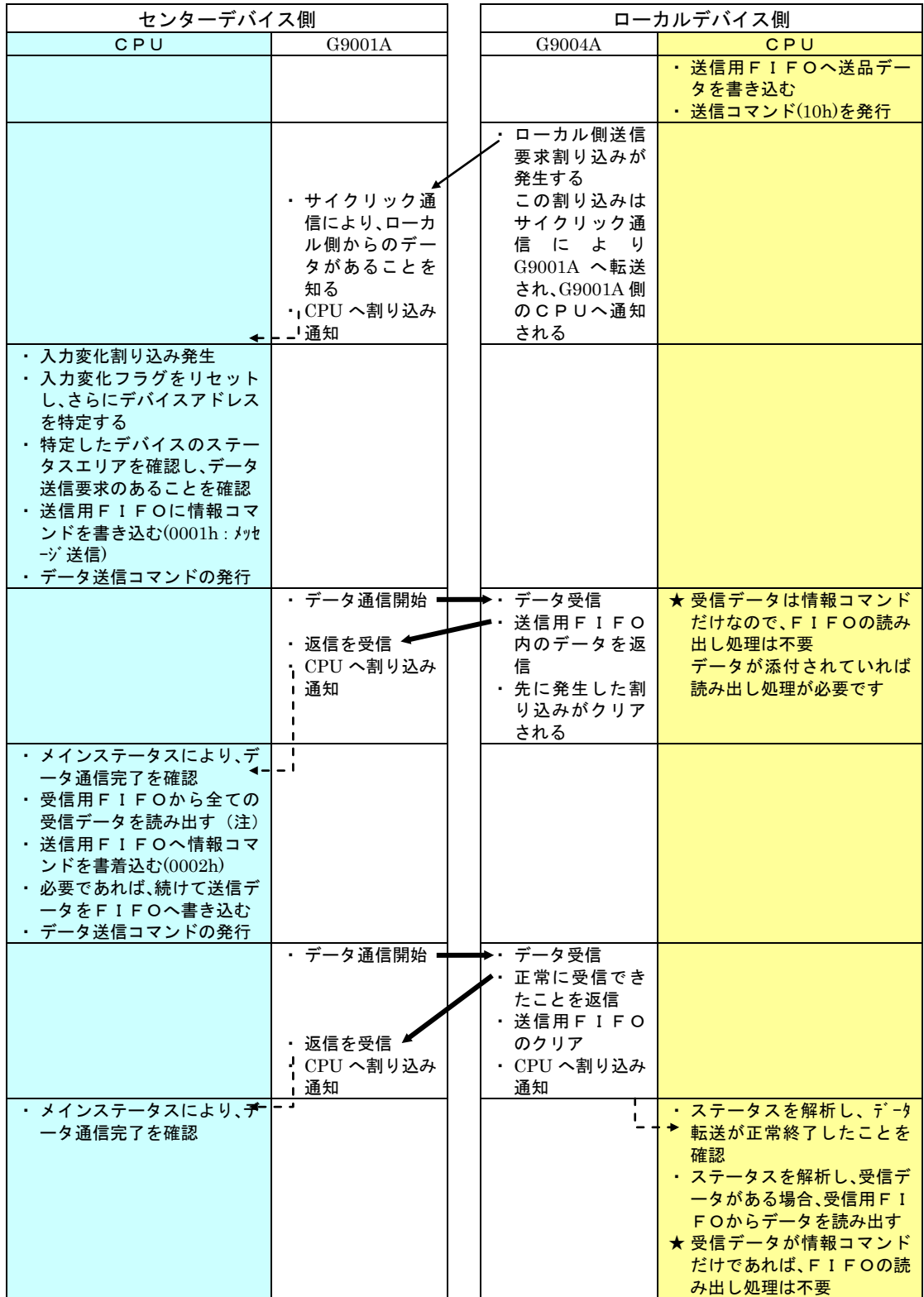
注 : FIFOから受信データを読み出す場合、以下のフラグが“0”になるまでリード処理を繰り返すことで全てのデータを取り出せます。

G9001A : メインステータスのビット10

G9004A : ステータスのビット5

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。細線はサイクリック通信です。
点線はCPUに対する割り込みの発生を表します。

1 2 - 4. メッセージ通信モード : 3 (G 9 0 0 4 A 側 CPU からセンター側 CPU ヘデータを渡す)



注 : F I F O から受信データを読み出す場合、以下のフラグが “ 0 ” になるまでリード処理を繰り返すことで全てのデータを取り出せます。

G 9 0 0 1 A : メインステータスのビット 1 0

G 9 0 0 4 A : ステータスのビット 5

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。細線はサイクリック通信です。

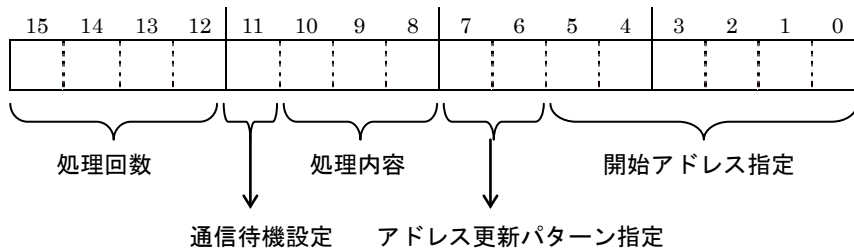
点線は CPU に対する割り込みの発生を表します。

1 2 - 5. CPU エミュレーションモード時のコマンドについて

G 9 0 0 1 A (センターデバイス) 側から、G 9 0 0 4 A 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。

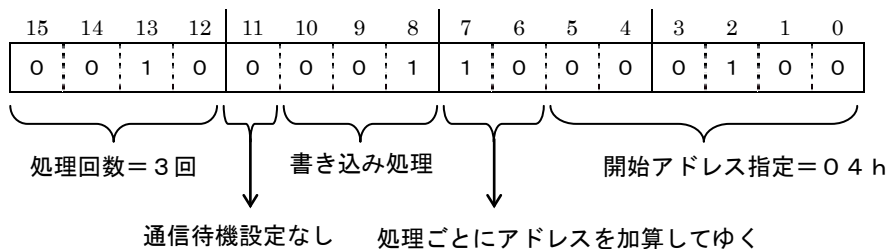
これに対してCPUエミュレーションモード時のコマンドは若干複雑です。以下に詳細を示します。



- 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 通信待機設定
コマンドを受け取ったG 9 0 0 4 A の応答パターンを設定します。詳細は後述します。
- ③ 処理内容
以下が利用できます。

000 : NOP	無効コマンド
001 : 書き込み処理	G9004A が制御する周辺デバイスへデータをライトをする
010 : 読み出し処理	G9004A が制御する周辺デバイスからデータリードをする
011 : 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
100 : 受信処理完了割り込みリセット	G9004A に発生した“受信処理完了割り込み”のクリア
101 : LIRQ 入力割り込みリセット	G9004A に発生した“LIRQ 入力割り込み”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。
 - 0x : アドレス固定
 - 10 : 処理ごとにアドレスを加算してゆく (16bit アクセスなら+2)
 - 11 : 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、G 9 0 0 4 A の動作を説明します。

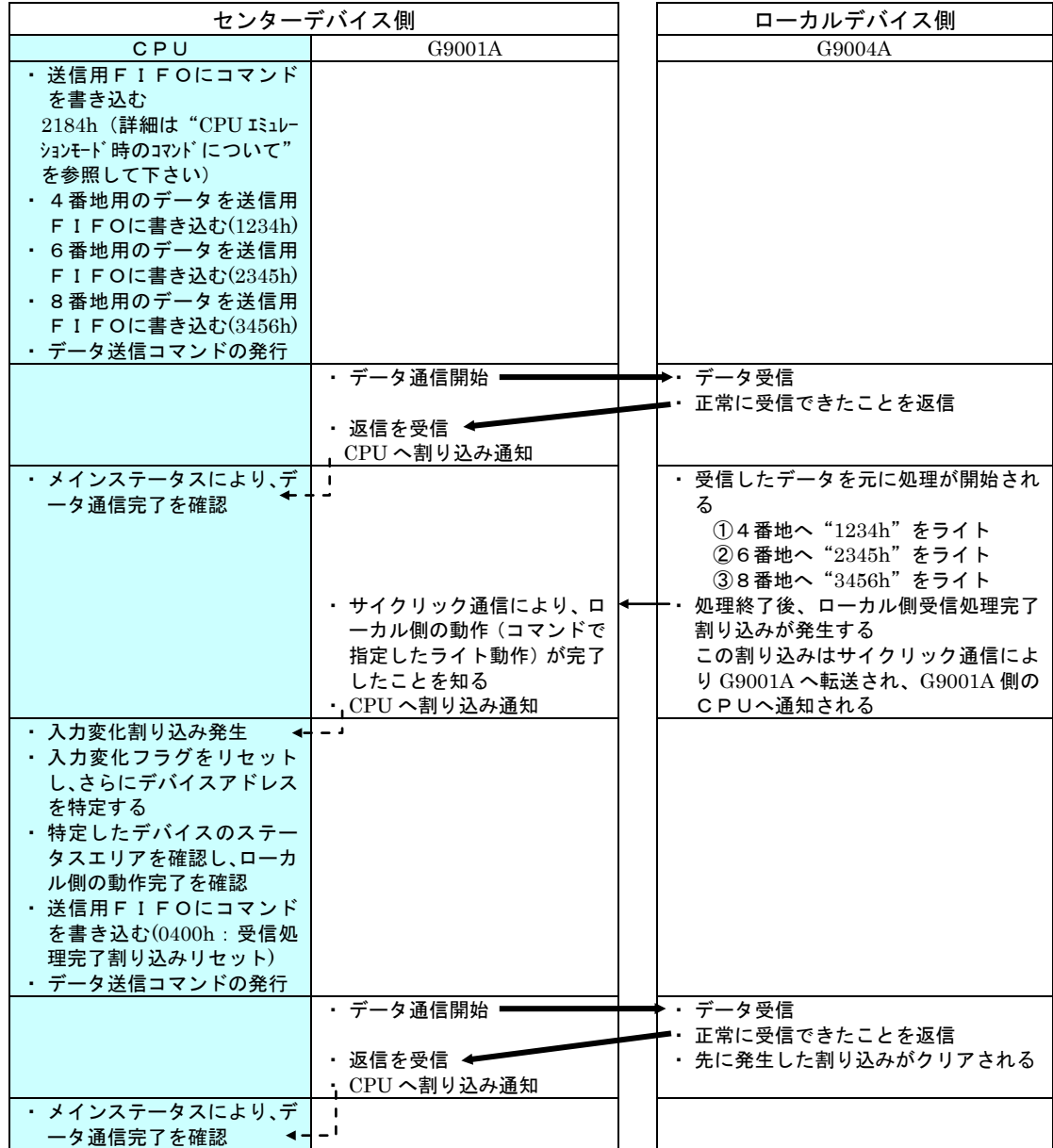


アドレス=4番地から順番に、3回の書き込み処理を、応答処理を待機させないで行います。

この時書き込むデータは、このコマンドの後に続けて指定します。詳細は次項をご参照下さい。

1 2 - 6. CPU エミュレーションモード : 1 (連続データの書き込み)

G9004Aが制御する周辺デバイスの、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3ワード分の処理を行う動作を想定します。通信待機機能は使用しません。



注: FIFOから受信データを読み出す場合、以下のフラグが“0”になるまでリード処理を繰り返すことで全てのデータを取り出せます。

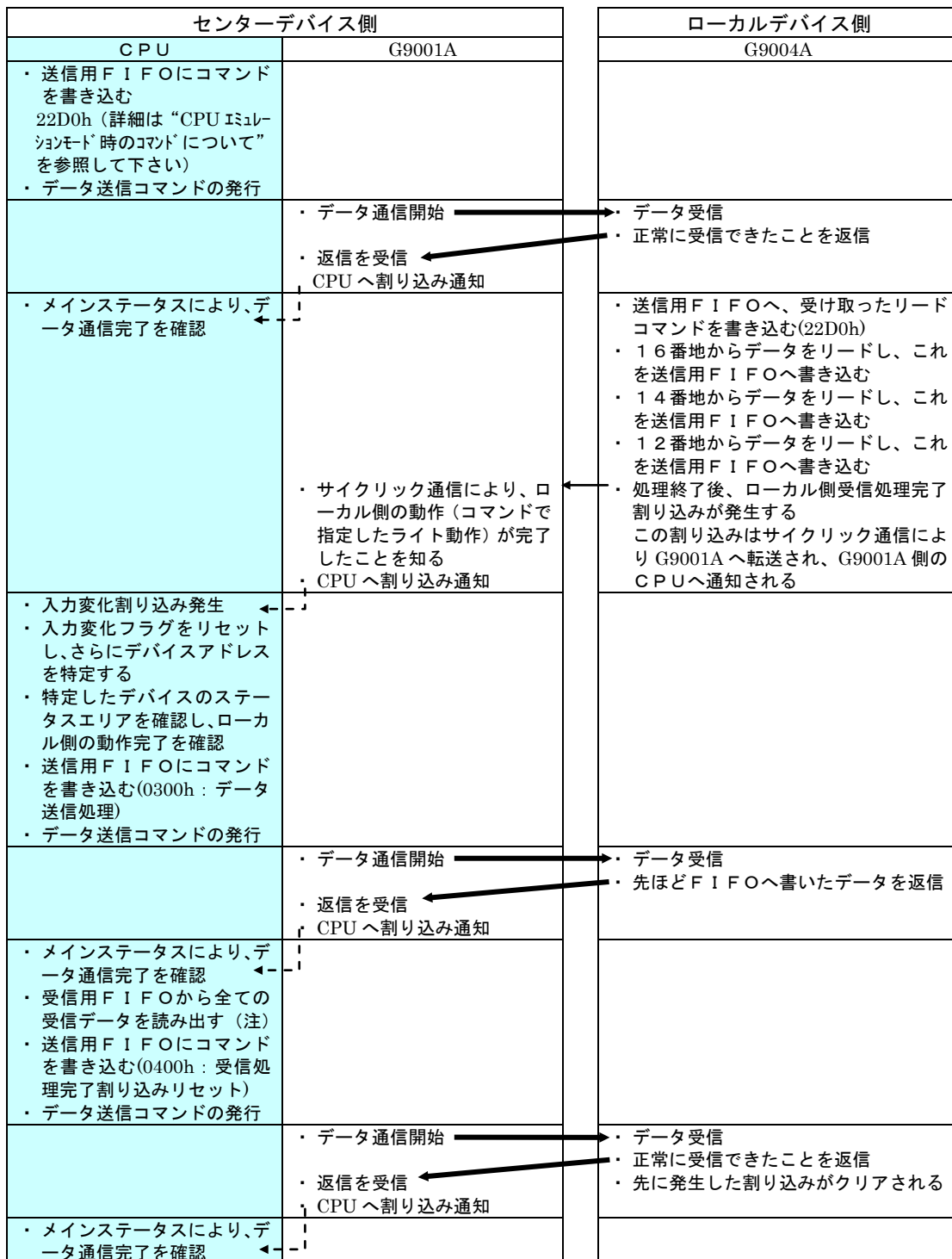
G9001A: メインステータスのビット10

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。細線はサイクリック通信です。

点線はCPUに対する割り込みの発生を表します。

1 2 - 7. CPU エミュレーションモード : 2 (連続データの読み出し)

G9004Aが制御する周辺デバイスの、16番地(アドレスバス=010000b)、14番地、12番地から、合計で3ワード分のデータをリードする処理を行う動作を想定します。通信待機機能は使用しません。



注：FIFOから受信データを読み出す場合、以下のフラグが“0”になるまでリード処理を繰り返すことで全てのデータを取り出せます。

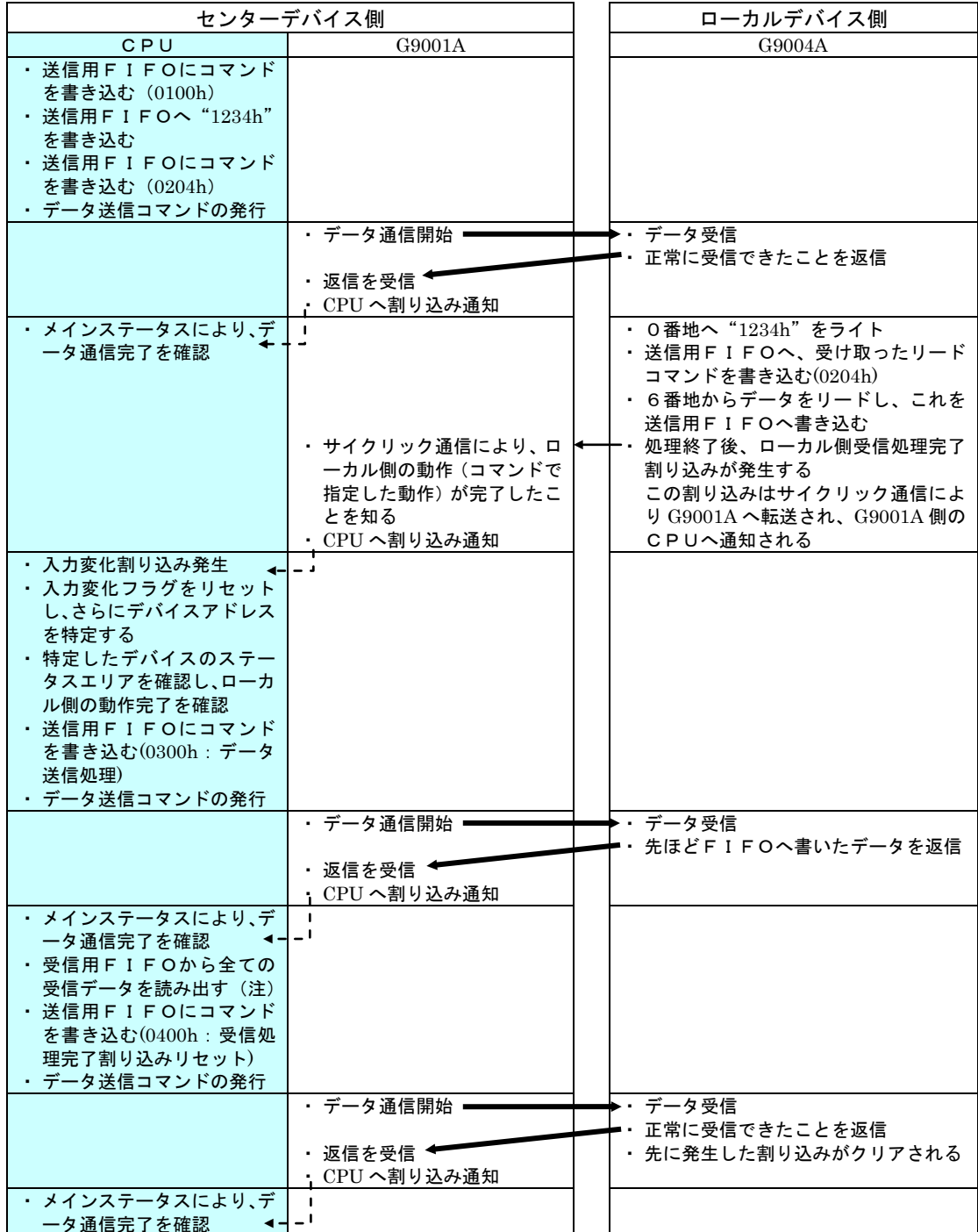
G9001A：メインステータスのビット10

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。細線はサイクリック通信です。

点線はCPUに対する割り込みの発生を表します。

1 2 - 8 . CPU エミュレーションモード : 3 (組み合わせ処理)

G9004Aが制御する周辺デバイスの、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1ワード分のデータをリードする処理を行う動作を想定します。通信待機機能は使用しません。



注: FIFOから受信データを読み出す場合、以下のフラグが“0”になるまでリード処理を繰り返すことで全てのデータを取り出せます。

G9001A:メインステータスのビット10

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。細線はサイクリック通信です。

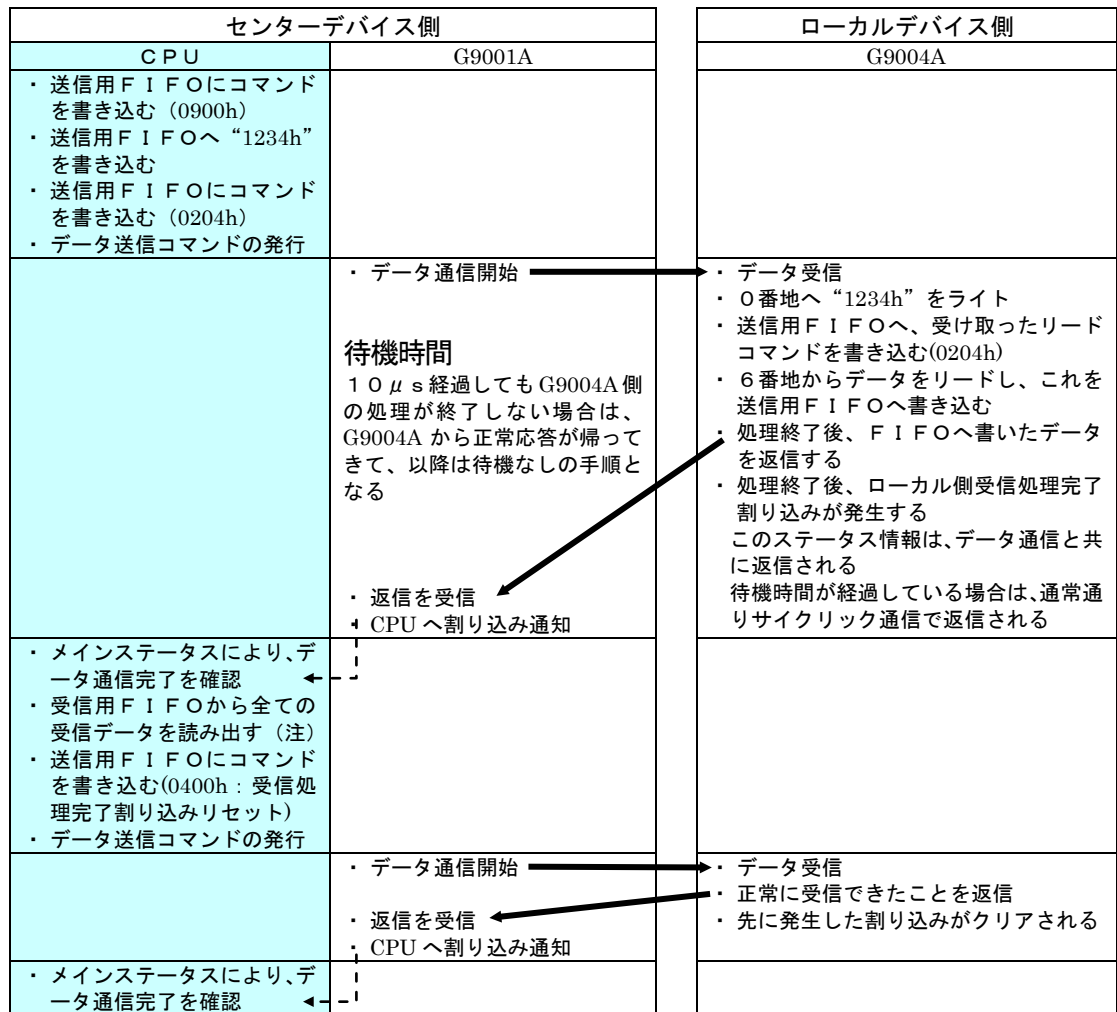
点線はCPUに対する割り込みの発生を表します。

12-9. CPU エミュレーションモード：通信待機設定とは？

待機処理を使用しない場合、G9001Aからコマンドを受け取ったG9004Aは、すぐに正常受信したという応答をG9001Aへ返し、その後でコマンドを解析し、実行します。

これに対して待機処理を使用する場合は、コマンドを受け取るとすぐには応答しないで、コマンド解析と実行を行います。そしてコマンド実行が終了した後で、正常終了の応答（リードデータがあればそれを付けて）の通信を返します。ただしコマンド実行処理時間が10μs（20Mbps時）以上かかる場合は、とりあえず正常応答を返し、待機処理なしと同じ手順となります。

前項での例を、待機処理で行った場合の手順として示します。



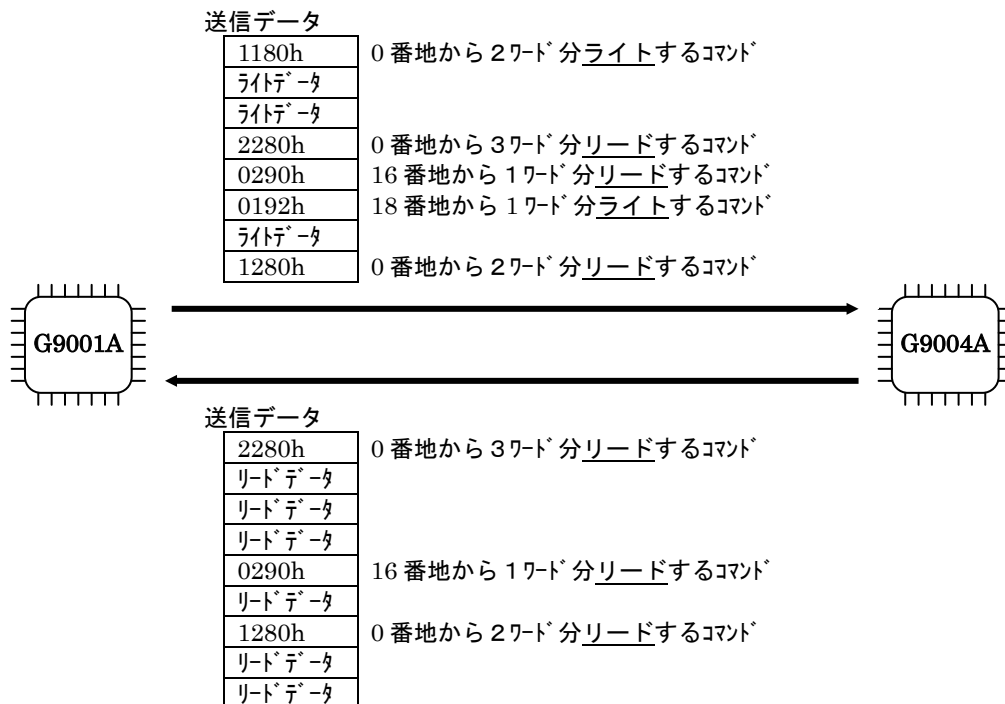
G9004Aが通信待機機能を使用するかどうかの判断は、最初のコマンドで決定されます。

したがって、最初のコマンドで通信待機機能を使用しないようになっていると、次のコマンドで通信待機を設定しても無視されます（逆でも同様です）。

12-10. CPU イミュレーションモード：リードコマンド使用時にG9004Aから返信されるデータ形式
G9004Aが制御する周辺デバイスからデータをリードするわけですから、当然リードしたデータが返ってきます。しかし返信データにはリードデータ以外にも、そのデータのリード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下さい。

例えば以下のように、ライトコマンドとリードコマンドが混在した場合（リード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、リードデータがG9001Aへ返信されるのですが、その返信データは以下に示すような内容となります。



まずリードコマンドがそのまま入り、その後実際にリードされたデータが続きます。

G9001A側のCPUでは、発行したリードコマンドと、リードデータの対比を行いながら処理を進められます（発行したリードコマンドを覚えておく必要がなくなります）。

12-11. CPU イミュレーションモード：リードコマンドでの返送データが128ワードを越える場合

リードコマンドでG9004Aが制御している周辺デバイスからデータをリードする場合、リードデータの前にリード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、リードコマンドによって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（処理回数設定などを使用するとさらに大きくなります）。

これらのデータが128ワードを越えた場合、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、十分ご注意下さい。

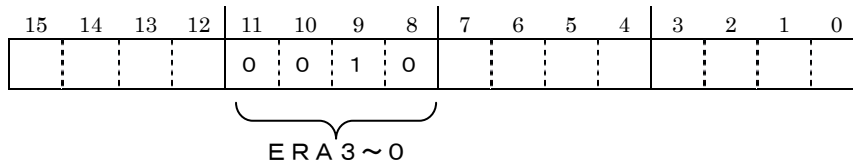
1 3. I/Oデバイスに対してデータ通信を行うとどうなるか？

G9002 (I/Oデバイス) に対してデータ通信を行ってはいけません。

G9002は受信用FIFOを持っていませんので、送信データを受信できません。

間違ってG9002へデータ通信を行った場合、G9002からはエラー応答のデータが返ってきます。この場合、G9001A (センターデバイス) の割り込みステータスが以下のようになり (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)



1 4. I/Oデバイスとセンターデバイスで、異なるPMD設定をするとどうなるか？

G9002 (I/Oデバイス) は、PMD外部端子の設定で、4ポートある入出力端子のI/O属性を決定します。

一方G9001A (センターデバイス) は、内部RAMに、各G9002のPMD端子状態を記憶するエリアがあります。

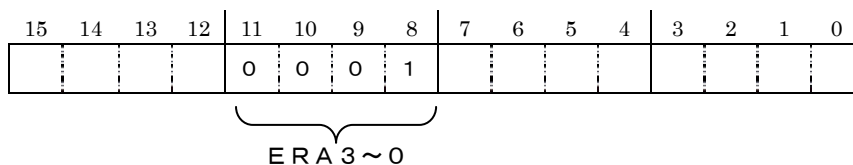
サイクリック通信時には、この内部RAMのPMD状態記憶エリアを参考にしてポートデータのやりとりが行われます。つまりG9001Aの内部RAMのPMD情報と、実際のG9002のPMD端子状態が一致していなければなりません。

通信処理開始前に、G9001Aがシステム通信を行い、バスに接続されているローカルデバイスの情報を自動収集した場合、各G9002のPMD端子状態も収集されるので、内部RAMのPMD情報と実際のG9002のPMD端子状態が食い違うことはまずありません。

しかし、CPU側からこの内部RAMのPMDエリアを手動で設定した場合などは食い違う可能性が残ります。食い違った状態でサイクリック通信が行われると、間違っただサイクリック通信を受け取ったG9002はエラー応答を返します。

この場合、G9001A (センターデバイス) の割り込みステータスが以下のようになり (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)



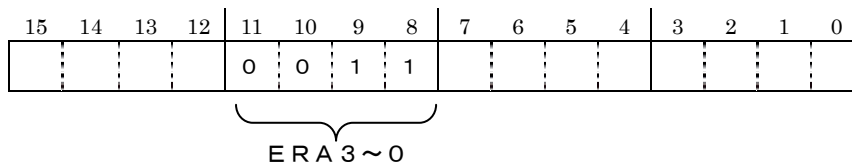
15. PCLデバイスが、受信FIFO容量以上のデータを送信するとどうなるか？

G9001A（センターデバイス）の送信用FIFOは128ワード（256バイト）あります。これに対し、G9003（PCLデバイス）の受信用FIFOは4ワード（8バイト）しかありません。

仮にG9001Aの送信用FIFOに9バイト以上のデータを書き込んで、これをG9003へ送信した場合、G9003はエラー応答を返します。

この場合、G9001A（センターデバイス）の割り込みステータスが以下のようになり（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）



注！：G9003のFIFOは4ワードですが、実際に使用しているのは最大で3ワードだけです。4ワードのデータを送ってもエラーにはなりません、動作は保証されませんが、PCLデバイスへのデータ転送は、基本的にワンコマンド単位で行ってください。

16. G9004Aで、受信用FIFOのデータをリードしないでいるとどうなるか？

G9004Aをメッセージ通信モードで使用する場合、G9004A自体は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内のデータのリード処理を行わ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のリード処理を保留にするとどうなるのでしょうか？

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側から一生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からデータをリードして下さい（受信用FIFOのリセットでも良いです）。

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。

17. 通信が不可能な状態になると各デバイスはどうか？

通信回線に何らかの障害が発生し、G9001A（センターデバイス）から、各ローカルデバイス側へ通信ができなくなるとどうなるでしょうか？

各ローカルデバイスは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、各ローカルデバイスに内蔵されているウォッチドックタイマが働きます。

このタイマーは各デバイス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時はタイムアップとなり、以下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外部端子	処 理
H	タイムアップ時のローカルデバイスの状態を保持したままで、自分宛の通信の再開を待つ。 この場合、外部端子の出力状態はタイムアップ時の状態を保持します。 PCLデバイスなどは、内部が動作している場合、動作しっぱなしです。
L	デバイス回路をリセットし、自分宛の通信の再開を待つ。 外部端子の状態も初期化されます。

18. ローカルデバイスが機能を停止したことを認識するには？

サイクリック通信稼働中に、どこかのローカルデバイスの電源が落ちたり、通信回線から抜けたり（事故、または人為など、要因はいろいろです）した場合、G9001A（センターデバイス）が、そのデバイスアドレスへ送信したサイクリック通信は無応答となります（返信が返ってこない）。

G9001Aは、3周期連続でサイクリック通信に 응답しないデバイスがある場合、通信回線上にローカルデバイスが存在しなくなったと見なします（サイクリック通信からそのデバイスが除外されることはありません。削除の必要があれば、CPU側から削除の操作を行って下さい）。

この時、G9001Aのメインステータスのビット2（EIOE）は“1”となります。

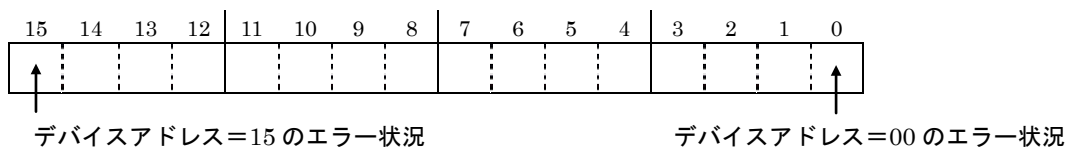
エラーとなったデバイスアドレスは、G9001Aの内部RAMの以下の領域（I/O通信エラーフラグエリア）のチェックで確認できます。

アドレス	内容 (16bit)
000h ∩ 0B6h	他の目的で使用
0B8h	デバイスアドレス=15~00のI/O通信エラーフラグ
0BAh	デバイスアドレス=31~16のI/O通信エラーフラグ
0BCh	デバイスアドレス=47~32のI/O通信エラーフラグ
0BEh	デバイスアドレス=63~48のI/O通信エラーフラグ
0C0h ∪ 1FEh	他の目的で使用

注：I/Fモード3（IF1=H,IF0=L）のアドレスマップです

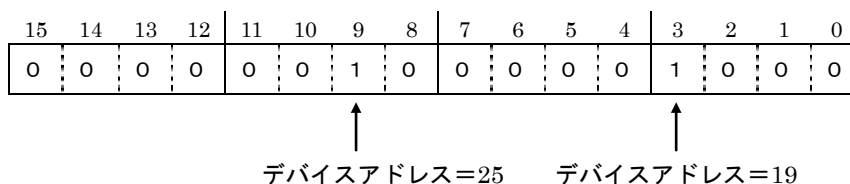
このデータは以下のように解釈します。

0B8hのデータの場合



このように、データの下位側がデバイスアドレスの若いもののI/O通信エラーの状況を示します。このビットが“1”だと、該当するローカルデバイスとのサイクリック通信ができなくなったことを示します。

例えば、0BAhが以下のような状態になったとします。



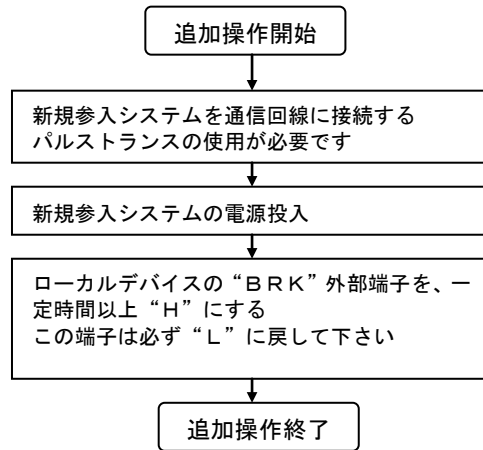
0BAhのデータの最下位ビットは、デバイスアドレス=16に相当します。ここから順番に数えると、“1”となっているところは、デバイスアドレス=19と、25になることがわかります。

“1”になったビットをクリアするには、該当するビットへ“1”を書き込んで下さい。

19. 稼働中に新たなデバイスを参入させるには？

何らかの原因で通信回線から脱落してしまったデバイス（前項参照）や、全く新しいローカルデバイスのサイクリック通信への参入は、サイクリック通信を中断しなくとも可能です。

この時の手順を示します。



以上です。

G9001A（センターデバイス）は、一定周期で新規参入デバイスがないかどうかを監視しています。

ここで、BRK外部端子に一定時間（3.2ms以上）の“H”信号を入力したデバイスがあると、G9001Aは新規参入デバイスがあると確認します。

新規参入デバイス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下さい。

これで新規参入したローカルデバイスの情報を取得できます。取得後は、この新規デバイス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：通信回線とローカルデバイスのシステムが、パルストランスなどにより絶縁されていなければ、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、通信回線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下さい。

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（センターデバイス）と、それ以外（ローカルデバイス）で端子状況が異なります。以下の表をご参照下さい。

デバイス種類	端子名	内容
センターデバイス	MCRY	通信回線上の状態を示します。 回線上にデータが載っている場合に“L”となります。
	MERR	以下のケースで $3.2\mu\text{s}$ だけ“L”となります。 ① ローカルデバイスからのデータを正常に受け取れなかった場合（チェックコードでエラーがあった場合など）。 ② ローカル側への通信完了後、 $20\mu\text{s}$ 待っても応答がなかった場合（無応答と判断）。
ローカルデバイス	MSEL	受信成功 センターデバイスからの通信を正常に受信できた場合、 $3.2\mu\text{s}$ だけ“L”となります。
	MRER	受信失敗 センターデバイスからの通信を正常に受け取れなかった場合（チェックコードでエラーがあった場合など）、 $3.2\mu\text{s}$ だけ“L”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がタイムアップ、それと共に本端子が“L”となります。サイクリック通信が再開されると“H”になります。

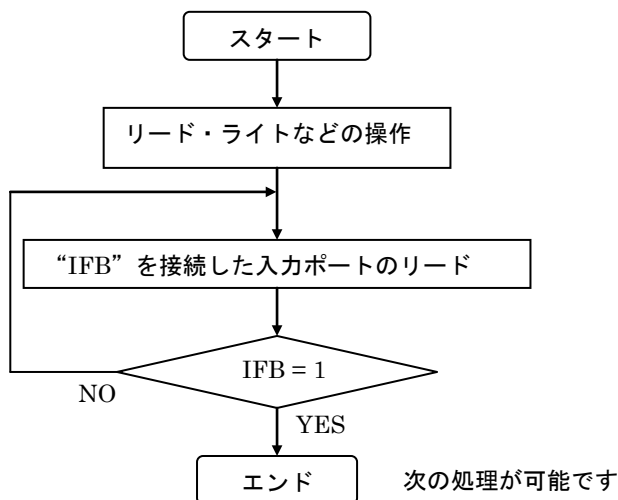
2.1. ウェイト制御用端子の無いCPUを使用するには？

G9001A（センターデバイス）とG9004A（CPUエミュレーションデバイス）はCPUにより制御されます。この時CPUからのリード操作やライト操作に対して、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“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を待たせるようにして下さい（大きな無駄ループで無理矢理待っても良いですが・・・）。

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）へ接続して下さい



G9001Aのユーザズマニュアルの、“ウェイト機能のないCPUとの接続”（第三章の、1-1-5. 項）に詳細がありますので、ご参照下さい。

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（センターデバイス）はアドレスバスを9本持っています。

これは512バイトある内部RAMへ直接アクセスできるようにしたため、通常はCPUからも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）しか接続しないでもG9001Aの制御は可能です。

これはG9001Aへのコマンド操作にて内部RAMへアクセスする機能を利用したもので、従来の弊社製ASICであるPCLシリーズの操作と同じものです。

G9001Aのユーザズマニュアルの、“コマンド”（第一章の、4-8. 項）に詳細がありますので、ご参照下さい。

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と80MHz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デバイスに接続されているクロックは専用クロックで、1個のG9000デバイス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デバイスに接続されているクロックは兼用クロックですが、2～3個程度のデバイス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デバイスに接続されているクロックは兼用クロックで、複数個のデバイスを駆動している

対処1 → 新たに40MHzの専用クロックを設けて下さい

対処2 → 兼用クロックを80MHzにして下さい（無理かな？）

2 4 . 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を左右します。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、おおざっぱな基準を述べてゆきます。

2 4 - 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下さい）。

弊社ではカテゴリ5以上の製品を推奨します。

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

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言えません。最低限でも減衰量の保証が欲しいです）。

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。

2 4 - 2 . RS485デバイスの選定

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（標準で12k Ω ）を選定して下さい。この値は24k Ω 以上が理想です。

5Vと、3.3Vのデバイスが選択できますが、5Vの製品が理想です（G9000シリーズは3.3Vですが）。

またフェイルセーフ機能付きのものを選択して下さい。

2 4 - 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 μ H前後のものを使用して下さい（これより大きい分には問題ないと思われませんが、数百 μ H程度だと64個ものローカルデバイスの接続は不可能です。また5VのRS485デバイスの使用を前提としています）。

2 4 - 4 . 終端抵抗の抵抗値の決定

通信回線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。

25. レベルシフタが必要なケースとは？

G9000シリーズは3.3V単一電源で動作するデバイスです。

しかし、たいいていの場合5Vデバイスを直接接続することが可能です（詳しくはユーザーズマニュアルでご確認下さい）。

しかし、通信に関わる端子（“SO”、“SOEL”、“SOEH”などの出力信号、“SI”などの入力信号）に関しては、高速性が必要なため、基本的には5Vデバイスを直接接続できません。

つまり、RS485デバイスとして5V製品を選定した場合、レベルシフタなどにより適切な電圧への変換が必要となります。

以下に、Motionnetの特長をまとめました。

◆通信周期とI/O点数

I/O点数は2048点と多く、通信速度も20Mbpsと高速です。

◆I/Oデータの保存

MotionnetはRAMを内蔵していますので、部品点数の削減と値段の面で有利になります。

◆ローカルデバイスのアドレス設定

ローカルデバイスのアドレスを、システム通信により自動取得できますので、アドレスをランダムに選択できます。

◆プラグ&プレイ機能

ローカルデバイスの増設時、ブレーク端子にパルス信号を入力する事により、センター側のCPU側にデバイス増設を認識させられますので、プラグ&プレイ機能を実現できます。

◆通信状態の管理

ウォッチドックタイマ機能により通信状態の管理ができます。

G9001A（センターデバイス）から本ローカルデバイスへの送信間隔が、規定時間以上になった時にタイムアップします。タイムアップ時に現在の状態を維持、またはリセット状態の設定を端子で設定できます。

◆入出力ポートの設定

Motionnetは、ポート単位で入出力の設定が出来ますので、効率よく使用でき、変更・追加も容易になります。

◆軸制御

・Motionnetには、1軸制御用としてG9003があり、定速・直線加減速・S字加減速により、多種多様な連続動作、位置決め動作、原点復帰動作等の制御が行えます。

ローカルデバイスを全てG9003にすれば、省配線による64軸のモーションコントロールが可能です。

・G9004AをCPUエミュレーションモードにして使用すると、G9004Aに対してPCL6045B（弊社製の補間機能付き4軸制御LSI）であれば2ヶ接続出来ます。従って、ローカルデバイスを全てG9004Aにすれば、省配線による512（64×4×2）軸のモーションコントロールも可能です。

◆CPU間データ通信

G9004Aをメッセージ通信モードで使用すると、1回の通信で最長127ワードのメッセージデータ（フォーマット自由）を通信できますので、G9001AとG9004Aに接続したCPU間で自由にデータ通信ができます。

索引

BRK	45	サイクリック通信	5
CPUエミュレーションデバイス.....	3	システムクロック	48
EIA-485規格	10	システム通信	16
FIFO	18	ステータス	8
G9004Aのステータス	26	センターデバイス	2
I/Fモード	12	タイムアップ	43
I/Oデバイス.....	2	データ通信	6
I/O通信エラーフラグエリア	44	データ通信コマンド.....	17
I/O通信の開始	16	デバイスアドレス	4
IF	12	デバイス情報エリア.....	22
LWRQ	47	トランシーバ機能	10
MCRY	46	トランス.....	10
MERR	46	パルストランス.....	10
MRER	46	ポーリング/セレクトイング方式.....	5
MSEL	46	マルチドロップ形式.....	4
PCLデバイス.....	2	メッセージ通信モード.....	3
PMD	2	ローカルデバイス	2
RS485	10	終端抵抗.....	11
SIA.....	13	通信周期.....	8
SIB.....	13	通信待機設定	35,39
SO	13	電源分離.....	10
TOUT	46	特性インピーダンス.....	49
TUD	43	入力変化割り込み	23
WRQ	47	入力変化割り込みフラグエリア	24
ウェイト制御用端子	47	入力変化割り込み設定エリア	23
ウォッチドックタイマ	43	反射	11

