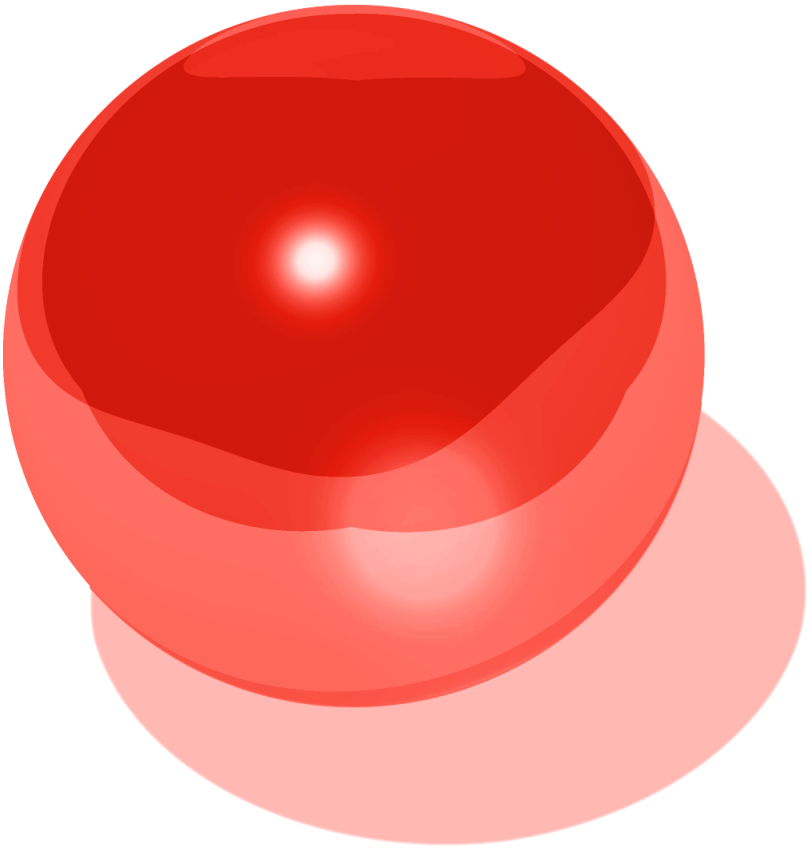
TOYODENKI

μGPCSX series

プログラミングマニュアル

命令語編



IGJ057A

このたびは、TOYO FA ディジタルコントローラμGPCsx をお買い上げいただきまことにありがとうございます。

このプログラミングマニュアル　命令語編は、プログラミングの考え方、リレーおよびレジスターの説明、各命令語について解説したものです。μGPCsx を正しくお使いいただくために、このプログラミングマニュアルをよくお読みください。

また、下表に示す関連マニュアルも併せてお読みくださるようお願いいたします。

名称	マニュアル番号	記載内容
μGPCsx シリーズ　プログラミングマニュアル（オペレーション編）	IGJ058A	TdsxEditor のメニュー、アイコンなどの説明および TdsxEditor のオペレーションのすべてを解説
μGPCsx シリーズ　プログラミングマニュアル（技術編）	IGJ059A	プログラムの組み方、作成方法について解説
μGPCsx シリーズ　ユーザーズマニュアル（ハードウェア編）	IGJ060A	μGPCsx シリーズのシステム構成、各モジュールのハードウェア仕様などを解説

ご注意

（1） 本書の内容の一部または全部を無断で転載、複製することは禁止されております。

（2） 本書の内容に関しては、改良のため予告なしに仕様などを変更することがありますのでご了承ください。

（3） 本書の内容に関しては万全を期しておりますが、万一ご不審な点や誤りなどお気付きのことがありましたら、お手数ですが巻末記載の弊社営業所までご連絡ください。その際、表紙記載のマニュアル番号も併せてお知らせください。

本製品をご使用の前に「安全上のご注意」をよくお読みの上、正しくご使用ください。
ここでは、安全上の注意事項のレベルを「危険」および「注意」として区分しており、意味は下記のとおりです。

危険 ：取り扱いを誤った場合に、死亡または重傷を受ける可能性があります。

注意 ：取り扱いを誤った場合に、中程度の障害や軽傷を受ける可能性、あるいは物的損傷が発生する可能性があります。

なお、 注意 に記載した事項でも、状況によっては重大な結果に結びつく可能性があります。

いずれも重要な内容を記載しておりますので、必ず守ってください。
特に注意していただきたい点を以下に示しますが、マニュアルの本文中にも上記記号で示します。

危険
● 非常停止回路・インタロック回路などは、ＰＣの外部で構成してください。 ＰＣの故障により、機械の破損や事故のおそれがあります。

注意
● 運転中のプログラム変更、強制出力、起動、停止などの操作は十分安全を確認してから行ってください。 操作ミスにより機械が動作し、機械の破損や事故のおそれがあります。

※マニュアル番号は、このマニュアルの表紙の右下に記載しております。

印刷日付	※マニュアル番号	改訂内容
2001 年 5 月	IGJ060A	初版印刷（暫定版）

第１章　概要

第１章　概要

μGPCsx シリーズでは、アプリケーションプログラム用の言語として、コンピュータ用言語(アセンブラ言語，C 言語など)を用いることなく、制御用言語として新たにμ-GPC 言語を開発しました。

μ-GPC 言語は、論理演算にはシーケンサなどで従来から使用されてきたラダーネットワークを、数値演算にはアナログコンピュータなどで使用されてきた D・F・S（データ・フロー・シンボル）を用いており、パーソナルコンピュータを利用したプログラミングツール上でビジュアルなプログラミングを可能とする新しいプログラム手法です。

μ-GPC 言語は次の特徴があります。

(1) コンピュータ言語の概念を変えた制御用として最適な言語体系です。
（マイクロプロセッサの処理手順を記述するのではなく、データの加工手順を記述します。）

(2) 図式表示言語であり、プログラムが非常に分かり易く、誤りの少ないプログラミングが可能です。
（論理演算とデータ処理の両者を同一画面上でプログラミングすることが可能です。）

(3) 取り扱うデータの種類（整数、BCD 型、実数等）を自動変換するため、プログラムのなかで型変換命令を使用する必要がありません。
（データを分割して使用する場合は変換命令が使用可能）

(4) S 字演算等の制御向け時系列関数が豊富に利用できるため、多数のラダーシンボルで実現した機能が 1 個のシンボルで記述され、誰でも簡単にプログラムすることができます。
（プログラムの実行時間を計測しながら自動調整していますので、時間意識は全く不要です。）

(5) 3 つのインデックスレジスタ（X、Y、Z）によるインデックス修飾が可能であり、コンピュータ的な柔軟なプログラミングも可能です。
（ジャンプ命令を使用したプログラムループによるステップ数の減量にも効果あります。）

(6) サブプログラムを使った構造化プログラムを作成することが容易に可能です。
（アプリケーションプログラムの再利用・標準化に最適です。）

(7) 2 個のマルチタスクプログラムを作成することが可能で、効率的なシステムが構築できます。
（実行サイクルタイムを個別に設定可能ですので、実行周期を遅・速 2 分割することができます。）

(8) CPU 本体にプログラムに関するすべての情報をメモリしていますので、開発時に使用したパーソナルコンピュータが万一破損した場合でも別のパーソナルコンピュータで保守が可能です。
（プログラム上のコメントも再現されますので、プログラム・コメント・実行データのセットで保守可能）

(9) 便利な機能を豊富に盛り込んだプログラミングツール（TDsxEditor）を使用することによってシステム変更時の変更作業が極めて短時間に、誤りが少なく、確実に行うことができます
（RUN 中ローダ、モニター、デバッガ、トレンド、トレースバック機能等の詳細については TDsxEditor オペレーションマニュアルを参照してください。）

第２章　μ－GPC 言語での プログラミングの方法

第２章　μ－GPC 言語でのプログラミング方法

μGPCsx では 1 台の CPU にロードされるプログラムをプロジェクトと言う概念で構築します。
プロジェクトには名称が付けられ、自由に変更することが可能です。（最適な名称を決めます。）
1 つのプロジェクトはシステム定義、タスク 1、タスク 2、サブルーチンの 4 つの部分に分けられます。

(1) システム定義

CPU のハード的な条件を定義するためのもので、システム構成定義（I/O 割付）、システム動作定義、CPU 動作定義、冗長化定義の 4 つの構成になっています。

(2) タスク 1、タスク 2

優先度の高いタスクをタスク 1 とし、スキャンタイム、メモリ転送定義、トレースバック設定のほかに複数のサブプログラムから構成されます。各サブプログラムにはプログラムの名称が付けられ（指定ない場合は NoName）、プロジェクト内での適当な担当処理名等に変更可能です。

1 つのサブプログラムは横 12 カラム、縦 19 ラインで構成されるプログラミングシートにプログラミングします。1 枚のプログラミングシートを 1 ページとし、順次ページを追加していくことが可能です。

サブプログラム内ではローカルシンボルが使用可能となりますが、サブプログラム間の受け渡しはグローバルメモリのみ有効です。

(3) サブルーチン

タスク 1、タスク 2 のなかのサブプログラムと同様に共通で使用されるサブルーチンです。
サブルーチンの名称（6 桁の英数字）を決め、追加します。

(4) プログラミングシート

横 12 カラムのうち、それぞれのカラムはシンボル挿入部分とクロスポイント部分から構成されます。これらの部分にシンボルを置き、ラベル名称を入力することでプログラミングが完了します。

（END 命令や、コンパイル操作はなく、エディタ終了時点で自動的にコンパイルされます。）

1～11 カラムはラダーシンボルでの接点およびデータフローシンボルが置けます。

12 カラムはラダーシンボルでのコイル専用でコイル以外は置けません。

また 11 カラムにはクロスポイントがありませんので加算命令やラダ－シンボルの交点を挿入することはできません。

通常クロスポイントは 2 項演算子（加算、減算、乗算、等）を置きますが、C 接点のみは接点名称を入力するため、シンボル挿入部分に置きます。

縦 19 ラインのうち、それぞれの行（ライン）はラベル名称部分、シンボル挿入部分とデータ・コメント部分に分かれます。

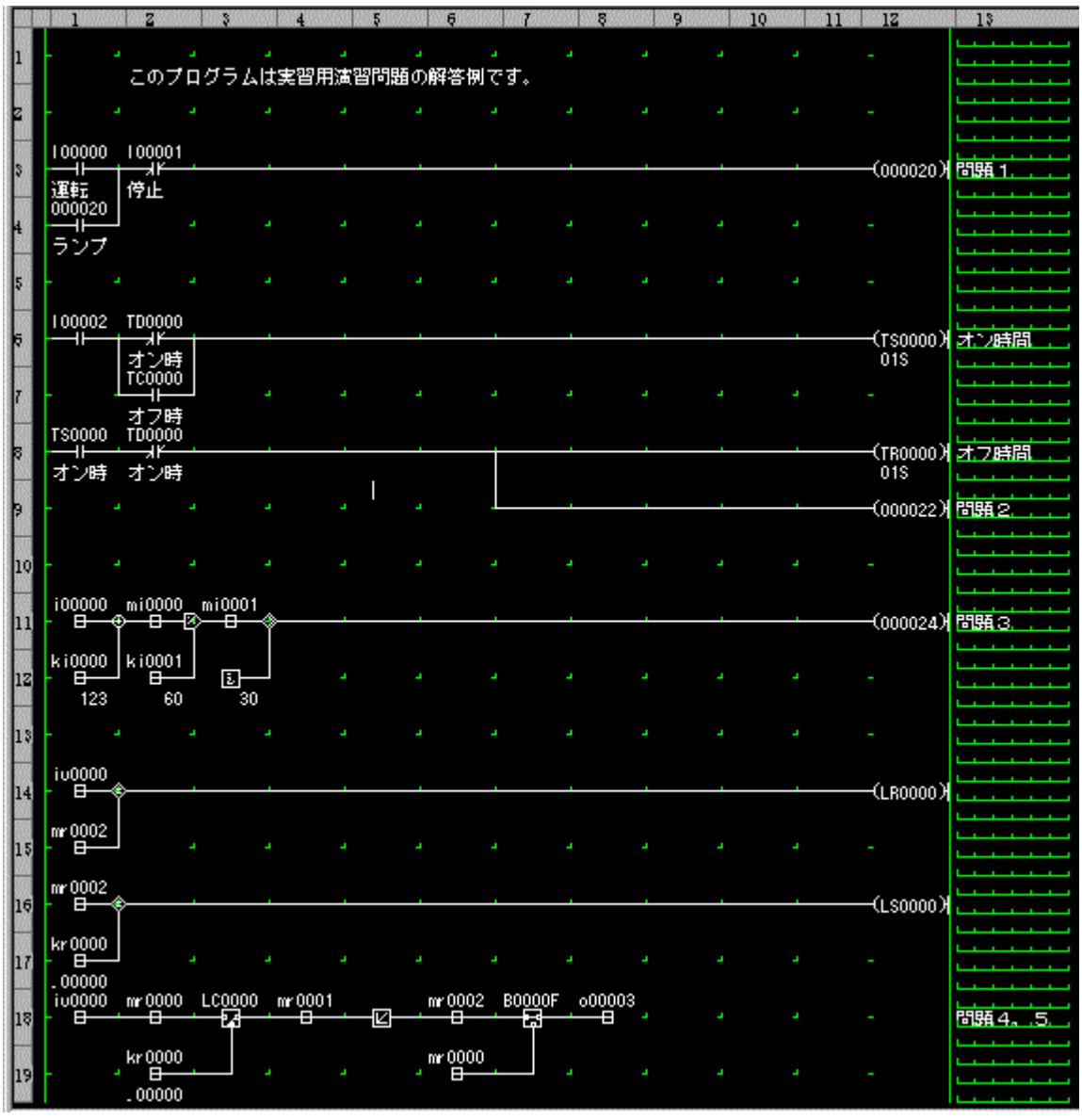
クロスポイントを使用したプログラムでは複数行に渡ってプログラムしますが、19 ラインを超えるプログラムはテンポラリラベルを使用してページ分割します。

(5) プログラムコメント

プログラミングシートは下図プログラミング例のように 13 カラム目をコメント用に使用することができ、ラダーシンボルでコイルを置いた場合には該当接点部分のコメント位置に反映されます。（接点側で入力した場合以外、自動的に表示されます。）

ただし、全角で 3 桁（半角 6 桁）までですので、極力判別できる文字列を検討してください。

また、1 行目のようにシンボルがない部分のコメント位置はすべてコメントとして使用できます。



(6) サンプルプログラムの説明

参考までに前記実習用演習問題のプログラム例を説明します。

1 ライン目はコメント行です。この例のようにプログラムの内容等を予め記述します。

2 ライン目は空白行です。プログラムリストを見やすくするために必要に応じて入れておきます。

3 ライン目～4 ライン目は典型的な 2 操作スイッチを使用したホールド回路のラダーシンボルです。

I00000 である入力スイッチをオンすることによって 000020 であるランプ回路を点灯させ、ホールドします。

I00001 は上記ホールドを解く B 接点入力スイッチです。オンすると上記ランプは消灯します。

5 ライン目は空白行です。

6 ライン目～9 ライン目はオンディレイタイマとオフディレイタイマを組み合わせたランプの点滅回路です。オン時間とオフ時間をそれぞれ独立に変更することができます。

各タイマの設定時間は 12 カラム目のコイルの下側に時間設定します。前記の例では 1.0Ｓ（秒）となっていますが、2 時間までの設定が可能で、H で時を、M で分を、S で秒を表します。最小単位は 10mS ですが、0.01Ｓと記述します。

10 ライン目は空白行です。

11 ライン目～12 ライン目は 16 ビットの入力モジュールから数値データを読み取り定数 123 を加算しさらに 60 で除算した余りを求め、その値が 30 を超えた場合ランプを点灯させる回路です。

途中の演算結果はレジスタにセーブしてありますので、デバッグ時はこれを確認しながら結果をモニターすることができます。比較命令シンボルの右側は論理演算シンボルとなります。

13 ライン目は空白行です。

14 ライン目～19 ライン目はラッチリレーと変化率制限関数（弊社では ARC と称しています。）を使用したパターン発生回路の例です。連続的に三角波を発生します。波高値は入力モジュールから数値を BCD 型で設定することができます。周期は ARC 関数の変化率パラメータを変更することで間接的に変更することが可能です。18 ライン目と 19 ライン目で実数演算と整数演算、BCD 演算が混在していますが、ARC の入力値を C 接点で切り替えることによってパターンを連続的に発生させています。

B0000F の C 接点はテスト用で、デバッガでオンすることによって入力値を直接出力します。

第３章　取り扱えるデータの型と範囲

第３章　取り扱えるデータの型と範囲

μGPCsx で取り扱うデータは種別 2 桁＋16 進番号 4 桁のラベル名称で表します。
また 16 進番号の先頭 1 桁はインデックスラベル X、Y、Z に置き換えできます。
　ラベル例 ： I0X123　　b0y234　　mr02AF

３－１　データの種類

μGPCsx で取り扱うデータは大別すると、「論理データ」と「数値データ」の 2 種類があります。

３－１－１　論理データ

・論理データは、1 ビットの論理すなわち「1」または「0」を表すデータです。
・論理データは、論理演算などにより処理されます。
・論理データは、「リレー」に蓄えられており、プログラムの中では、「リレー番号」を指定することにより参照されます。
・比較演算シンボルの演算結果は論理データとなります。

要点　・μGPCsx では論理データを蓄えておくところを「リレー」といいます。
　　・論理データの「1」はリレーの「ON」の状態に相当し、論理データの「0」はリレーの「OFF」の状態に相当します。

３－１－２　数値データ

・数値データは、16 ビット（1 ワード）または 32 ビット（2 ワード）を 1 単位として表すデータです。
・数値データは、「レジスタ」に蓄えられており、プログラムの中では、「レジスタ番号」を指定することにより参照されます。
・比較演算シンボルの入力条件は数値データとなります。

要点　・μGPCsx では数値データを蓄えておくところを「レジスタ」といいます。

論理データのリレー番号の頭文字は大文字を使います。
（例）
　　I00000

数値データのレジスタ番号の頭文字は小文字を使います。
（例）
　　i00000

３－２　データの型の種類

３－２－１　論理データの型

特に型の区別はありません。
取り扱えるデータは１（ON）または０（OFF）です。

３－２－２　数値データの型

下記の５種類があり、３－３以降に説明します。

① 16 ビット整数型（i 形式）
② 16 ビット BCD 型（u 形式）
③ 32 ビット整数型（w 形式）
④ 32 ビット BCD 型（v 形式）
⑤ 32 ビット実数型（r 形式）

３－３　16 ビット整数型（i 形式）

16 ビットの符号付き整数値データを１単位（1 ワード）として表します。
内部的に取り扱われるデータの範囲は

－32,768　～　32,767（8000H ～ 7FFFH）

このような数値データを「16 ビット整数データ」といいます。

３－４　16 ビット BCD 型（u 形式）

16 ビットの BCD（2 進化 10 進コード）４桁のデータを１単位（1 ワード）として表します。
内部的に取り扱われるデータの範囲は

0000 ～ 9999（0000H ～ 270FH）

このような数値データを「16 ビット BCD データ」といいます。

注意　16 ビット BCD データは、入出力ユニット（I／O）との間でやりとりするデータ（入出力データ）についてのみ使用可能です。

３－５　32 ビット整数型（W 形式）

32 ビットの符号付き整数値データを 1 単位（2 ワード占有）として表します。
内部的に取り扱われるデータの範囲は、

－2147483648 ～ 2147483647（80000000H ～ 7FFFFFFFH）

このような数値データを「32 ビット整数データ」といいます。

注意　32 ビット整数データは、入出力ユニット（I／O）との間でやりとりするデータ（入出力データ）についてのみ使用可能です。

３－６　32 ビット BCD 型（V 形式）

32 ビットの BCD（2 進化 10 進コード）8 桁のデータを 1 単位（2 ワード占有）として表します。
内部的に取り扱われるデータの範囲は

00000000 ～ 99999999（00000000H ～ 05F5E0FFH）

このような数値データを「32 ビット BCD データ」といいます。

注意　32 ビット BCD データは、入出力ユニット（I／O）との間でやりとりするデータ（入出力データ）についてのみ使用可能です。

３－７　32 ビット実数型（r 形式）

32 ビットの浮動小数点フォーマットのデータを 1 単位（2 ワード占有）として表します。
内部的に取り扱われるデータの範囲は

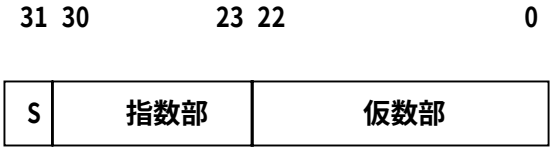
$$-6.2573187\times10^{38} \quad \sim \quad 6.2573187\times10^{38}$$

このような数値データを「32 ビット実数データ」といいます。

参考　32 ビット実数データは、内部的に次のように取り扱われます。
（ユーザは気にする必要はありません。）

$$(-1)^{S} \times 2^{e-127} \times 1.f$$

s：符号部の値
e：指数部の値
f：仮数部の値（23 ビット 2 進数で正規化）

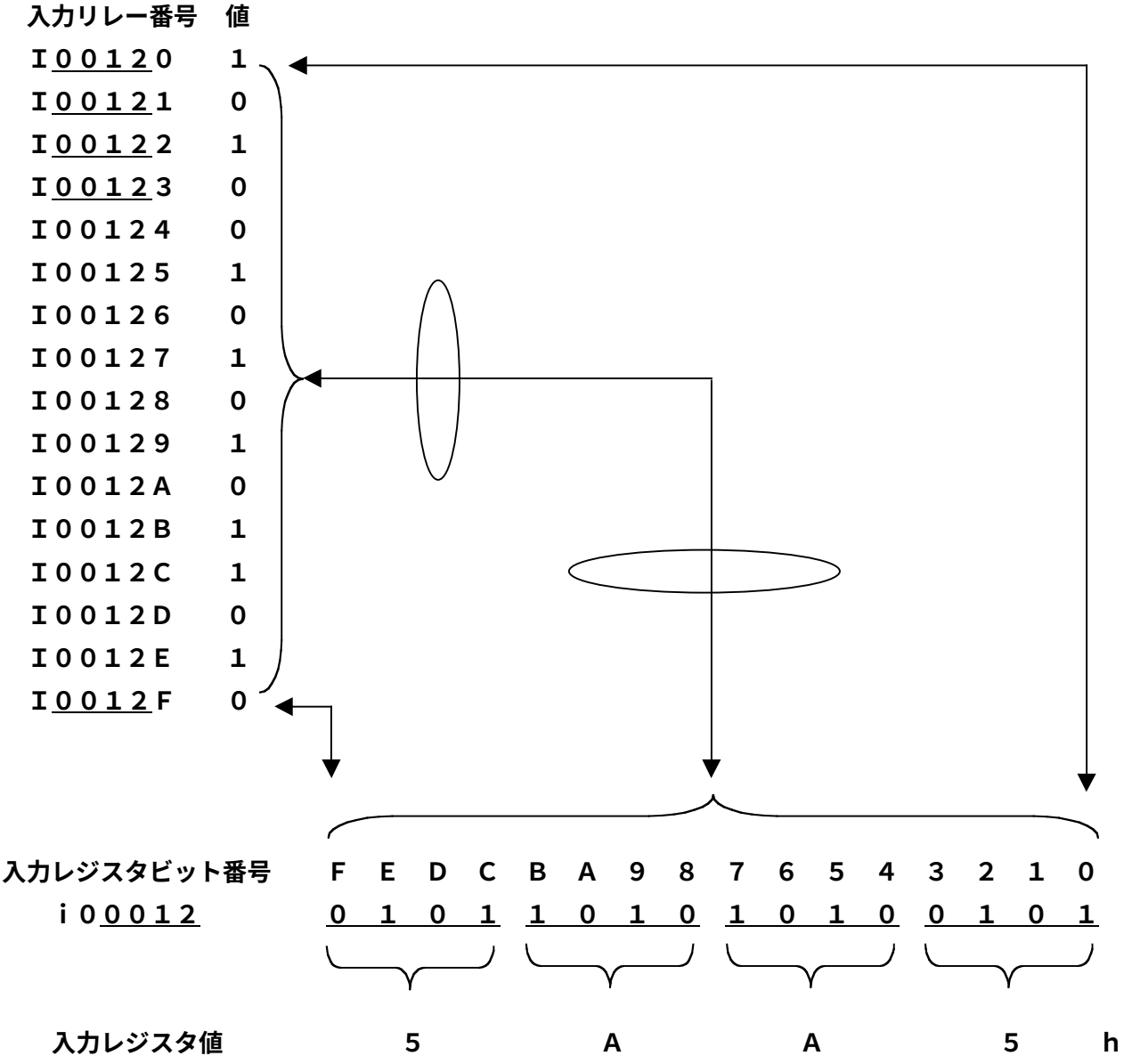


３－７　論理データと 16 ビット整数データ（i 形式）との関係

μGPCsx で取り扱う「論理データ」は、16 ビットずつまとめて 1 つの「16 ビット整数データ（i 形式）」として対応づけることができます。

この場合の論理データと 16 ビット整数データ、およびこれらのデータを格納するリレーとレジスタ、リレー番号とレジスタ番号には次のような関係があります。

（例）連続するリレー番号 I00120、I00121、～I0012F は、16 個の論理データを格納した入力リレーに対応します。一方レジスタ番号 i00012 は 1 個の 16 ビット整数データを格納する入力レジスタに対応します。両者の関係を図で示すと図 3.1 のようになります。
この図は入力レジスタ i00012 の内容 5AA5（16 進）が入力リレーI00120、I00121、～I0012F に展開される様子を表したものです。



同様に 16 ビットずつまとめられる入力リレーと入力レジスタとの対応関係は次のようになります。

入力リレー番号			入力レジスタ番号
I00000，I00001，	～，	I0000F	i00000
I00010，I00011，	～，	I0001F	i00001
I00020，I00021，	～，	I0002F	i00002

この他、出力リレー、リンクリレー、補助リレー等それぞれリレーの種類別に、同様に出力レジスタ、リンクレジスタ、補助レジスタ等を対応づけることができます。

要点　リレー番号とレジスタ番号の対応関係

（例）

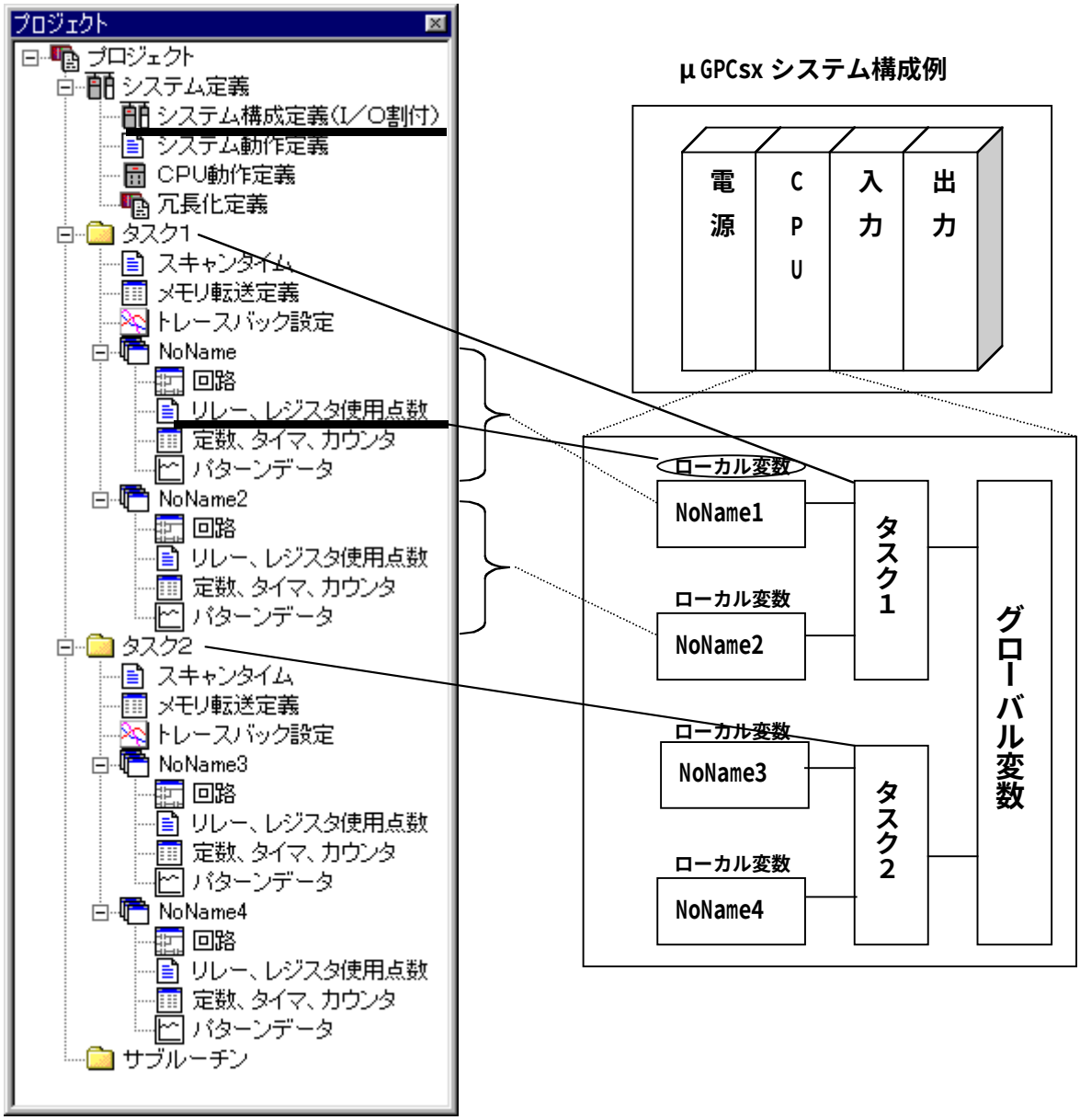
リレー番号 I00123 はレジスタ番号 i00012 のビット番号 3 を表します。

注意　リレー番号とレジスタ番号の取り得る範囲はリレーやレジスタの種類によって異なります。
リレーに展開しても意味がなく、展開できないレジスタ（kr、mr、mi 等）もあります。

第４章　リレーとレジスタの種類

第４章　リレーとレジスタの種類

４－１　ローカル変数、グローバル変数とサブプログラムの関係



・ローカル変数 ----- １つのサブプログラム内でのみ参照可能な変数(他のサブプログラムからは参照できません)。
各サブプログラムの“リレー、レジスタ使用点数”で使用する点数を設定します。
処理機能により分割して作成します。
(例)　mi、BO など

・グローバル変数 --- １つのプロジェクト内のどのサブプログラムからも参照可能な変数。
“システム構成定義”の CPU のパラメータで使用する点数を設定します。
(例)　GO、fi、RI など

４－２　リレー・レジスタ使用可能点数

①グローバル変数

プロジェクト内のどの POU(プログラム)でも使用可能な変数の最大使用できる点数を下表にあらわします。

名称	点数 (最大)	種別	データ番号	データ方向	備考
入力リレー	8,192	接点	I00000 ～ I01FFF	ロード	*1 *3
入力レジスタ	512	入力データ	i□0000 ～ i□01FF		
出力リレー	(8,192)	コイル、接点	O00000 ～ O01FFF	ストア	*1 *3
出力レジスタ	(512)	出力データ	o□0000 ～ o□01FF		
アナウンスリレー アナウンスレジスタ	32,768	システム情報	Z00000 ～ Z07FFF	ロード	
	2,048		z00000 ～ z007FF		
グローバルリレー グローバルレジスタ	131,072	コイル、接点	G00000 ～ G1FFFF	ロード ストア	
	8,192	グローバル データ	g00000 ～ g01FFF		
	4,096		gr0000 ～ gr1FFE		*2
リテインリレー リテインレジスタ	65,536	コイル、接点	RI0000 ～ RIFFFF	ロード ストア	
	4,096	リテイン データ	ri0000 ～ ri0FFF		
	2,048		rr0000 ～ rr0FFF		*2
ネットワークリレー ネットワークレジスタ	65,536	コイル、接点	FI0000 ～ FIFFFF	ロード ストア	
	4,096	ネットワーク データ	fi0000 ～ fi0FFF		
	2,048		fr0000 ～ fr0FFE		*2

***1：入力と出力の合計点数となります。**

***2：奇数番号は使用できません。**

***3：□内は入出力レジスタの型式を表す u(BCD4 桁)、v(BCD8 桁)、w(32 ビット整数）があります。**

②ローカル変数

各サブプログラム単位に最大使用できる点数を下表にあらわします。

名称	点数(最大)	種別	データ番号	データ方向	備考
補助リレー	512	コイル、接点	B00000 ～ B001FF	ロード ストア	
補助レジスタ	32	補助データ	b00000 ～ b0001F		
ラッチリレー ラッチレジスタ	512	セットコイル	LS0000 ～ LS01FF	ロード ストア	
			ls0000 ～ ls001F		
		リセットコイル	LR0000 ～ LR01FF	ロード ストア	
			lr0000 ～ lr001F		
	32	ラッチ接点	LC0000 ～ LC01FF	ロード	
			lc0000 ～ lc001F		
オン微分リレー オン微分レジスタ	512	コイル	US0000 ～ US01FF	ロード ストア	
			us0000 ～ us001F		
	32	微分接点	UC0000 ～ UC01FF	ロード	
			uc0000 ～ uc001F		
オフ微分リレー オフ微分レジスタ	512	コイル	DS0000 ～ DS01FF	ロード ストア	
			ds0000 ～ ds001F		
	32	微分接点	DC0000 ～ DC01FF	ロード	
			ds0000 ～ ds001F		
オンタイマ オンタイマ レジスタ	224	コイル、 瞬時接点	TS0000 ～ TS00DF	ロード ストア	
			ts0000 ～ ts0009		
	14	限時接点	TD0000 ～ TD00DF	ロード	
			td0000 ～ td0009		
	224	経過時間	tn0000 ～ tn00DF	ロード	
オフタイマ オフタイマ レジスタ	224	コイル、 瞬時接点	TR0000 ～ TR00DF	ロード ストア	
			tr0000 ～ tr0009		
	14	限時接点	TC0000 ～ TC00DF	ロード	
			tc0000 ～ tc0009		
	224	経過時間	tf0000 ～ tf00DF	ロード	

名称	点数（最大）	種別	データ番号	データ方向	備考
カウンタ	192	リセットコイル	NR0000 ～ NR00BF	ロードストア	
			nr0000 ～ nr000B		
		プリセットコイル	NP0000 ～ NP00BF	ロードストア	
			np0000 ～ np000B		
		UP コイル	NU0000 ～ NU00BF	ロードストア	
			nu0000 ～ nu000B		
		DOWN コイル	ND0000 ～ ND00BF	ロードストア	
			nd0000 ～ nd000B		
カウンタレジスタ	12	ゼロ検出接点	NZ0000 ～ NZ00BF	ロード	
			nz0000 ～ nz000B		
	192	カウント現在値		ロード	
			N00000 ～ n000BF		
演算データ	512	整数	mi0000 ～ mi01FF	ロードストア	
	256	実数	mr0000 ～ mr00FF		
定数データ	512	整数	ki0000 ～ ki01FF	ロード	
	256	実数	kr0000 ～ kr00FF		
パターンデータ	10	整数	pi0000 ～ pi0009	ロード	*1
	10	実数	pr0000 ～ pr0009		*1
スタックレジスタ	256	整数	si0000 ～ si00FF	ロードストア	
	128	実数	sr0000 ～ sr007F		*2
インデックスレジスタ	3	整数	indx_x、indx_y、indx_z	ロードストア	

*1：パターンデータのポイント数の設定によって使用可能なパターン数も変わります。

*2：奇数番号は使用できません。

③レジスタの共有体構造

グローバルレジスタやスタックレジスタは取り扱いを良くするために共有体関係になっています。
下表にグローバルメモリのリレー、整数レジスタ、実数レジスタの共有体関係を表します。
特に sr0000 は活線データを、sr0002 は引数の 1 番目をあらわします。

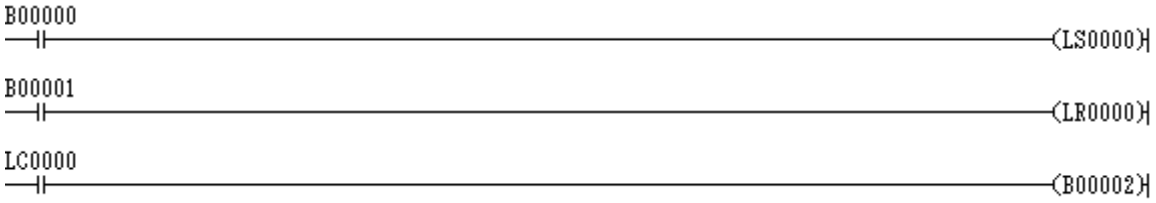
<table>
<tr><th>リレー名称</th><th>整数レジスタ</th><th>実数レジスタ</th></tr>
<tr><td>G00000</td><td rowspan="3">g00000</td><td rowspan="6">gr0000</td></tr>
<tr><td>G00001</td></tr>
<tr><td>G00002
～
G0000F</td></tr>
<tr><td>G00010</td><td rowspan="3">g00001</td></tr>
<tr><td>G00011</td></tr>
<tr><td>G00012
～
G0001F</td></tr>
<tr><td>G00020
～
G0002F</td><td>g00002</td><td rowspan="2">gr0002</td></tr>
<tr><td>g00030
～
G0003F</td><td>g00003</td></tr>
</table>

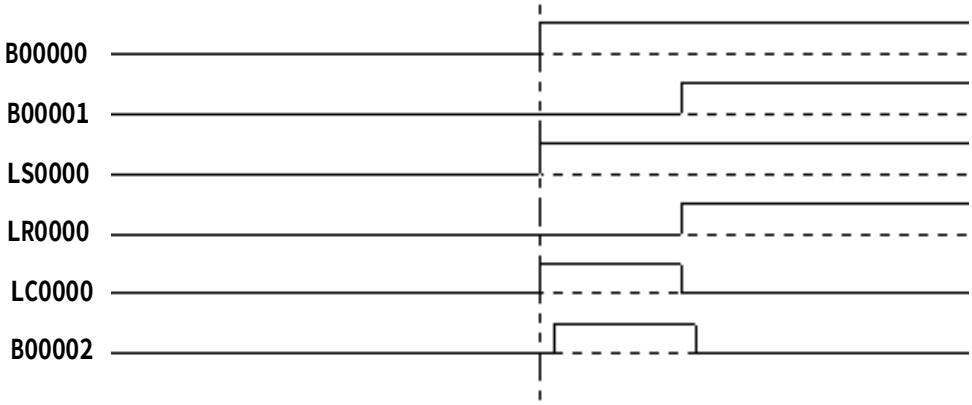
<table>
<tr><th>リレー名称</th><th>整数レジスタ</th><th>実数レジスタ</th></tr>
<tr><td>SI0000</td><td rowspan="2">Si0000</td><td rowspan="2">sr0000</td></tr>
<tr><td></td></tr>
<tr><td>SI0020</td><td rowspan="2">Si0002</td><td rowspan="2">sr0002</td></tr>
<tr><td></td></tr>
</table>

注意：共有体関係はいずれのレジスタからも操作できますので使用の場合は特に注意してください。

４－３　特殊リレーの概要

①ラッチリレー／レジスタ





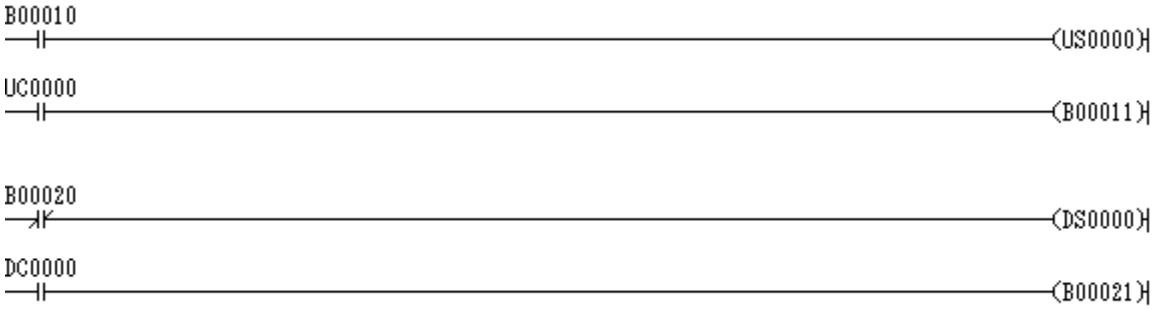
セットコル LS0000 が ON すると、ラッチ接点 LC0000 が ON し、000020 は ON し続けます。
リセットコイル LR0000 が ON すると、ラッチ接点 LC0000 が OFF し、000020 は OFF し続けます。
ラッチ接点 LC0000 はラッチコイルより１スキャン分遅れます。
ラッチコイルは通常電源を開放すると、OFF します。
ラッチコイルを電源開放時にも保持したい場合は、リテインメモリを使用しメモリ転送定義で転送するか、SET　RESET 関数を使用して（パラメータにリテインリレーを設定）ください。

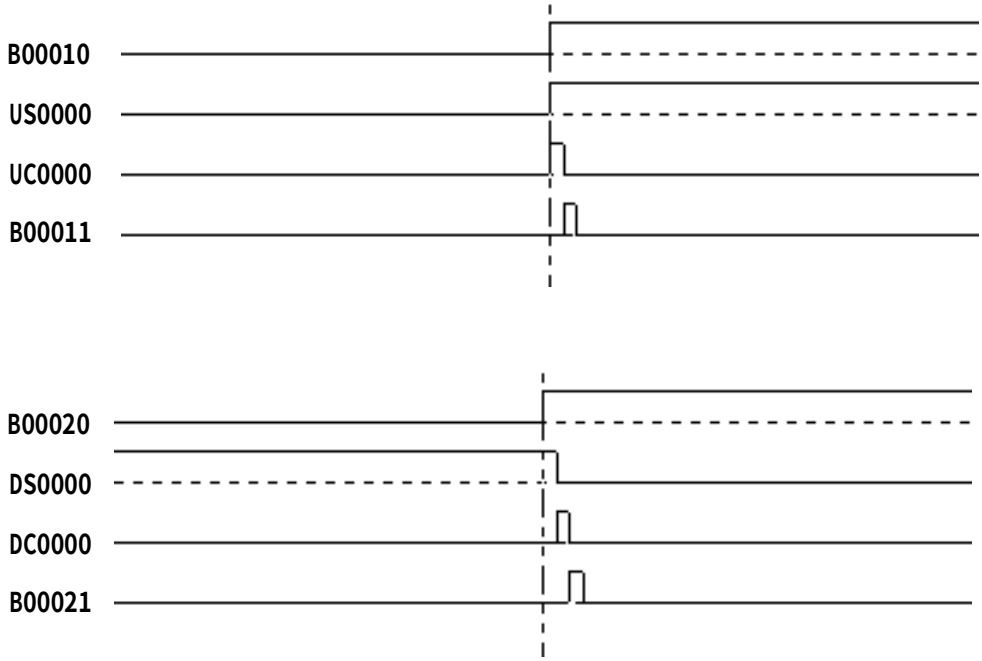
（メモリ転送定義：演算前）
RI0000　－＞　LC0000

（メモリ転送定義：演算後）
LC0000　－＞　RI0000

サブルーチン内で同様の機能を実現するにはサブルーチン内で SI0000 を使用して SET　RESET 関数を使います。

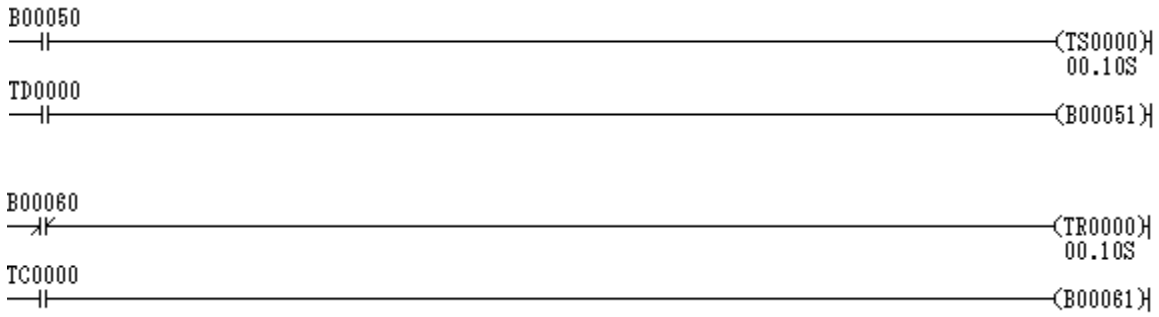
②オン／オフ微分リレー／レジスタ

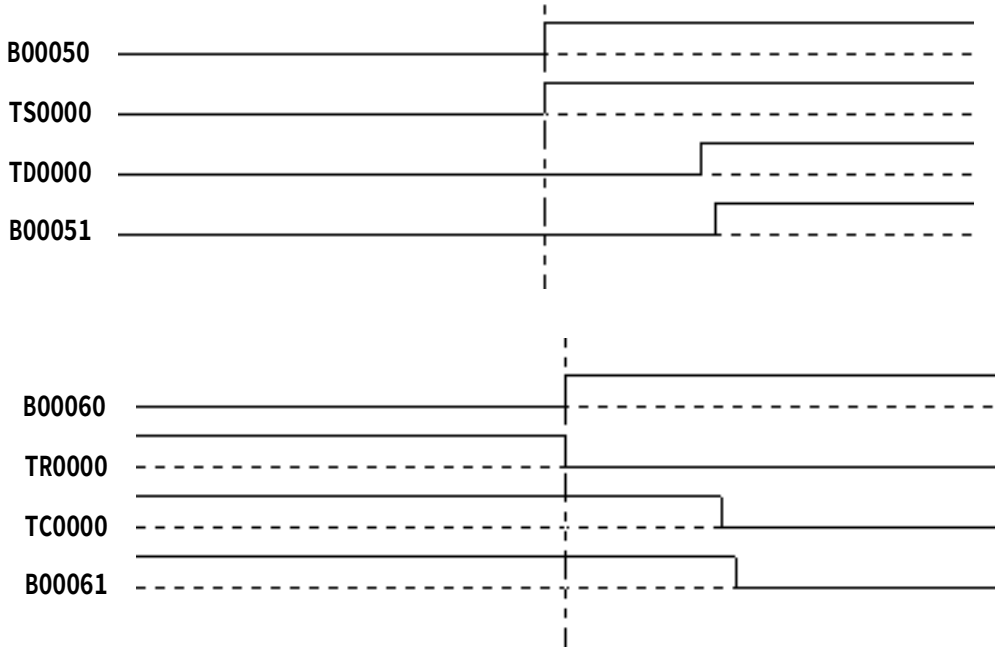




コイル US0000 が ON すると、1 スキャン分遅れて微分接点 UC0000 が 1 スキャン分 ON になります。
コイル DS0000 が OFF すると 1 スキャン分遅れて微分接点 DC0000 が１スキャン分 ON になります。
この他に同様の機能を実現するために USUC 関数と DSDC 関数があります。

③オン／オフタイマリレー／レジスタ





コイル TS0000 が ON すると、設定時間経過後に限時接点 TD0000 が ON になります。TD0000 は TS0000 が OFF になって 1 スキャン以内に OFF になります。

（タイマ設定値は TS コイルの下側に入力します。）

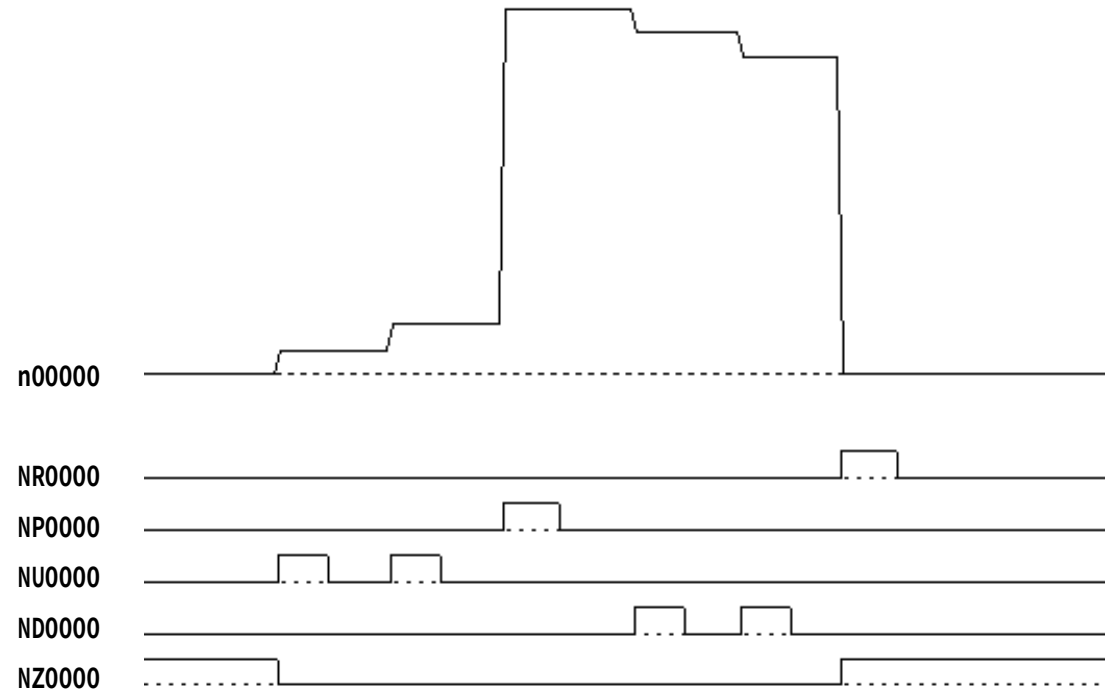
ここで S は秒を、M は分を、H は時間を意味し、0.01 秒から 2 時間までの設定が可能です。

コイル TR0000 が ON すると、限時接点 TD0000 は TR0000 が ON になってから 1 スキャン以内に ON になります。そして設定時間経過後 OFF になります。

（タイマ設定値は TR コイルの下側に入力します。）

ここで S は秒を、M は分を、H は時間を意味し、0.01 秒から 2 時間までの設定が可能です。

④カウンタリレー／レジスタ



カウンタの初期値は 0 です。次にアップコイルが ON となり、カウント値は 1 加えられます。またゼロ検出接点は始め 0 で ON ですが 1 が加えられたので、0 ではなく OFF となります。

さらにアップコイルが ON となり、カウント値は 1 加えられ 2 となります。

プリセットコイルが ON となりカウント値は 15 となります。

プリセット値は NP コイルの下側に設定します。

ダウンコイルが ON となりカウント値から 1 減らします。

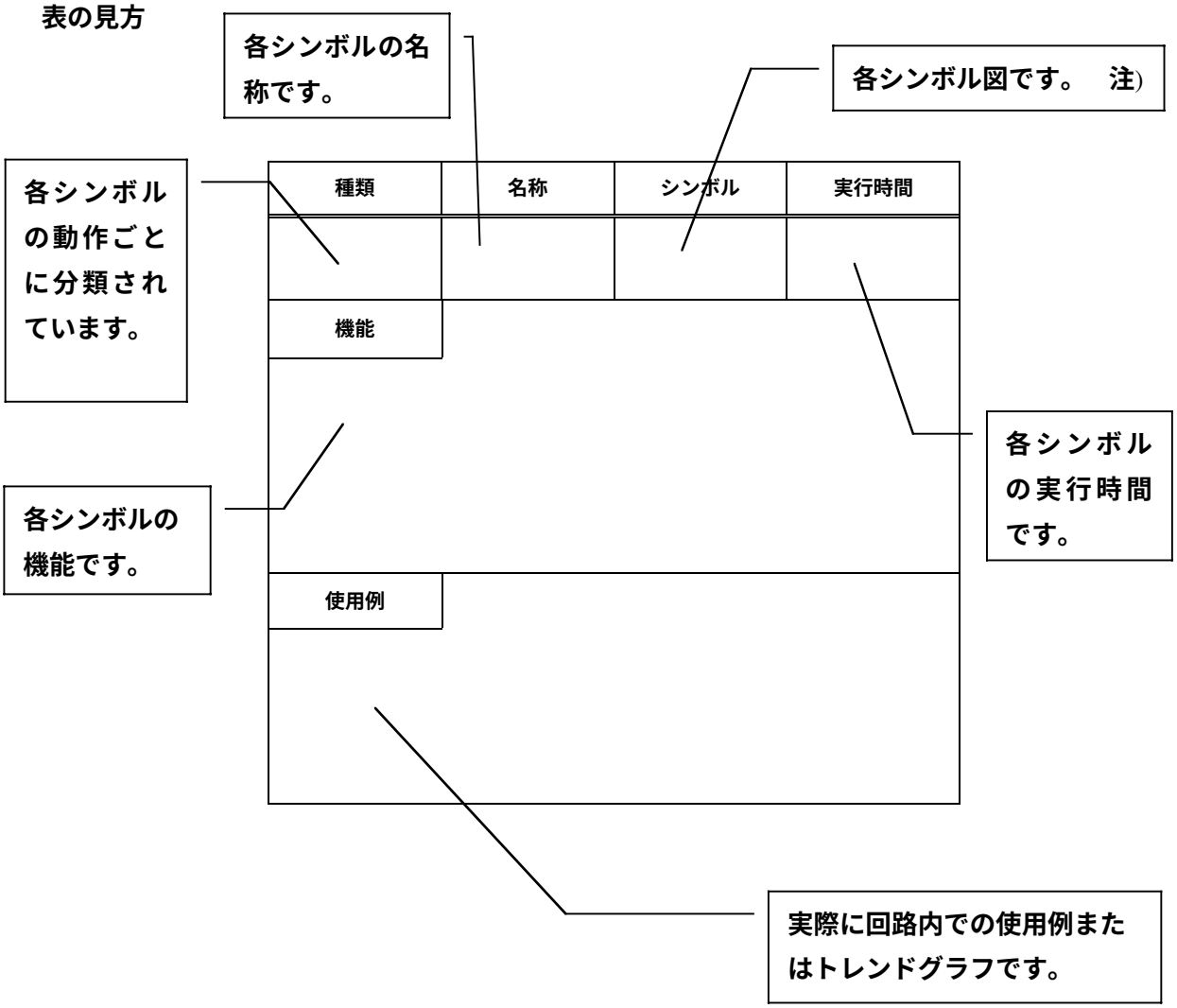
さらにダウンコイルが ON となりカウント値から 1 減らします。

リセットコイルが ON となりカウント値は 0 となり、ゼロ検出接点は ON となります。

第５章　命令語説明

第５章　命令語説明

表の見方

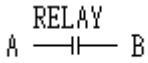


注)　ここではこれ以降のシンボル欄に表示される Relay、Reg について説明します。

左図はリレーを示します。ここでは簡略化するために RELAY という言葉を用いて表します。RELAY には、G0、I0、B0 などのすべてのリレーが設定可能です。

左図はレジスタを示します。ここでは簡略化するために REG という言葉を用いて表します。REG には、g0、mi、kr などのすべてのレジスタが設定可能です。

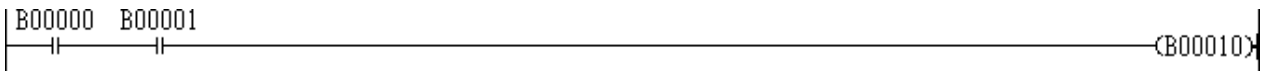
種類	名称	シンボル	実行時間
LD 言語	A 接点		0.02 [μs]
機能	RELAY が ON ならば入力論理値を出力します。 OFF ならば出力論理値を OFF にします。		



RELAY	A	B
ON	ON	ON
ON	OFF	OFF
OFF	X	OFF

X： don t care

使用例



リレーB00000 とリレーB00001 がともに ON の時は、リレーB00010 が ON になります。
これ以外では、リレーB000010 が OFF になります。

種類	名称	シンボル	実行時間
LD 言語	B 接点	RELAY	0.02 [μs]
機能	RELAY が OFF ならば入力論理値を出力します。 ON ならば出力論理値を OFF にします。		

RELAY
A — B

RELAY	A	B
OFF	ON	ON
OFF	OFF	OFF
ON	X	OFF

X ： don t care

使用例

リレーB00000 が ON、リレーB00001 が OFF の時はリレーB00010 が ON になります。
これ以外ではリレーＢ00010 が OFF になります。

種類	名称	シンボル	実行時間
LD 言語	B 接点	RELAY	0.02 [μs]

機能	RELAY が OFF ならば入力論理値を出力します。 ON ならば出力論理値を OFF にします。

RELAY
A — B

RELAY	A	B
OFF	ON	ON
OFF	OFF	OFF
ON	X	OFF

X ： don t care

使用例

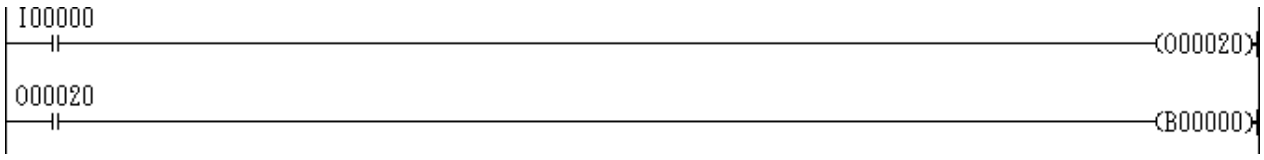
リレーB00000 が ON、リレーB00001 が OFF の時はリレーB00010 が ON になります。
これ以外ではリレーB00010 が OFF になります。

種類	名称	シンボル	実行時間
LD 言語	コイル	─(RELAY)|	0.10 [μs]
機能	入力論理値を RELAY に出力します。		

A ─(RELAY)|

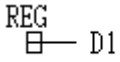
A	RELAY
ON	ON
OFF	OFF

使用例

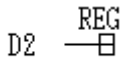


リレーI00000 が ON の時は、リレーO00020=ON とリレーB00000 がともに ON になります。
リレーI00000 が OFF の時は、リレーO00020 とリレーB00000 がともに OFF になります。

種類	名称	シンボル	実行時間
データフロー言語 (基本)	ロード		整数　0.48 [μs] 実数　0.45 [μs]
	ストア		
機能	ロード：REG のデータを出力数値にします。 ストア：入力数値を REG に出力します。		

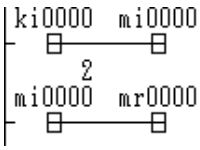


D1 = REG



REG = D2

使用例



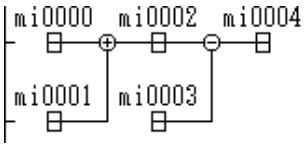
　レジスタ ki0000 のデータ(2)がロードされ、レジスタ mi0000 にストアされます。
　次にレジスタ mi0000 のデータがロードされ、レジスタ mr0000 にストアされます。
　レジスタ mr0000 は実数型のレジスタですので、整数・実数の型変換が行われデータ(2.0)がストアされます。

種類	名称	シンボル	実行時間
データフロー言語 (基本)	ストア＆ロード ストア	REG	整数　0.48 [μs] 実数　0.45 [μs]
機能	入力数値を REG に出力し、REG のデータを出力の数値にします。 演算の途中のデータを REG に保持する時に使用します。		

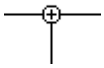
D1 —REG— D2

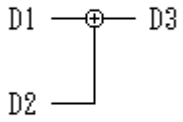
REG = D1
D2 = REG

使用例



レジスタ mi0000 のデータとレジスタ mi00001 のデータが加算され、結果がレジスタ mi0002 にストアされます。
次にレジスタ mi0002 のデータからレジスタ mi0003 のデータが減算され、結果がレジスタ mi0004 にストアされます。
レジスタ mi0002 には演算の途中の加算データが保持されることになります。

種類	名称	シンボル	実行時間
データフロー言語 (基本)	加算		整数　1.15 [μs] 実数　1.13 [μs]
機能	2 つの入力数値を加算し結果を出力します。 型が違っても演算できます。ただし、整数は実数に変換後、実数演算します。		



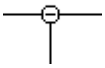
D3 = D1 + D2

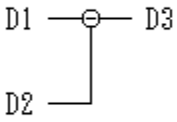
型変換について

1 つの演算ブロックで使用しているレジスタの型式が整数型と 16 ビット BCD 型の場合は 16 ビット整数型に変換して演算しますが、実数型、32 ビット整数型、32 ビット BCD 型のレジスタを使用している場合は実数型に変換して演算します。
(以後、減算、乗算、除算、剰余、上位優先、下位優先についても型変換を行います。)

使用例

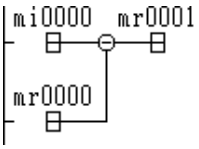
レジスタ mi0000 のデータとレジスタ mr0000 のデータが加算され、結果がレジスタ mr0001 にストアされます。
レジスタ mi0000 のデータは整数ですが、レジスタ mr0000 のデータが実数ですので整数/実数の型変換を行ってから加算されます。

種類	名称	シンボル	実行時間
データフロー言語 (基本)	減算		整数　1.27 [μs] 実数　1.25 [μs]
機能	2 つの入力数値を減算し結果を出力します。 型が違っても演算できます。ただし、整数は実数に変換後、実数演算します。		

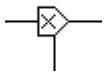


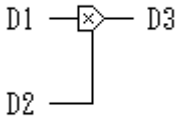
D3 = D1 - D2

使用例



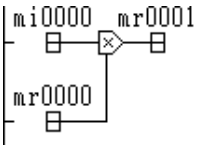
レジスタ mi0000 のデータからレジスタ mr0000 のデータが減算され、結果がレジスタ mr0001 にストアされます。
レジスタ mi0000 のデータは整数ですが、レジスタ mr0000 のデータが実数ですので整数/実数の型変換を行ってから減算されます。

種類	名称	シンボル	実行時間
データフロー言語 (基本)	乗算		整数　1.17 [μs] 実数　1.13 [μs]
機能	2 つの入力数値を乗算し結果を出力します。 型が違っても演算できます。ただし、整数は実数に変換後、実数演算します。		

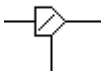


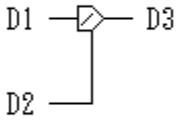
D3 = D1 * D2

使用例



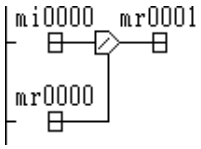
レジスタ mi0000 のデータとレジスタ mr0000 のデータが乗算され、結果がレジスタ mr0001 にストアされます。
レジスタ mi0000 のデータは整数ですが、レジスタ mr0000 のデータが実数ですので整数/実数の型変換を行ってから乗算されます。

種類	名称	シンボル	実行時間
データフロー言語 (基本)	除算		整数　2.48 [μs] 実数　2.32 [μs]
機能	2 つの入力数値を除算し結果を出力します。 型が違っても演算できます。ただし、整数は実数に変換後、実数演算します。		

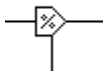


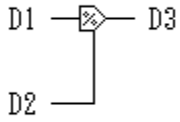
D3 = D1 / D2

使用例



レジスタ mi0000 のデータとレジスタ mr0000 のデータが除算され、結果がレジスタ mr0001 にストアされます。
レジスタ mi0000 のデータは整数ですが、レジスタ mr0000 のデータが実数ですので整数/実数の型変換を行ってから除算されます。

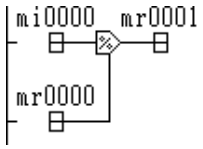
種類	名称	シンボル	実行時間
データフロー言語 (基本)	剰余		2.48 [μs]
機能	2 つの入力数値を除算し結果(剰余)を出力します。		



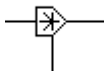
D3 = D1 % D2

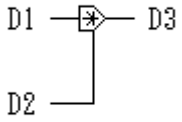
注)　整数演算のみ有効です。

使用例



レジスタ mi0000 のデータがレジスタ mi0001 のデータで除算され、結果(剰余)がレジスタ mi0002 にストアされます。

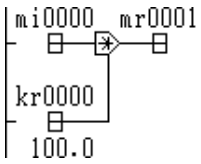
種類	名称	シンボル	実行時間
データフロー言語 (基本)	上位優先		整数　1.52 [μs] 実数　1.45 [μs]
機能	2 つの入力数値を比較し大きい数値を出力します。 型が違っても演算できます。ただし、整数は実数に変換後、実数演算します。		



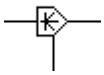
D1 > D2　なら　D3 = D1

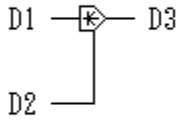
D1 <= D2　なら　D3 = D2

使用例



　レジスタ mi0000 のデータとレジスタ kr0000 のデータ 100.0 が比較され、大きい方のデータがレジスタ mr0001 にストアされます。
　レジスタ mi0000 のデータは整数ですが、レジスタ kr0000 のデータが実数ですので整数/実数の型変換を行ってから比較されます。
　レジスタ kr0000 のデータ(100.0)を下限値とするリミッタになります。

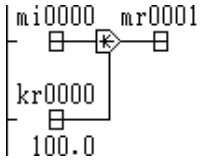
種類	名称	シンボル	実行時間
データフロー言語 (基本)	下位優先		整数　1.43 [μs] 実数　1.64 [μs]
機能	2 つの入力数値を比較し小さい数値を出力します。 型が違っても演算できます。ただし、整数は実数に変換後、実数演算します。		



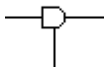
D1 > D2　なら　D3 = D2

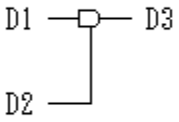
D1 <= D2　なら　D3 = D1

使用例



　レジスタ mi0000 のデータとレジスタ kr0000 のデータ 100.0 が比較され、小さい方のデータがレジスタ mr0001 にストアされます。
　レジスタ mi0000 のデータは整数ですが、レジスタ kr0000 のデータが実数ですので整数/実数の型変換を行ってから比較されます。
　レジスタ kr0000 のデータ(100.0)を上限値とするリミッタになります。

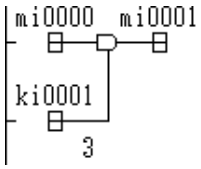
種類	名称	シンボル	実行時間
データフロー言語 (基本)	数値積		1.15 ［μs］
機能	2 つの入力数値の論理積演算を行い、結果を出力します。		



D3 = D1 & D2

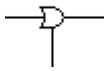
注)　整数の演算のみ有効です。

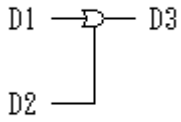
使用例



レジスタ mi0000 のデータとレジスタ ki0001 のデータ(3)の論理積演算が行われ、結果がレジスタ mi0001 にストアされます。
レジスタ mi0000 のデータが(10)ならばレジスタ mi0001 には(2)がストアされます。

mi0000	0000	0000	0000	1010	(10)
ki0000	0000	0000	0000	0011	(3)
mi0001	0000	0000	0000	0010	(2)

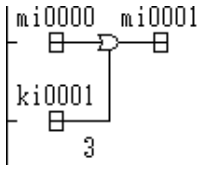
種類	名称	シンボル	実行時間
データフロー言語 (基本)	数値和		1.15 ［μs］
機能	2 つの入力数値の論理和演算を行い、結果を出力します。		



D3 = D1 ¦ D2

注)　整数の演算のみ有効です。

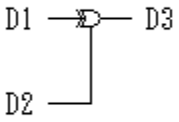
使用例



レジスタ mi0000 のデータとレジスタ ki0001 のデータ(3)の論理和演算が行われ、結果がレジスタ mi0001 にストアされます。
レジスタ mi0000 のデータが(10)ならばレジスタ mi0001 には(11)がストアされます。

mi0000	0000	0000	0000	1010	(10)
ki0000	0000	0000	0000	0011	(3)
mi0001	0000	0000	0000	1011	(11)

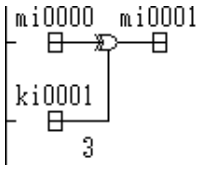
種類	名称	シンボル	実行時間
データフロー言語 (基本)	数値排他和		1.15 ［μs］
機能	2 つの入力数値の排他的論理和演算を行い、結果を出力します。		



D3 = D1 ^ D2

注)　整数の演算のみ有効です。

使用例



レジスタ mi0000 のデータとレジスタ ki0001 のデータ(3)の排他的論理積演算が行われ、結果がレジスタ mi0001 にストアされます。
レジスタ mi0000 のデータが(10)ならばレジスタ mi0001 には(9)がストアされます。

mi0000	0000	0000	0000	1010	(10)
ki0000	0000	0000	0000	0011	(3)
mi0001	0000	0000	0000	1001	(9)

種類	名称	シンボル	実行時間
データフロー言語 (基本)	a 接点		整数　1.52 [μs] 実数　1.33 [μs]
機能	RELAY が ON ならば入力の数値を出力します。 OFF ならば出力の数値を 0 にします。		

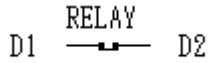
RELAY=ON　なら　D2 = D1

RELAY=OFF　なら　D2 = 0

使用例

リレーI00000 が ON の時は、レジスタ mi0000 のデータがレジスタ mi0001 にストアされます。
リレーI00000 が OFF の時は、レジスタ mi0001 に(0)がストアされます。

種類	名称	シンボル	実行時間
データフロー言語 (基本)	b 接点	RELAY	整数　1.52 [μs] 実数　1.33 [μs]
機能	RELAY が OFF ならば入力の数値を出力します。 ON ならば出力の数値を 0 にします。		

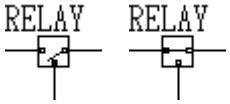


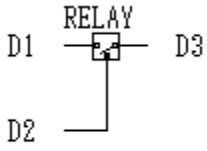
RELAY=ON　　なら　D2 = 0

RELAY=OFF　　なら　D2 = D1

使用例

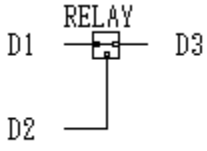
リレーI00000 が OFF の時は、レジスタ mi0000 のデータがレジスタ mi0001 にストアされます。
リレーI00000 が ON の時は、レジスタ mi0001 に(0)がストアされます。

種類	名称	シンボル	実行時間
データフロー言語 (基本)	c 接点		整数　1.31 [μs] 実数　1.15 [μs]
機能	RELAY の論理値により 2 つの入力数値の一方を選択し出力します。		



RELAY=ON　なら　D3 = D1

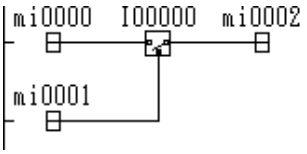
RELAY=OFF　なら　D3 = D2



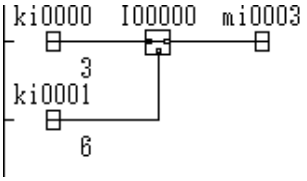
RELAY=ON　なら　D3 = D2

RELAY=OFF　なら　D3 = D1

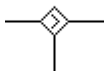
使用例

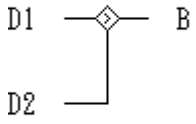


リレーI00000 が OFF の時はレジスタ mi0001 のデータが選択されてレジスタ mi0002 にストアされます。
リレーI00000 が ON の時はレジスタ mi0000 のデータが選択されてレジスタ mi0002 にストアされます。



リレーI00000 が OFF の時はレジスタ ki0000 のデータ(3)が選択されてレジスタ mi0003 にストアされます。
リレーI00000 が ON の時はレジスタ ki0001 のデータ(6)が選択されてレジスタ mi0003 にストアされます。

種類	名称	シンボル	実行時間
データフロー言語 (基本)	コンペア・ハイ		整数　1.17 [μs] 実数　1.21 [μs]
機能	2 つの入力数値の比較を行い、判定結果を論理値で出力します。		



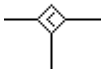
D1 > D2　なら　B = ON

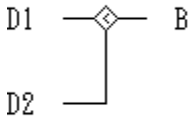
D1 <= D2　なら　B = OFF

使用例

レジスタ mi0000 のデータがレジスタ mi0001 のデータよりも大きければリレー000020 が ON になります。
これ以外ではリレー00020 が OFF になります。

論理反転と組み合わせて、論理を変更することができます。
レジスタ mi0002 のデータがレジスタ mi0003 のデータと同じかレジスタ mi0003 のデータよりも小さければリレー000021 が ON になります。
これ以外ではリレー00020 が OFF になります。

種類	名称	シンボル	実行時間
データフロー言語 (基本)	コンペア・ロウ		整数　1.17 [μs] 実数　1.21 [μs]
機能	2 つの入力数値の比較を行い、判定結果を論理値で出力します。		



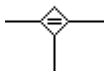
D1 < D2　なら　B = ON

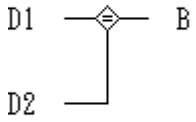
D1 >= D2　なら　B = OFF

使用例

レジスタ mi0000 のデータがレジスタ mi0001 のデータよりも小さければリレー000020 が ON になります。
これ以外ではリレー00020 が OFF になります。

論理反転と組み合わせて、論理を変更することができます。
レジスタ mi0002 のデータがレジスタ mi0003 のデータと同じかレジスタ mi0003 のデータよりも大きければリレー000021 が ON になります。
これ以外ではリレー00020 が OFF になります。

種類	名称	シンボル	実行時間
データフロー言語 (基本)	コンペア・イコール		整数　1.17 [μs] 実数　1.21 [μs]
機能	2 つの入力数値を比較し、判定結果を論理値で出力します。		



D1 = D2　なら　B = ON

D1 ≠ D2　なら　B = OFF

注)　使用するレジスタが実数の場合、表れない細かな数値のため ON しない場合があります。

使用例

レジスタ mi0000 のデータがレジスタ mi0001 のデータと同じならばリレー000020 が ON になります。
これ以外ではリレー00020 が OFF になります。

論理反転と組み合わせて、論理を変更することができます。
レジスタ mi0002 のデータがレジスタ mi0003 のデータと等しくないならばリレー000021 が ON になります。
これ以外ではリレー00021 が OFF になります。

種類	名称	シンボル	実行時間
データフロー言語 (基本)	ロード　局所定数 （整数、実数）		整数　0.91　[μs] 実数　0.85　[μs]
機能	局所的な定数（整数、実数）をロードします。		

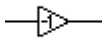
定数はプログラム内（パラメータでなく）に確保されます。

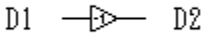
ロード局所定数（整数）は i 形式のみの演算ブロック中でのみ使用できます。

1 つの演算ブロック内で（整数）と（実数）の混在はできません。

使用例

レジスタ mi0000 には整数値(10)がロードされます。
レジスタ mr0000 には実数値(5.0000)がロードされます。

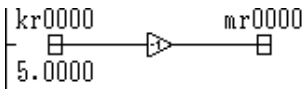
種類	名称	シンボル	実行時間
データフロー言語 (関数 1)	符号変換		整数　0.38 [μs] 実数　0.15 [μs]
機能	入力数値の正負符号の反転を行い出力します。		



D2 = - (D1)

使用例

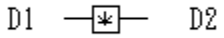
レジスタ ki0000 のデータ(-10)が正数に符号変換されてレジスタ mi0000 に(10)がストアされます。



レジスタ kr0000 のデータ(5.0000)が負数に符号変換されてレジスタ mr0000 に(-5.0000)がストアされます。

種類	名称	シンボル	実行時間
データフロー言語 (関数 1)	絶対値変換		整数　0.40 [μs] 実数　0.15 [μs]

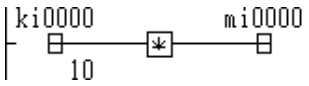
機能	入力数値の絶対値をとり出力します。



D1 < 0　なら　D2 = -(D1)

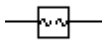
D1 >= 0　なら　D2 = D1

使用例



レジスタ ki0000 のデータ(10)が絶対値変換されてレジスタ mi0000 に(10)がストアされます。

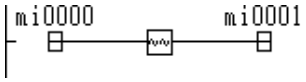
レジスタ kr0000 のデータ(-5.0000)が絶対値変換されてレジスタ mr0000 に(5.0000)がストアされます。

種類	名称	シンボル	実行時間
データフロー言語 (関数 1)	1‘ 補数		0.40 [μs]
機能	入力数値の補数演算を行い、結果を出力します。		

D2 = NOT (D1)

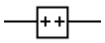
注)　整数演算のみ有効です。

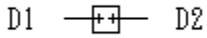
使用例



レジスタ mi0000 のデータに対して 1 の補数演算を行い、結果をレジスタ mi0001 にストアします。
レジスタ mi0000 のデータが(10)ならば、レジスタ mi0002 に(-11)をストアします。

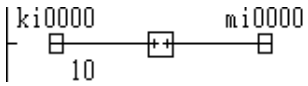
mi0000	0000	0000	0000	1010	(10)
mi0001	1111	1111	1111	0101	(-11)

種類	名称	シンボル	実行時間
データフロー言語 (関数 1)	インクリメント		整数　0.04 [μs] 実数　0.17 [μs]
機能	入力数値に 1 を加算して結果を出力します。		

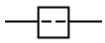


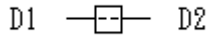
D2 = D1 + 1
(D2 = D1 + +)

使用例



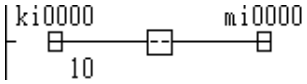
レジスタ ki0000 のデータ(10)に(1)を加算し、演算結果(11)をレジスタ mi0000 にストアします。

種類	名称	シンボル	実行時間
データフロー言語 (関数 1)	デクリメント		整数　0.04 [μs] 実数　0.17 [μs]
機能	入力数値から 1 を減算して結果を出力します。		



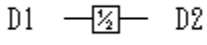
D2 = D1 - 1
(D2 = D1 - -)

使用例



レジスタ ki0000 のデータ(10)から(1)を減算し、演算結果(9)をレジスタ mi0000 にストアします。

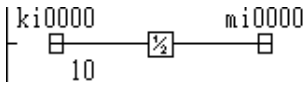
種類	名称	シンボル	実行時間
データフロー言語 (関数 1)	2 分の 1		0.42 [μs]
機能	入力数値の 2 分の 1 倍した結果を出力します。		



D2 = D1 / 2

注)　整数演算のみ有効です。

使用例



レジスタ ki0000 のデータ(10)を 2 分の 1 にし、演算結果(5)をレジスタ mi0000 にストアします。
この命令は整数レジスタのデータを符号付で 1/2 倍する場合に使用します。

種類	名称	シンボル	実行時間
データフロー言語 (関数 1)	2 倍		0.08 [μs]
機能	入力数値の 2 倍した結果を出力します。		

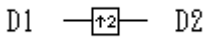
D2 = D1 * 2

注)　整数演算のみ有効です。

使用例

レジスタ ki0000 のデータ(10)を 2 倍にし、演算結果(20)をレジスタ mi0000 にストアします。
この命令は整数レジスタのデータを符号付で 2 倍する場合に使用します。

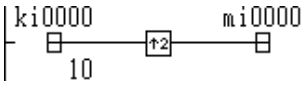
種類	名称	シンボル	実行時間
データフロー言語 (関数 1)	二乗		整数　0.16 [μs] 実数　0.27 [μs]
機能	入力数値の二乗した結果を出力します。		



D2 = D1 * * 2

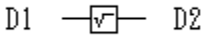
(D2 = $D1^2$)

使用例



レジスタ ki0000 のデータ(10)を 2 乗し、演算結果(100)をレジスタ mi0000 にストアします。

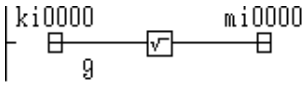
種類	名称	シンボル	実行時間
データフロー言語 (関数 1)	平方根		整数　2.04 [μs] 実数　1.10 [μs]
機能	入力数値の平方根を出力します。		



D2 = SQRT (D1)

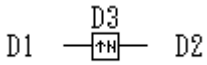
注)　入力値が負の値の場合、出力も負の値となります。

使用例



レジスタ ki0000 のデータ(9)の平方根を演算し、演算結果(3)をレジスタ mi0000 にストアします。

種類	名称	シンボル	実行時間
データフロー言語 (関数 1)	指数関数		3.74 ［μs］
機能	入力数値を指数演算し結果を出力します。		

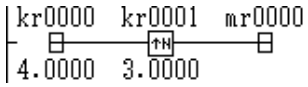


D2 = D3 * * D1

(D2 = $D3^{D1}$)

注)　実数演算のみ有効です。

使用例

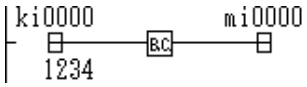


レジスタ kr0000 のデータ(4.0000)に対して、レジスタ kr0001(3.0000)のデータで指数演算を行い、演算結果(64)をレジスタ mr0000 にストアします。

種類	名称	シンボル	実行時間
データフロー言語 (関数 1)	ビットカウント		2.99 ［μs］
機能	入力数値を 16 ビット 2 進数として読み込み、ON しているビットの数を出力します。		

注)　整数演算のみ有効です。

使用例



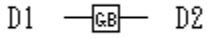
レジスタ ki0000 のデータ(1234)を 16 ビット 2 進数として読み込み、ON しているビット(1 になっている) の数を計算し、演算結果(5)をレジスタ mi0000 にストアします。

ki0000	0000	0001	1010	1010	(1234)
mi0001	0	+ 1	+ 2	+ 2	=

種類	名称	シンボル	実行時間
データフロー言語 (関数 1)	グレイコード バイナリー		15.1 ［μs］

機能	入力数値(グレイコード)を変換し、2 進数で結果を出力します。

グレイコードは、数値が 1 つ変化するのに対して、1 ビットしか変化しないため、位置決め制御などで使います。



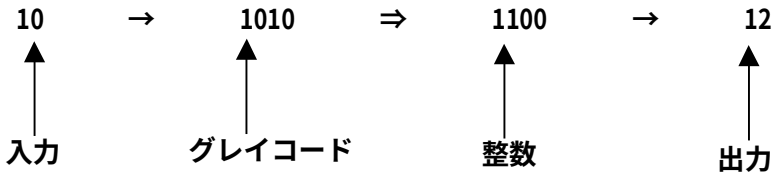
0～15 までのビットパターンは下記のようになります。

D2 整数	D1 グレイ	D2 整数	D1 グレイ	D2 整数	D1 グレイ	D2 整数	D1 グレイ
0000	0000	0100	0110	1000	1100	1100	1010
0001	0001	0101	0111	1001	1101	1101	1011
0010	0011	0110	0101	1010	1111	1110	1001
0011	0010	0111	0100	1011	1110	1111	1000

注)　整数演算のみ有効です。

使用例

レジスタ mi0000 のデータをグレイコード変換して、演算結果を mi0001 にストアします。
レジスタ mi0000 のデータが(10)ならば、レジスタ mi0001 に(12)をストアします。



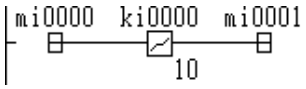
種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	不感帯		整数　7.06 [μs] 実数　6.50 [μs]
機能	入力数値が不感帯範囲内なら 0 を出力します。 入力数値が範囲外なら不感帯値（絶対値）を減算し、結果を出力します。		

-D3 < D1 < +D3　なら　D2 = 0

+D3 <= D1　なら　D2 = D1 — D3

-D3 >= D1　なら　D2 = D1 + D3

使用例



レジスタ mi0000 のデータがレジスタ ki0000 の符号変換したデータ(-10)より大きく、正数データ(10)より小さい場合には、(0)をレジスタ mi0001 にストアします。

レジスタ mi0000 のデータがレジスタ ki0000 のデータ(10)と等しいか大きい場合には、レジスタ mi0000 のデータからレジスタ ki0000 のデータ(10)を減算した結果をレジスタ mi0001 にストアします。

レジスタ mi0000 のデータがレジスタ ki0000 の符号変換したデータ(-10)と等しいか小さい場合には、レジスタ mi0000 のデータとレジスタ ki0000 のデータ(-10)を加算した結果をレジスタ mi0001 にストアします。

種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	パターン		整数　12.4 [μs] 実数　15.3 [μs]
機能	入力数値をパターンメモリにより折れ線近似変換し、結果を出力します。		

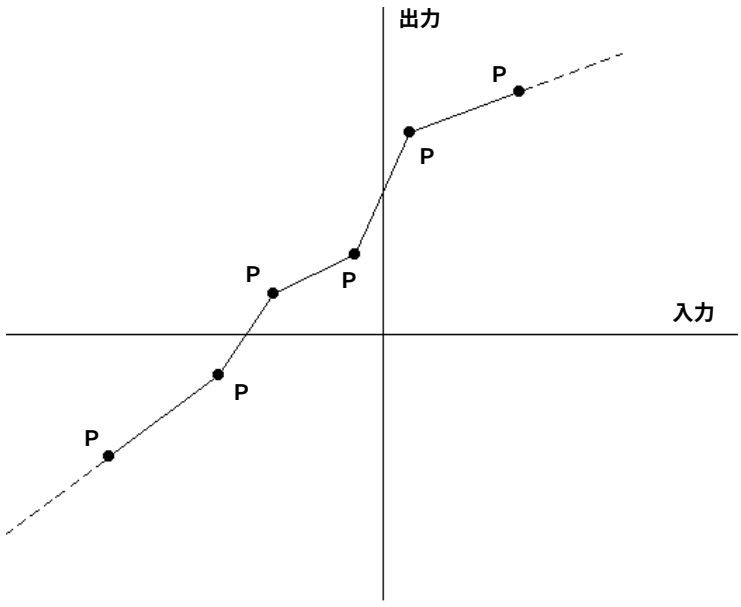
パターンデータはあらかじめツールのパターンデータで設定しておきます。

横軸のデータは必ず小さいほうから大きいほうに順に並べてください。

横軸は関数の入力値に相当し、パターンデータから離脱したデータが入力された場合でもパターンデータの傾斜で延長され、変換し出力します。

グラフ

入力が P1 より小さい場合は、直線 P1・P2 で延伸された近似直線に変換し出力します。
大きい場合も同様に直線 P5・P6 で延伸された近似直線に変換し出力します。



	入力	出力
P1/Q1	-10	-3
P2/Q2	-6	-1
P3/Q3	-4	1
P4/Q4	-1	2
P5/Q5	1	5
P6/Q6	5	6

種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	微分補償		10.2 ［μs］
機能	入力数値の時間微分値の 3 回平均をとり、結果を出力します。		

関数引数設定内容

①微分ゲイン：秒単位系での微分係数（入力変化が毎秒 1.0 の時、1.0 を出力）

急激な変化量に対しては安全のため平均化しています。

演算パラメータは krxxxx 以外に mrxxxx も使用可能ですが、この場合各パラメータをユーザプログラムでセットしてください。

注）　実数演算のみ有効です。

グラフ

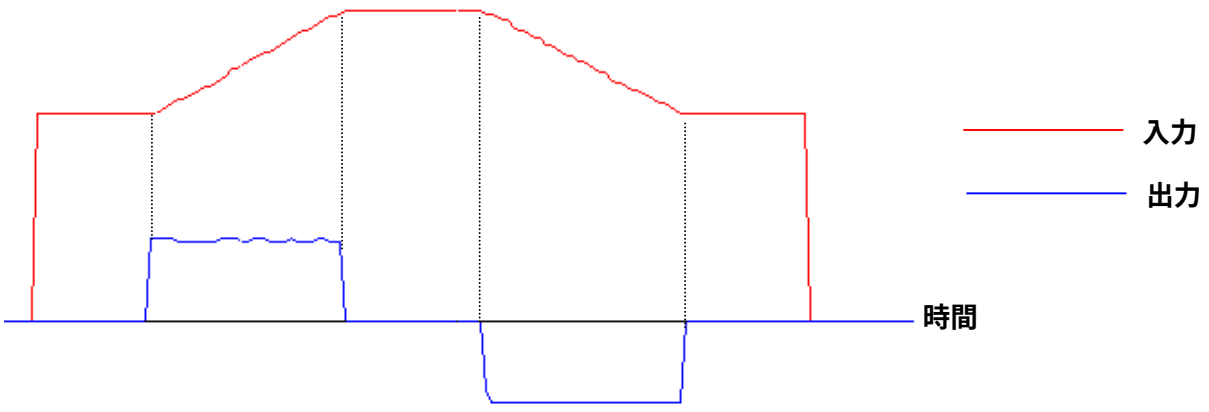
右記のように関数の引数を設定してトレンドグラフを採したのが下になります。

微分補償

微分ゲイン	kr0000	10.000

入力値が一定（傾き 0）のところでは、微分値も 0 なので出力が 0 となります。
入力値が常に変化している部分しか出力値も変化しません。

注）　下のトレンドグラフには急激な変化分についてはグラフ上で現れていません。



種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	位相補償		10.2　[μs]

機能	入力数値に対して位相補償を行い、結果を出力します。

関数引数設定内容

①リセット：入出力短絡リセット動作を指令します。
②位相ゲイン（A）：1.0 との大小関係で進み、位相または遅れ位相となります。
③時間ゲイン（T）：秒単位での時間係数（出力値が入力値に到達するまでの時間：秒）

演算パラメータは krxxxx 以外に mrxxxx も使用可能ですが、この場合各パラメータをユーザプログラムでセットしてください。

リセットを ON すると入出力間を短絡しますので、任意の値にプリセットすることができます。

注)　実数演算のみ有効です。

グラフ

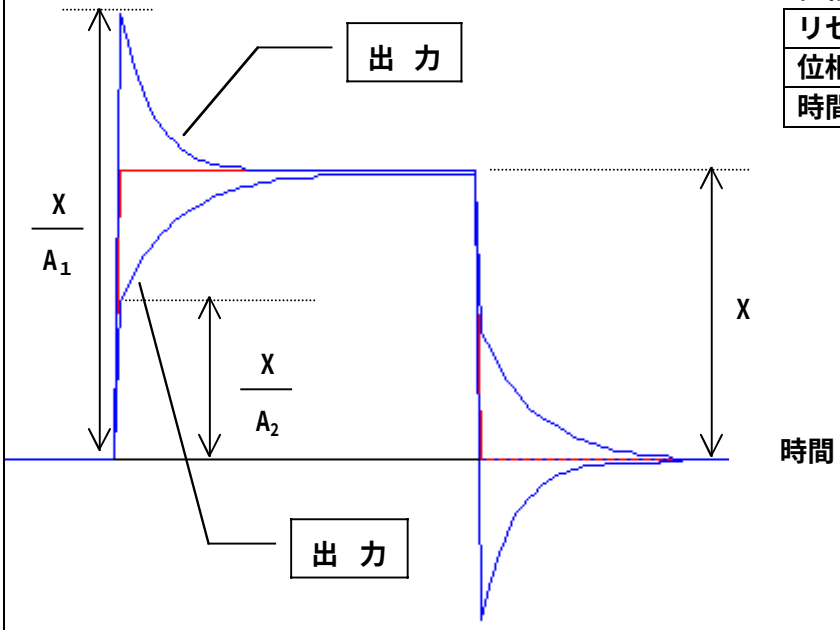
右記のように関数の引数を設定してトレンドグラフを採取したのが下になります。
時間ゲインにより出力値が入力値に近づいていくカーブの大きさが変化します。
小さければ小さな弧を描き、大きければ大きな弧を描きます。

位相補償　　出力 1

リセット	G00000	
位相ゲイン(A_1)	kr0000	2.0000
時間ゲイン	kr0001	-0.8000

位相補償　　出力 2

リセット		
位相ゲイン(A_2)	kr0002	0.6000
時間ゲイン	kr0003	1.0000



種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	PI 補償		12.6 ［μs］
機能	入力数値に対して PI 補償（比例・積分）を行い、結果を出力します。		

関数引数設定内容

①リセット：入出力短絡リセット動作を指令します。
②ホールド：積分ホールド SW（積分停止）
③比例ゲイン：
④積分ゲイン：秒単位系での積分係数（出力値が入力値に到達するまでの時間：秒）
⑤上限値：出力する上限値を指定します。
⑥下限値：出力するか現地を指定します。

演算パラメータは krxxxx 以外に mrxxxx も使用可能ですが、この場合各パラメータをユーザプログラムでセットしてください。

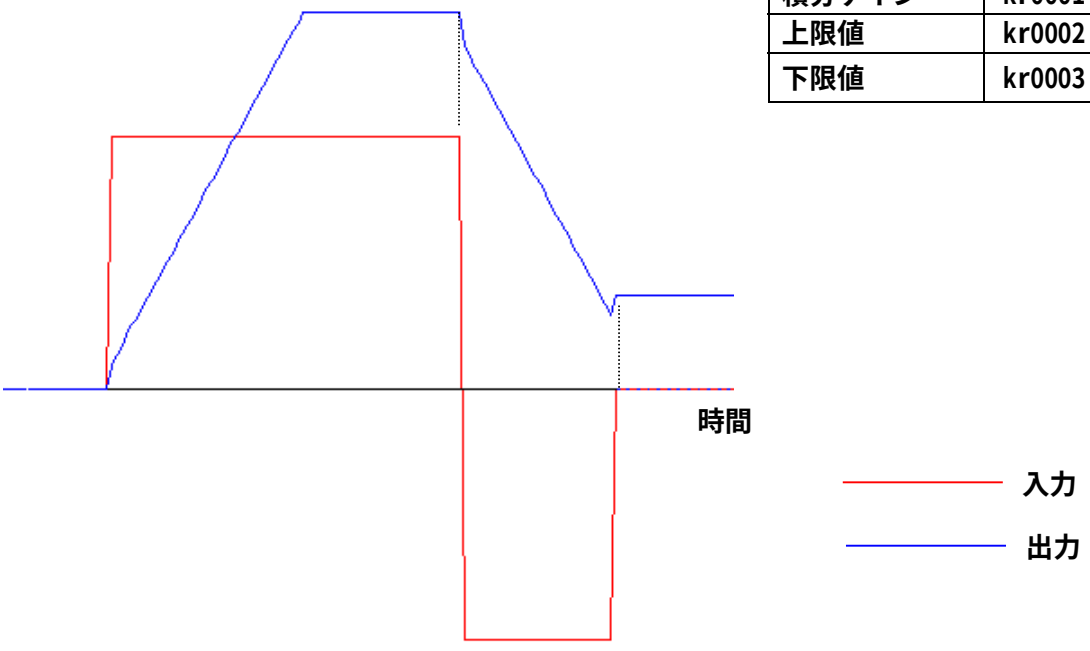
リセットを ON すると入出力間を短絡しますので、任意の値にプリセットすることができます。
注)　実数演算のみ有効です。

グラフ

右記のように関数の引数を設定してトレンドグラフを採取したのが下になります。
比例ゲインによって始めの出力値の高さが変化し、積分ゲインによって出力値の傾きが変化します。

PI 補償

リセット	G00000	
ホールド	G00001	
比例ゲイン	kr0000	0.1000
積分ゲイン	kr0001	3.0000
上限値	kr0002	30.000
下限値	kr0003	-30.000



種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	直線形変化率制限		8.4 ［μs］
機能	入力数値の時間的変化率制限を行い、結果を出力します。		

関数引数設定内容

①リセット：入出力短絡リセット動作を指令します。
②最大上昇率（＞0.0：正の値）：毎秒当たりの出力上昇率制限値
（例：10.0＝毎秒 10 以下の上昇を許可）
③最大下降率（＜0.0：負の値）：毎秒当たりの出力下降率制限値
（例：-10.0＝毎秒 10 以下の下降を許可）

演算パラメータは krxxxx 以外に mrxxxx も使用可能ですが、この場合各パラメータをユーザプログラムでセットしてください。

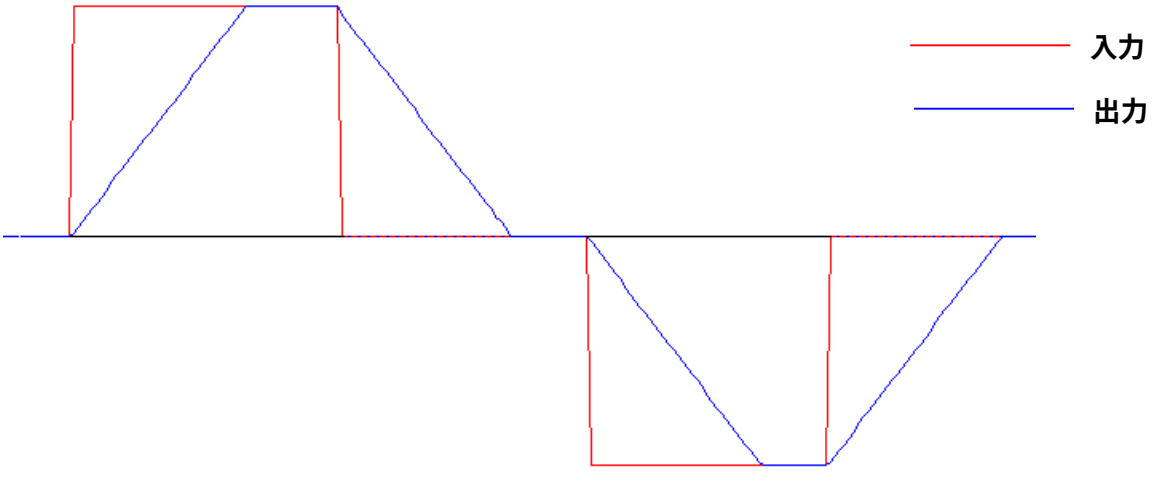
リセットを ON すると入出力間を短絡しますので、任意の値にプリセットすることができます。
注)　実数演算のみ有効です。

グラフ

右記のように関数の引数を設定してトレンドグラフを採したのが下になります。
上昇・下降率によって出力値の傾きを設定できます。(スップ入力を加えた場合)

直線変化率制限

リセット	G00000	
最大上昇率	kr0000	0.1000
最大下降率	kr0001	-0.1000



種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	S 字形変化率制限 (S-ARC)		23.4　[μs]

機能	入力数値に対し S 字時間的変化率制限を行い、結果を出力します。

関数引数設定内容

①リセット：入出力短絡リセット動作を指令します。
②最大上昇率（＞0.0）：毎秒当たりの出力上昇率制限値
③最大下降率（＜0.0）：毎秒当たりの出力下降率制限値
④加・上昇率（＞0.0）：加速開始時の毎秒当たりの加加速度値
⑤減・上昇率（＜0.0）：加速終了時の毎秒当たりの減加速度値
⑥減・減少率（＞0.0）：減速終了時の毎秒当たりの減減速度値
⑦加・減少率（＜0.0）：減速開始時の毎秒当たりの加減速度値
⑧S 字速終了係数(＞0.0)：加減速終了時の変化率制限値
　　通常は④～⑦の絶対値の一番大きい値の 2 倍位で設定する。

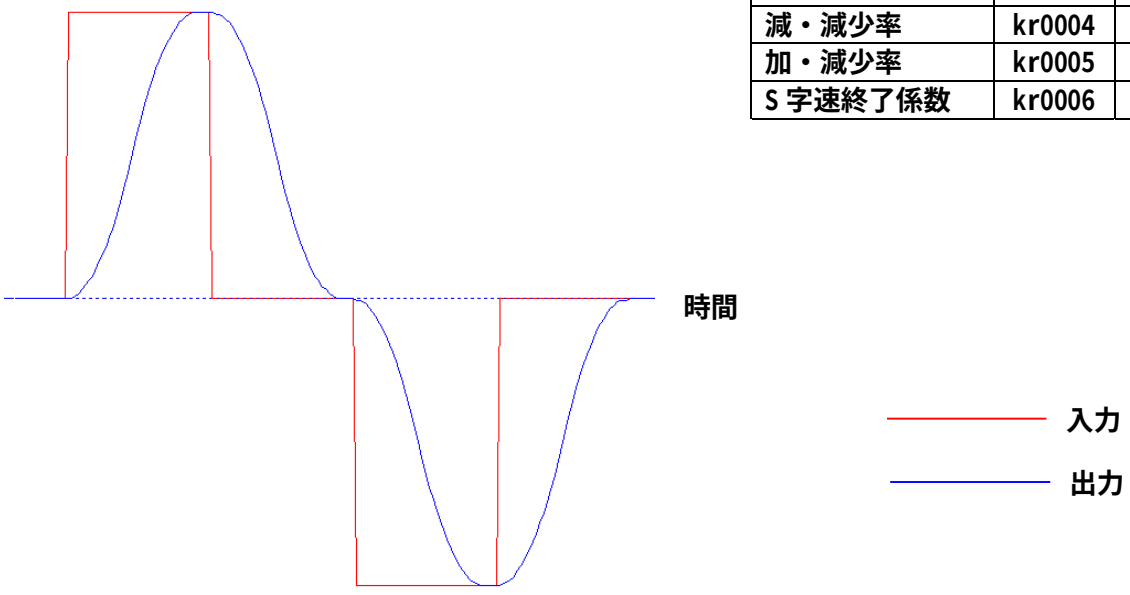
リセットを ON すると入出力間を短絡しますので、任意の値にプリセットすることができます。
注)　実数演算のみ有効です。

グラフ

右記のように関数の引数を設定してトレンドグラフを採取したのが下になります。
ARC と同様ですが、直線になる前のカーブ（B1～4）も設定しますので、S 字形のような波形を出力します。
(注意)　加減速中に入力値を変更するとオーバシュートする場合があります。

S 字型変化率制限

リセット	G00000	
最大上昇率	kr0000	10.000
最大下降率	kr0001	-10.000
加・上昇率	kr0002	0.020
減・上昇率	kr0003	-0.020
減・減少率	kr0004	0.0020
加・減少率	kr0005	-0.0020
S 字速終了係数	kr0006	0.0040



種類	名称	シンボル	実行時間
データフロー言語 (関数 3)	三角関数 逆三角関数		SIN　18.6 [μs] COS　16.8 [μs] TAN　30.4 [μs] ATAN　20.5 [μs]

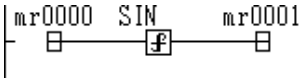
機能

入力数値を三角（逆三角）関数演算を行い、結果を出力します。

sin 関数	D1 —[SIN]— D2	D2 = sin (D1)
cos 関数	D1 —[COS]— D2	D2 = cos (D1)
tan 関数	D1 —[TAN]— D2	D2 = tan (D1)
asin 関数	D1 —[ASIN]— D2	D2 = $\sin^{-1}$ (D1)
acos 関数	D1 —[ACOS]— D2	D2 = $\cos^{-1}$ (D1)
atan 関数	D1 —[ATAN]— D2	D2 = $\tan^{-1}$ (D1)

注)　実数演算のみ有効です。

使用例



mr0001 ＝ SIN(mr0000)

レジスタ mr0000 のデータの正弦を演算し、結果をレジスタ mr0001 にストアします。

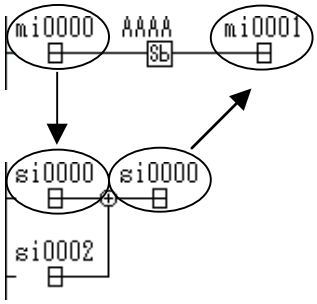
種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	無条件サブルーチン	XXXXXX Sb	14.0 [μs]

機能	サブルーチンを無条件で実行します。

シンボルをダブルクリックすると引数設定画面が出てユーザがサブルーチンに対して引数を設定することができます。
サブルーチン内ではスタックレジスタ（sr0000、si0000、SI0000）を使用してデータの受け渡しを行います。スタックレジスタには引数設定画面から設定します。
実際には以下の様なデータの流れになります。

入力データ		スタックレジスタ		出力データ
整数データ	_	si0000	_	整数データ
実数データ	_	sr0000	_	実数データ
リレー/コイル	_	SI0000	_	リレー／コイル
呼び出し元		サブルーチン		呼び出し元

使用例



無条件でサブルーチン AAAA を実行します。レジスタ mi0000、mi0001 は慣例的に使用しますが、このデータも使用したい時には、レジスタ mi0000 のデータはサブルーチン AAAA のスタックレジスタ si0000 に渡されます。そしてサブルーチン AAAA で計算されたデータがスタックレジスタ si0000 にストアされたら、レジスタ mi0001 にデータがストアされます。
ただし、サブルーチン AAAA で使用しない場合には、レジスタ mi0000 のデータがレジスタ mi0001 にストアされます。

種類	名称	シンボル	実行時間
LD 言語	ジャンプ命令	―(JPXXXX)―	――――
	ラベル命令	XXXXXX L―	
機能	ジャンプ：指定回路、指定ラベルへジャンプします。 ラベル：ジャンプ先ラベルに使用します。		

論理回路の 1 つとみなされます。

XXXX は回路番号またはラベル名（4 桁）です。

注 1)　サブプログラム・サブルーチン間でのジャンプはできません。
注 2)　また 1 ヵ所でループするようプログラムも作成可能ですが、永久ループにならないようにしてください。
注 3)　ラベルの右側にはレジスタのストアにしてください。

使用例

リレーB00000 が ON の時は、ラベル ABCD ラインにジャンプし、ラベル ABCD との間にあるプログラムを実行しません。

リレーB00000 が ON の時は、レジスタ kr0000 のデータ(10.000)がレジスタ mr0000 にストアされ、レジスタ mi0000 には 1 がストアされます。
リレーB00000 が OFF の時は、レジスタ kr0000 のデータ(10.000)がレジスタ mr0000 にストアされず、レジスタ mi0000 には 0 がストアされます。

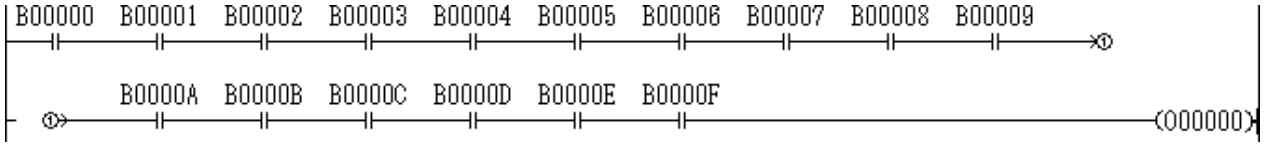
種類	名称	シンボル	実行時間
LD 言語	結合子 (ストア)	⟶①	0.10 [μs]
	結合子 (ロード)	①⟶	0.06 [μs]
機能	論理演算、数値演算結果の中間メモリのストアおよびロードを行います。		

直列に 12 個以上論理記号、数値記号がある時に使用します。

必ずネットワークとネットワークの間に入れてください。

1 回路に 10 組までシンボルが入れられますが、必ずストアの後にロードしてください。

使用例

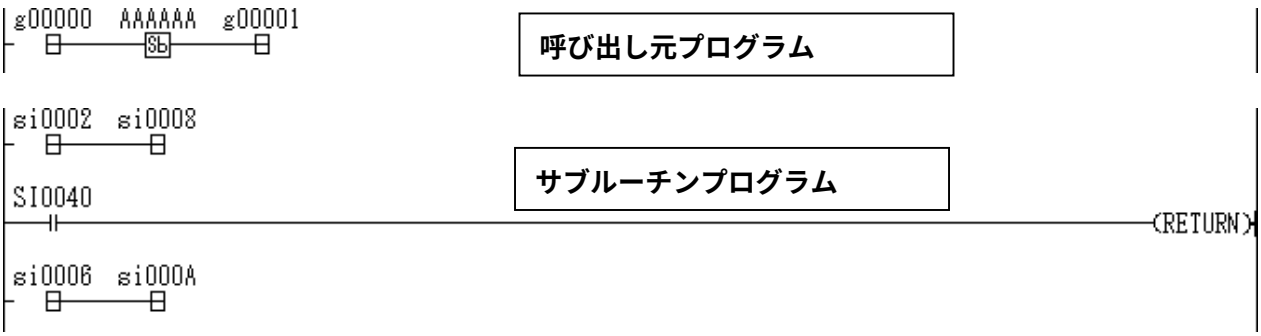


種類	名称	シンボル	実行時間
LD 言語	サブルーチン プログラム処理終了	―(RETURN)―	14.0 ［μs］

機能	サブルーチンプログラムを終了します。

サブルーチンプログラム内で、ある条件で終了させたいときに使用します。

使用例



リレーI00000 が OFF の時は、スタックレジスタ si0002 のデータ=ki0000(5)のデータがスタックレジスタ si0008 にストアされ、レジスタ mi0000 にデータ(5)がロードされます。スタックレジスタ si0006 のデータ=z00009 のデータはスタックレジスタ si000A にストアされ、レジスタ mi0001 にロードされます。

しかし、リレーI00000 が ON の時は、スタックレジスタ si0002 のデータ(5)はそのままスタックレジスタ si0008 にストアされますが、スタックレジスタ si0006 のデータは I00000 が ON した時のデータが si000A にストアされたままになります。(z00009 は 1 ミリカウンタなので 100 の時 ON したなら si000A には 100 がストアされます。また I00000 を ON すれば、si0006 のデータがストアされます。)

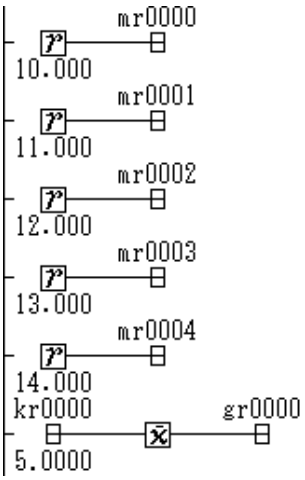
引数名	ラベル名	値
si0002	ki0000	5
SI0040	I00000	
si0006	z00009	
si0008	mi0000	
si000A	mi0001	

種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	（算術）平均		――――――
機能	引数で設定した先頭アドレスから入力数値分のデータの算術平均値をとり、 結果を出力します。		

関数引数設定内容

①バッファアドレス先頭（mrXXXX）：入力が 1 より少ない場合は 1 とみなし 1 個目のデータの値を返します。

使用例



算術平均の引数　バッファアドレス先頭：mr0000

以上のように設定すると、算術平均はレジスタ kr0000 のデータ(5.0000)と引数を読み込み、
(mr0000 + mr0001 + mr0002 + mr0003 + mr0004)/5
の演算結果(12.000)をレジスタ gr0000 にストアします。

種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	フィルタ		18.8　[μs]
機能	入力数値に対し周波数制限を行い、結果を出力します。		

関数引数設定内容

①リセット：入出力短絡リセット動作を指令します。
②下限周波数（＞0.0：正の値）：3db 低下の下限周波数
③上限周波数（＞0.0：正の値）：3db 低下の上限周波数

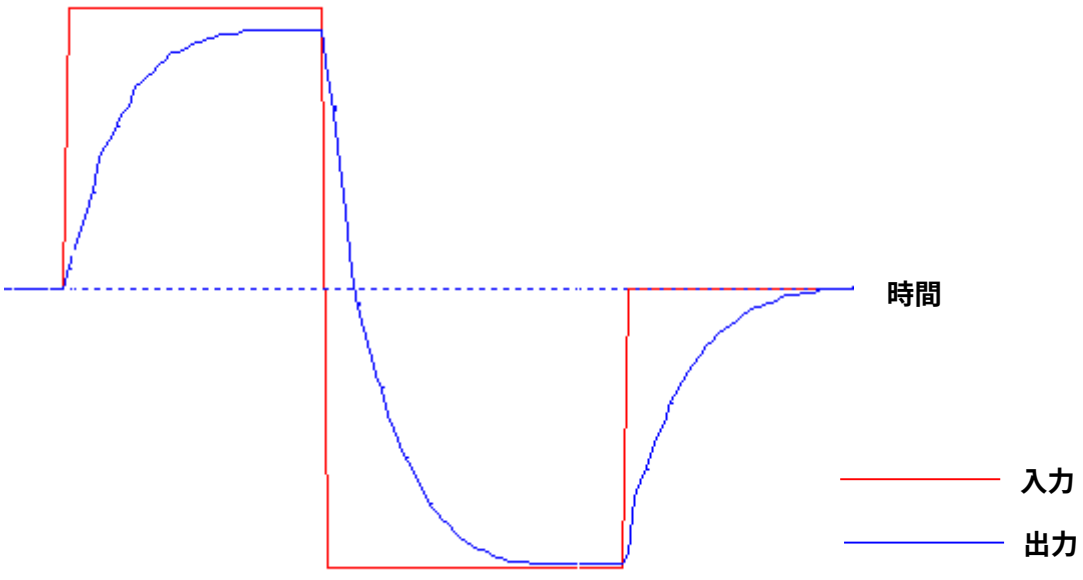
注)　実数演算のみ有効です。

グラフ

右記のように関数の引数を設定してトレンドグラフを採したのが下になります。

フィルタ

リセット	G00000	
下限周波数	kr0000	0.0001
上限周波数	kr0001	0.0500



種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	PID 補償		14.8 ［μs］

機能	入力数値に対して PID 補償を行い、結果を出力します。

関数引数設定内容

①リセット：入出力短絡リセット動作を指令します。
②ホールド：積分停止 SW
③ゼロクリア：ゼロリセット指令するリレーを指定します。
④比例ゲイン：
⑤積分ゲイン：秒単位系での積分係数（出力値が入力値に到達するまでの時間：秒）
⑥微分ゲイン：秒単位系での微分係数（入力変化が毎秒 1.0 の時、1.0 を出力）
⑦MAX リミット：出力する上限値を指定します。
⑧MIN リミット：出力する下限値を指定します。

リセットを ON すると入出力間を短絡しますので、任意の値にプリセットすることができます。

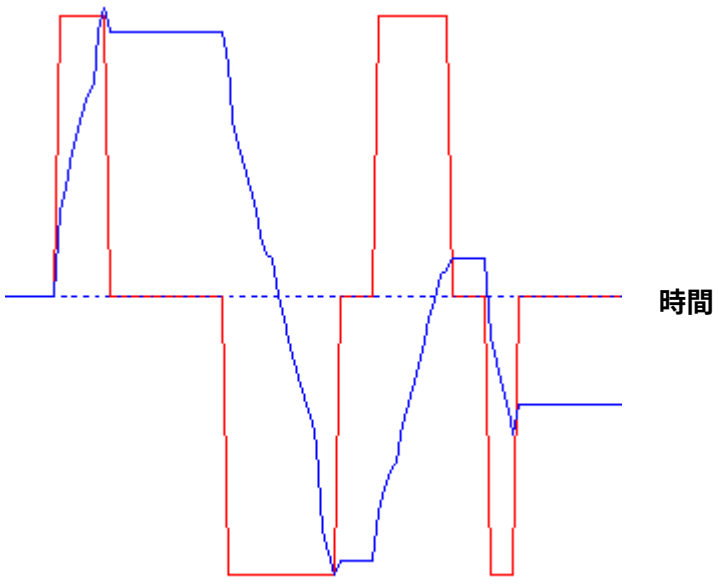
注)　実数演算のみ有効です。

グラフ

右記のように関数の引数を設定してトレンドグラフを採取したのが下になります。

フィルタ

リセット	G00000	
ホールド	G00001	
ゼロクリア	G00002	
比例ゲイン	kr0000	0.1000
積分ゲイン	kr0001	3.0000
微分ゲイン	kr0002	0.0100
MAX リミット	kr0003	30.000
MIN リミット	kr0004	-30.000



種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	一時遅れ		9.8 ［μs］

機能	入力数値に対し一次遅れ応答を出力します。

関数引数設定内容
①リセット：入出力短絡リセット動作を指令します。
②時定数：T 秒

演算開始時に必ずリセット SW を ON してください。

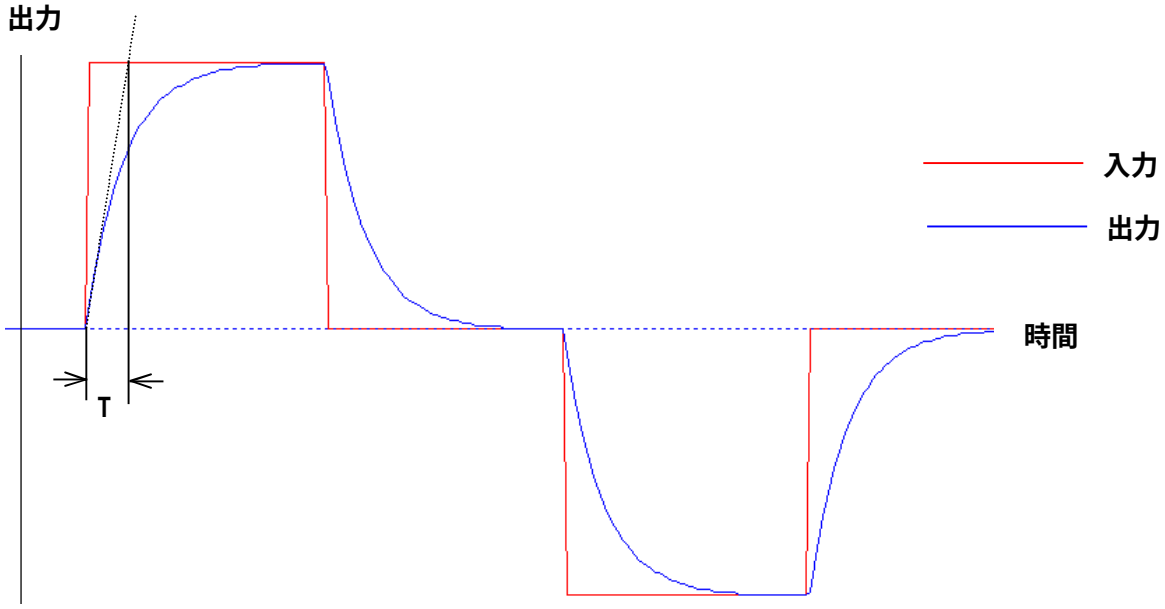
注)　実数演算のみ有効です。

グラフ

右記のように関数の引数を設定してトレンドグラフを取したのが下になります。
時定数によって入力の変化した間を出力値は弧を描きがら出力します。

フィルタ

リセット	G00000	
時定数	kr0000	1.0000



種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	ディレー （時間遅れ）		9.6 ［μs］
機能	入力数値に設定した遅れ時間を付加し出力します。		

関数引数設定内容

①リセット：入出力短絡リセット動作を指令します。

②遅れ時間：T（秒）

③サンプリング時間：⊿T（秒）　　サンプル数（T/⊿T）は 1000 以下で有効です。

リセット SW を ON するとディレーは無くなります。

注)　実数演算のみ有効です。

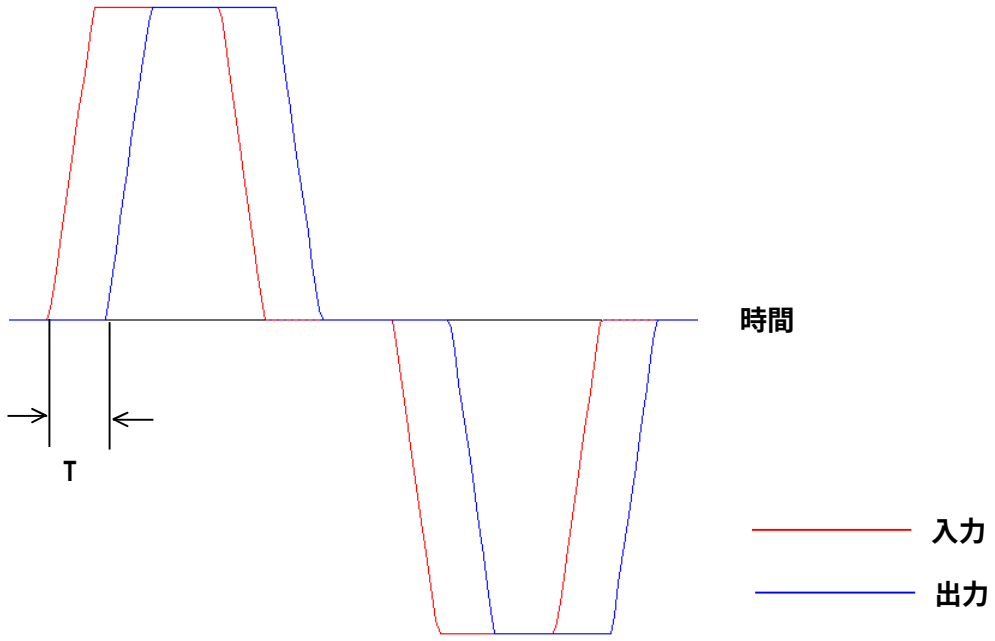
グラフ

右記のように関数の引数を設定してトレンドグラフを取したのが下になります。

遅れ時間により入力波形を T（秒）だけ遅らせて出力ます。

ディレー

リセット		
遅れ時間	kr0000	5.0000
サンプリング時間	kr0001	1.0000



種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	定周期パルス		8.0 ［μs］

機能	入力数値を設定した時間でオン・オフさせ出力します。

関数引数設定内容

①リセット：入出力短絡リセット動作を指令します。

②オン時間（秒）：出力を ON させる時間を指定します。

③オフ時間（秒）：出力を OFF させる時間を指定します。

注)　実数演算のみ有効です。

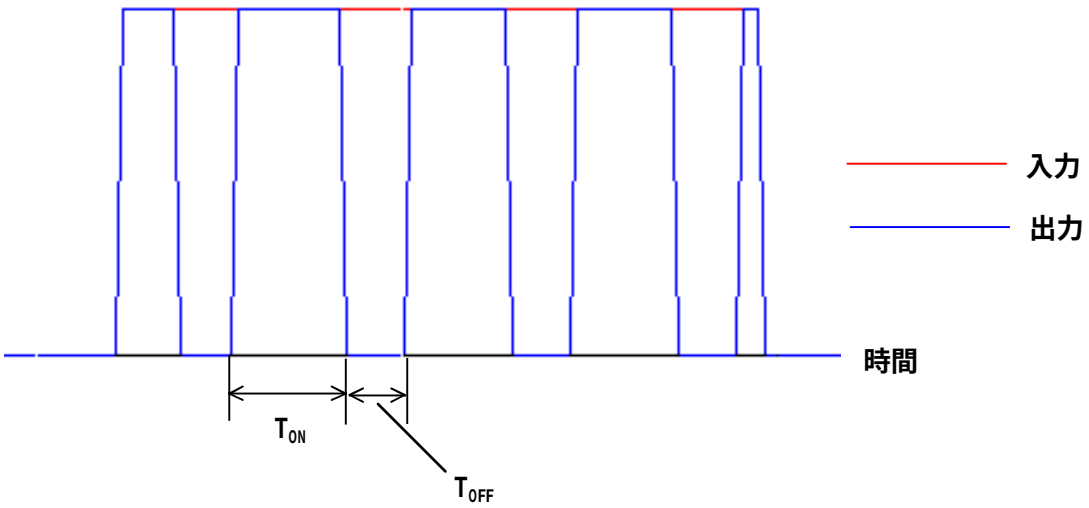
グラフ

右記のように関数の引数を設定してトレンドグラフを取したのが下になります。

入力された波形をオン・オフ時間によって出力します。

定周期パルス

リセット	G00000	
オン時間	kr0000	5.0000
オフ時間	kr0001	3.0000



種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	変数設定パターン		12.7 ［μs］

機能	入力数値をパターンメモリにより折れ線近似変換し出力します。

関数引数設定内容

①ポイント数(≧2：整数)：入力パターン数

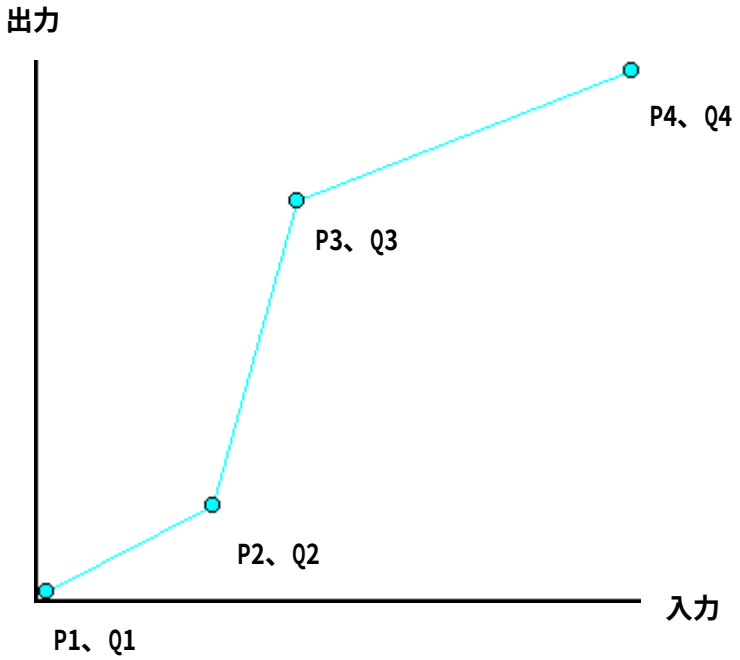
②パターンバッファ先頭(mrXXXX)：入力バッファ先頭アドレス

パターンではあらかじめパターンデータで初期値を設定しましたが、ここでは回路中の実数値を変更することができます。

プロセス制御で得られデータを蓄積することによって学習制御に応用できます。

注)　実数演算のみ有効です。

グラフ



P1/Q1	mr0000	mr0001
P2/Q2	mr0002	mr0003
P3/Q3	mr0004	mr0005
P4/Q4	mr0006	mr0007

種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	上下限リミッタ		7.45　[μs]

機能	入力数値に上下限リミッタを付加し出力します。

関数引数設定内容

①上限値：出力の上限値を指定します。

②下限値：出力の下限値を指定します。

注)　実数演算のみ有効です。

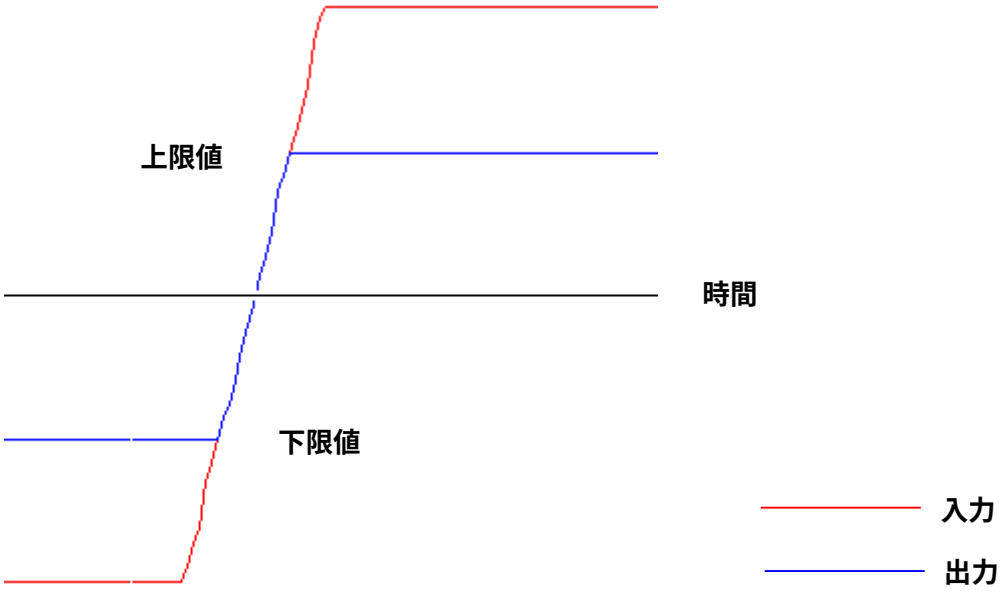
グラフ

右記のように関数の引数を設定してトレンドグラフを取したのが下になります。

入力された波形を上限値、下限値によって出力します。

上下限リミッタ

上限値	kr0000	10.000
下限値	kr0001	-10.000



種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	ヒステリシス		8.4 ［μs］
機能	入力数値にヒステリシス（上昇下降時の 2 ゲイン増幅器）を付加し出力します。		

関数引数設定内容：

①リセット：出力値　＝　入力値　×　G1　とします。

②ロウ側ゲイン：G1（0.0＜G1＜G2）

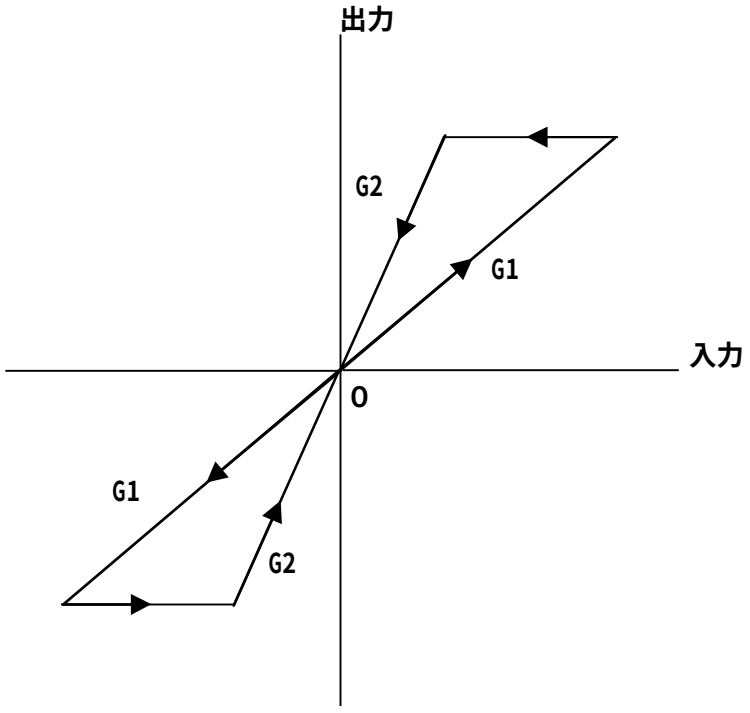
③ハイ側ゲイン：G2（0.0＜G1＜G2）

入力データが上昇時には G1 が有効となり下降時には G2 が有効となります。
上昇・下降または下降・上昇の切り替え時には一定の出力に留まります。

演算開始時に必ずリセット SW を ON してください。
注)　実数演算のみ有効です。

グラフ

入力データの変化の履歴により出力データが図に示すカーブとなります。



種類	名称	シンボル	実行時間
データフロー言語 (関数 3)	スケーリング	SCAL f	7.27　[μs]

機能	入力数値をスケーリング（積和演算）を付加し出力します。

関数引数設定内容：

①ゲイン：積和演算の乗算係数

②オフセット：積和演算の加算係数

出力　＝　入力　＊　ゲイン　＋　オフセット

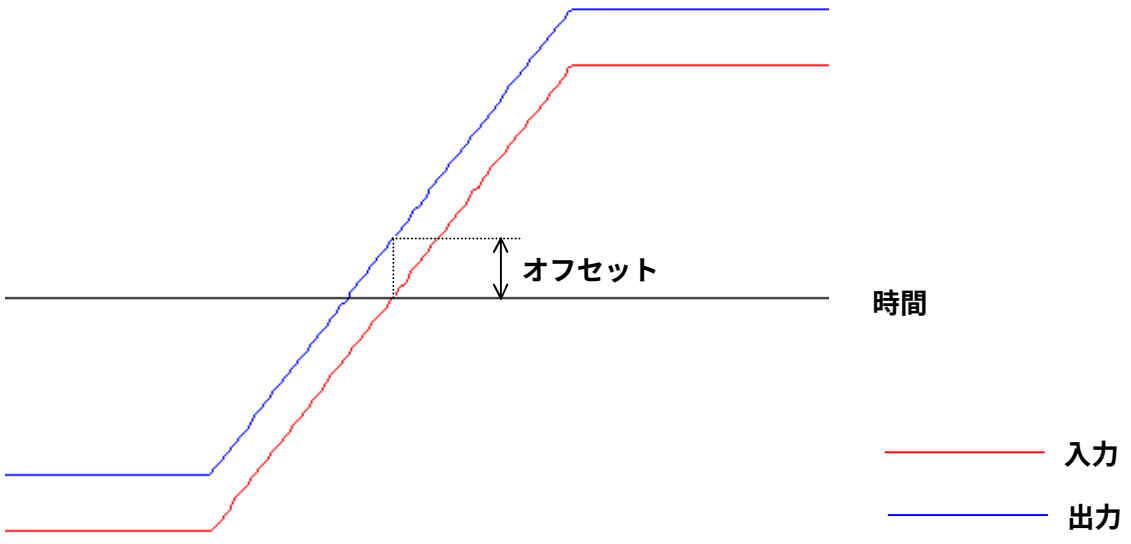
注)　実数演算のみ有効です。

グラフ

右記のように関数の引数を設定してトレンドグラフを採取したのが下になります。
入力された波形をゲイン・オフセットによって出力します。

スケーリング

ゲイン	kr0000	1.0000
オフセット	kr0001	5.0000



種類	名称	シンボル	実行時間
データフロー言語 (関数 3)	バックラッシュ	BKLS	8.8 ［μs］
機能	入力数値にバックラッシュ（一種の積分補償）を付加し出力します。		

関数引数設定内容

①リセット：入出力短絡リセット動作を指令します。

②バックラッシュの幅：W

演算開始時に必ずリセット SW を ON してください。

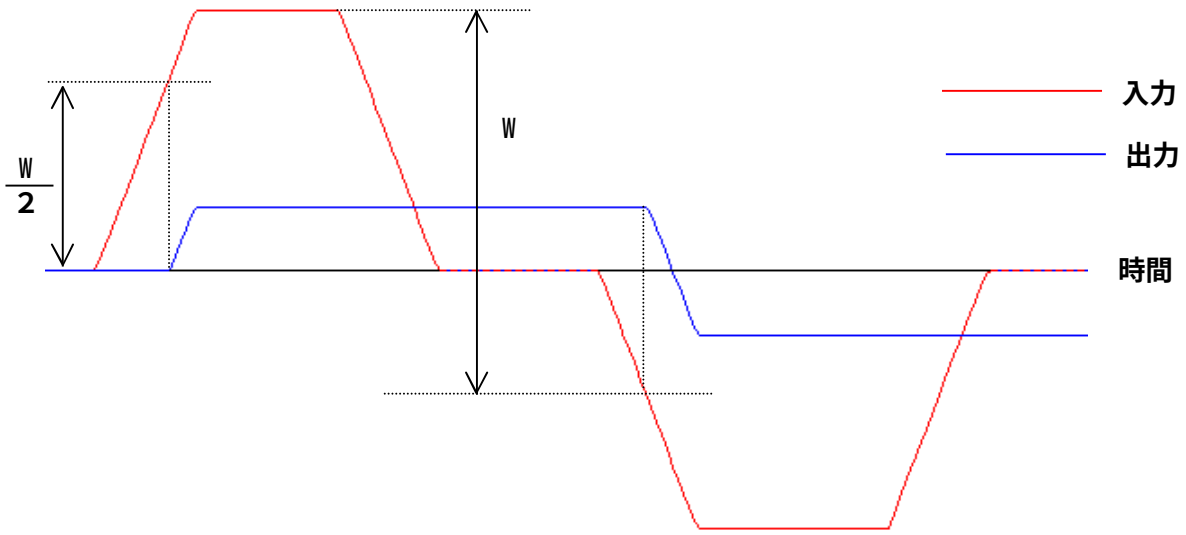
注)　実数演算のみ有効です。

グラフ

右記のように関数の引数を設定してトレンドグラフを取したのが下になります。

バックラッシュ

リセット	G00001	
バックラッシュの幅	kr0000	20.000



種類	名称	シンボル	実行時間
データフロー言語 (関数 3)	バックラッシュ補正	BKLC f	8.2 ［μs］
機能	入力数値をバックラッシュ補正（一種の微分補償）し出力します。		

関数引数設定内容

①リセット：入出力短絡リセット動作を指令します。

②バックラッシュの幅：W

演算開始時に必ずリセット SW を ON してください。

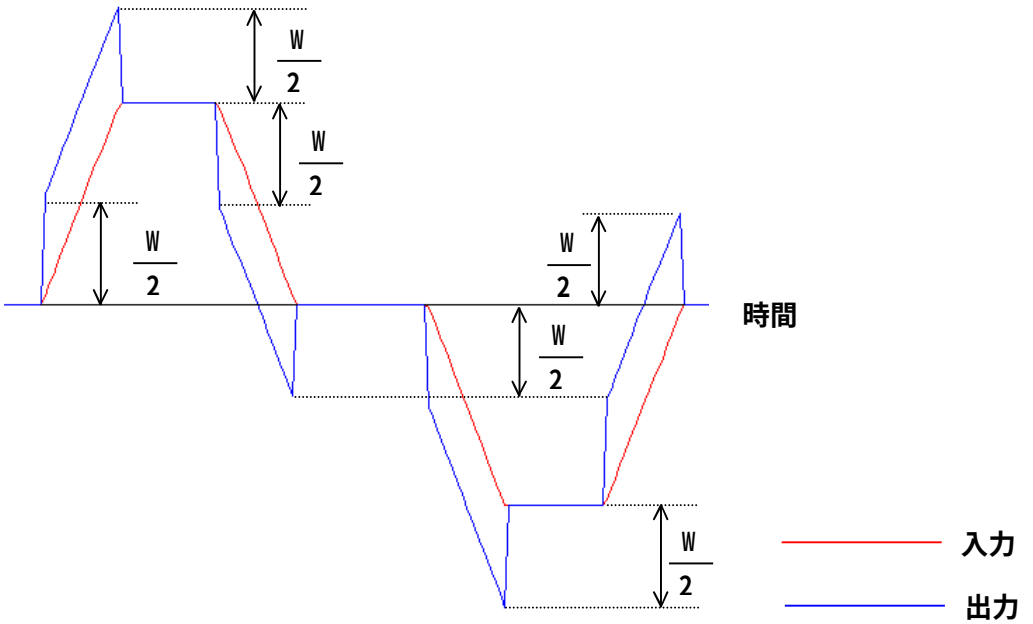
注)　実数演算のみ有効です。

グラフ

右記のように関数の引数を設定してトレンドグラフを取したのが下になります。

バックラッシュ補正

リセット	G00001	
バックラッシュの幅	kr0000	20.000



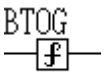
種類	名称	シンボル	実行時間
データフロー言語 (関数 2)	条件付サブルーチン	XXXXXX SB	――――――

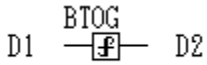
機能	入力の論理条件によりサブルーチンを実行します。

入力が ON の時サブルーチンを実行し、OFF の時は実行しません。
その他の内容は無条件サブルーチンと同様です。

使用例

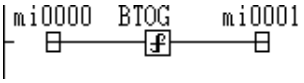
リレーB00000 が ON の時は、サブルーチン AAAA を実行します。
リレーB00000 が OFF の時は、サブルーチン AAAA を実行しません。

種類	名称	シンボル	実行時間
データフロー言語 (関数 3)	バイナリー グレイコード		――――――
機能	入力数値を整数データとして読み込み、グレイコード変換して出力します。		



注）　グレイコード変換　  　とは逆の操作を行います。間違えないでください。

使用例



レジスタ mi0000 のデータを 16 ビット整数として読み込み、グレイコードに変換し出力します。
レジスタ mi0000 のデータが(10)の場合、レジスタ mi0001 には(15)がストアされます

D1	D2	D1	D2	D1	D2	D1	D2
整数	グレイ	整数	グレイ	整数	グレイ	整数	グレイ
0000	0000	0100	0110	1000	1100	1100	1010
0001	0001	0101	0111	1001	1101	1101	1011
0010	0011	0110	0101	1010	1111	1110	1001
0011	0010	0111	0100	1011	1110	1111	1000

10 → 1010 → 1111 → 15

入力　整数　グレイコード　出力

種類	名称	シンボル	実行時間
データフロー言語 (関数 3)	割り余り	DIVMOD	――――――

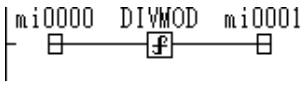
機能	入力値の除算値と余りを出力します。

関数引数設定内容

①除数（整数）：入力値を割る数

②余り（整数）：余りをストアするレジスタ

使用例



右のように DIVMOD の引数を設定すると、レジスタ mi0000 のデータを割る数 ki0000(7)で割った余りがレジスタ mi0002 にストアされます。またレジスタ mi0001 には商がストアされます。

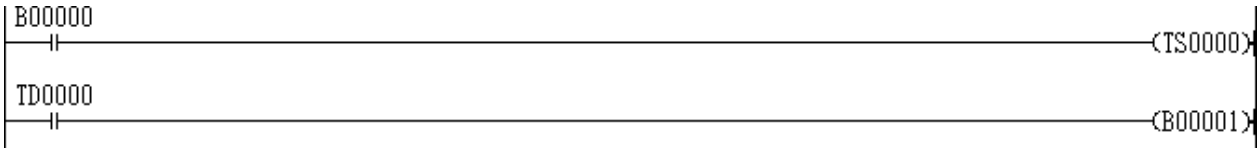
DIVMOD

引数	ラベル	値
除数(整数)	ki0000	7
余り(整数)	mi0002	

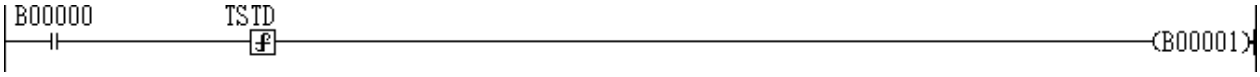
レジスタ mi0000 のデータが(10)の場合、レジスタ mi0001 には商として(1)、レジスタ mi0002 には余りとして(3)がストアされます。

種類	名称	シンボル	実行時間
データフロー言語 (関数 3)	オンタイマ (TSTD)	TSTD f	――――
	オフタイマ (TRTC)	TRTC f	
機能	オンタイマリレー(TS、TD)とオフタイマリレー(TR、TC)を 1 行にまとめたもので動作は同じです。		

TSTD：入力ビットが ON すると引数で設定した時間経過後にコイルが ON します。



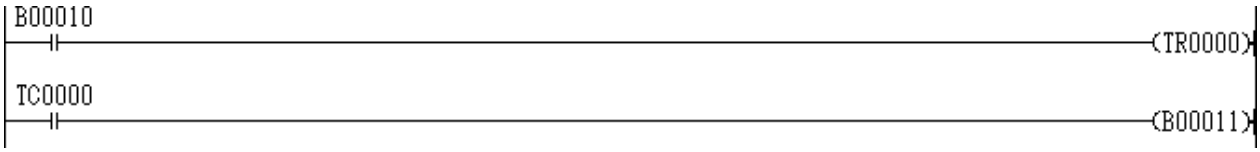
↓2 行書いていたところを 1 行で書けます。



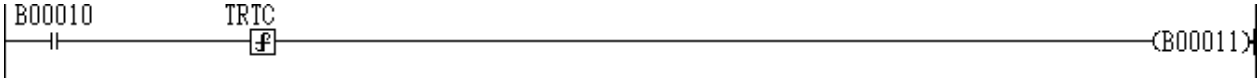
関数引数設定内容

① タイマ値（実数）：指定時間経過後に ON させる時間を設定します。

TRTC：入力ビットが OFF すると引数で設定した時間経過後にコイルが OFF します。



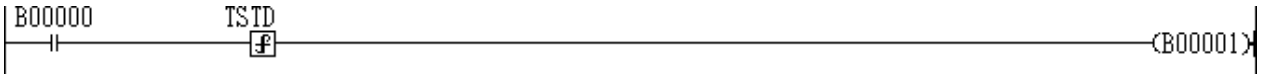
↓2 行書いていたところを 1 行で書けます。



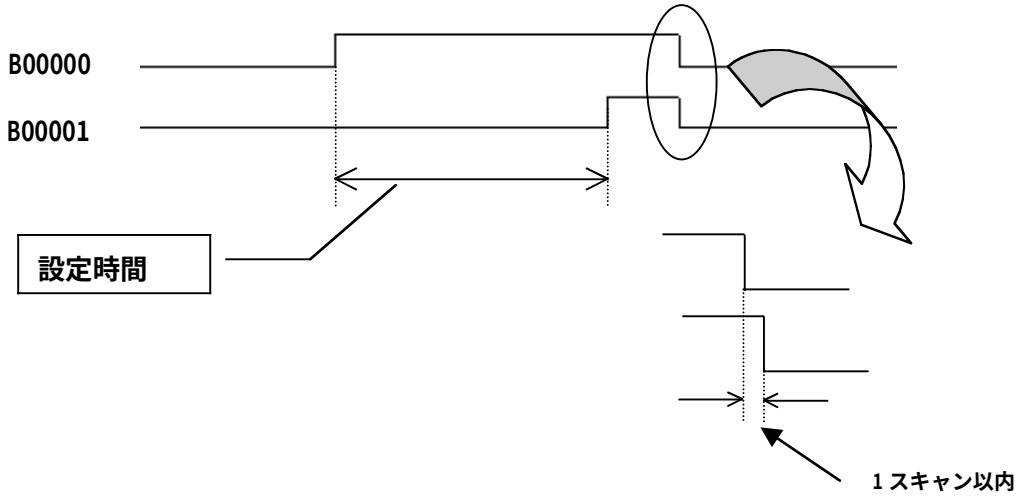
関数引数設定内容

①タイマ値（実数）：指定時間経過後に OFF させる時間を設定します。

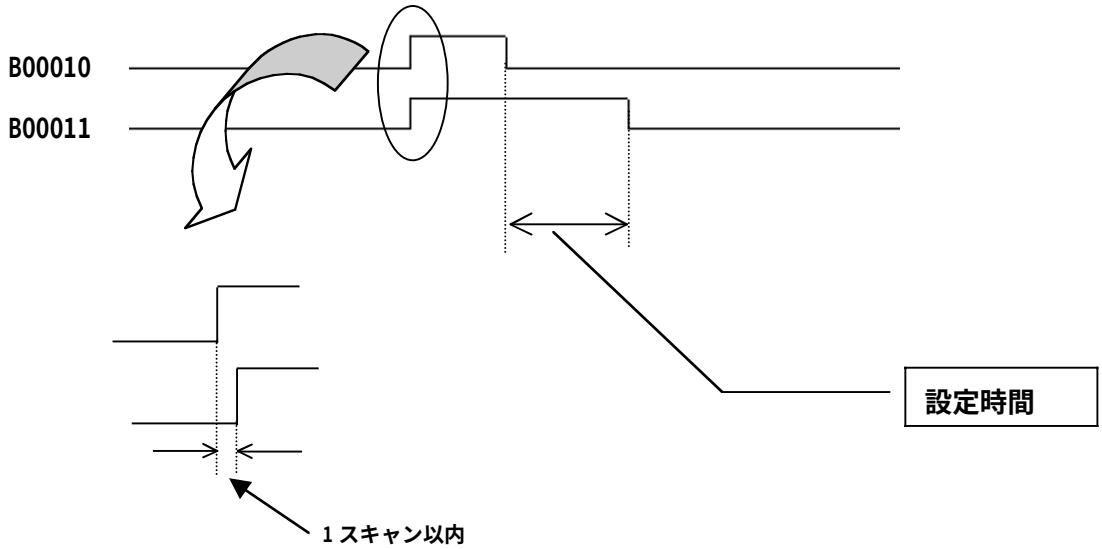
使用例



リレーB00000 が ON してから TSTD で設定した時間経過後にリレーB00001 が ON します。

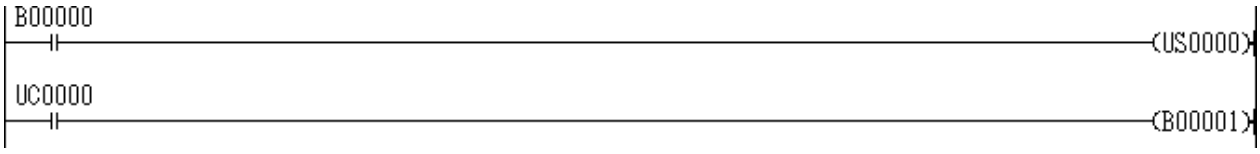


リレーB00010 が OFF してから TRTC で設定した時間経過後にリレーB00011 が OFF します。

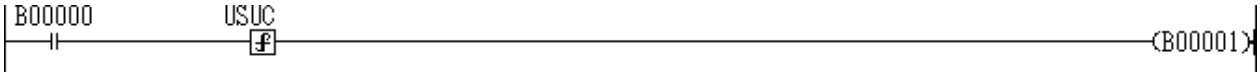


種類	名称	シンボル	実行時間
データフロー言語 (関数 3)	オン微分(USUC)	USUC	―――――
	オフ微分(DSDC)	DSDC	
機能	オン微分リレー(US、UC)とオフ微分リレー(DS、DC)を 1 行にまとめたもので動作は同じですが 1 スキャンは遅れません。		

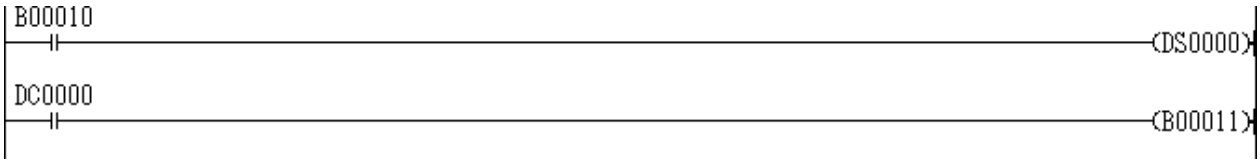
USUC：入力ビットが ON すると 1 スキャン遅れずに 1 スキャン ON します。



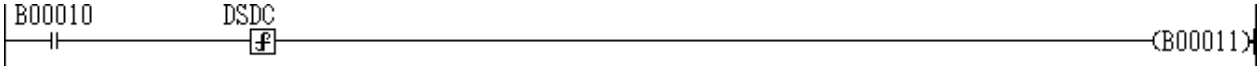
↓2 行で書いていたところを 1 行で書けます。



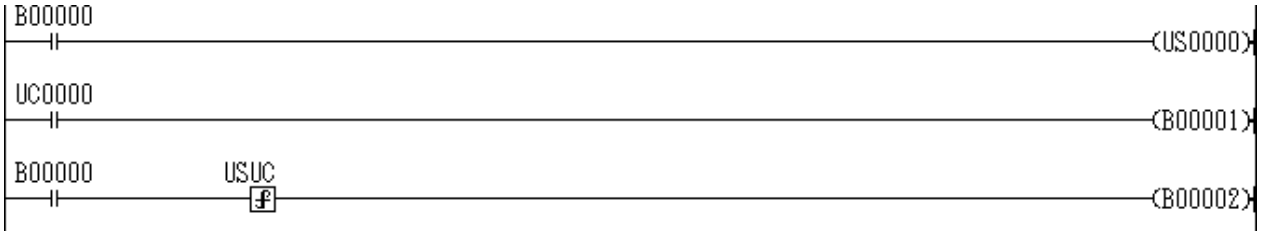
DSDC：入力ビットが OFF すると 1 スキャン遅れずに 1 スキャン ON します。



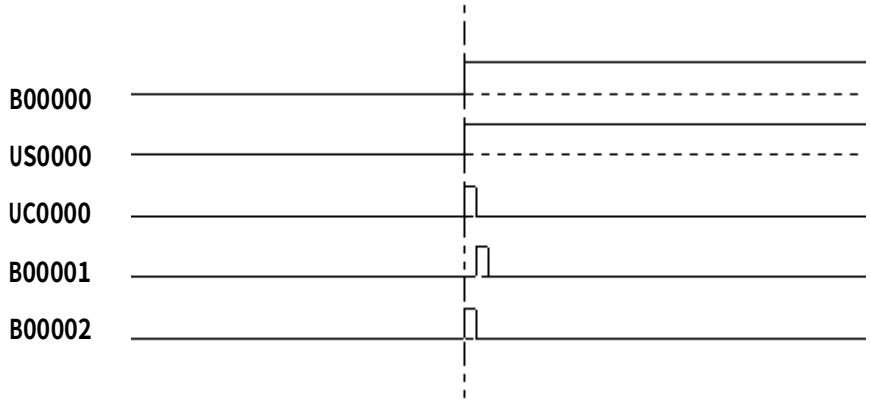
↓2 行で書いていたところを 1 行で書けます。

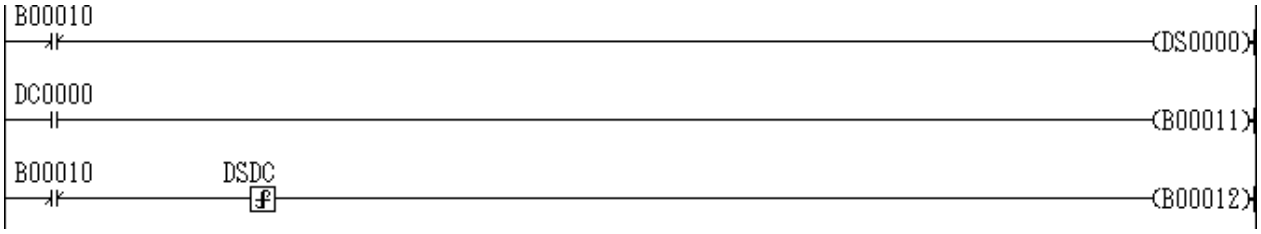


使用例

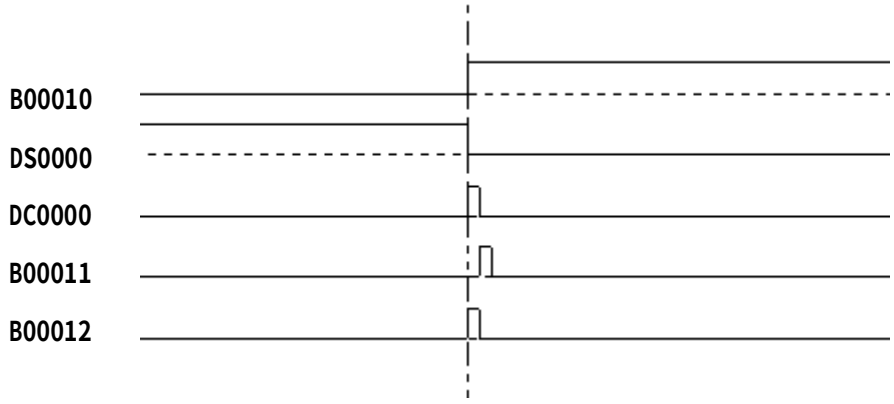


B00000 が ON したら B00001 は 1 スキャン遅れて、1 スキャン分 ON しますが、
B00002 は 1 スキャン遅れず B00000 が ON したらすぐに 1 スキャン分 ON します。





B00010 が ON したら B00011 は 1 スキャン遅れて、1 スキャン分 ON しますが、
B00012 は 1 スキャン遅れず B00010 が ON したらすぐに 1 スキャン分 ON します。



種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	セット(SET) リセット(RESET)	SET F　RESET F	―――――
機能	セット：入力ビットが ON したら指定した出力ビットを ON し続けます。 リセット：入力ビットが OFF したら指定した出力ビットを OFF し続けます。		

セット(SET) ：　注)セットが ON している時は RESET が ON すると引数で設定した接点が OFF します。

関数引数設定内容

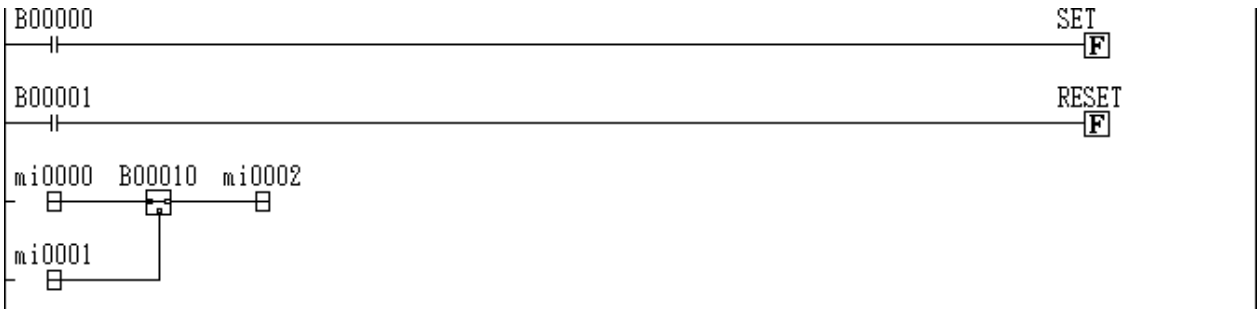
①セットコイル：ON し続けるリレーを指定します。

リセット(RESET)：　注)リセットが ON している時は SET が ON しても引数で設定した接点は ON しません。

関数引数設定内容

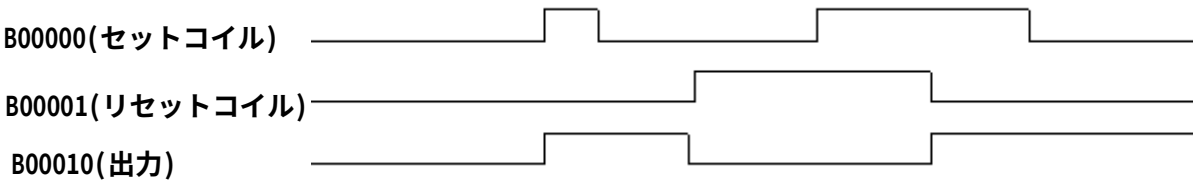
①リセットコイル：OFF し続けるリレーを指定します。

使用例



B00000=ON となると、B00010=ON となり、mi0002 には mi0001 の値がストアされます。

B00001=ON となると、B00010=OFF となり、mi0002 には mi0000 の値がストアされます。



B00000=ON となると、B00010=ON となります。(B00000=OFF となっても B00010=OFF とはなりません)

B00001=ON となると、B00010=OFF となります。(B00000=ON となっても B00010=ON とはなりません)

B00001=OFF となると、B00000=ON となっているので、B00010=ON となります。

種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	カウンタ (UPDOWN)	UPDOWN F	――――――
機能	カウンタ(NR、NP、NU、ND、NZ、nO)を 1 行にまとめたもので動作は同じです。		

関数引数設定内容

①リセットコイル：カウント現在値を 0 にするリレーを設定します。

②プロセットコイル：カウント現在値をカウントプリセット値で設定した値にするリレーを設定します。

③アップコイル：カウント現在値をインクリメントするにするリレーを設定します。

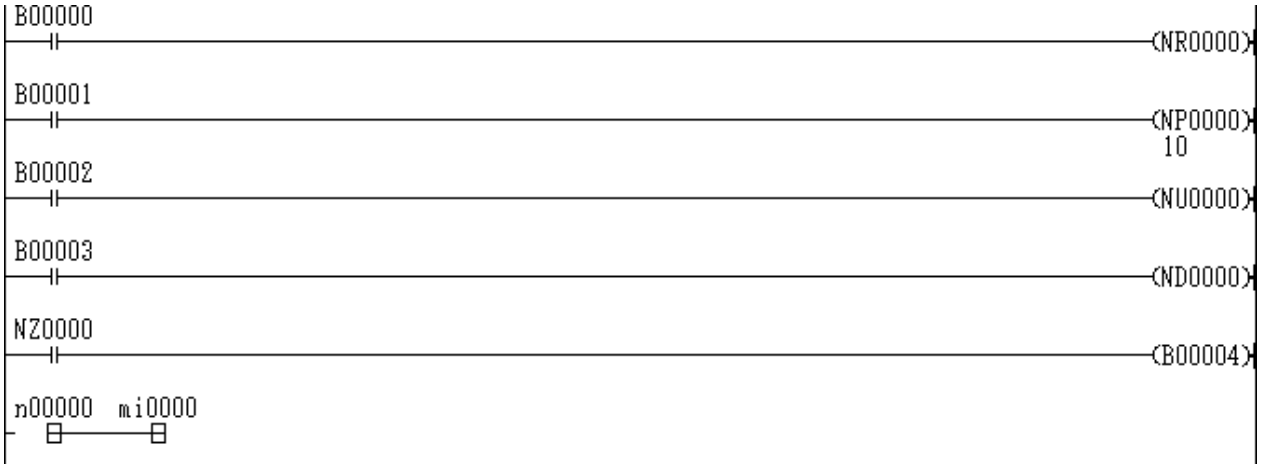
④ダウンコイル：カウント現在値をデクリメントするにするリレーを設定します。

⑤ゼロ検出接点：カウント現在値がゼロになったことを知らせるリレーを設定します。

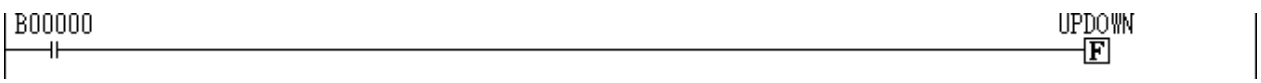
⑥カウント現在値：現在値をストアするレジスタを設定します。

⑦カウントプリセット値：プリセットコイルを ON した時にカウント現在値にセットする値を設定します。

使用例



↓5 行で書いていたのが 1 行になります。



種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	データ転送 (MOVW/MOVWD)	MOVW F　MOVWD F	――――――

機能	指定したデータを指定したラベルへワード単位で転送します。

関数引数設定内容

①転送元ラベル：データを送信する先頭アドレスを指定します。

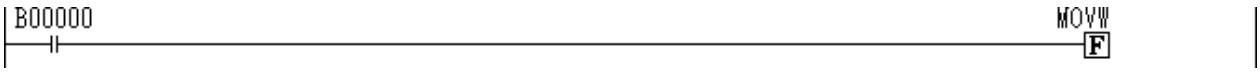
②転送先ラベル：データを受信する先頭アドレスを指定します。

③転送元オフセット：転送元ラベルの何番目からデータを送信するか指定します。（MOVW のみ）

④転送先オフセット：転送先ラベルの何番目からデータを受信するか指定します。（MOVW のみ）

⑤転送個数：送信するデータ個数を指定します。

使用例



右記のように設定すると mi000A から b00004 へデータを 5 ワード分転送します。

mi000A → b00004
mi000B → b00005
mi000C → b00006
mi000D → b00007
mi000E → b00008

MOVW

引数	ラベル	値
転送元ラベル	mi0000	
転送先ラベル	b00000	
転送元オフセット	ki0000	10
転送先オフセット	ki0001	4
転送個数	ki0002	5

種類	名称	シンボル	実行時間
データフロー言語 (関数 3)	整数変換	TODINT	――――――
	実数変換	TOREAL	

機能	指定したデータを指定した型に変換し、結果を出力します。

TODINT(実数入力を 32 ビット整数に変換)

関数引数設定内容

①転送先(2 点使用：偶数アドレス)：入力実数データを 32 ビット整数に変換して出力するアドレスを指定します。

②転送先(2 点使用：偶数アドレス+1)：入力実数データを 32 ビット整数に変換した符号を出力するアドレスを指定します。

TOREAL(32 ビット整数入力を実数に変換)

関数引数設定内容

①転送元(2 点使用：偶数アドレス)：入力 32 ビット整数データを実数に変換して出力するアドレスを指定します。

②転送元(2 点使用：偶数アドレス+1)：入力 32 ビット整数データを実数に変換した符号を出力するアドレスを指定します。

使用例

TODINT の場合、右記のように設定し、入力実数レジスタ mr0000 のデータが(-12.5600)の場合

mi0010 = -13　　mi0011 = -1　　となります。

TODINT

引数	ラベル	値
転送先 (偶数アドレス)	mi0010	
転送先 (偶数アドレス+1)	mi0011	

TOREAL の場合、右記のように設定すると、

mr0011 = 131082　　となります。

mr0011 = ki0000 + ki0001 * 65536

= 10 + 2 * 65536

= 10 + 131072

= 131082

TOREAL

引数	ラベル	値
転送元 (偶数アドレス)	ki0000	10
転送元 (偶数アドレス+1)	ki0001	2

種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	バンク切換 (F_BANK)	F_BANK	――――
機能	FL-net モジュールで使用するブロードキャスト通信領域のデータの同期をとるために使用します。		

関数引数設定内容

①切換対象 SX バス局番(整数):バンク切換対象モジュール(FL-net)の SX バス局番

②ステータス(整数):正常なら 0 が、異常なら以下のエラーコードが入力されます。

64:対象モジュールでない SX バス局番を指定しています。

65:1CPU からのバンク切換要求が多重発生しています。

66:バンク切換処理中、プロセッサバスにアクセスエラー発生しています。

③処理中フラグ(ビット):バンク切換の処理が行われている時 ON します。

④エラーフラグ(ビット):エラー発生時に 1 スキャン ON

注）バンク切換を使用する時はシステム構成定義で FL-net モジュールのパラメータを正しく設定してください。正しく設定しないと正常な動作の保証はできません。

F_BANK 関数の内部の動き

1 スキャン目で処理中フラグが ON します。

2 スキャン目で完了フラグが ON して、その直後に処理中フラグが OFF します。

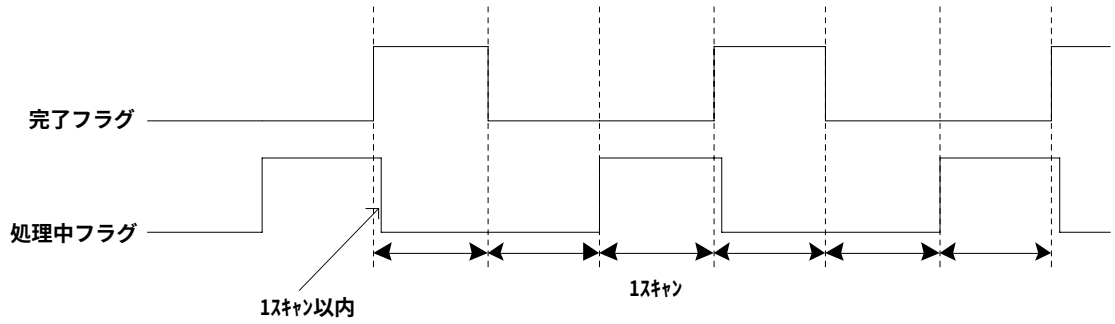
データの送受信を行うときはこの完了フラグが ON している時に行ってください。

3 スキャン目で完了フラグは OFF します。

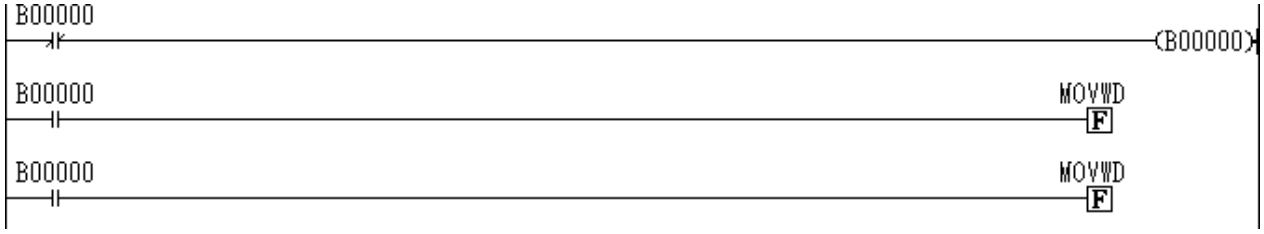
4 スキャン目で処理中フラグが ON します。

あとはこの繰り返しです。

連続したデータの送受信において、実際に渡るデータは 3 スキャンに 1 回の完了フラグが ON した時のデータです。



使用例



右記のように各関数のパラメータを設定します。

F_BANK

引数	ラベル	値
切替対象 SX バス局番	ki0000	7
ステータス	mi0000	
処理中フラグ	B00001	
エラーフラグ	B00002	

B00020 は B 接点なので、最初に F_BANK 関数が実行され処理中フラグが ON します。

次のスキャンで完了フラグ(B00020)が ON するのでこのタイミングでデータの転送(MOVWD)を行ってください。B00020 が ON すると最初は B 接点なので B00020 は OFF し、また初期状態に戻ります。

この繰り返しでデータ転送(MOVWD)を行います。

MOVWD

転送元ラベル	g00000	
転送先ラベル	fi0000	
転送個数	ki0001	10

MOVWD

転送元ラベル	fi0100	
転送先ラベル	g00100	
転送個数	ki0002	10

種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	リモートデータリード		――――――
機能	通信モジュールを経由して、ネットワークに接続されている機器のデータを直接アドレス指定で読み出します。		

関数引数設定内容

①SX バス局番：経由する通信モジュールの SX バス局番

②チャンネル番号：通信モジュールのチャンネル番号

③ノード番号：通信相手のノード番号

④変数指定方式：通信相手先のアクセス対象毎に指定します。(<変数指定方式について>参照)

⑤変数指定先頭アドレス：読み込むデータの種類を指定する先頭のアドレスを指定します。
(<サポートメッセージ一覧>参照)

⑥読み込みデータサイズ：読み込みデータのワードサイズを指定します。

⑦読み込みデータ先頭アドレス：読み込みデータの先頭アドレスを指定します。

⑧エラーフラグ：読み込みが正常に行われなかった時、1 スキャン ON します。

⑨ステータス：エラーフラグの内容を表示します。以下に示します。

細かい内容は、使用例で説明します。

以下はエラーフラグが ON した時に、ステータスに入力される値です。

コード	名称	原因
68	メモリアドレス指定異常	⑤で指定したアドレスに誤りがある場合。
69	メモリサイズオーバ	⑤で指定したアドレス+⑥がアドレスの有効範囲を超えている場合。このときの読み込みデータの値は保障されません。
160	通信相手指定異常	④＝0 の時、通信相手の CPU 番号が存在しない場合。
171	内部資源枯渇	R_READ、R_WRIT を実行するための内部資源の枯渇が発生した場合。または複数個同時に起動した場合に、内部資源枯渇が発生することがあります。この場合は、しばらくたってから再起動してください。
193	チャンネルオープン異常	②に異常な値を設定した場合。
195	メッセージ送信異常	②に異常な値を設定した場合。 ③に異常な値を設定した場合。 メモリ種別に種別コード以外の値を設定した場合。
201	空きポートなし	1 つの通信モジュール内に規定を超えるポートをオープンしようとした時。
206	転送サイズオーバ	変数指定方式に"0"以外を設定した時、経由する通信モジュールのメッセージデータサイズ制限値を超えた場合。

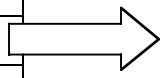
<変数指定方式について>

ここでは、R_READ および R_WRIT の変数指定方式に設定する内容を表示します。変数指定方式は通信相手先のアクセス対象毎に定められています。

①変数指定方式=0 のとき

ネットワーク(種別には依存しません)を経由したμGPCsx システムの CPU 上のメモリを使用するとき指定します。

F	0
CPU 番号	
メモリ種別	
アドレス下位	
アドレス上位	



名前	メモリ種別コード
標準メモリ	1
リテインメモリ	3
ユーザ FB メモリ	5
システム FB メモリ	9
システムメモリ	10

注)　メモリ種別コードに 1、3、5、9、10 以外を指定しないでください。

②変数指定方式=2 のとき

JPCN1 などの OPEN 規格のネットワークに接続されている機器を使用するとき指定します。

F	0
有効サイズ　n	
—	Address　1
—	⋮
—	⋮
—	Addrress　n

注)　この場合は、16 ビット幅の配列の下位 8 ビットに有効データを入力します。
μGPCsx システムでは 8 ビットのデータは扱えないためです。

<サポートメッセージ一覧>

以下に変数指定アドレスに設定するメッセージ伝送のサポートメッセージ一覧を表示します。実際に関数で使用する値は処理コードの要求部分になります。先頭アドレスにパラメータ数を、2 番目に要求コマンドの下位 8 ビット、3 番目に上位 8 ビットを設定します。

NO	メッセージの種類		処理コード(TCD コード　注1)		使用するメッセージ関数	メッセージ データサイズ	パラメータ数 注5)
			要求	応答			
①	バイトブロック読み出し　注2)		65003(FDEB)	65203(FEB3)	R_READ(変数指定方式=2)	476 バイト	6
②	バイトブロック書き込み　注2)		65004(FDEC)	65204(FEB4)	R_WRIT(変数指定方式=2)	476 バイト	6
③	ワードブロック読み出し		65005(FDED)	65205(FEB5)	R_READ(変数指定方式=2)	476 バイト	6
④	ワードブロック書き込み		65006(FDEE)	65206(FEB6)	R_WRIT(変数指定方式=2)	476 バイト	6
⑤	ネットワークパラメータ読み出し		65007(FDEF)	65207(FEB7)	R_READ(変数指定方式=2)	56 バイト	2
⑥	ネットワークパラメータ書き込み		65008(FDF0)	65208(FEB8)	R_WRIT(変数指定方式=2)	20 バイト	2
⑦	停止		65009(FDF1)	65209(FEB9)	R_WRIT(変数指定方式=2)	—	2
	起動		65010(FDF2)	65210(FEBA)	R_WRIT(変数指定方式=2)	—	2
⑧	プロファイル読み出し		65011(FDF3)	65211(FEBB)	R_READ(変数指定方式=2)	480 バイト	2
⑨	通信ログの読み出し		65013(FDF5)	65213(FEBD)	R_READ(変数指定方式=2)	480 バイト	4
⑩	通信ログのクリア		65014(FDF6)	65214(FEBE)	R_WRIT(変数指定方式=2)	—	2
⑪	メッセージ折り返し試験用		65015(FDF7)	65215(FEBF)	R_WRIT(変数指定方式=2)	1024 バイト	2
⑫	透過型メッセージ		00000～59999(0000～EA5F)		M_SEND/M_RECEIVE	1026 バイト　注3)	—
	SX 予約	アドレス読み出し	100(64)	150(96)	R_READ(変数指定方式=0)	—　注4)	—
		アドレス書き込み	101(65)	151(97)	R_WRIT(変数指定方式=0)	—　注4)	—
		ローダコマンド	200(C8)	250(FA)	—	492 バイト	—

注 1)　（　）は 16 進表現です。

注 2)　μGPCsx は byte のデータ型をサポートしておりませんので、相手ノードからの"バイトブロック読み出し"、"バイトブロック書き込み"要求を受け付けることはできません。

注 3)　TCD コードを含んだ値です。

注 4)　最大サイズは各 CPU モジュールの指定するメモリ領域の最大値になります。

注 5)　パラメータ数は変数指定で設定するパラメータの数です。

例)　例えば、ネットワークパラメータの読み込みを使用するときは、変数指定先頭アドレスにおいて 1 番目にパラメータの数(この場合は 2)、2 番目に処理コードの要求部分の値(FDEF)の下位 8 ビット(EF)を設定し、3 番目に上位 8 ビット(FD)を設定します。

<仮想アドレス空間>

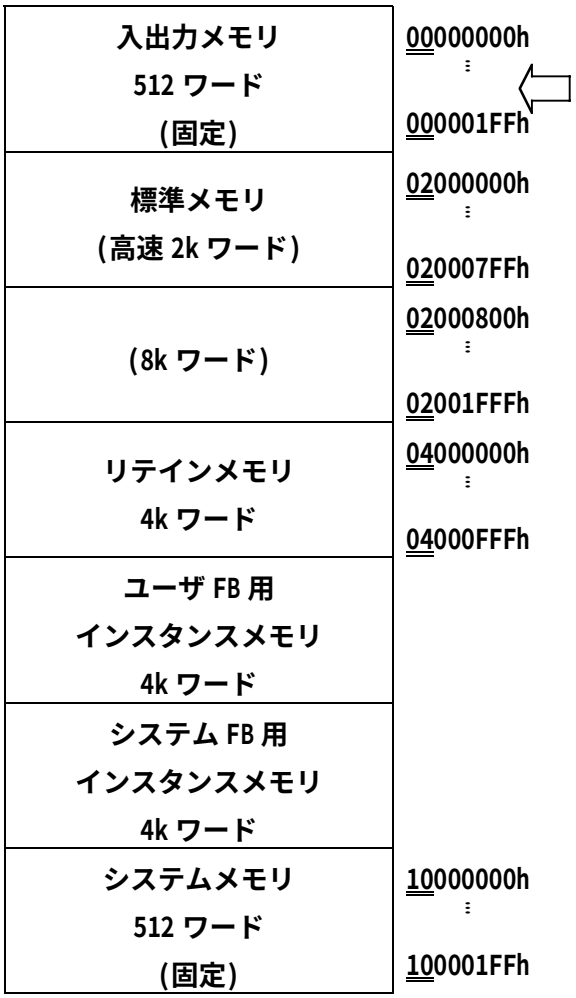
μGPCsx のメモリ	仮想アドレス空間
入出力メモリ	00□□□□□□h～
標準メモリ	02□□□□□□h～
リテインメモリ	04□□□□□□h～
システムメモリ	08□□□□□□h～

高性能 CPU モジュール NP1PS-32 のメモリマップ(デフォルト値)の場合、下図のように各メモリをアクセスします。

メモリマップ

15

0



注 1)入出力の仮想アドレスはシステム構成により異なり、複雑です。簡単にするためにアクセスしたい領域を 1 度標準メモリなどの内部メモリへ転送し、その内部メモリをアクセスされることをお奨めします。

注 2)各メモリのサイズはご使用になる CPU の型式によっても異なります。

注 3)システムメモリへ誤ってデータを書き込んだ場合、誤動作したり重故障で停止する場合があります。(システム動作は保証できません。)

<サポートメッセージ詳細>

① バイトブロック読み出し

FL-net を介して、相手ノードが持つ仮想アドレス空間(32 ビットアドレス空間)に対して、バイト単位(1 アドレス 8 ビット単位)でデータを読み出す機能です。仮想アドレス空間のアドレスマップは、各ノードの仕様を参照してください。(変数指定方式=2、読み出し要求コード=FDEB)

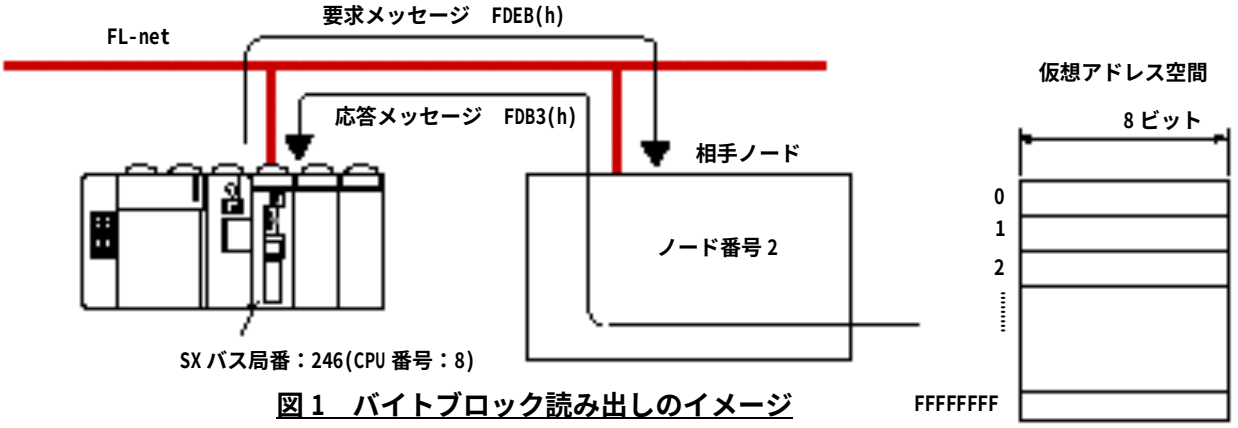


図 1　バイトブロック読み出しのイメージ

<バイトブロック読み出しプログラム例>

ノード番号"2"の FL-net ユニットに接続された CPU の仮想アドレス 00000000(h)から 12 ワード、データを読み出す例です。下に示す変数指定フォーマットの値を変数指定先頭アドレス mi0000 から順に設定していきます。

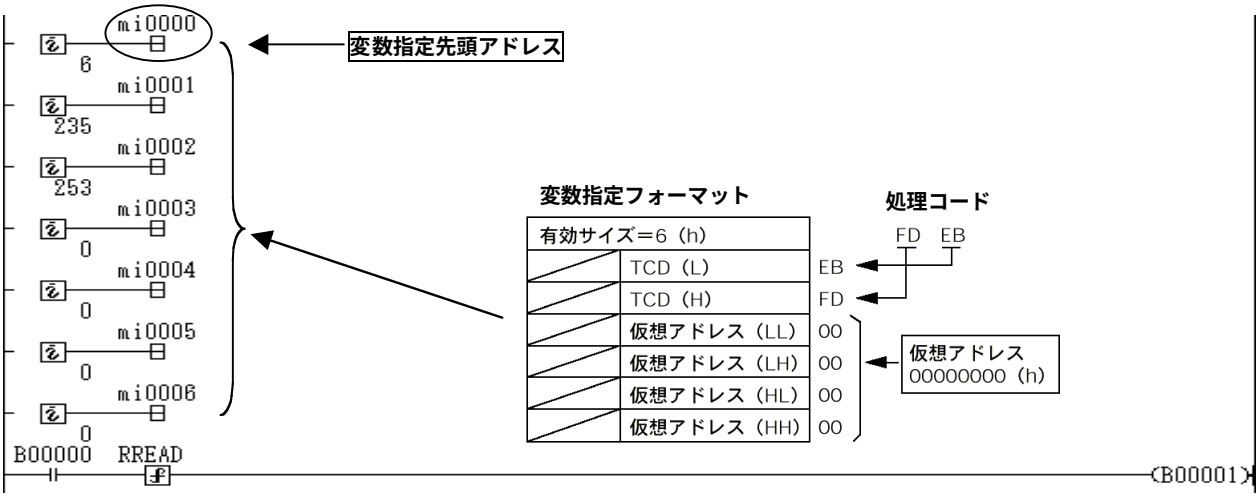


図 2　バイトブロック読み出し回路図と変数指定フォーマットの関係

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	12
読み出しデータ先頭アドレス	b00001	

注 1)NP1L-FL1 のチャンネル番号は"0"固定です。

注 2)読み出しデータのサイズは、

(読み出すデータ量(ワード数)≦(受信データのサイズ)

となるようにしてください。

② ワードブロック読み出し

ネットワークから相手ノードが持つ仮想アドレス空間(32 ビットアドレス空間)に対して、ワード単位(1 アドレス 16 ビット単位)でデータを読み出すメッセージ機能です。仮想アドレス空間のアドレスマップは、各ノードの仕様を参照して下さい。(変数指定方式=2、読み出し要求コード=FDED)

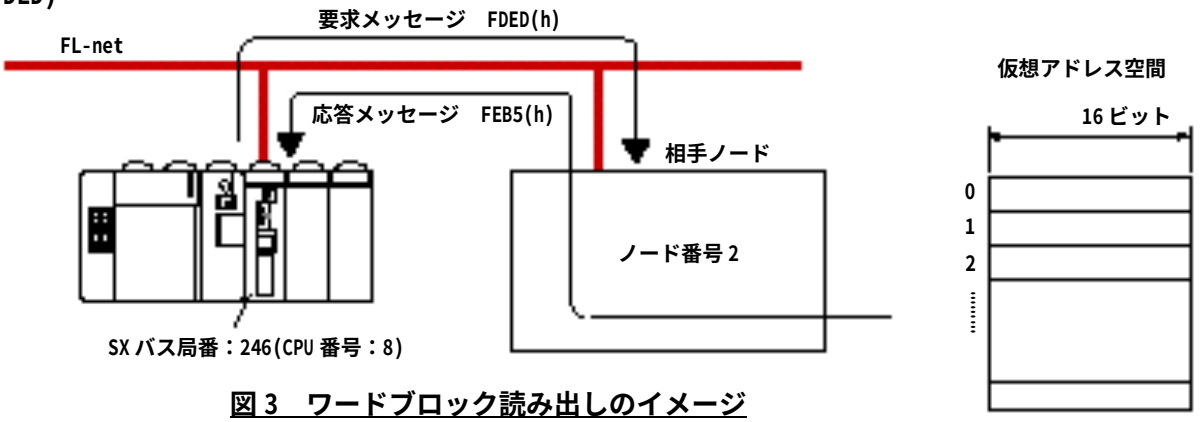


図 3　ワードブロック読み出しのイメージ

<ワードブロック読み出しプログラム例>

ノード番号"2"の FL-net ユニットに接続された CPU の仮想アドレス 00000000(h)から 10 ワード、データを読み出す例です。下に示す変数指定フォーマットの値を変数指定先頭アドレス mi0000 から順に設定していきます。

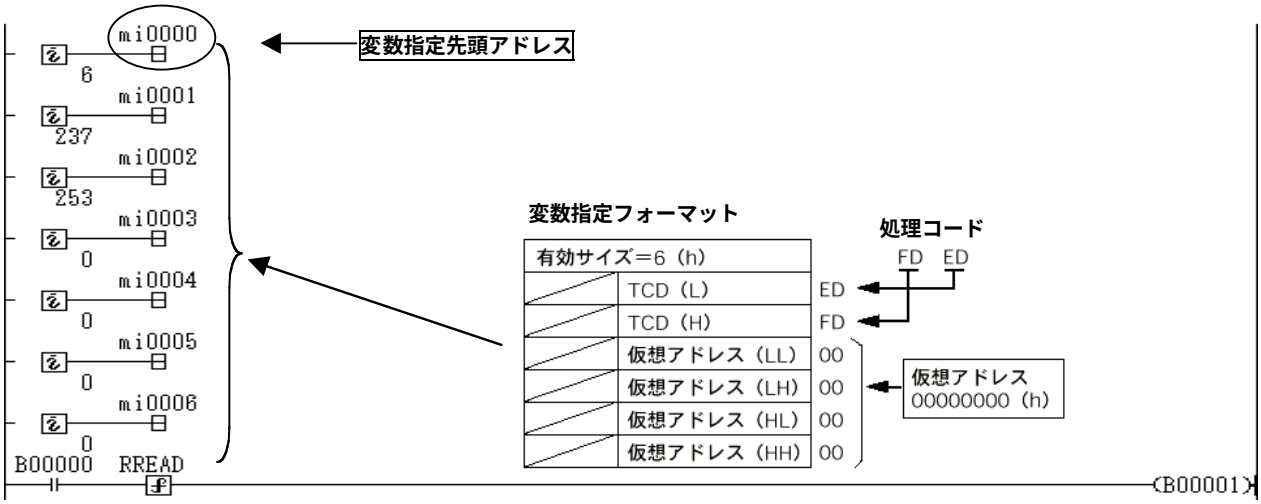


図 4　ワードブロック読み出しの回路図と
変数指定フォーマットの関係

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	10
読み出しデータ先頭アドレス	b00001	

注 1)NP1L-FL1 のチャンネル番号は"0"固定です。

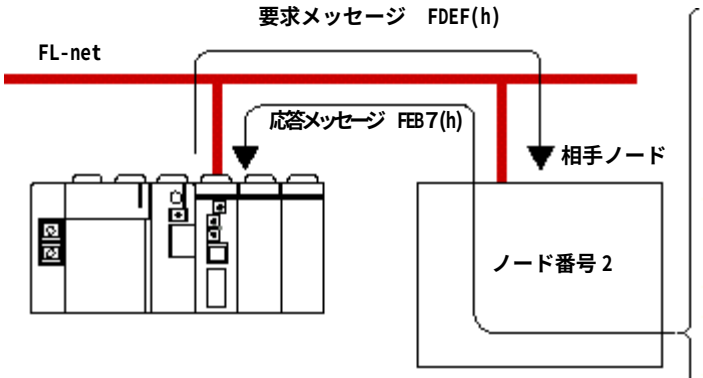
注 2)読み出しデータのサイズは、
(読み出すデータ量(ワード数)≦(受信データのサイズ)
となるようにしてください。

③　ネットワークパラメータ読み出し

ネットワークから相手ノードのネットワークパラメータを読み出すメッセージ機能です。

(変数指定方式=2、読み出し要求コード=FDEF)

ネットワークパラメータ読み出しでは次の情報が読み出されます。



0	ノード	
+4		
+5	未使用	
+9		
+10	メーカ形式	
+14		
+15	コモンメモリ領域 1 の先頭アドレス	
+16	コモンメモリ領域 1 のサイズ	
+17	コモンメモリ領域 2 の先頭アドレス	
+18	コモンメモリ領域 2 のサイズ	
+19	未使用	トークン監視タイムアウト
+20	未使用	最小許容フレーム間
+21	未使用	FA リンクの状態
+22	未使用	プロトコルバージョン
+23	上位層の状態	
+24	リフレッシュサイクル許容時間 RCT 設定値	
+25	リフレッシュサイクル測定値　(現在値)	
+26	リフレッシュサイクル測定値　(最大値)	
+27	リフレッシュサイクル測定値　(最小値)	

図 5　ネットワークパラメータ読み出しのイメージ

<ネットワークパラメータ読み出しプログラム例>

ノード番号"2"の FL-net ユニットのネットワークパラメータを読み出します。

変数指定先頭アドレス mi0000 から順番に変数指定フォーマットのパラメータを入力します。

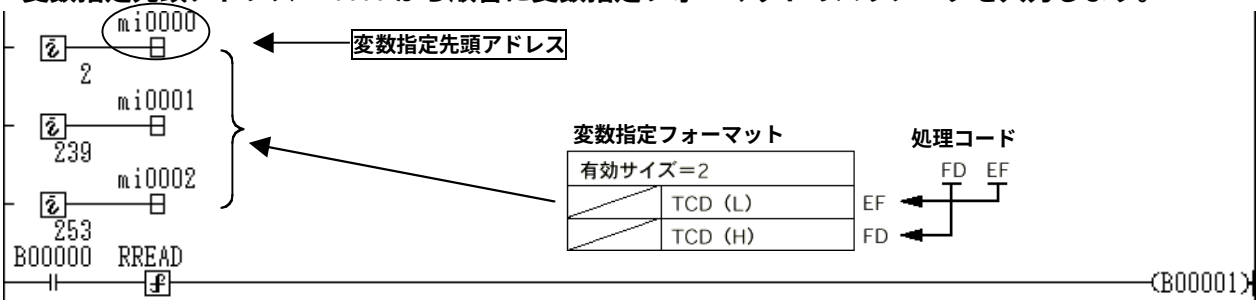


図 6　ネットワークパラメータ読み出しの回路図と変数指定フォーマットの関係

注 1)NP1L-FL1 のチャンネル番号は"0"固定です。

注 2)ネットワークパラメータは 28 ワードあるため、読み出しデータサイズは 28 以上を設定してください。

注 3)FLRAS 関数が参照するコモンメモリとはアドレスが違うので注意してください。

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	28
読み出しデータ先頭アドレス	b00001	

④　プロファイル読み出し

ネットワークから相手ノードのシステムパラメータ(固有の情報)を読み出します。

システムパラメータには 2 つのパラメータがあります。

・共通パラメータ(必須)　→　NP1L-FL1 は共通パラメータのみ用意しています。

・デバイス固有パラメータ(任意)

(変数指定方式=2、読み出し要求コード=FDF3)

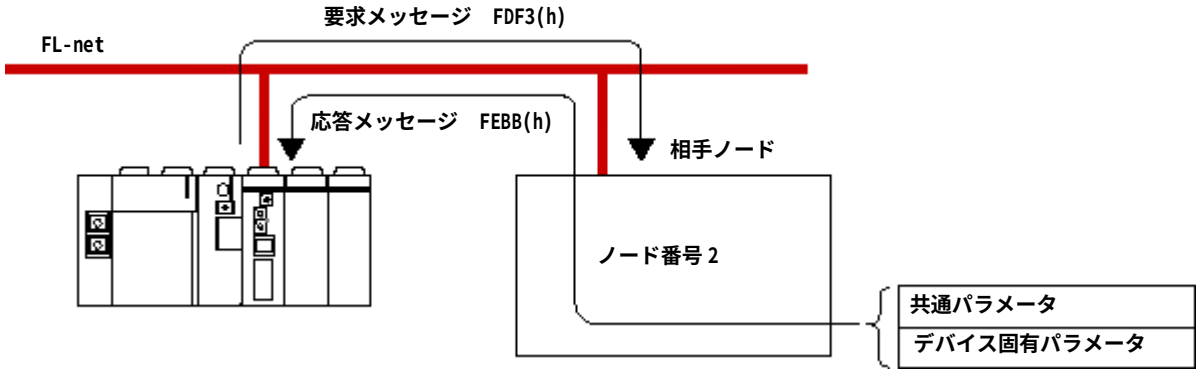


図 7　プロファイル読み出しのイメージ

<プロファイル読み出しプログラム例>

ノード番号"2"の FL-net ユニットのプロファイルを読み出します。

変数指定先頭アドレス mi0000 から順番に変数指定フォーマットのパラメータを入力します。

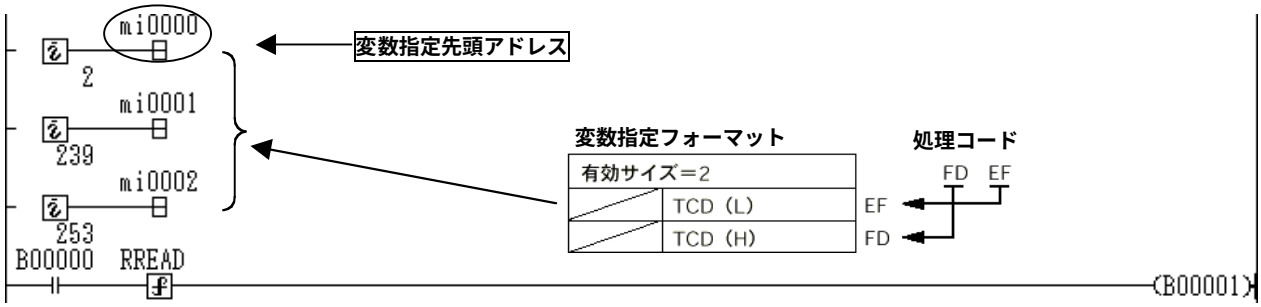


図 8　プロファイル読み出しの回路図と変数指定フォーマットの関係

注 1)NP1L-FL1 のチャンネル番号は"0"固定です。

注 2)読み出しデータのサイズは、

(読み出すデータ量(ワード数)≦(受信データのサイズ)

となるようにしてください。

注 3)読み出しデータサイズについては、相手ノードのプロファイル使用を参照してください。

NP1L-FL1 の場合、プロファイルのサイズは 113 バイトです。

したがって、読み出すワードサイズは、57 ワードと指定します。

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	28
読み出しデータ先頭アドレス	b00001	

⑤　通信ログデータの読み出し

ネットワークから相手ノードのログ情報を読み出す機能です。

(変数指定方式=2、読み出し要求コード=FDF5)

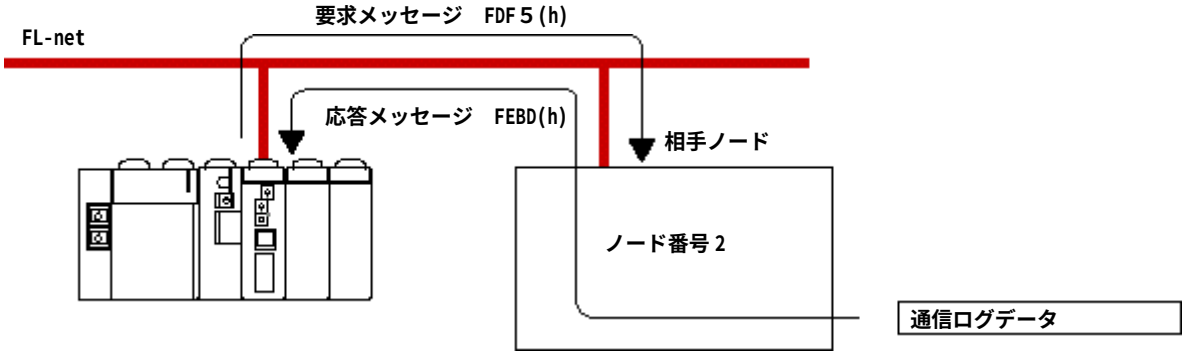


図９　通信ログデータの読み出しのイメージ

<通信ログデータ読み出し引数設定例>

ノード番号"2"の FL-net ユニットの通信ログデータ(512 バイト)を読み出します。

変数指定先頭アドレス mi0000 から順番に変数指定フォーマットのパラメータを入力します。

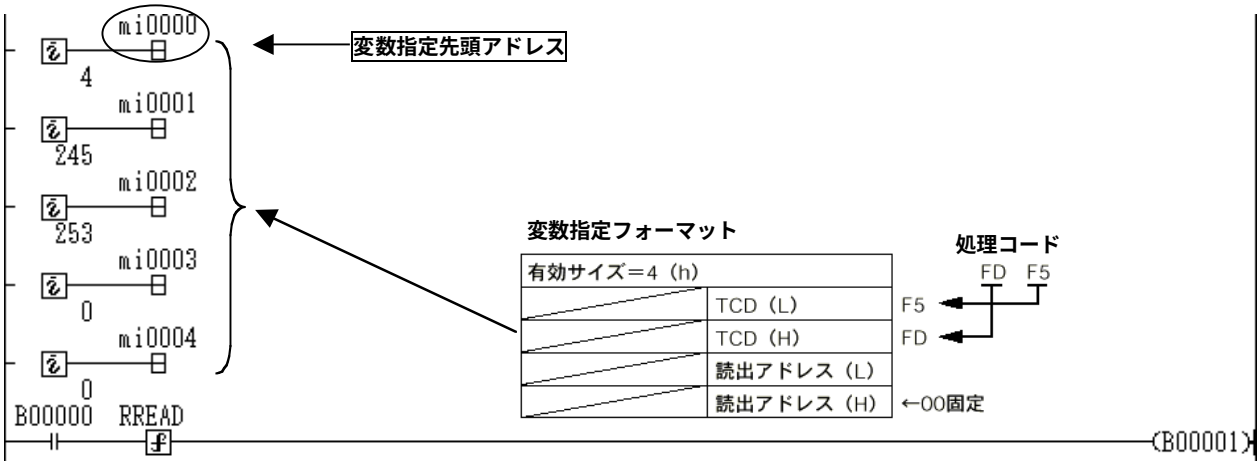


図 10　通信ログデータ読み出しの回路図と変数指定フォーマットの関係

注 1)NP1L-FL1 のチャンネル番号は"0"固定です。

注 2)通信ログデータは、FL-net のどのノードも 512 バイト(固定)となっています。ただし、提供する項目は必須項目と任意項目があります。詳しくは各ノードの仕様を確認してください。
また 1 度に読み出せるデータ量は 239 ワード(480 バイト)です。したがって、2 回に分けて読み出す必要があります。

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	17
読み出しデータ先頭アドレス	b00001	

使用例

下図はネットワークパラメータの読み出しの例です。

1 つのベースに CPU と FL-net モジュールを 1 台ずつ実装し、2 台のベースを FL-net で通信する構成です。

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	48
読み出しデータ先頭アドレス	b00001	
エラーフラグ	G00000	
ステータス	mi0010	

SX バス局番には通信相手の SX バス局番を設定します。(この場合、相手は FL-net モジュールなので ki0000=246 となります。)

チャンネル番号は FL-net モジュールの場合"0"に固定です。

ノード番号は 2 台目の FL-net モジュールなので"2"になります。

変数指定方式には、この場合 FL-net モジュールを使用しているので"2"となります。

変数指定先頭アドレスには、設定するパラメータをどのラベルから読み込むかその先頭ラベルを指定します。(この場合、mi0000。)次に指定したラベルにこの場合、設定するパラメータ数が 2 個なので mi0000=2 とし、サポートメッセージ一覧を参考に該当する要求コマンド 65007(FDEF)の下位 8 ビット 239(EF)を mi0001 に、上位 8 ビット 253(FD)を mi0002 に設定します。

読み出しデータサイズには、ネットワークパラメータの場合 28 ワードあるため、それより小さくならない値を設定します。

読み出しデータ先頭アドレスには、読み込んできたデータをどのラベルから読み込ませるか、その先頭アドレスを指定します。

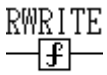
エラーフラグは、読み出しに失敗したときに 1 スキャンだけ ON します。

ステータスは、エラーフラグが ON したときにエラーの内容を表示します。

以上のように設定すると、他のベースボードに実装されているモジュール情報を取得することができます。

注 1)ネットワークパラメータの値は FLRAS 関数と似ていますが、アドレスが違っているので注意してください。

注 2)読み込みと書き込みではデータの内容が違うので注意してください。

種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	リモートデータライト (RWRITE)		――――――
機能	通信モジュールを経由して、ネットワークに接続されている機器にデータを直接アドレス指定で書き込みます。		

関数引数設定内容

①SX バス局番：経由する通信モジュールの SX バス局番

②チャンネル番号：通信モジュールのチャンネル番号

③ノード番号：通信相手のノード番号

④変数指定方式：通信相手先のアクセス対象毎に指定します。(次ページ参照)

⑤変数指定先頭アドレス：書き込むデータの種類を指定する先頭のアドレスを指定します。

⑥書き込みデータサイズ：書き込むデータのワードサイズを指定します。

⑦書き込みデータ先頭アドレス：書き込みデータの先頭アドレスを指定します。

⑧エラーフラグ：書き込みが正常に行われなかった時、1 スキャン ON します。

⑨ステータス：エラーフラグの内容を表示します。以下に示します。

細かい内容は、使用例で説明します。

コード	名称	原因
35	伝送インタロック異常	通信相手のモジュールがインタロックされている場合。伝送インタロックはインスタンス画面を開くと、ダウンロードなどの操作があると行われます。このエラーが発生した場合、リトライしてください。
68	メモリアドレス指定異常	⑤で指定したアドレスに誤りがある場合。
69	メモリサイズオーバ	⑤で指定したアドレス+⑥がアドレスの有効範囲を超えている場合。このときの読み込みデータの値は保障されません。
160	通信相手指定異常	④＝0 の時、通信相手の CPU 番号が存在しない場合。
171	内部資源枯渇	R_READ、R_WRIT を実行するための内部資源の枯渇が発生した場合。または複数個同時に起動した場合に、内部資源枯渇が発生することがあります。この場合は、しばらくたってから再起動してください。
177	パラメータ異常	⑥に 0 が入力された場合。 変数指定方式に指定された値以外の値が入力された場合。 SX バス局番にとりうる値の範囲を超える値が入力された場合。
193	チャンネルオープン異常	②に異常な値を設定した場合。
201	空きポートなし	1 つの通信モジュール内に規定を超えるポートをオープンしようとした時。
206	転送サイズオーバー	変数指定方式に”0”以外を設定した時、経由する通信モジュールのメッセージデータサイズ制限値を超えた場合。

① バイトブロック書き込み

FL-net を介して、相手ノードが持つ仮想アドレス空間(32 ビットアドレス空間)に対して、バイト単位(1 アドレス 8 ビット単位)でデータを書き込む機能です。仮想アドレス空間のアドレスマップは、各ノードの仕様を参照してください。(変数指定方式＝2、書き込み要求コード=FDEC)

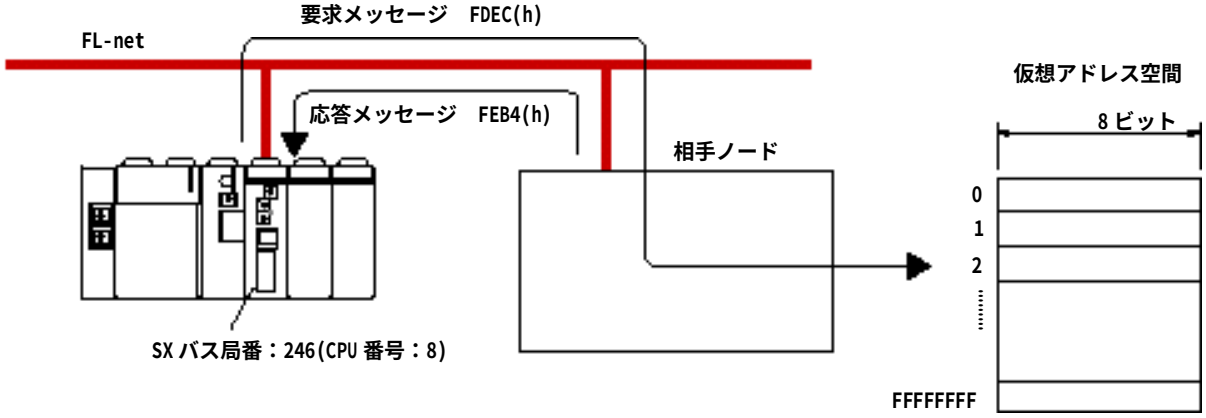


図１　バイトブロック書き込みのイメージ

<バイトブロック書き込みプログラム例>

ノード番号"2"の FL-net ユニットに接続された CPU の仮想アドレス 64(h)から 5 ワード、データを書き込む例です。変数指定先頭アドレス mi0000 から順番に変数指定フォーマットのパラメータを入力します。

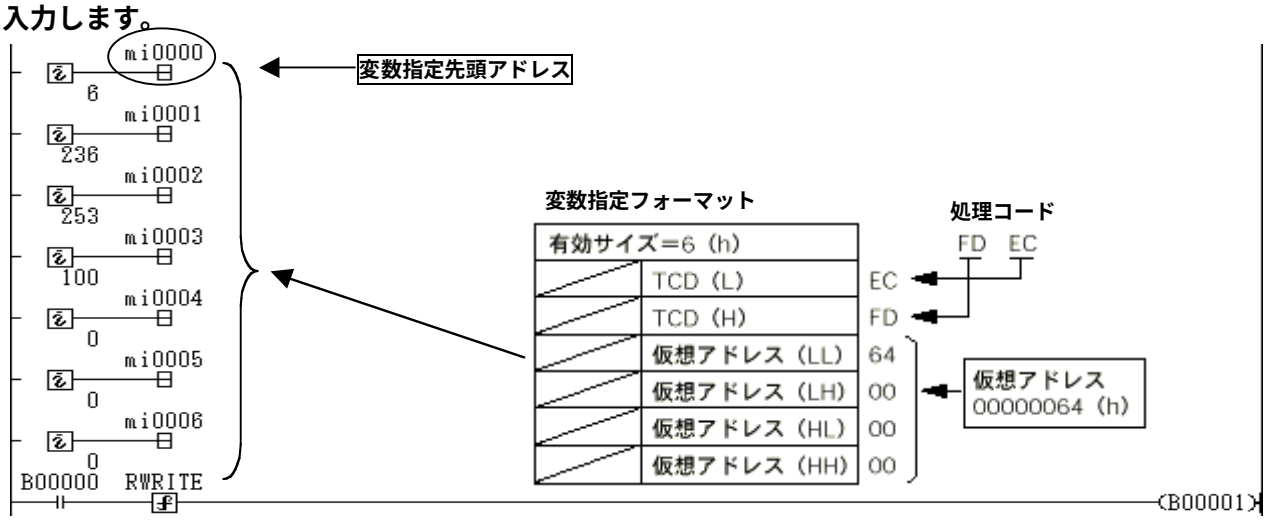


図２　バイトブロック書き込みの回路図と変数指定フォーマットの関係

注 1)NP1L-FL1 のチャンネル番号は"0"固定です。

注 2)読み出しデータ先頭アドレスに指定した b00001 から読み出しデータサイズで指定した 5 ワードまで(b00005)に書き込むデータを入力します。

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	5
読み出しデータ先頭アドレス	b00001	

② ワードブロック書き込み

ネットワークから相手ノードが持つ仮想アドレス空間(32 ビットアドレス空間)に対して、ワード単位(1 アドレス 16 ビット単位)でデータを書き込むメッセージ機能です。仮想アドレス空間のアドレスマップは、各ノードの仕様を参照して下さい。(変数指定方式=2、書き込み要求コード=FDEE)

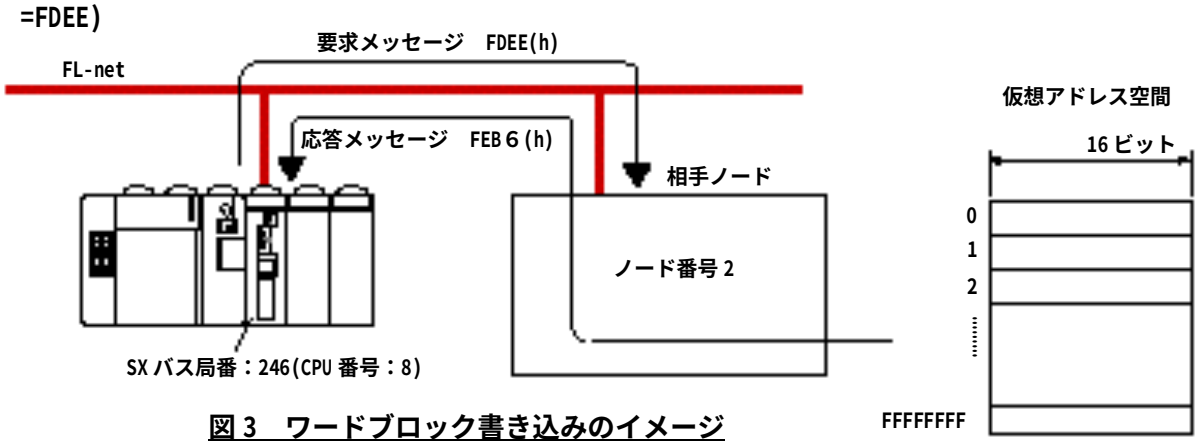


図３　ワードブロック書き込みのイメージ

<ワードブロック書き込みプログラム例>

ノード番号"2"の FL-net ユニットに接続された CPU の仮想アドレス 00000200(h)から 5 ワード、データを書き込む例です。変数指定先頭アドレス mi0000 から順番に変数指定フォーマットのパラメータを入力します。

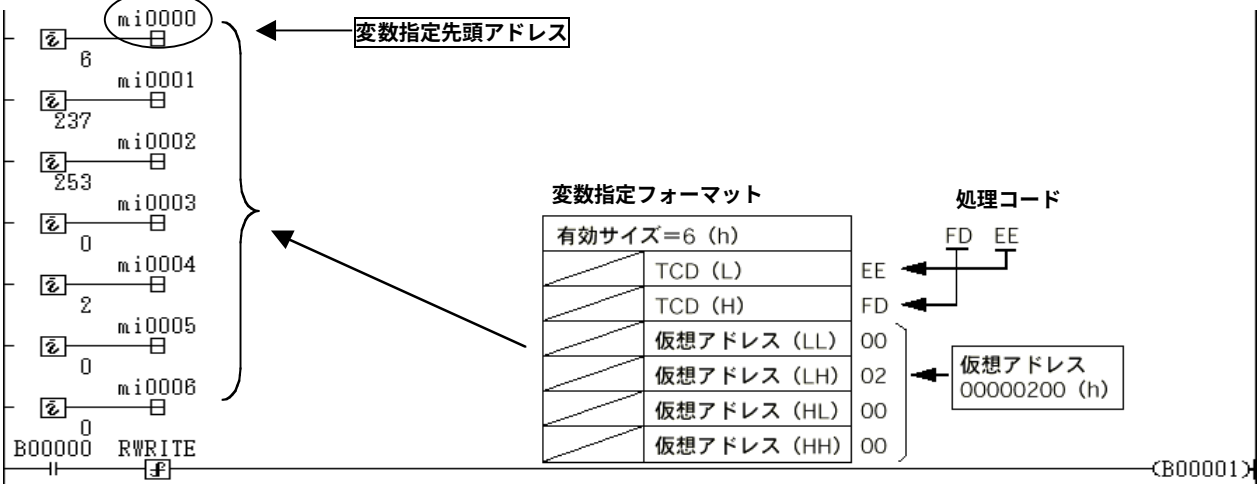


図４　ワードブロック書き込みの回路図と変数指定フォーマットの関係

注 1)NP1L-FL1 のチャンネル番号は"0"固定です。

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	5
読み出しデータ先頭アドレス	b00001	

③　ネットワークパラメータ書き込み

ネットワークから相手ノードのネットワークパラメータ情報を変更する機能です。

次の情報を変更することができます。

・ノード名　　・コモンメモリのアドレスとサイズ

コモンメモリのアドレスとサイズを変更した場合、相手ノードはネットワークから一度離脱し、再加入します。ノード名のみ変更した場合、相手ノードは離脱しません。

(変数指定方式=2、書き込み要求コード=FDF0)

	未使用	設定パラメータ
0	コモンメモリ領域１の先頭アドレス	
	コモンメモリ領域１のサイズ	
	コモンメモリ領域２の先頭アドレス	
+1	コモンメモリ領域２のサイズ	
+2	ノード名(設備名)	
+3		
+4		
+5		
+9		

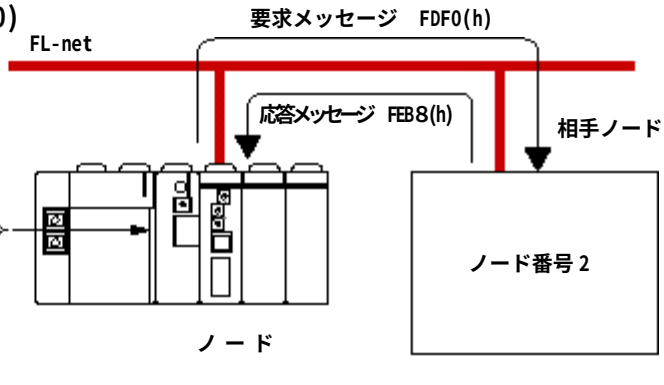


図５　ネットワークパラメータ書き込みのイメージ

注)　設定パラメータ

01（h）：コモンメモリのアドレスとサイズのみ書き込まれます。

02（h）：ノード名のみ書き込まれます。

03（h）：コモンメモリのアドレスとサイズ、ノード名の両方書き込まれます。

<ネットワークパラメータ書き込みプログラム例>

ノード番号"2"の FL-net ユニットのネットワークパラメータを書き込む例です。

変数指定先頭アドレス mi0000 から順番に変数指定フォーマットのパラメータを入力します。

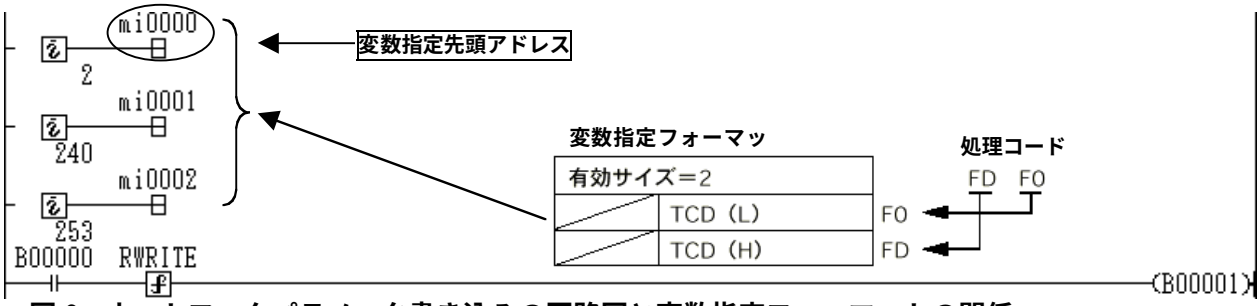


図６　ネットワークパラメータ書き込みの回路図と変数指定フォーマットの関係

注 1)NP1L-FL1 のチャンネル番号は"0"固定です。

注 2)FLRAS 関数が参照するコモンメモリとはアドレスが違うので注意してください。

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	10
読み出しデータ先頭アドレス	b00001	

④　起動・停止指令

ネットワークから相手ノードをリモート起動・停止させる機能です。

(変数指定方式=2、停止要求コード=FDF1／起動要求コード＝FDF2)

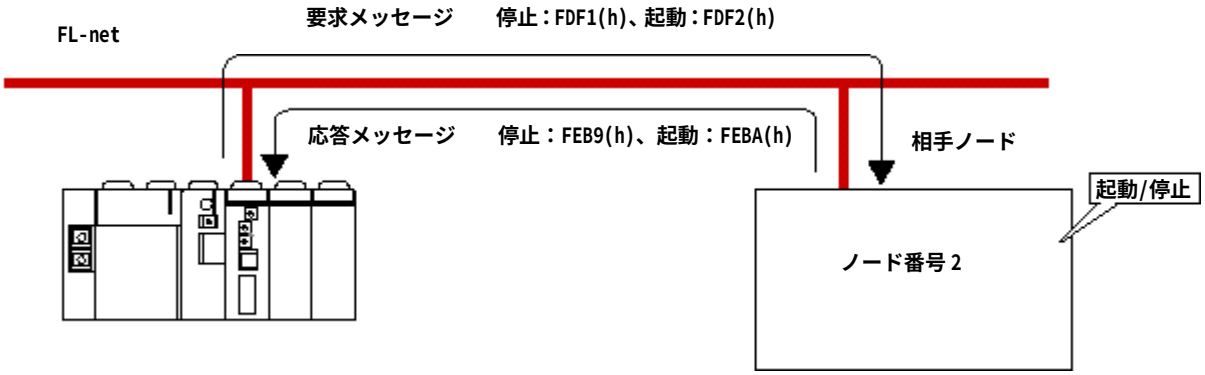


図７　起動・停止指令のイメージ

<停止指令プログラム例>

変数指定先頭アドレス mi0000 から順番に変数指定フォーマットのパラメータを入力します。

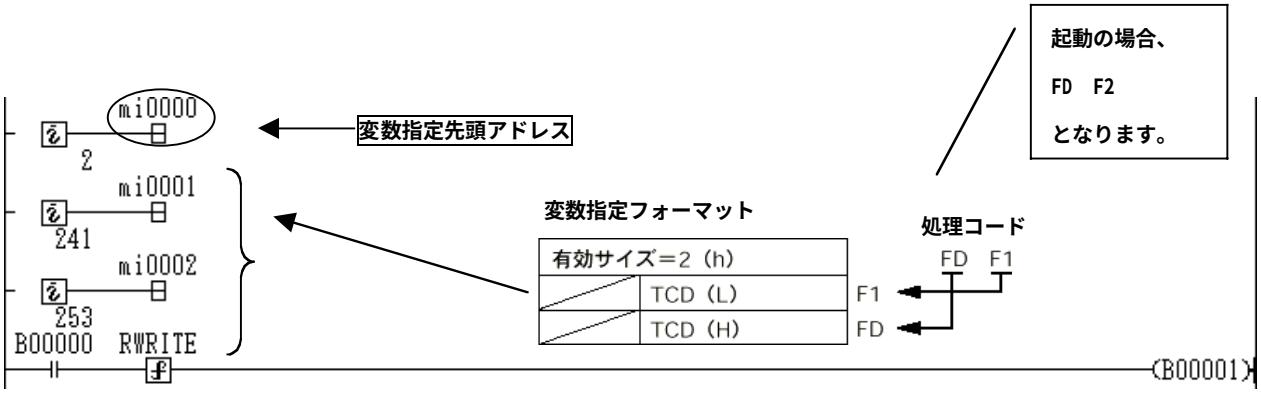


図８　停止指令の回路図と変数指定フォーマットの関係

注 1)NP1L-FL1 のチャンネル番号は"0"固定です。

注 2)実際に書き込むデータはありませんが読み出しデータ先頭アドレスには設定する必要があります。

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	10
読み出しデータ先頭アドレス	b00001	

⑤　通信ログデータのクリア

ネットワークから相手ノードのログ情報をクリアする機能です。

(変数指定方式=2、クリア要求コード=FDF6)

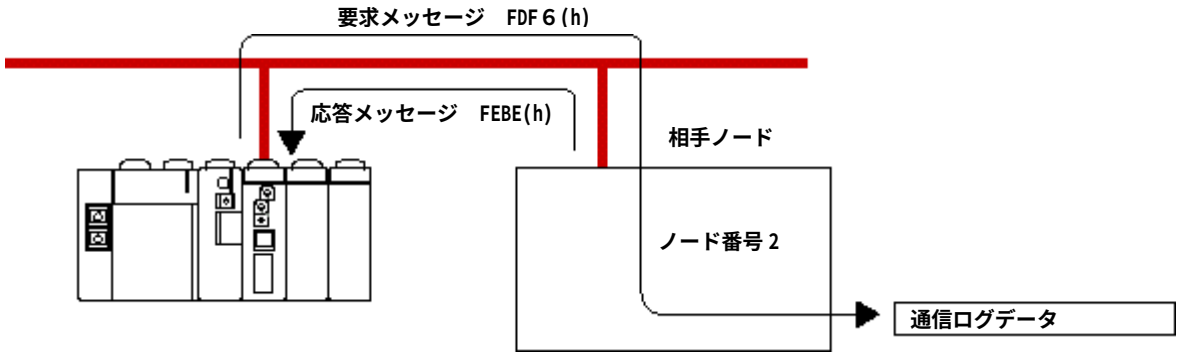


図９　通信ログデータのクリアのイメージ

<通信ログデータのクリアプログラム例>

ノード番号"2"の FL-net ユニットの通信ログデータをクリアします。

変数指定先頭アドレス mi0000 から順番に変数指定フォーマットのパラメータを入力します。

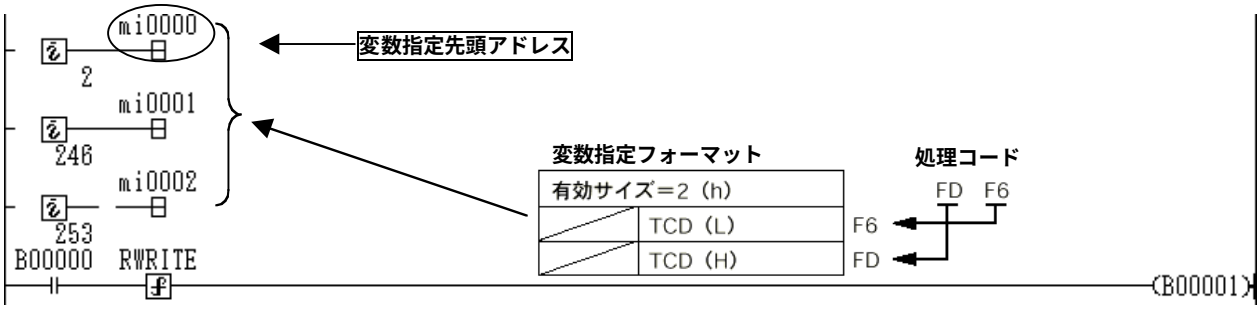


図 10　通信ログデータのクリアの回路図と変数指定フォーマットの関係

注 1)NP1L-FL1 のチャンネル番号は"0"固定です。

注 2)読み出しデータサイズは"10"固定です。

注 3)実際に書き込むデータはありませんが読み出しデータ先頭アドレスには設定する必要があります。

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	10
読み出しデータ先頭アドレス	b00001	

⑥　メッセージ折り返し

受信したメッセージを折り返す機能です。折り返しは FL-net モジュール/ユニット内で自動的に行われます。（変数指定方式=2、折り返し要求コード=FDF7）

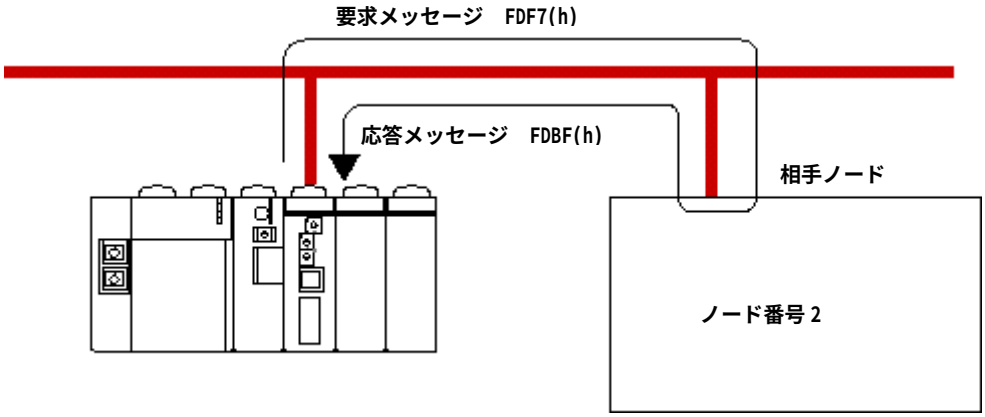


図 11　メッセージ折り返しイメージ

<メッセージ折り返しプログラム例>

ノード番号"2"の FL-net ユニットにメッセージ折り返し要求を出します。
変数指定先頭アドレス mi0000 から順番に変数指定フォーマットのパラメータを入力します。

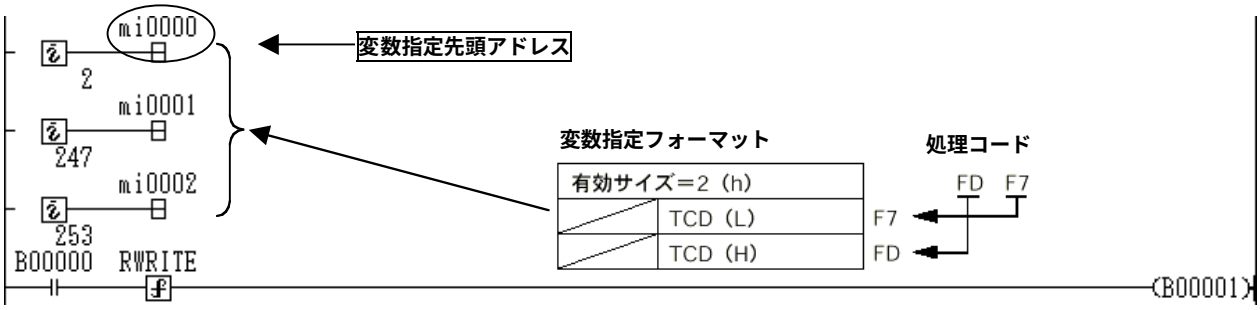


図 12　メッセージ折り返しの回路図と変数指定フォーマットの関係

注 1)読み出しデータサイズは"10"固定です。

注 2)自動的に 512 ワードの送信、受信、照合を行っています。
照合エラーを検出した場合、05(h)のエラーステータスが出ます。

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	10
読み出しデータ先頭アドレス	b00001	

使用例

下図はネットワークパラメータの書き込みの例です。

1 つのベースに CPU と FL-net モジュールを 1 台ずつ実装し、2 台のベースを FL-net で通信する構成です。

SX バス局番には通信相手の SX バス局番を設定します。(この場合、相手は FL-net モジュールなので ki0000=246 となります。)

チャンネル番号は FL-net モジュールの場合"0"に固定です。

ノード番号は 2 台目の FL-net モジュールなので"2"になります。

変数指定方式には、この場合 FL-net モジュールを使用しているので"2"となります。

変数指定先頭アドレスには、設定するパラメータをどのラベルから読み込むかその先頭ラベルを指定します。(この場合、mi0000。)次に指定したラベルにこの場合、設定するパラメータ数が 2 個なので mi0000=2 とし、サポートメッセージ一覧を参考に該当する要求コマンド 65008(FDF0)の下位 8 ビット 240(F0)を mi0001 に、上位 8 ビット 253(FD)を mi0002 に設定します。

引数	ラベル	値
SX バス局番	ki0000	246
チャンネル番号	ki0001	0
ノード番号	ki0002	2
変数指定方式	ki0003	2
変数指定先頭アドレス	mi0000	
読み出しデータサイズ	ki0004	10
読み出しデータ先頭アドレス	b00001	
エラーフラグ	G00000	
ステータス	mi0010	

読み出しデータサイズには、ネットワークパラメータの場合 10 ワードあるため、その値を設定します。

読み出しデータ先頭アドレスには、書き込むデータをどのラベルから読み込ませるか、その先頭アドレスを指定します。

エラーフラグは、読み出しに失敗したときに 1 スキャンだけ ON します。

ステータスは、エラーフラグが ON したときにエラーの内容を表示します。

以上のように設定すると、現在つなっがていないベースのモジュール情報を取得することができます。

注 1)ネットワークパラメータの値は FLRAS 関数と似ていますが、アドレスが違っているので注意してください。

注 2)また読み込みと書き込みではデータの内容が違うので注意してください。

種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	チャンネルオープン		――――――
機能	メッセージの通信相手の設定をするための関数です。この設定は次ページ以降で説明する M_SEND（メッセージ送信）、M_RECV（メッセージ受信）で使われます。通信相手との接続の確認はおこないません。		

関数引数設定内容

①通信 SX バス局番：

コンフィグレーション外通信　→　経由する通信モジュールの SX バス局番

コンフィグレーション内通信　→　通信相手の CPU の SX バス局番

②チャンネル番号：通信モジュール内のチャンネル番号

(複数のチャンネルがある場合は対象チャンネルを、ない場合は"0"を設定します。)

③ステーション番号（L）：通信相手のネットワーク上のステーション番号(下位 16 ビット)

④ステーション番号（H）：通信相手のネットワーク上のステーション番号(上位 16 ビット)

(コンフィグレーション内通信の場合は意味を持ちません)　<引数詳細参照>

⑤モジュール種別番号：

0　→　コンフィグレーション内のモジュールとメッセージ通信

1　→　コンフィグレーション外のモジュールとメッセージ通信

⑥通信モード：コネクションの通信条件を設定します。<引数詳細参照>

⑦通信サブモード：<引数詳細参照>

0　→　相手ノードでの送達確認なしを設定します。

1　→　相手ノードでの送達確認ありを設定します。

⑧送信ポート番号：通信相手のポート番号を設定します。　注 1、2)

⑨受信ポート番号：受信するポート番号を設定します。　注 1、2)

⑩エラーフラグ：オープン処理が異常終了したとき 1 スキャンだけ ON します。

⑪エラーステータス：エラー内容を表示します。<引数詳細参照>

⑫コネクション番号：チャンネルオープン処理が完了したらコネクション番号が割り付けられます。

注 1) SX バス上に本関数で設定できるポート番号は１～127 です。

注 2)コンフィグレーション外通信で経由する通信モジュールが PC カードインターフェースモジュールの場合、システム構成定義の PC カードモジュールのパラメータにおいて自己ポート基準番号/通信相手ポート基準番号で指定された値がオフセット値としてポート番号に加算されます。

<命令の動作>

①入力リレー(B00000)の立ち上がり（OFF→ON）により通信 SX バス局番で指定されたモジュールのオープン処理が開始されます。（オープン処理は 1 スキャンでは終了しません）

②オープン処理が正常完了した場合正常フラグが ON となり、コネクション番号にコネクション番号が出力されます。この状態で M_SEND、M_RECV の使用ができるようになります。

③オープン処理が正常に行われない場合は、エラーフラグが 1 スキャン ON となり、ステータスにエラーコードが出力されます。

④入力リレーを OFF にするとクローズ処理を行います（クローズ処理も 1 スキャンでは終了しません）。

⑤クローズ処理が終了すると正常フラグが OFF になります（クローズ処理は異常終了することはありません）

＜命令の注意事項＞

①オープン方法は受信用の「Passive 方式」と送信用の「Active 方式」があります。通信を行うためには受信用のオープン処理、送信用のオープン処理があります。
送信するためには送信相手先が受信可能状態となっている必要があるため受信用の「Passive 方式」のオープン処理を先に完了しておく必要があります。

②オープン中に入力リレーを ON → OFF にするとクローズ処理を行います。

③クローズ処理の後、再オープンを行うときは、通信相手側を一旦クローズした後、再オープンの処理を行う必要があります。

＜引数詳細＞

1)ステーション番号(L)、(H)―(2 ワード)

通信相手先の IP アドレスを設定します。IP アドレスは、16 進数または 10 進数で設定します。

下位 16 ビットをステーション番号(L)に、上位 16 ビットをステーション番号(H)に設定します。

例)　IP アドレスが 172．16．0．1 のときには、次のように設定します。

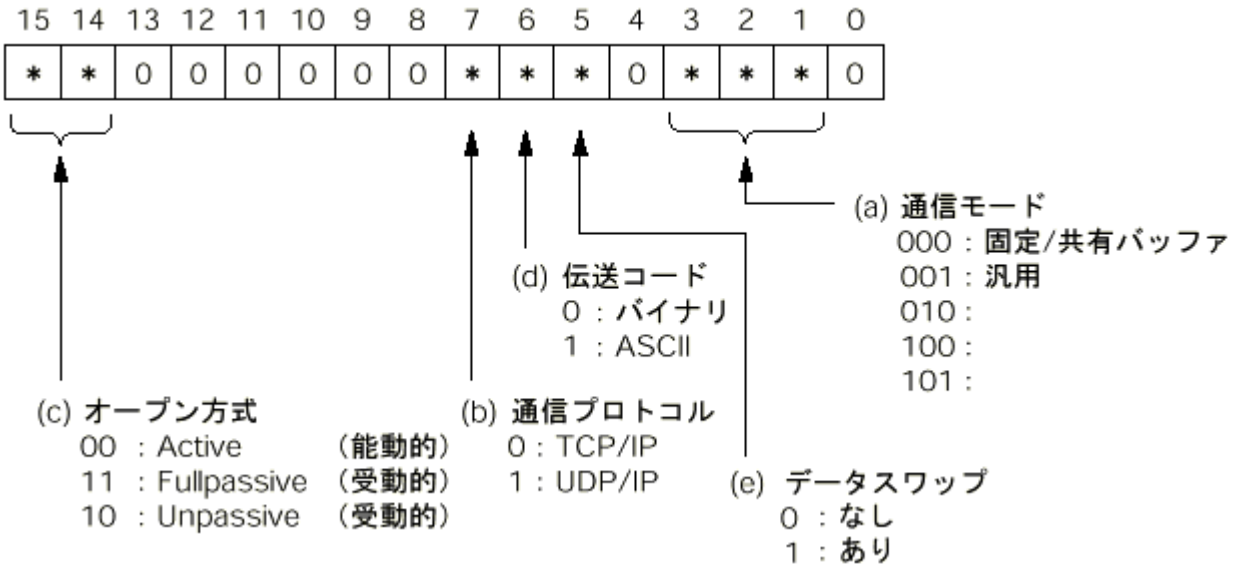
ACh	10h	00h	01h
172	16	0	1

ステーション番号(L)＝0001(h)または１

ステーション番号(H)＝AC10(h)または-21488

2)通信モード

チャンネルオープンするコネクションの通信条件を、ビット情報として 1 ワードのデータにそれぞれ設定します。1 ワードの内容については次のとおりです。



(a)通信モード

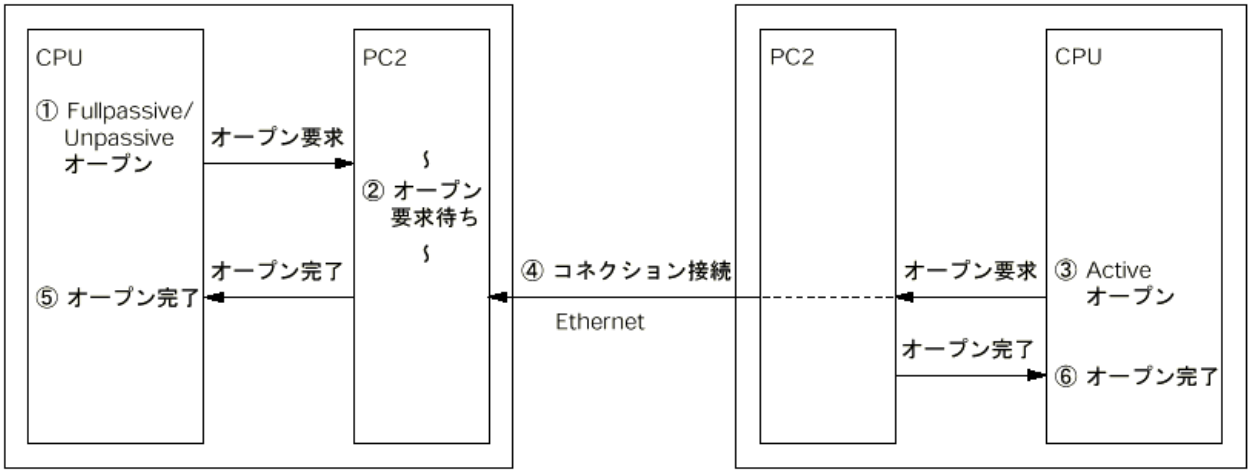
オープンするチャンネルの通信モードを設定します。

(b)通信プロトコル

各コネクション別の通信プロトコルで、TCP/IP を使用するのか、UDP/IP を使用するのかを設定します。

(c)オープン方式

TCP/IP でオープンするときは、Fullpassive/Unpassive オープン(受動的オープン)するノードのオープン処理完了後に、Active オープン(能動的オープン)するノードのオープンを行います。



①Active オープン方式

TCP コネクションのオープン受動状態となっている他ノードに対して能動的なオープン処理を行います。

②Fullpassive オープン方式

通信アドレス設定エリアに設定した特定ノードに対してのみ、TCP コネクションの受動的なオープン処理を行います。通信アドレス設定エリアに設定した他ノードからの Active なオープン要求待ち状態となります。

③Unpassive オープン方式

ネットワークに接続されているすべての他ノードに対して、TCP コネクションの受動的なオープン処理を行います。ネットワーク内のすべての他ノードに対して、Active なオープン要求待ち状態となります。

(d)伝送ノード

他ノードとデータ通信を行う場合のデータコード種別（バイナリ、ASCII）を選択します。

(e)データスワップ

すべての通信モードで伝送コードをバイナリに指定しているときの、伝送データの上位バイト/下位バイトの扱いを反転させます。伝送コードが ASCII の場合、本指定は意味を持ちません。

通信モードのデータ設定例（伝送コードをバイナリとした場合の例）

通信方式＼通信モード		汎用	固定/共有バッファ
TCP	Active	0002h	0000h
	Fullpasive	C002h	C000h
	Unpasive	8002h	8000h
UDP	Active	0082h	0080h
	Fullpasive	C082h	C080h
	Unpasive	8082h	8080h

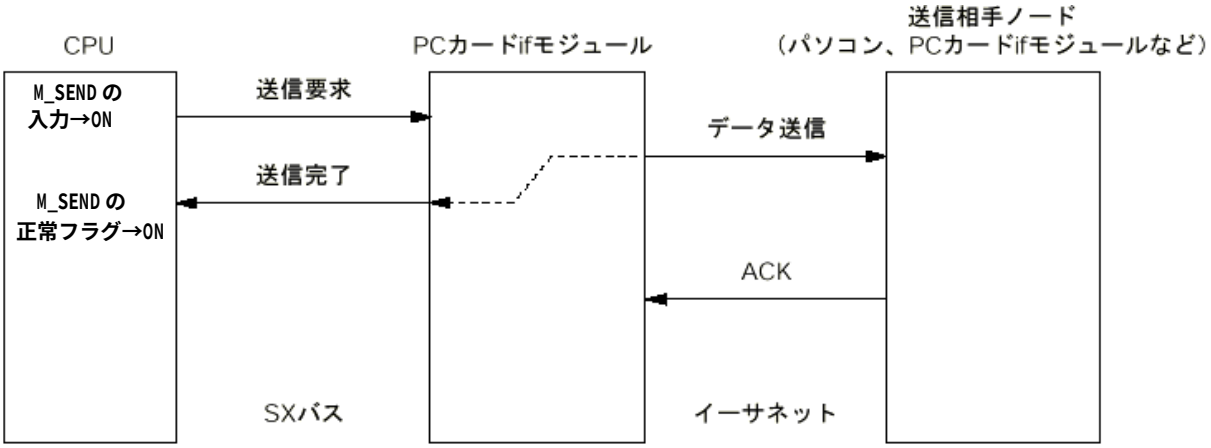
3)通信サブモード

0：相手ノード（相手モジュールもしくは相手ノードアプリケーション）での送達確認なしを設定します。

1：相手ノード（相手モジュールもしくは相手ノードアプリケーション）での送達確認ありを設定します。

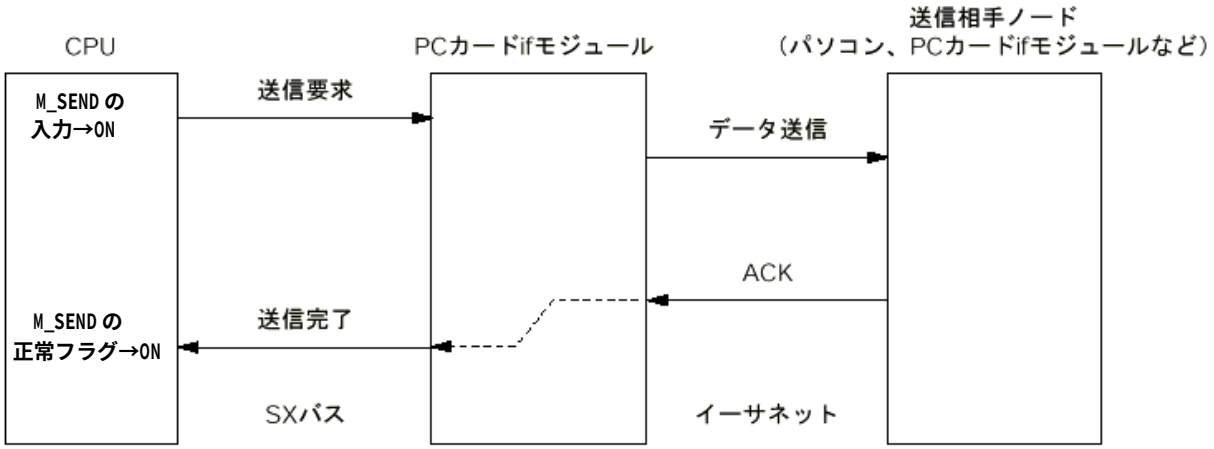
<通信サブモードの動作について＞

①　通信サブモード=0 の時



※ 相手ノードからのACKを待たずにイーサネット上にデータ送信した時点で送信完了となります。

②　通信サブモード=1 の時



※ 相手ノードからのACKを待って送信完了となります。

4)エラーステータス

名称	コード	内容
パラメータ異常	177(B1h)	通信バス局番で指定した局番に、モジュールが存在しない場合、またはモジュール種別番号で指定したコードが、通信モジュールのネットワーク種別と一致しない場合
チャンネルオープン異常	193(C1h)	ステーション番号に異常な値を設定した場合 通信モードの設定に異常な値を設定した場合 通信モードをアクティブ側(送信側)に設定したとき、相手のステーション番号(IP アドレス、送信ポート番号)がネットワーク上にない場合 その他コネクションが確立しなかった場合
ポート指定異常	200(C8h)	受信ポート番号で指定したコードが 1～127 の範囲にない場合 リソース内で既に同じ受信ポート番号が指定されている場合 同じ通信モジュールに、同じ送信ポート番号と受信ポート番号の組み合わせで登録されている場合
コネクション番号・クライアントポート番号 FULL	201(C9h)	リソース内で同時に 57 ポート以上オープンしようとした場合 1 つの通信モジュールに内に規定を超えるポートをオープンしようとした場合

注１)メッセージ関数の共通ステータスは(付録 4)を参照してください。

注 2)具体的な使用例は M_RECV の所でまとめて表示します。

種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	メッセージ送信	M_SEND	――――――
機能	M_OPEN で設定した通信相手にメッセージ送信を行います。		

関数引数設定内容

①コネクション番号：M_OPEN により開設したコネクション番号を設定します。

②送信データ格納変数：送信データが格納されている先頭アドレスを設定します。

③送信データ格納変数サイズ：送信するデータが格納されているデータサイズを設定します。(ワード単位)

④エラーフラグ：メッセージ送信が正常に行われなかったとき、1 スキャン ON します。

⑤ステータス：メッセージ送信が正常に行われなかったとき、その内容を出力します。

＜命令の動作＞

①入力リレーの立ち上がり（OFF→ON）でコネクション番号に設定したコネクション番号のステーションへメッセージ送信を行います（送信処理は 1 スキャンでは終了しません）。

②メッセージ送信が正常に終了すると、正常フラグが 1 スキャンの間 ON します。

③メッセージ送信が正常に行われない場合は、エラーフラグ”が 1 スキャン ON となり、ステータスにエラーコードが出力されます。

＜命令の注意事項＞

①1 回のメッセージ送信で送信可能なデータ量は 1017 ワードです。（汎用通信モード）その他は各モードで確認願います。

②メッセージ送信中（リレー入力の立ち上がりから正常フラグまたはエラーフラグが立ち上がるまで）入力リレーは無効です。

③メッセージ送信中は送信データ格納変数を変更しないでください。変更した場合の送信データは保障されません。

④送信データ格納変数サイズで指定したデータ数が送信データ格納変数で指定した変数サイズを超過する場合、超過分のデータは不定となる場合があります。送信データ格納変数サイズには、必ず指定した変数のサイズを入力してください。

⑤入力リレーには M_OPEN の正常フラグが ON してから ON フラグが入力されるようにプログラムしてください。

＜M_SEND 使用時の注意事項＞

①UDP/IP の汎用通信モードでは送達確認およびフロー制御は行いません。受信側の受信処理が間に合わなくなった場合、受信バッファが一杯になり次に送られてきたデータは破棄されます。したがって、送信側の送信完了数と受信側の受信完了数は不一致になります。また受信バッファが一杯になった場合、バッファ解放に約 10 秒要するため、その間受信動作が停止することがあります。

②Full Passive オープンにて IP アドレス、ポート番号が一致しない相手からオープン要求を受信した場合、1 度コネクション確立してから Full Passive 側が Active 側へクローズ要求を行います。そのため Active 側では、オープン正常完了してデータ送信を行ったときにエラーステータス C7h（強制クローズ）となります。

③送信側のポート番号が受信側と一致しない場合は、送信異常となり送信側から強制クローズが行われ、エラーステータス“C7h：（強制クローズ）”が発生します。

④μGPCsx 同士で通信を行う場合は、連続 1 ワード送信を行うと、受信側では M_RECV のタイミングにより最初に受信した 1 ワードと次に受信した 1 ワードが結合されて 2 ワード受信として CPU に応答を返す場合があります。したがって、送信ワード数が 1 ワードの場合は、受信側のバッファ領域は 2 ワードをとってください。送信ワード数が 2 ワード以上の場合は、受信側のバッファ領域は送信ワード数と同じ数をとります。

⑤UDP/IP の汎用通信モードにて ASCII 変換してデータ送信を行うと、データ数 1019 バイトを超えた場合、送信側は 2 回の分割送信を行います。そのため受信側は 2 回の受信要求が必要となります。また、受信側のバッファ領域は送信データより大きくとる必要があります。

＜エラーステータス＞

名称	コード	内容
パラメータ異常	177(B1h)	送信データ格納変数サイズに 0 が入力された場合
メッセージ送信異常	195(C3h)	通信相手の通信モジュールにメッセージを送信できない場合 通信相手の通信モジュールから応答がない場合 (送信は完了したが送信 ACK が返信されないとき)
チャンネルクローズ	199(C7h)	通信相手がクローズしていた場合 注)このコードを受け取った場合には、1 度該当チャンネルをクローズし、再度オープン要求を行ってください
ポート指定異常	200(C8h)	通信相手がオープンしていない場合
バッファオーバ	206(CEh)	送信データ数が 1017 ワードを超えた場合(汎用通信モード)
コネクション番号異常	207(CFh)	オープンされていないコネクション番号を使用した場合 送信中のコネクション番号で送信しようとした場合(2 つの M_SEND を同じコネクション番号で並列に使用したときに起きます)

種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	メッセージ受信	M_RECV f	――――――
機能	M_OPEN で設定した通信相手にメッセージ受信を行います。		

関数引数設定内容

①コネクション番号：M_OPEN により開設したコネクション番号を設定します。

②受信データ格納変数：受信データが格納されている先頭アドレスを設定します。

③受信データ格納変数サイズ：受信するデータが格納されているデータサイズを設定します。(ワード単位)

④エラーフラグ：メッセージ受信が正常に行われなかったとき、1 スキャン ON します。

⑤ステータス：メッセージ受信が正常に行われなかったとき、その内容を出力します。

＜命令の動作＞

①入力リレーの立ち上がり（OFF→ON）でコネクション番号に設定したコネクション番号のステーションからメッセージ受信を行います（受信処理は 1 スキャンでは終了しません）。

②メッセージ受信が正常に終了すると、正常フラグが 1 スキャン ON します。

③メッセージ受信が正常に行われない場合は、“ ERROR ”が 1 スキャンの間“ 1 ”となり、“ STATUS ”にエラーコードが出力されます。

＜命令の注意事項＞

①1 回のメッセージ送信で送信可能なデータ量は 1017 ワードです。（汎用通信モード）その他は各モードで確認願います。

②メッセージ受信中（入力リレーの立ち上がりから正常フラグまたはエラーフラグが立ち上がるまで）入力リレーは ON に保持してください。入力リレーを OFF にすることは受信一時中断を意味します。

③受信一時中断後、入力リレーを立上げる（OFF→ON）と受信を再開します。このときにコネクション番号、受信データ格納変数、受信データ格納変数サイズを変更しても、中断前の入力値で再開します。変更はメッセージ受信処理には反映されません。

④メッセージ受信処理の終了後、次のスキャンでも入力リレーが ON に保持されていると、新たなメッセージ受信処理を開始します。

⑤受信処理中は受信データ格納変数を保持しておいてください。書き換えた場合の受信メッセージデータは保障されません。

⑥受信データ格納変数サイズで指定した数が受信データ格納変数で指定した変数のサイズを超過する場合、他の変数領域を書き換えてしまう場合があります。受信データ格納変数サイズには、必ず指定した変数サイズを入力してください。

⑦入力リレーには M_OPEN の正常フラグが ON になってから入力されるようにプログラムしてください。

＜M_RECV 使用時の注意事項＞

M_SEND と同様です。「M_SEND 使用時の注意事項」を参照してください。

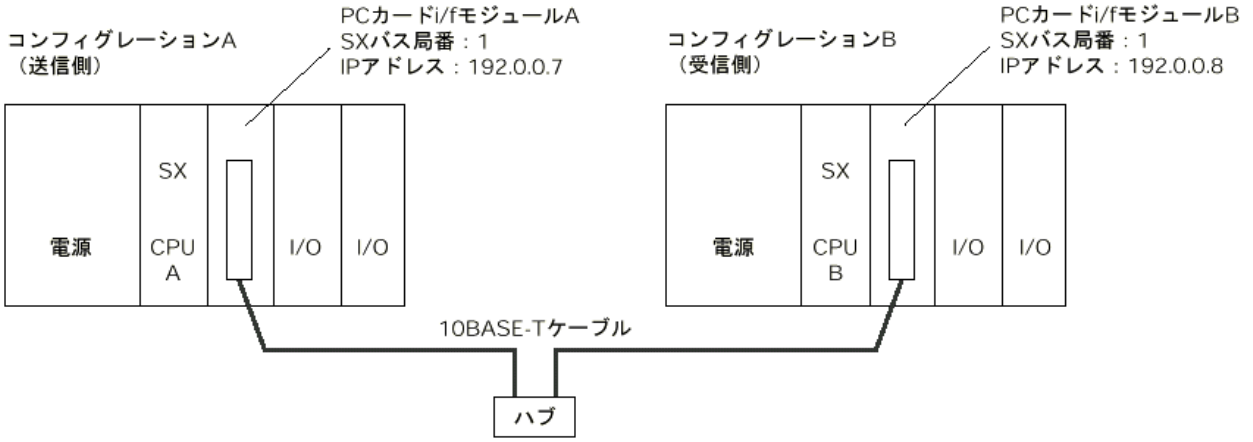
＜エラーステータス＞

名称	コード	内容
パラメータ異常	177(B1h)	受信データ格納変数サイズに 0 が入力された場合
チャンネルクローズ	199(C7h)	通信相手がクローズしていた場合 注)このコードを受け取った場合には、1 度該当チャンネルをクローズし、再度オープン要求を行ってください
ポート指定異常	200(C8h)	通信相手がオープンしていない場合
バッファオーバ	206(CEh)	指定した受信データサイズを超えてデータ受信した場合 このとき、受信データ格納変数には有効受信データが格納されています
コネクション番号異常	207(CFh)	オープンされていないコネクション番号を使用した場合 受信中のコネクション番号で受信しようとした場合（2 つの M_RECV を並列に使用したときに起こります）

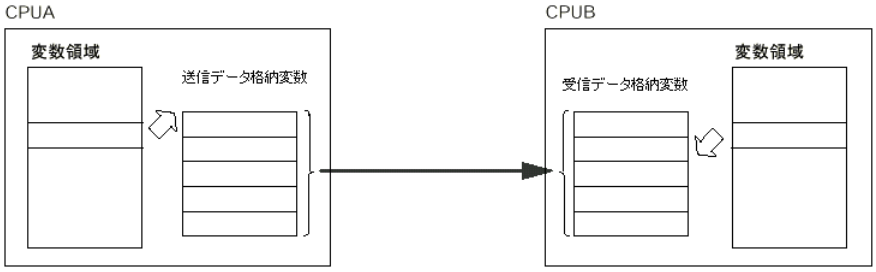
＜プログラム使用例１＞

下図のようなシステムにおいて、チャネルのオープン“M_OPEN”、メッセージ送信“M_SEND”、メッセージ受信“M_RECV”命令を使用して、A→B へデータを送る場合のプログラム例を紹介します。

＜構成図＞



＜プログラムのイメージ図＞



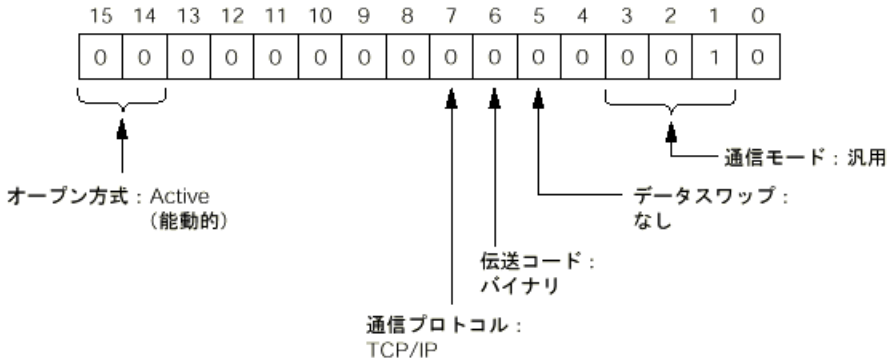
プログラムの内容

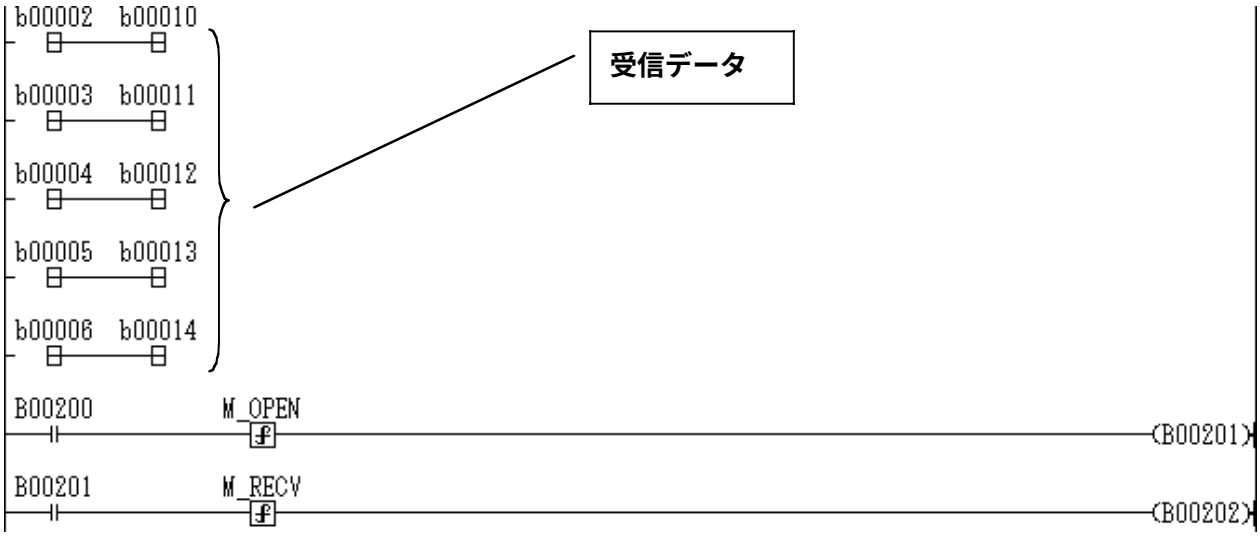
（1）受信側 CPU B の“M_OPEN”を実行しチャネルを開きます。

（2）受信側 CPU B の“M_RECEIVE”を実行し受信待ち状態にします。

（3）送信側 CPU A の“M_OPEN”を実行しチャネルを開きます。

（4）送信側 CPU A の“M_SEND”を実行し CPU B へデータを送ります。





(1) M_OPEN の引数

通信 SX バス局番には通信相手の SX バス局番"1"を設定します。

チャンネル番号は"0"固定です。

ステーション番号は相手の IP アドレス"192.0.0.9"を 16 進数に変換し"C000"を(H)に、"0009"を(L)に設定します。

モジュール種別番号はコンフィグレーション外通信なので"0"を設定します。

通信モードは前ページを参考に設定します。

サブモードは相手ノードでの送達確認ありを設定します。

送信ポート番号は受信ポート番号と重ならないように設定します。

エラーフラグ B00000 には M_OPEN 関数を実行したときエラーになれば ON します。その結果がステータスに出力されます。

コネクション番号はオープン処理が成功したとき番号が割り付けられます。

M_OPEN

引数	ラベル	値
通信 SX バス局番	ki0000	1
チャンネル番号	ki0001	0
ステーション番号(L)	ki0002	0009(H)
ステーション番号(H)	ki0003	C000(H)
モジュール種別番号	ki0004	1
通信モード	ki0005	2
サブモード	ki0006	1
送信ポート番号	ki0007	1
受信ポート番号	ki0008	2
エラーフラグ	B00000	
ステータス	mi0000	
コネクション番号	mi0001	

(2)M_RECV の引数

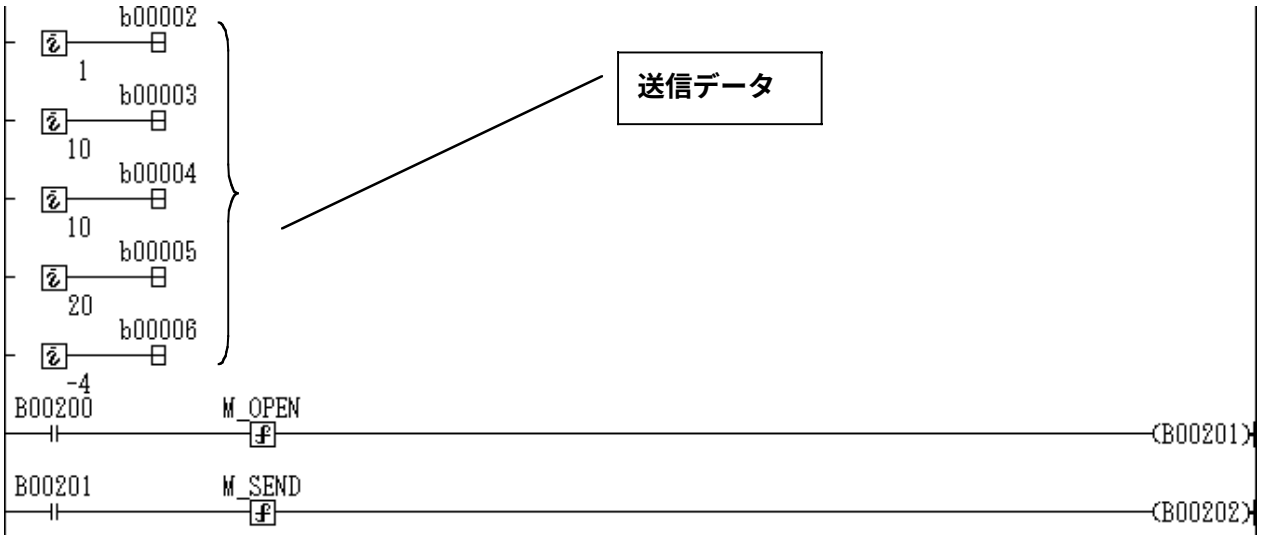
コネクション番号は M_OPEN で取得したコネクション番号をそのまま使用します。

受信データ格納変数は受信するラベルの先頭アドレスを設定し、受信データ格納変数サイズは受信するデータのワード数を設定します。

エラーフラグ B00010 には M_RECV を実行したときにエラーになれば ON します。その結果がステータスに出力されます。

M_RECV

コネクション番号	mi0001	
受信データ格納変数	b00002	
受信データ格納変数サイズ	ki0010	5
エラーフラグ	B00010	
ステータス	mi0010	



(3)M_OPEN の引数

通信 SX バス局番には通信相手の SX バス局番"1"を設定します。

チャンネル番号は"0"固定です。

ステーション番号は相手の IP アドレス"192.0.0.8"を 16 進数に変換し"C000"を(H)に、"0008"を(L)に設定します。

モジュール種別番号はコンフィグレーション外通信なので"0"を設定します。

通信モードは前ページを参考に設定します。

サブモードは相手ノードでの送達確認ありを設定します。

送信ポート番号は受信ポート番号と重ならないように設定します。

エラーフラグ B00000 には M_OPEN 関数を実行したときエラーになれば ON します。その結果がステータスに出力されます。

コネクション番号はオープン処理が成功したとき番号が割り付けられます。

M_OPEN

引数	ラベル	値
通信 SX バス局番	ki0000	1
チャンネル番号	ki0001	0
ステーション番号(L)	ki0002	0008(H)
ステーション番号(H)	ki0003	C000(H)
モジュール種別番号	ki0004	1
通信モード	ki0005	2
サブモード	ki0006	1
送信ポート番号	ki0007	2
受信ポート番号	ki0008	1
エラーフラグ	B00000	
ステータス	mi0000	
コネクション番号	mi0001	

(4)M_SEND の引数

コネクション番号は M_OPEN で取得したコネクション番号をそのまま使用します。

送信データ格納変数は送信するラベルの先頭アドレスを設定し、送信データ格納変数サイズは送信するデータのワード数を設定します。

エラーフラグ B00010 には M_SEND を実行したときにエラーになれば ON します。その結果がステータスに出力されます。

M_SEND

コネクション番号	mi0001	
送信データ格納変数	b00002	
送信データ格納変数サイズ	ki0010	5
エラーフラグ	B00010	
ステータス	mi0010	

(5)

以上（1）から（4）まで無事エラーが表示されば CPU　A から CPU　B へとデータ渡ります。

もしデータが何もこなければ、関数の引数で設定したパラメータの値にミスがあると考えられます。もう一度見直してください。

＜プログラム使用例 2＞

・透過型メッセージ伝送

透過型のメッセージを受信すると、FL-net モジュール/ユニットは、受信したメッセージを FL-net 上位層へ通知し、通知を受けた FL-net 上位層は、ユーザインタフェースレベルへそのまま通知します。ユーザインタフェースレベルへ通知された場合、アプリケーションプログラムなどにより、対応する応答を返す必要があります。μGPCsx では M_SEND/M_RECV 関数を使用します。

また、使用機器により透過型メッセージに固有のサービスを提供している場合があります。

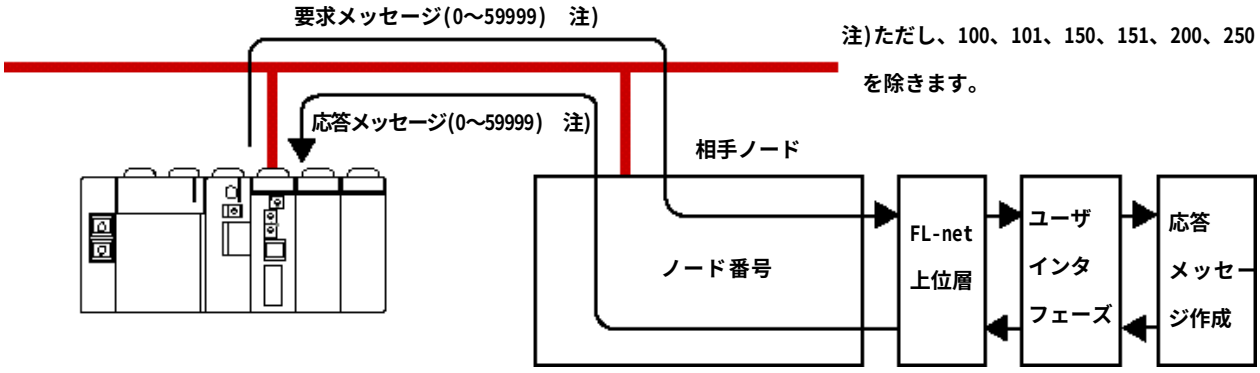


図１　透過型メッセージ折り返しイメージ

＜透過型メッセージの送信プログラム例＞

ノード番号”2”の FL-net ユニットにメッセージ折り返し要求を出します。

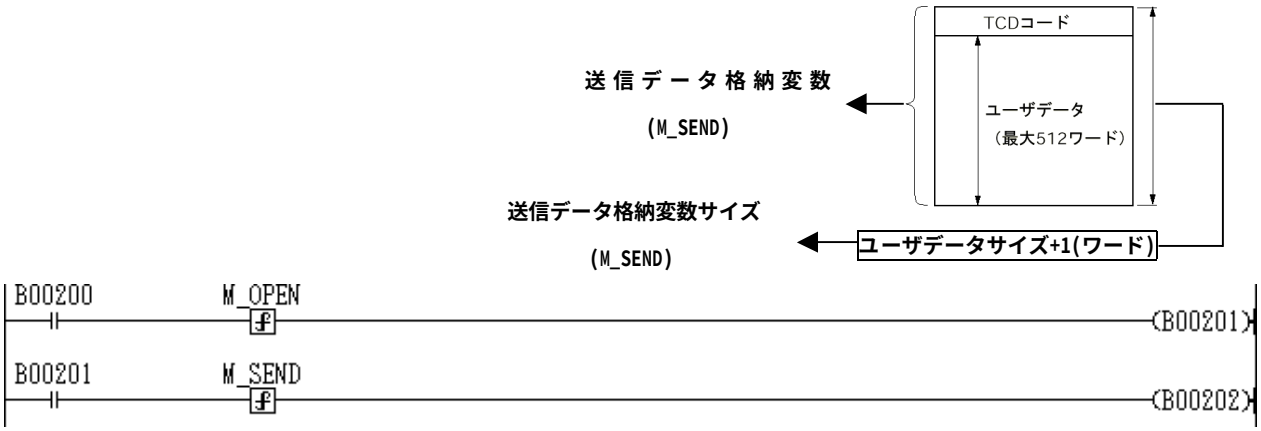


図 2　透過型メッセージの送信プログラムと変数指定フォーマットの関係

注 1)通常“３”（送受信共用オープン）を指定してください。同じノードに対して複数のオープン要求はできません（動作は保証されません）。ただし、1：送信専用オープンと 2：受信専用オープンでオープンすることは可能です。

1：送信専用オープン
2：受信専用オープン
3：送受信共用オープン
その他は使用できません。

注 2)送信ポート番号、受信ポート番号は 1～127 をつけることができます。他の M_OPEN_関数が使用しているポート番号と重ならないようにしてください。

M_OPEN

引数	ラベル	値
通信 SX バス局番	Ki0000	246
チャンネル番号	Ki0001	0
ステーション番号(L)	Ki0002	0002(H)
ステーション番号(H)	Ki0003	0000(H)
モジュール種別番号	Ki0004	1
通信モード　注 1)	Ki0005	3
サブモード	Ki0006	0
送信ポート番号　注 2)	Ki0007	2
受信ポート番号　注 2)	Ki0008	1
エラーフラグ	B00000	
ステータス	mi0000	
コネクション番号	mi0001	

M_SEND

コネクション番号	mi0001	
受信データ格納変数	b00002	
受信データ格納変数サイズ	Ki0010	5
エラーフラグ	B00010	
ステータス	mi0010	

＜透過型メッセージの受信プログラム例＞

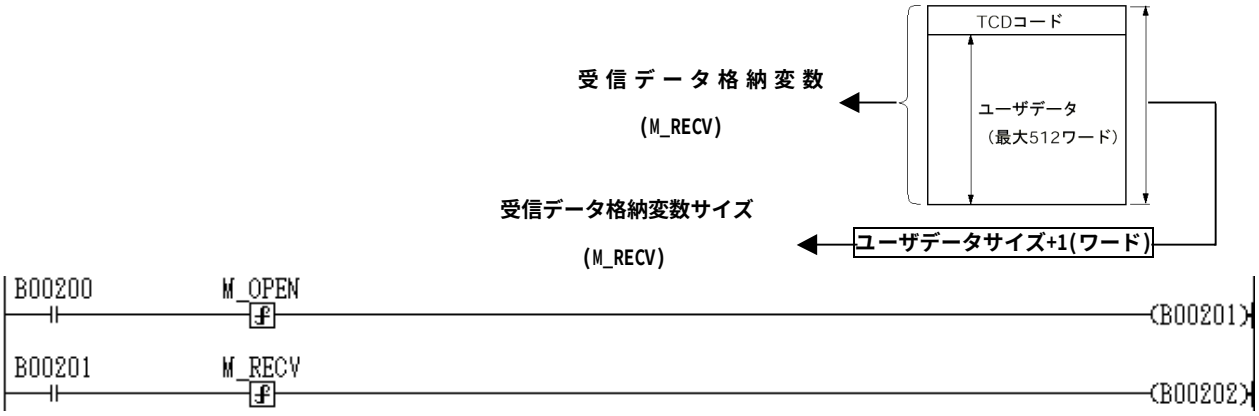


図３　透過型メッセージの受信プログラムと変数指定フォーマットの関係

注 1)通常“３”（送受信共用オープン）を指定してください。同じノードに対して複数のオープン要求はできません（動作は保証されません）。ただし、1：送信専用オープンと 2：受信専用オープンでオープンすることは可能です。

1：送信専用オープン
2：受信専用オープン
3：送受信共用オープン
その他は使用できません。

注 2)送信ポート番号、受信ポート番号は 1～127 をつけることができます。他の M_OPEN_関数が使用しているポート番号と重ならないようにしてください。

M_OPEN

引数	ラベル	値
通信 SX バス局番	Ki0000	246
チャンネル番号	Ki0001	0
ステーション番号(L)	Ki0002	0002(H)
ステーション番号(H)	Ki0003	0000(H)
モジュール種別番号	Ki0004	1
通信モード　注 1)	Ki0005	3
サブモード	Ki0006	0
送信ポート番号　注 2)	Ki0007	2
受信ポート番号　注 2)	Ki0008	1
エラーフラグ	B00000	
ステータス	mi0000	
コネクション番号	mi0001	

M_RECV

コネクション番号	mi0001	
受信データ格納変数	b00002	
受信データ格納変数サイズ	Ki0010	5
エラーフラグ	B00010	
ステータス	mi0010	

種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	マトリクス (MATRIX)	MATRIX f	――――

機能	マトリクス入力をする関数です。

関数引数設定内容

①入力レジスタ：ストローブにより出力データが切り替わる外部機器を接続します。

②出力レジスタ：ストローブ出力(外部機器ストローブ入力に接続します。)

③マトリクス入力レジスタ先頭名：ストローブ出力により入力されたデータを順次格納していく
レジスタ名の先頭を指定します。

使用例

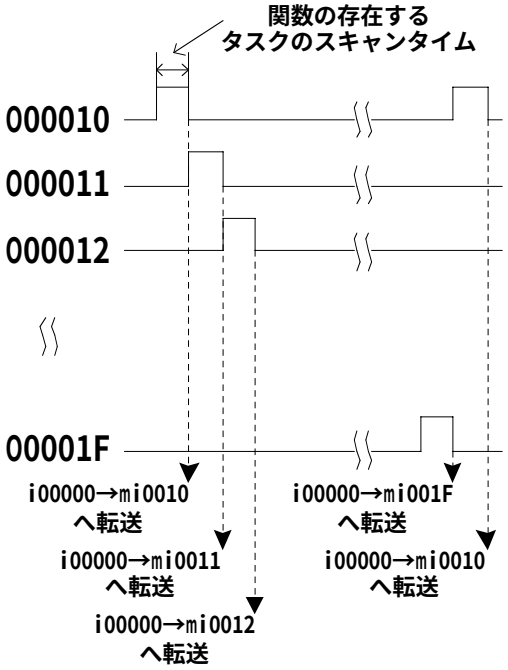
入力レジスタ:i00000(1 ワード分のデータ入力用レジスタ名)

出力レジスタ:o00001(ストローブパルス発生用の出力レジスタ名)

マトリクス入力レジスタ先頭名:mi0010

o00001(000010～00001F)のストローブ出力により入力された i00000 データを mi0010 から mi001F へ順次格納します。

```
i00000=1    000010=ON   mi0010=1
i00000=2    000011=ON   mi0011=2
i00000=3    000012=ON   mi0012=3
                ↓
i00000=16   00001F=ON   mi001F=16
i00000=17   000010=ON   mi0010=17
i00000=18   000011=ON   mi0011=18
```



種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	FL-net の RAS 情報取得		――――――
機能	FL-net の RAS 情報を取得します。		

FLRAS1⇒引数で指定したワードの 1 ビットのみの情報取得が可能です。

関数引数設定内容

①転送元ワードオフセット:何ワード目の情報を取得したいか指定します。
デフォルトは 0(詳細は下記参照)

②転送元ビットオフセット:デフォルトは 0(詳細は下記参照)

③CPU フラグ(8 または 9):

8 → 1 台目の FL-net モジュール

9 → 2 台目の FL-net モジュール

1)ビットオフセットに 0 を指定した場合、ストアが

a)コイルなら 0 ビット目の情報(ON、OFF)が出力されます。

b)レジスタなら指定したワードオフセットの全ビットの情報が出力されます。

2)ビットオフセットに 0 以外を指定した場合、ストアが

a)コイルなら指定したビットの情報(ON、OFF)が出力されます。

b)レジスタなら指定したワードオフセットのビット値が数値で出力されます。

FLRAS8,9⇒指定した複数ワード単位での情報取得が可能です。

関数引数設定内容

①転送元オフセット:デフォルトは 0(詳細は下記参照)

②転送先アドレス:FL-net の RAS 情報を取得する先頭アドレスを指定します。

③転送する個数:転送するワード数を指定します。

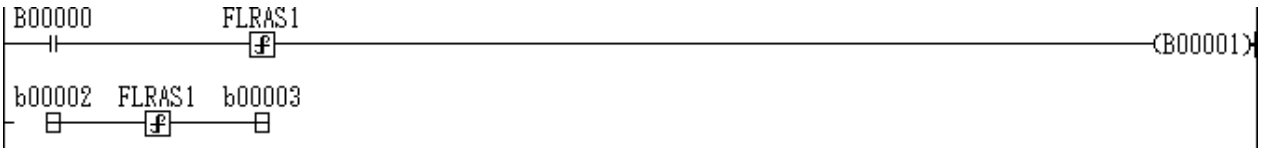
注)1 つのベースボードに 2 台の FL-net モジュールを装着した場合、FLRAS8 は 1 台目、FLRAS9 は 2 台目の FL-net モジュールの情報を取得します。

下記に示す転送元ワードオフセット値を参照し、転送する個数を設定してください。

(各ビットの詳細情報は FL-net モジュールのマニュアルを参照してください。)

ワードオフセット値	FL-net のRAS 情報内容	ワードオフセット値	FL-net のRAS 情報内容
0～15	参加フラグ	74～79	ネットワーク管理テーブル
16～31	構成フラグ	80～1103	参加ノード管理＃C テーブル
32～47	異常フラグ	1104～2896	参加ノード管理＃M テーブル
48～73	自ノード管理テーブル	2897～2962	FL-net エラーログ

使用例



1 番目の FLRAS 関数を右記のようにを設定すると、B00001 には FL-net モジュールの CPU 番号 8 上におけるノード番号 1 の参加状況が出力されます。

B00001=ON となるとノード番号 1 は参加しており、B00001=OFF なら参加していません。

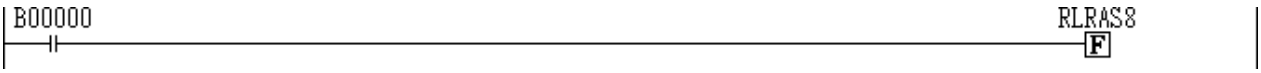
FLRAS1　1 番目

引数	ラベル	値
転送元ワードオフセット	ki0000	0
転送元ビットオフセット	ki0001	1
CPU フラグ	ki0002	8

2 番目の FLRAS 関数を右記のように設定すると、b00003 には FL-net モジュールの CPU 番号 8 上におけるノード番号 1 の参加状況が数値でストアされます。参加しているのであれば、ノード番号 1 は 1 ビット目なので 2 が、参加していないのであれば 0 がストアされます。もし、転送元ビットオフセット値を ki0000=0 に変更した場合、b00003 にはノード番号 1 から 15 までの参加フラグが数値としてストアされます。

FLRAS1　2 番目

転送元ワードオフセット	ki0003	0
転送元ビットオフセット	ki0004	1
CPU フラグ	ki0005	8



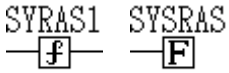
FLRAS8 関数を右記のように設定すると、

mi0000 にはノード番号１～15
mi0001 にはノード番号 16～31
mi0002 にはノード番号 32～47
mi0003 にはノード番号 48～63

の参加しているノード番号の情報が数値でストアされます。

FLRAS8

引数	ラベル	値
転送元オフセット	ki0010	0
転送元オフセット	mi0000	
転送個数	ki0011	4

種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	システムメモリ の RAS 情報取得		――――――
機能	システムメモリの RAS 情報を取得します。		

SYRAS1⇒引数で指定したワードの 1 ビットのみの情報取得が可能です。

関数引数設定内容

①転送元ワードオフセット:何ワード目の情報を取得したいか指定します。
デフォルトは 0(詳細は次ページ参照)

②転送元ビットオフセット:デフォルトは 0(詳細は次ページ参照)

1)ビットオフセットに 0 を指定した場合、ストアが

a)コイルなら 0 ビット目の情報(ON、OFF)が出力されます。

b)レジスタなら指定したワードオフセットの全ビットの情報が出力されます。

2)ビットオフセットに 0 以外を指定した場合、ストアが

a)コイルなら指定したビットの情報が出力されます。

b)レジスタなら指定したワードオフセットのビット値が出力されます。

SYSRAS⇒指定した複数ワード単位での情報取得が可能です。

関数引数設定内容

①転送元オフセット:デフォルトは 0(詳細は次ページ参照)

②転送先アドレス:システムメモリの RAS 情報を取得するアドレスを指定します。

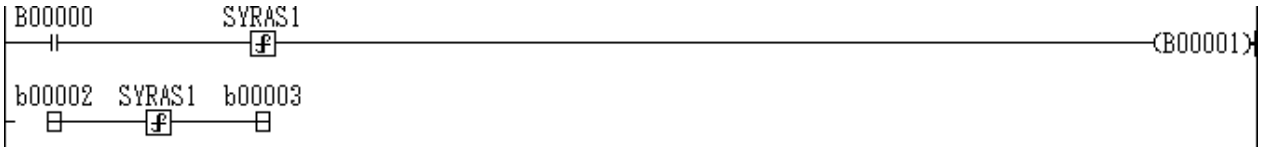
③転送する個数:転送するワード数を指定します。

次ページに示す転送元ワードオフセット値を参照し、転送する個数を設定してください。
(各ビットの詳細情報は付録 3 を参照してください。)

ワードオフセット値	システムメモリのRAS情報内容	ワードオフセット値	システムメモリのRAS情報内容
0	リソース運転スタート	22～29	システム定義異常要因
1	リソーススイッチ設定情報	38～39	アプリケーションプログラム異常要因
2	リソース重故障要因	42～43	アナウンスリレー
4	リソース軽故障要因	49	リソース運転情報
6	CPU異常要因	50	リソース構成情報
8	メモリ異常要因	51	リソース異常情報
10～11	SXバス異常要因	52～67	SXバス構成情報 (コンフィグレーション構成情報)
12	アプリケーション異常要因(重故障)	68～83	SXバス異常情報 (コンフィグレーション異常情報)
13	アプリケーション異常要因(軽故障)	128～255	リモートIOマスタ(0～7) （I/Oモジュール構成/異常構成）
14～16	ユーザ重故障　要因0-要因47	508～511	SXバス伝送エラーレート情報
18～20	ユーザ軽故障　要因0-要因47		

注）3、5、7、9、17、21、30～37、40、41、44～48、84～127、256～507 のワードオフセット値は使用しません。
ただし、例えば 0 ワード目から 8 ワード目までの情報を 1 度に取得したい場合、転送個数を 9 個にすると、3、5、7 ワード目にも値が入りますが、ユーザは特に気にする必要はありません。

使用例



1 番目の SYRAS1 関数を右記のように設定すると、
B00001 には CPU モジュールの軽故障情報が出力されます。
B00001=ON となると CPU モジュールに軽故障が発生、
B00001=OFF なら軽故障は発生していません。

SYRAS1	1 番目	
引数	ラベル	値
転送元ワードオフセット	ki0000	0
転送元ビットオフセット	ki0001	3

2 番目の SYRAS1 関数を右記のように設定すると、
b00003 には CPU モジュールの軽故障情報が数値でストアされます。軽故障になれば、3 ビット目なので 8 が、軽故障でなければ 0 がストアされます。

SYRAS1	2 番目	
転送元ワードオフセット	ki0003	0
転送元ビットオフセット	ki0004	3

もし、転送元ビットオフセット値を ki0004=0 に変更した場合、b00003 には CPU の運転フラグから S
スマスタまでの情報が数値としてストアされます。

SYSRAS 関数を右記のように設定すると、
　mi0000 にはリソース運転スタート
　mi0001 にはリソーススイッチ設定情報
　mi0002 にはリソース重故障要因
　mi0004 にはリソース軽故障要因
　mi0006 には CPU 異常要因
の値がストアされます。

SYSRAS		
引数	ラベル	値
転送元オフセット	ki0010	0
転送元アドレス	mi0000	
転送個数	ki0011	6

注)　mi0003、mi0005 には値が入りますがユーザはこれを特に気にする必要はありません。

種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	汎用通信		――――――
機能	汎用通信用の関数です。		

関数引数設定内容

①送信要求：データ送信を開始します。送信終了時にはアプリケーションで OFF する必要があります。

②送信データ長：送信するデータ長をバイト数で指定します。

③送信データアドレス：送信データの先頭アドレスを指定します。

④受信データアドレス：受信データの先頭アドレスを指定します。

⑤パラメータアドレス：ポート初期化用パラメータの先頭アドレスを指定します。

⑥RAS 情報アドレス：C_FREE 動作情報の先頭アドレスを指定します。

⑦オープンステータス：ポート初期化の結果を示すコードです。

⑧送信完了：送信が完了すると ON します。(1 スキャン)

⑨送信異常：送信に異常が発生すると ON します。(1 スキャン)

⑩送信ステータス：送信の結果を示すコードです。

⑪受信完了：受信が完了すると ON します。(1 スキャン)

⑫受信異常：受信に異常が発生すると ON します。(1 スキャン)

⑬受信ステータス：受信の結果を示すコードです。

⑭受信データ長：受信したデータ長が格納されます。

⑮RS-485 局番：汎用通信モジュールの局番が格納されます。

注)本関数を使用するには関数インスタンスメモリを 3500 ワード分確保してください。
システム構成定義より CPU モジュールのプロパティ － パラメータにて設定できます。
本関数の詳細については別途マニュアルを参照してください。

(1)RAS 情報アドレスのフォーマット

RAS 情報先頭アドレスで指定した先頭アドレスから以下の順でパラメータが入力されます。

RAS	RAS 情報
0	ポートステータス
1	通信モジュールステータス
2	送信要求回数
3	送信完了回数
4	受信回数
5	フレーム検出回数
6	M_OPEN ステータス
7	M_SEND ステータス
8	M_RECV ステータス
9	M_SEND エラー回数
10	M_RECV エラー回数

(2)送信データ及び受信データのフォーマット

15　　8
0

ワード数	送信データ	
0	データ 2	データ 1
1	データ 4	データ 3
⋮	⋮	
511	データ n	データ n-1

n は先頭コード、終了コード BCC 等含む

15　　8
0

ワード数	受信データ	
0	データ 2	データ 1
1	データ 4	データ 3
⋮	⋮	
511	データ n	データ n-1

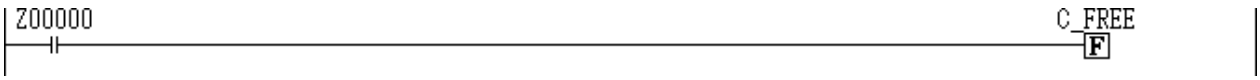
n は先頭コード、終了コード BCC 等含む

(3)通信パラメータのフォーマット

ワード数	項目	内容	
0	汎用通信モジュール局番	汎用通信モジュール SX バス上の局番を設定します。	
1	ポート No.	汎用通信モジュールのインターフェースポートを指定します。 0：RS-232C ポート　　1：RS-485 ポート	
2	メッセージポート No.	汎用通信モジュールとのメッセージ送受信用ポート No.を指定します。（１～127） 注）　他のメッセージ送受信用ポート No.と重複しないでください。	
3	伝送速度	伝送速度を指定します。 0：1200　1：2400　2：4800　3：9600　4：19200　5：38400　6：57600　bps	
4	データビット	データビット長を指定します。7 は 7 ビットで 1 データ、8 は 8 ビットで 1 データを表します。 0：7 ビット　1：8 ビット	
5	パリティビット	データに付加する誤り検出用のビットです。相手機器の設定に合わせて指定してください。 0：なし　1：奇数　2：偶数	
6	ストップビット	データの終わりを示すビットです。相手機器の設定に合わせて指定してください。 0：1 ビット　2：2 ビット	
7	DCE 指定	信号線の制御を行わない場合、DCE/DTE モード共に同一の動きをします。汎用通信モジュールの RS-232C は DTE 仕様ですが、信号線を読み替えることにより、DCE 仕様として使用することができます。 4 番ピン(RS) → CS　5 番ピン(CS) → RS　6 番ピン(DR) → ER 20 番ピン(ER) → DR 0：DTE　1：DCE　2：モデム DTE	
8	ER/DR 信号制御	0：なし　1：あり	
9	信号フロー制御	DTE モード	0：なし → RS：常時 ON　送信：無条件 1：あり → RS：送信時 ON　送信：CS　ON 時
		DCE モード	0：なし → CS：常時 ON　送信：無条件 1：あり → CS：RS　ON にて ON　送信：ER　ON 時
10	XON/XOFF 制御	送信側と受信側とは非同期で接続しているため、フロー制御が必要となることがあります。受信側は XOFF を送ってしばらくデータを受け取れないことを通し、XON を送ってこれを解除します。"XON/XOFF 制御"は接続する相手の機器にこの機能があることが必要です。 0：なし　1：あり	
11	RS-485 モード	RS-485 を使用時に、4 線式か 2 線式かを選択します。 0：4 線式　1：2 線式	
12	コード変換	バイナリデータを文字列変数に変換します。 0：なし　1：ASCII 変換　2：EBCDIC 変換	

ワード数	項目	内容
13	フレーム検出	データの受信方法を指定します。 0：なし　　データが受信されれば受信完了となります。 1：可変長　　先頭コードと終了コードで囲まれたデータを検出した時、受信完了となります。 2：固定長　　受信データが受信バイト数に達した時、受信完了となります。
14	受信バイト数	固定長の時、受信バイト数を指定します。可変長の時は、"0"と指定します。
15	先頭コードバイト数	可変長の時、先頭コードバイト数を指定します。
16	先頭コード 1	可変長の時、先頭コードを指定します。
⋮	⋮	
20	先頭コード 5	
21	終了コードバイト数	可変長の時、終了コードバイト数を指定します。
22	終了コード 1	可変長の時、終了コードを指定します。
⋮	⋮	
26	終了コード 5	
27	BCC 指定	テキストデータの伝送誤りをチェックするための水平パリティを付加するか否かの設定です。 0：なし 1：上位/下位順に設定　BCC の上位バイト｜BCC の下位バイト 2：下位/上位順に設定　BCC の下位バイト｜BCC の上位バイト
28	計算範囲、位置	BCC の位置と計算範囲を設定します。　←→：計算範囲 0：テキスト部を計算して、終了コードの前に入れます。 先頭コード　TEXT　BCC　終了コード　注） 1：テキスト部と終了コードを計算して、終了コードの後ろに入れます。 先頭コード　TEXT　終了コード　BCC 2：先頭コードとテキスト部を計算して、終了コードの前に入れます。 先頭コード　TEXT　BCC　終了コード　注） 3：先頭コードとテキスト部と終了コードを計算して、終了コードの後ろに入れます。 先頭コード　TEXT　終了コード　BCC 注)　この場合、BCC コード形態にはバイナリの指定はできません。
29	BCC の計算式	伝送誤りをどのようにチェックするかの計算方法です。 D1　D2　・　・　Dn 0：加算　　D1+D2+　・　・　・　+Dn 1：加算反転　　(D1+D2+　・　・　・　+Dn)の反転 2：EOR　　D1 EOR D2 EOR　・　・　・　EOR Dn 3：CRC　　CRC-16：$X^{16}+X^{15}+X^2+1$
30	BCC のコード形態	BCC データのコード形態を指定します。 0：バイナリ　　1：ASCII　　2：EBCDIC
31	送信タイマ値	CPU モジュールが RS-232C 回線にデータ送信要求を行ってから、送信が完了するまでの送信監視タイマです。 通常 100(1 秒)に設定します。(0.01 秒単位)

使用例



右記のように設定すると B00000 の ON で外部機器に対し g00000 から mi0010 分のデータを送信します。

また外部機器からの受信データは g00200 に格納され mi0011 にデータ長が格納されます。

C_FREE

引数	ラベル	値
送信要求	B00000	
送信データ長	mi0010	
送信データアドレス	g00000	
受信データアドレス	g00200	
パラメータアドレス	ki0000	
RAS 情報アドレス	g00400	
オープンステータス	mi0000	
送信完了	B00001	
送信異常	B00002	
送信ステータス	mi0001	
受信完了	B00003	
受信異常	B00004	
受信ステータス	mi0002	
受信データ長	mi0011	
RS-485 局番	mi0012	

種類	名称	シンボル	実行時間
データフロー言語 (関数 4)	AIP インターフェース		――――――
機能	汎用通信モジュールを用い、コマツ製 AIP とのインターフェースを行います。		

関数引数設定内容

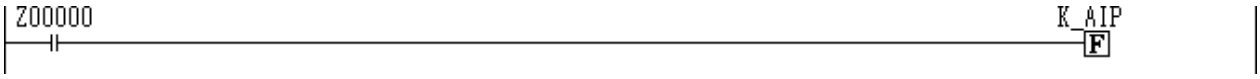
①通信パラメータアドレス：ポート初期化用パラメータの先頭アドレスを指定します。
②RAS 情報アドレス：K_AIP 動作情報の先頭アドレスを指定します。
③通信可：ポートの初期化が正常に完了すると ON し、AIP との通信が可能であることを示します。
④オープンステータス：ポート初期化の結果を示すコードです。
⑤送信異常：送信に異常が発生すると ON します。(1 スキャン)
⑥送信ステータス：送信の結果を示すコードです。
⑦受信異常：受信に異常が発生すると ON します。(1 スキャン)
⑧受信ステータス：受信の結果を示すコードです。

通信パラメータの詳細

ワード数	項目	内容
0	汎用通信モジュール局番	汎用通信モジュールの SX バス局番を設定します。
1	ポート No.	汎用通信モジュールのインターフェースポートを設定します。 0：RS-232C ポート　1：RS-422 ポート
2	メッセージポート No.	汎用通信モジュールとメッセージ送受信用ポート No.を指定します。(1～127) 注)　他のメッセージ送受信用ポート No.と重複しないでください。
3	伝送速度	伝送速度 bps を指定します。 0：1200　1：2400　2：4800　3：9600　4：19200　5：38400 6：57600bps

注)本関数を使用するには関数インスタンスメモリを 3500 ワード分確保してください。
システム構成定義より CPU モジュールのプロパティ － パラメータにて設定できます。
本関数の詳細については別途マニュアルを参照してください。

使用例



K_AIP

引数	ラベル	値
通信パラメータアドレス	ki0000	
RAS 情報アドレス	mi0010	
通信可	B00000	
オープンステータス	mi0000	
送信異常	B00001	
送信ステータス	mi0001	
受信異常	B00002	
受信ステータス	mi0002	

通信パラメータの内容

ki0000	1
ki0001	0
ki0002	1
ki0003	4

上記の設定は、汎用通信モジュールを SX 局番 1 に挿入し、RS-232C、19200bps で AIP と接続する設定です。

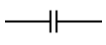
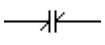
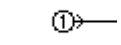
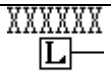
付　録

付録

(付録１) シンボルと各名称

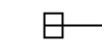
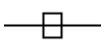
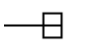
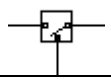
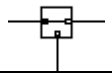
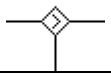
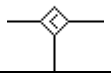
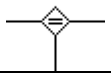
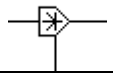
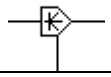
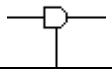
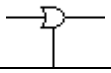
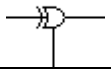
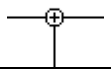
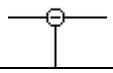
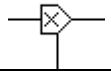
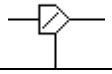
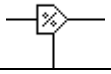
(1) LD 言語

表５.１

A 接点	B 接点	論理反転	コイル	結合子ロード	結合子ストア
					
ラベル	ジャンプ	リターン			
					

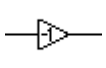
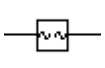
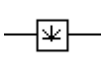
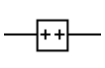
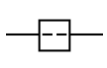
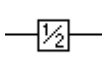
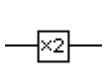
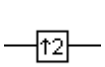
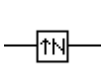
(2) データフロー言語(基本)

表５.２

ロード	ストア&ロード	ストア	a 接点	b 接点	c 接点
					
c 接点	コンペアハイ	コンペアロウ	コンペアイコール	上位優先	下位優先
					
論理積	論理和	論理排他和	加算	減算	乗算
					
除算	剰余	局所定数：整数	局所定数：実数		
					

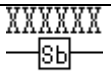
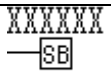
(3) データフロー言語(関数 1)

表５．３

符号変換	1‘補数	絶対値変換	インクリメント	デクリメント	2 分の 1
					
2 倍	2 乗	指数	平方根	ビットカウント	グレイコード バイナリー
					

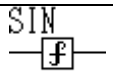
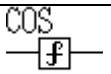
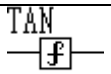
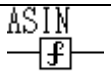
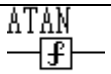
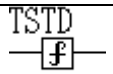
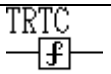
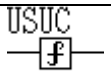
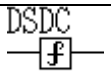
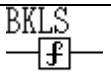
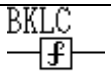
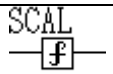
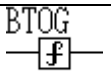
(4) データフロー言語(関数 2)

表５．４

不感帯	パターン	微分補償	位相補償	PI 補償	ARC
					
S-ARC	算術平均	フィルタ	PID 補償	一時遅れ	ディレー
					
定周期パルス	変数設定 パターン	上下限リミッタ	ヒステリシス	無条件 サブルーチン	条件付 サブルーチン
					

(5) データフロー言語(関数 3)

表５．５

正弦	余弦	正接	逆正弦	逆余弦	逆正接
					
オンタイマ	オフタイマ	オン微分	オフ微分	バックラッシュ	バックラッシュ 補正
					
スケーリング	バイナリーグレイ 変換	割り余り	整数変換	実数変換	
					

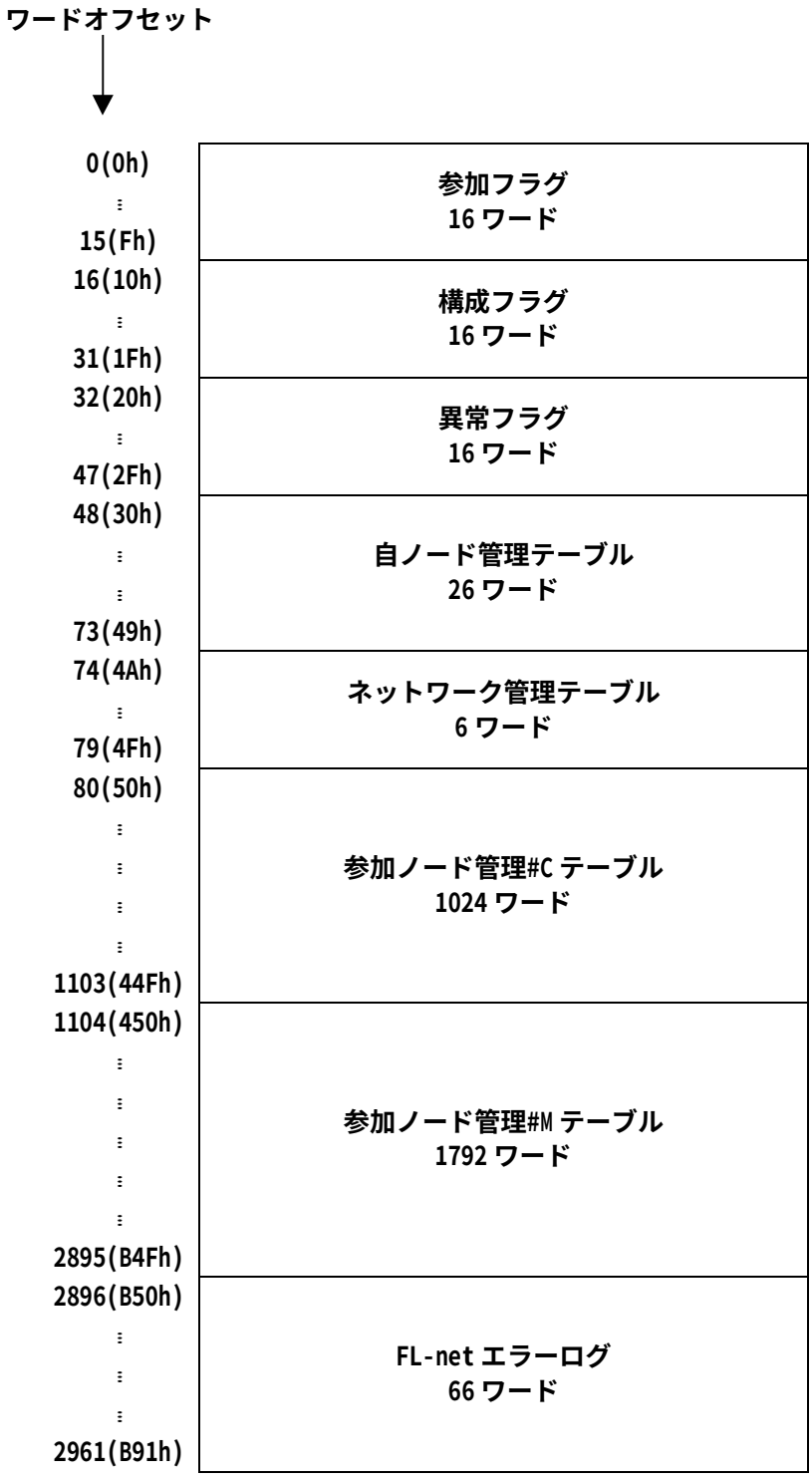
(6) データフロー言語(関数 4)

表５．６

バンク切換	リモートデータリード	リモートデータライト	チャンネルオープン	メッセージ送信	メッセージ受信
F_BANK	RREAD	RWRITE	M_OPEN	M_SEND	M_RECV
マトリクス	FL-net の RAS 情報取得	システムメモリの RAS 情報取得	セット	リセット	データ転送
MATRIX	FLRAS1	SYRAS1	SET	RESET	MOVW
データ転送	カウンタ	FL-net の RAS 情報取得	FL-net の RAS 情報取得	システムメモリの RAS 情報取得	汎用通信
MOVWD	UPDOWN	FLRAS8	FLRAS9	SYSRAS	C_FREE
AIP インターフェース					
K_AIP					

(付録 2)　FL-net モジュール内リンクデータ領域

FL-net モジュールの参加フラグ、構成フラグなどは、次のように FL-net モジュール内メモリに割り付けられています。これらのデータは FLRAS1、8、9 関数を使用して参照することができます。ワードオフセット値は 10 進表現です。(　)内は 16 進表現です。



(1) 参加フラグ/構成フラグ/異常フラグ(ワードオフセット値:0(0h)～47(2F))

FL-net上に接続されている各ノードの状態を示します。各ノードの状態は参加フラグ/構成フラグ/異常フラグとシステム構成定義内ノード構成登録状態との組み合わせで判断します。

＜ノードの状態によるフラグの変化＞

構成登録	参加	構成	異常	ノードの状態
なし	OFF	OFF	OFF	登録なしノード未接続中
	ON	OFF	OFF	登録なしノード接続中(参加)
	OFF	OFF	ON	登録なしノード脱落
あり	OFF	OFF	ON	該当ノード未接続または脱落
	ON	ON	OFF	該当ノード正常接続(参加)

①参加フラグ (ワードオフセット値:0(0h)～15(Fh)) (読み出し専用)

FL-net上に該当するノードが参加しているときONします。表中の数字はノード番号を表しています。

＜各ノードの参加フラグ＞

ワードオフセット ↓　　ビットオフセット ↓

	(Fh)	(Eh)	(Dh)	(Ch)	(Bh)	(Ah)	(9h)	(8h)	(7h)	(6h)	(5h)	(4h)	(3h)	(2h)	(1h)	(0h)
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0(0h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
1(1h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
2(2h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
3(3h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
4(4h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
5(5h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
6(6h)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
7(7h)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112
8(8h)	143	142	141	140	139	138	137	136	135	134	133	132	131	130	129	128
9(9h)	159	158	157	156	155	154	153	152	151	150	149	148	147	146	145	144
10(Ah)	175	174	173	172	171	170	169	168	167	166	165	164	163	162	161	160
11(Bh)	191	190	189	188	187	186	185	184	183	182	181	180	179	178	177	176
12(Ch)	207	206	205	204	203	202	201	200	199	198	197	196	195	194	193	192
13(Dh)	223	222	221	220	219	218	217	216	215	214	213	212	211	210	209	208
14(Eh)	239	238	237	236	235	234	233	232	231	230	229	228	227	226	225	224
15(Fh)		254	253	252	251	250	249	248	247	246	245	244	243	242	241	240

　　の部分は未使用です。

②構成フラグ　(ワードオフセット値:16(10h)～31(1Fh))　(読み出し専用用)

FL-net上のノードがシステム構成登録されている状態でかつ実際にFL-net上に参加しているときONします。

＜各ノードの構成フラグ＞

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
16(10h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
17(11h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
18(12h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
19(13h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
20(14h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
21(15h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
22(16h)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
23(17h)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112
24(18h)	143	142	141	140	139	138	137	136	135	134	133	132	131	130	129	128
25(19h)	159	158	157	156	155	154	153	152	151	150	149	148	147	146	145	144
26(1Ah)	175	174	173	172	171	170	169	168	167	166	165	164	163	162	161	160
27(1Bh)	191	190	189	188	187	186	185	184	183	182	181	180	179	178	177	176
28(1Ch)	207	206	205	204	203	202	201	200	199	198	197	196	195	194	193	192
29(1Dh)	223	222	221	220	219	218	217	216	215	214	213	212	211	210	209	208
30(1Eh)	239	238	237	236	235	234	233	232	231	230	229	228	227	226	225	224
31(1Fh)		254	253	252	251	250	249	248	247	246	245	244	243	242	241	240

③異常フラグ　(ワードオフセット値:32(20h)～47(2Fh))　(読み出し専用)

FL-net上のノード脱落または参加していないときONします。

＜各ノードの異常フラグ＞

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
32(20h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
33(21h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
34(22h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
35(23h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
36(24h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
37(25h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
38(26h)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
39(27h)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112
40(28h)	143	142	141	140	139	138	137	136	135	134	133	132	131	130	129	128
41(29h)	159	158	157	156	155	154	153	152	151	150	149	148	147	146	145	144
42(2Ah)	175	174	173	172	171	170	169	168	167	166	165	164	163	162	161	160
43(2Bh)	191	190	189	188	187	186	185	184	183	182	181	180	179	178	177	176
44(2Ch)	207	206	205	204	203	202	201	200	199	198	197	196	195	194	193	192
45(2Dh)	223	222	221	220	219	218	217	216	215	214	213	212	211	210	209	208
46(2Eh)	239	238	237	236	235	234	233	232	231	230	229	228	227	226	225	224
47(2Fh)		254	253	252	251	250	249	248	247	246	245	244	243	242	241	240

(2) 自ノード管理テーブル (ワードオフセット値:48(30h)～73(49h))

自ノードの設定に関するデータを管理します。各設定データは下図のように割り付けられています。

48(30h)	ノード番号(1 バイト)	未使用
49(31h)	ノード名(10 バイト)	
50(32h)		
51(33h)		
52(34h)		
53(35h)		
54(36h)	未使用	
55(37h)		
56(38h)		
57(39h)		
58(3Ah)		
59(3Bh)	メーカ形式(10 バイト)	
60(3Ch)		
61(3Dh)		
62(3Eh)		
63(3Fh)		
64(40h)	自ノード状態(1 バイト)	未使用
65(41h)	FL-net の状態(1 バイト)	未使用
66(42h)	上位層の状態(2 バイト)	
67(43h)	コモンメモリ領域 1　送信領域先頭アドレス(2 バイト)	
68(44h)	コモンメモリ領域 1　送信領域サイズ(2 バイト)	
69(45h)	コモンメモリ領域 2　送信領域先頭アドレス(2 バイト)	
70(46h)	コモンメモリ領域 2　送信領域サイズ(2 バイト)	
71(47h)	最小許容フレーム間隔(1 バイト)	未使用
72(48h)	トークン監視時間(1 バイト)	未使用
73(49h)	プロトコルバージョン(1 バイト)	未使用

・ノード番号

NP1L-FL1前面のノード番号設定スイッチに設定された番号が16進で表示されます。

・ノード名

システム構成定義中のFL-netパラメータに設定したノード名が表示されます。

例えばノード名を“TOYO DENKI”とした場合、次のように表示されます。

49(31h)	54(h) “T”	4F(h) “O”
50(32h)	59(h) “Y”	4F(h) “O”
51(33h)	20(h) “ “	44(h) “D”
52(34h)	45(h) “E”	4E(h) “N”
53(35h)	4B(h) “K”	49(h) “I”

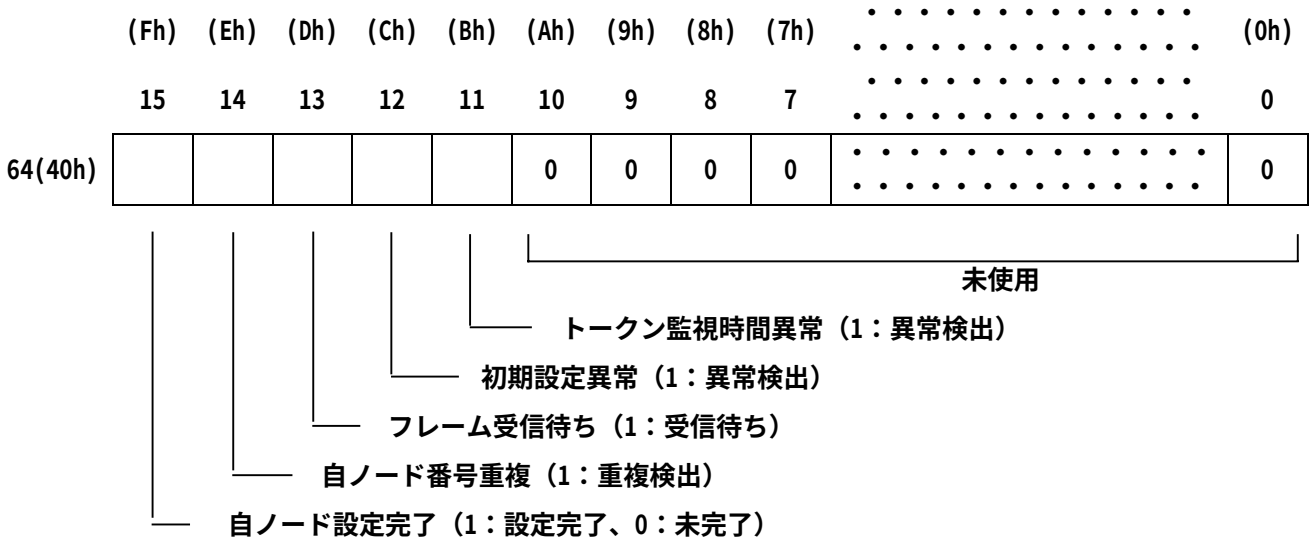
・メーカ形式

NP1L-FL1の場合、“NP1L-FL1 ”と規定されており、下図のように表示されます。

59(3Bh)	4E(h) “N”	50(h) “P”
60(3Ch)	31(h) “1”	4C(h) “L”
61(3Dh)	2D(h) “-“	46(h) “F”
62(3Eh)	4C(h) “L”	31(h) “1”
63(3Fh)	20(h) “ ”	20(h) “ ”

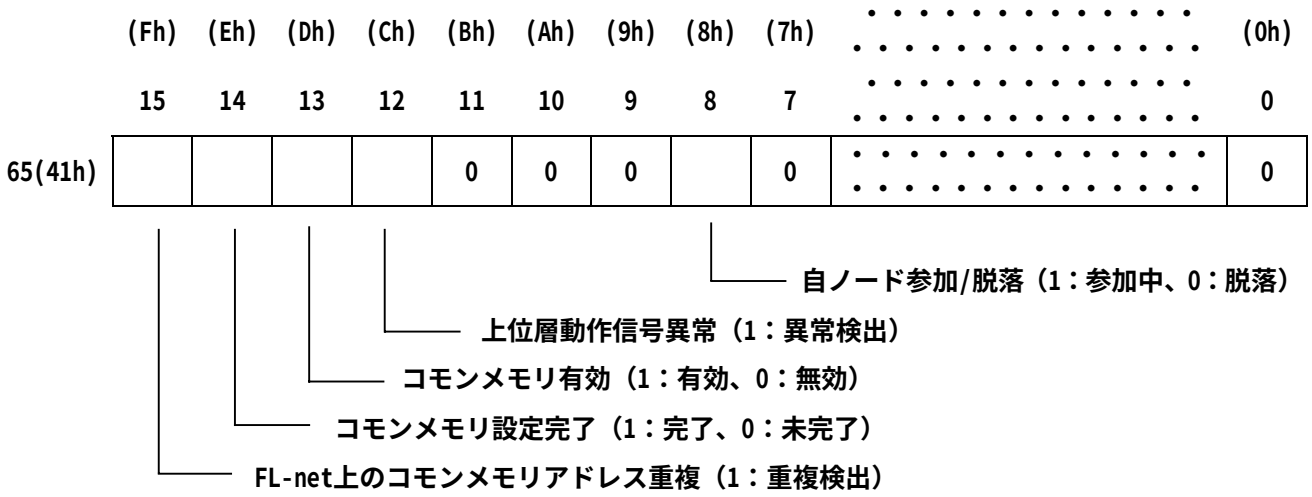
・自ノードの状態

自ノード（NP1L-FL1）の状態を示します。

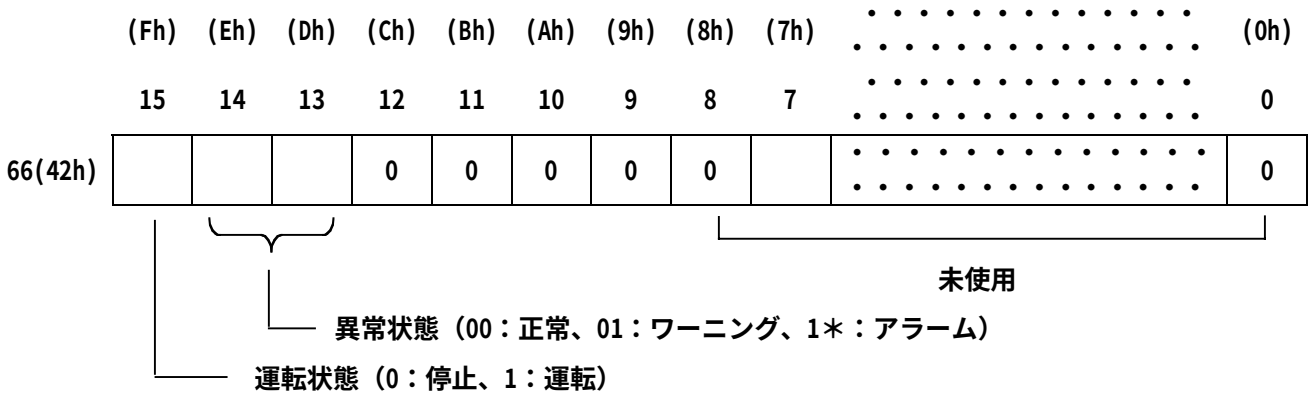


・FL-netの状態

FL-netの状態の情報は、ネットワーク上で共有される情報と、各ノードがそれぞれに管理する情報があります。



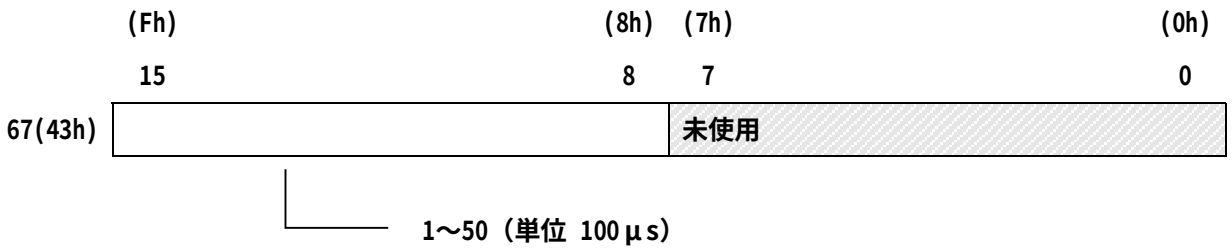
・上位層の状態



・最小許容フレーム間隔

他ノードからトークンを受信し自ノードがフレームを出すまでの時間をフレーム間隔と呼びます。

このとき、各ノードが最低限フレームを出すまで待たなければならない時間を最小許容フレーム間隔と呼びます。



・トークン監視時間

トークン保持ノードから自ノード（NP1L-FL1）がトークンを受け取り、トークンを次の保持ノードへ渡すまでの時間が表示されます。

	(Fh) 15	(8h) 8	(7h) 7	(0h) 0
68(44h)			未使用	

01～255（単位ms）

・プロトコルバージョン

プロトコルバージョンは80hex固定です。

	(Fh) 15	(8h) 8	(7h) 7	(0h) 0
69(45h)	80 (hex)		未使用	

(3)ネットワーク管理テーブル　(ワードオフセット値：74(4Ah)～79(4Fh))

74(4Ah)	トークン保持ノード番号(1 バイト)	未使用
75(4Bh)	最小許容フレーム間隔(1 バイト)	未使用
76(4Ch)	リフレッシュサイクル許容時間(2 バイト)	
77(4Dh)	リフレッシュサイクル測定時間　現在値(2 バイト)	
78(4Eh)	リフレッシュサイクル測定時間　最大値(2 バイト)	
79(4Fh)	リフレッシュサイクル測定時間　最小値(2 バイト)	

(4)参加ノード管理#Cテーブル　(ワードオフセット値：80(50h)～1103(44Fh))

FL-net上に参加している各ノードの送信領域が表示されます。1ノードの情報は4ワードで表示されます。

80(50h)	
⋮	⋮
+4×(ノード番号)	コモンメモリ領域 1　送信領域先頭アドレス(2 バイト)
+4×(ノード番号)+1	コモンメモリ領域 1　送信領域サイズ(2 バイト)
+4×(ノード番号)+2	コモンメモリ領域 2　送信領域先頭アドレス(2 バイト)
+4×(ノード番号)+3	コモンメモリ領域 2　送信領域サイズ(2 バイト)
⋮	⋮
1103(44Fh)	

(5)参加ノード管理#Mテーブル　(ワードオフセット値：1104(450h)～2895(B4Fh))

FL-net上に参加している各ノードのFL-netパラメータの設定内容が表示されます。1ノードの情報は7ワードで表示されます。

1104(450h)		
⋮	⋮	
+7×(ノード番号)	FL-net の状態(1 バイト)	未使用
+7×(ノード番号)+1	上位層の状態(2 バイト)	
+7×(ノード番号)+2	トークン監視時間(1 バイト)	未使用
+7×(ノード番号)+3	最小許容フレーム間隔(1 バイト)	未使用
+7×(ノード番号)+4	リフレッシュサイクル許容時間(2 バイト)	
+7×(ノード番号)+5	未使用	
+7×(ノード番号)+6	未使用	
⋮	⋮	
2895(B4Fh)		

(6)FL-netログ　(ワードオフセット値：2896(B50h)～2961(B91h))

FL-netの通信に関する履歴が格納されます。

2896(B50h)	着信回数(2ワード)	2930(B72h)	ACKエラー回数(2ワード)
2898(B52h)	ソケット部送信エラー回数(2ワード)	2932(B74h)	未使用(8ワード)
2900(B54h)	未使用(2ワード)	2940(B7Ch)	トークン多重認識回数(2ワード)
2902(B56h)	受信回数(2ワード)	2942(B7Eh)	トークン破棄回数(2ワード)
2904(B58h)	受信エラー回数(2ワード)	2944(B80h)	トークン再発行回数(2ワード)
2906(B5Ah)	未使用(8ワード)	2946(B82h)	未使用(2ワード)
2914(B62h)	サイクリック伝送エラー回数(2ワード)	2948(B84h)	トークン監視タイムアウト回数(2ワード)
2916(B64h)	未使用(2ワード)	2950(B86h)	未使用(2ワード)
2918(B66h)	メッセージ伝送再送信回数(2ワード)	2952(B88h)	フレーム待ち状態回数(2ワード)
2920(B68h)	メッセージ伝送再送信オーバ回数(2ワード)	2954(B8Ah)	加入回数(2ワード)
2922(B6Ah)	未使用(2ワード)	2956(B8Ch)	自己離脱回数(2ワード)
2924(B6Ch)	メッセージ受信エラー回数(2ワード)	2958(B8Eh)	スキップによる離脱回数(2ワード)
2926(B6Eh)	未使用(4ワード)	2960(B90h)	他ノード離脱認識回数(2ワード)

(付録3)　システムメモリ領域(512ワード)

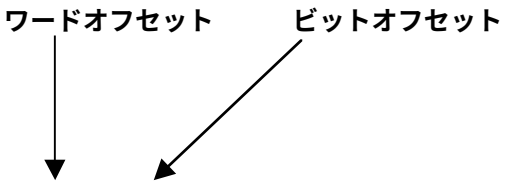
システムメモリは、μGPCｓｘシリーズのシステムの運転状態や異常状態を知らせるためのフラグなどが割り付けられており、用途が決められている領域です。これらのデータはSYRAS1、SYSRAS関数を使用して参照することができます。ワードオフセット値は10進表現です。(　)内は16進表現です。

ワードオフセット	
0(0h)	リソース運転ステータス
1(1h)	リソーススイッチ/ユーザ ROM 状態
2(2h)	リソース重故障要因
3(3h)	未使用
4(4h)	リソース軽故障要因
5(5h)	未使用
6(6h)	CPU 異常要因
7(7h)	未使用
8(8h)、9(9h)	メモリ異常要因
10(Ah)、11(Bh)	SX バス異常要因
12(Ch)	アプリケーション異常要因(重故障)
13(Dh)	アプリケーション異常要因(軽故障)
14(Eh)～16(10h)	ユーザ重故障　要因 0-要因 47
17(11h)	未使用
18(12h)～20(14h)	ユーザ軽故障　要因 0-要因 47
21(15h)	未使用
22(16h)～29(1Dh)	システム定義異常要因
30(1Eh)～37(25h)	未使用
38(26h)、39(27h)	アプリケーションプログラム異常要因
40(28h)、41(29h)	未使用
42(2Ah)、43(2Bh)	アナウンスリレー
44(2Ch)、45(2Dh)	未使用
46(2Eh)	冗長化アナウンスリレー
47(2Fh)	冗長化運転モード
48(30h)、49(31h)	リソース稼動/運転情報
50(32h)、51(33h)	リソース構成/異常情報
52(34h)～67(43h)	SX バス構成情報 (コンフィグレーション構成情報)
68(44h)～83(53h)	SX バス異常情報 (コンフィグレーション異常情報)
84(54h)～99(63h)	SX バス直結モジュール縮退モード情報
100(64h)～127(7Fh)	未使用

ワードオフセット	
128～135 (80h)～(87h)	リモート I/O マスタ 0 I/O モジュール構成情報
136～143 (88h)～(8Fh)	リモート I/O マスタ 0 I/O モジュール異常情報
144～151 (90h)～(97h)	リモート I/O マスタ 1 I/O モジュール構成情報
152～159 (98h)～(9Fh)	リモート I/O マスタ 1 I/O モジュール異常情報
160～167 (A0h)～(A7h)	リモート I/O マスタ 2 I/O モジュール構成情報
168～175 (A8h)～(AFh)	リモート I/O マスタ 2 I/O モジュール異常情報
176～183 (B0h)～(B7h)	リモート I/O マスタ 3 I/O モジュール構成情報
184～191 (B8h)～(BFh)	リモート I/O マスタ 3 I/O モジュール異常情報
192～199 (C0h)～(C7h)	リモート I/O マスタ 4 I/O モジュール構成情報
200～207 (C8h)～(CFh)	リモート I/O マスタ 4 I/O モジュール構成情報
208～215 (D0h)～(D7h)	リモート I/O マスタ 5 I/O モジュール構成情報
216～223 (D8h)～(DFh)	リモート I/O マスタ 5 I/O モジュール異常情報
224～231 (E0h)～(E7h)	リモート I/O マスタ 6 I/O モジュール構成情報
232～239 (E8h)～(EFh)	リモート I/O マスタ 6 I/O モジュール異常情報
240～247 (F0h)～(F7h)	リモート I/O マスタ 7 I/O モジュール構成情報
248～255 (F8h)～(FFh)	リモート I/O マスタ 7 I/O モジュール異常情報
256～507 (100h)～(1FBh)	未使用
508～511 (1FCh)～(1FFh)	SX バス伝送エラーレート 情報

(1)リソース運転ステータス（ワードオフセット値:0(0h)）（読み出し専用）

リソース（CPUモジュール）の運転状態および運転モードを示します。



W	B	名称	説明
0(0h)	0(0h)	運転中	CPUが運転中の時ONします。
	1(1h)	停止中	CPUが停止中の時ONします。
	2(2h)	重故障	リソースに重故障が発生した時ONします。
	3(3h)	軽故障	リソースに軽故障が発生した時ONします。
	4(4h)	冗長化稼動	冗長化運転時、稼動CPUの時ONします。
	5(5h)	冗長化待機	冗長化運転時、待機CPUの時ONします。
	6(6h)	1:1　冗長化	システムが1:1冗長化モードの時ONします。
	7(7h)	N:1　冗長化	システムがN:1冗長化モードの時ONします。
	8(8h)	非自動運転モード	非自動運転モードの時ONします。
	9(9h)	自動運転モード	自動運転モードの時ONします。
	10(Ah)	前回状態モード	前回状態モードの時ONします。
	11(Bh)	バッテリレス運転モード	バッテリレス運転の時ONします。
	12(Ch)	未使用	
	13(Dh)	SXバス直結モジュール縮退モード　注)	SXバス上に直結した全モジュールの縮退及び個別リセット対応モジュールの時ONします。
	14(Eh)	プロセッサバスマスタ	プロセッサバスを管理しているCPUモジュールである時ONします。
	15(Fh)	SXバスマスタ	SXバスを管理しているCPUモジュールである時ONします。

・非自動運転モード

CPUモジュール前面のキースイッチが“RUN”または“TERM”の位置でシステムの電源がONになってもCPUが運転を開始しないモードです。設定はリソースの“設定”内の電源投入時の動作指定で行います。

・自動運転モード

CPUモジュール前面のキースイッチが“RUN”または“TERM”の位置でシステムの電源がONになった場合、CPUが運転を開始するモードです。設定はリソースの“設定”内の電源投入時の動作指定で行います。

（デフォルトは自動運転モードです）。

・前回状態モード

CPUモジュール前面のキースイッチが“RUN”の位置でシステムの電源がONになった場合、CPUが運転を開始し、“TERM”の位置でシステムの電源がONになった場合は、前回、電源がOFFになる直前の状態（運転または停止）となります。

・バッテリレス運転モード

システムの電源投入時にメモリはすべて初期化（初期値の代入または0クリア）されます。また、電池の接続チェックおよび電圧のチェックは行いません。設定はリソースの“設定”内の電源投入時の動作指定で行います。また、本モードで、かつ、前回状態モードの場合は、自動運転モードとなります。

注） TDsxEditorでは、縮退の設定ができないため、あらかじめSXバス上のモジュールは縮退ありの設定になっています。したがって、ユーザは縮退の設定を変えることはできません。

(2)リソーススイッチ/ユーザROM状態 (ワードオフセット値:1(1h)) (読み出し専用)

リソースを制御するCPUモジュールのスイッチの状態を示します。

W	B	名称	説明
1(1h)	0(0h)	CPU番号	CPUモジュール前面のCPU番号設定スイッチに設定されている番号を4ビット(0～F)で表示します。ただし、CPUモジュールの設定範囲は、0-7です。
	1(1h)		
	2(2h)		
	3(3h)		
	4(4h)	未使用	
	5(5h)		
	6(6h)	ユーザROMカード装着状態 注1)	1:装着、0:未装着
	7(7h)	ユーザROMカードライトプロテクト 注1)	1:書き込み禁止 0:書き込み許可(1.6がONの時有効です。)
	8(8h)	STOP位置	キースイッチがSTOPの位置の時ONします。
	9(9h)	TERM位置(下)	キースイッチがTERMの位置(下側)の時ONします。
	10(Ah)	TERM位置(上) 注2) 注3)	キースイッチがTERMの位置(上側)の時ONします。
	11(Bh)	RUN位置	キースイッチがRUNの位置の時ONします。
	12(Ch)	未使用	
	～		
	15(Fh)		

注1)ユーザROMカード(コンパクトフラッシュカード)対応品のみが対象となります。

注2)キースイッチが不定状態のときもTERM位置フラグがONします。

注3)ユーザROMカード対応高性能CPUモジュールの場合、UR8M_TERM位置のときONします。

(3)リソース重故障要因（ワードオフセット値:2(2h)）　（読み出し専用）

リソース（1CPUシステム）の運転が停止する故障要因です。

W	B	名称	説明
2(2h)	0(0h)	CPU異常	自CPUモジュールに重故障が発生した時ONします。
	1(1h)	電源異常	電源断が発生した時ONします。
	2(2h)	メモリ異常	自CPUモジュールのメモリに異常が発生した時ONします。
	3(3h)	SXバス異常	ケーブル外れ、折り返しプラグ抜けなど、SXバスに異常が発生した時ONします。
	4(4h)	アプリケーション異常	アプリケーションプログラムまたはシステム定義に異常がある時ONします。
	5(5h)	未使用	
	6(6h)	共通モジュール異常	自CPUモジュール以外のSXバス上の共通モジュールに異常がある時ONします。
	7(7h)	冗長化連動切替実行異常	冗長化運転モード時、連動切替動作ができない時ONします。
	8(8h)	未使用	
	～		
	12(Ch)		
	13(Dh)	その他ハードウェア異常	CPU番号設定スイッチに異常が発生した時ONします。
	14(Eh)	未使用	
	15(Fh)	ユーザ重故障	アプリケーションプログラムでユーザ重故障フラグ(ワードオフセット:14～16)の何れかのビットをONさせた時ONします。

(4)リソース軽故障要因（ワードオフセット値:4(4h)）　（読み出し専用）

リソースが運転を継続する故障要因です。

W	B	名称	説明
4(4h)	0(0h)	未使用	
	1(1h)		
	2(2h)	メモリ異常	自CPUモジュールのメモリに異常が発生した時ONします。
	3(3h)	SXバス異常	SXバスに異常が発生した時ONします。
	4(4h)	アプリケーション異常	アプリケーションプログラムまたはシステム定義に異常がある時ONします。
	5(5h)	I/Oモジュール異常	自CPUモジュールの制御下にあるI/Oモジュールに異常があった時ONします。　注1)
	6(6h)	共通モジュール異常 注1)	自CPUモジュール以外のSXバス上の共通モジュールに異常がある時ONします。
	7(7h)	未使用	
	～		
	11(Bh)		
	12(Ch)	ユーザROMカード CPU照合不一致　注2)	ユーザROMカードとCPU内メモリの内容が異なる場合ONします。 照合する内容は、システム定義、プロジェクト、パスワードです。
	13(Dh)	その他ハードウェア異常	キースイッチ、ローダ/汎用通信切換スイッチに異常が発生した時ONします。 CPUモジュールはキースイッチが異常の時"TERM"として動作します。また、ローダ/汎用通信切換スイッチ異常の時、ローダ側として動作します。
	14(Eh)	バッテリ異常	データバックアップ用電池の電圧が低下した時、または電池がない時ONします。
	15(Fh)	ユーザ軽故障	アプリケーションプログラムでユーザ軽故障フラグ(ワードオフセット:18～20)の何れかのビットをONさせた時ONします。

注1) 共通モジュールとは入出力領域を占有しないSXバス直結モジュールです。

（CPUモジュール、通信モジュールなど）

注2) ユーザROMカード(コンパクトフラッシュカード)対応品のみが対象となります。

(5)CPU異常要因（ワードオフセット値:6(6h)）　（読み出し専用）

W	B	名称	説明
6(6h)	0(0h)	演算プロセッサ異常	CPUモジュール内の演算用LSIのハード異常
	1(1h)	OSプロセッサ異常	CPUモジュール内のOS制御用LSIのハード異常
	2(2h)	未使用	
	～		
	15(Fh)		

(6)メモリ異常要因　(ワードオフセット値:8(8h)、9(9h))　(読み出し専用)

W	B	名称	説明	故障レベル
8(8h)	0(0h)	システムROM異常	CPUモジュール内のシステムROMに異常が発生した時ONします。	重故障
	1(1h)	システムRAM異常	CPUモジュール内のシステムRAMに異常が発生した時ONします。	重故障
	2(2h)	アプリケーションROM異常	CPUモジュール内のアプリケーション格納用のROMに異常が発生した時ONします。	重故障 注1)
	3(3h)	アプリケーションRAM異常	CPUモジュール内のアプリケーション格納用のRAMに異常が発生した時ONします。	重故障
	4(4h)	未使用		
	～			
	14(Eh)			
	15(Fh)	メモリバックアップ異常	停電保持データが保持されていない時ONします。	重故障 注2)
9(9h)	0(0h)	未使用		
	～			
	14(Eh)			
	15(Fh)	メモリバックアップ異常	停電保持データが保持されていない時ONします。	軽故障 注2)

注1）ユーザROMカードに異常が発生したときもONします。

注2）高性能CPUはモジュールのバージョンによりメモリバックアップ異常時にONするビットが異なります。

V＊＊.25以前：.8.15がON、V10.30以降：.9.15がON。

メモリ異常後の動作について

メモリバックアップ異常が発生すると、ユーザメモリの全領域が0クリアされます。ただし、8.0～8.3まではハードウェア故障の可能性が高いため、電源のOFF→ONを行っても再びメモリ異常で重故障になる確率が高くなります。

(7)SXバス異常要因 (ワードオフセット:10(Ah)、11(Bh))

W	B	名称	説明	故障レベル
10(Ah)	0(0h)	SXバスLSI異常	SXバスを制御するLSIに異常が発生した時ONします。	重故障
	1(1h)	局番重複	1コンフィグレーション上に同じSXバス局番のモジュールが存在している時ONします。	重故障
	2(2h)	接続台数オーバ	SXバスに接続しているモジュールの台数が254台を超えた時ONします。	重故障
	3(3h)	未使用		
	～			
	12(Ch)			
	13(Dh)	SXバス伝送異常	SXバスの伝送に異常がある時ONします。	重故障
	14(Eh)	プロセッサバスアクセス異常	プロセッサバスのアクセスに異常がある時ONします。(自モジュールにアクセス異常要因がある場合)	重故障
	15(Fh)	I/Oリフレッシュ渋滞	SXバスによる入出力データの更新が128ms以上行われない場合ONします。	重故障
11(Bh)	0(0h)	未使用		
	～			
	13(Dh)			
	14(Eh)	プロセッサバスアクセス異常	プロセッサバスのアクセスに異常がある時ONします。(相手モジュールにアクセス異常要因がある場合) アプリケーションプログラムでOFFすることができます。	重故障
	15(Fh)	未使用		

(8)アプリケーション異常要因 (ワードオフセット値:12(Ch)、13(Dh)) (読み出し専用)

W	B	名称	説明	故障レベル
12(Ch)	0(0h)	システム定義異常	システム定義に異常がある時ONします。	重故障
	1(1h)	アプリケーションプログラム異常	アプリケーションプログラムに異常がある時ONします。	重故障
	2(2h)	未使用		
	～			
	15(Fh)			
13(Dh)	0(0h)	未使用		
	1(1h)	アプリケーションプログラム異常	アプリケーションプログラムに異常がある時ONします。	軽故障
	2(2h)	未使用		
	～			
	15(Fh)			

(9)ユーザ重故障　(ワードオフセット値:14(Eh)～16(10h))

W	B	名称	説明
14(Eh)	0(0h)	ユーザ重故障要因0	アプリケーションプログラムで何れかのビットをONさせると、CPUが重故障で停止します。
	≀		
	15(Fh)	ユーザ重故障要因15	
15(Fh)	0(0h)	ユーザ重故障要因16	
	≀		
	15(Fh)	ユーザ重故障要因31	
16(10h)	0(0h)	ユーザ重故障要因32	
	≀		
	15(Fh)	ユーザ重故障要因47	

(10) ユーザ軽故障　(ワードオフセット値:18(12h)～20(14h))

W	B	名称	説明
18(12h)	0(0h)	ユーザ軽故障要因0	アプリケーションプログラムで何れかのビットをONさせると、CPUが軽故障を発生させます。運転は継続します。 アプリケーションプログラムでONしているビットをOFFさせると、軽故障状態から復旧します。
	≀		
	15(Fh)	ユーザ軽故障要因15	
19(13h)	0(0h)	ユーザ軽故障要因16	
	≀		
	15(Fh)	ユーザ軽故障要因31	
20(14h)	0(0h)	ユーザ軽故障要因32	
	≀		
	15(Fh)	ユーザ軽故障要因47	

(11)システム定義異常要因　(ワードオフセット値:22(16h)～29(1Dh))　(読み出し専用)

W	B	名称	説明	故障レベル
22(16h)	0(0h)	未使用		
	1(1h)	システム構成定義異常	CPUモジュール内のシステム構成定義の内容と実際のシステム構成が一致しない時ONします。	重故障
	2(2h)	システム動作定義異常	1コンフィグレーション内に複数の共通モジュールを接続しているシステムや、標準CPUを使用しているシステムにおいて、タクト周期を0.5msに設定するとONします。	重故障
	3(3h)	システムDO設定異常	システムDO(出力)の設定されたSXバス直結モジュールがディジタル出力モジュールでない時ONします。	重故障
	4(4h)	冗長化設定異常	システム冗長化定義において、等値化範囲の指定に誤りがある時ONします。	重故障
	5(5h)	縮退立上げ設定異常	システム内に縮退機能未対応モジュールが存在する時縮退立上げ設定をするとONします。	重故障
	6(6h)	未使用		
	～			
	9(9h)			
	10(Ah)	CPU動作定義異常	システム構成定義に設定されたCPU番号と、CPUモジュール上のスイッチ設定が一致しない時ONします。	重故障
	11(Bh)	CPUメモリ境界定義異常	アプリケーションプログラムで使用されているメモリが、メモリの総容量を超えている時ONします。	重故障
	12(Ch)	未使用		
	～			
	15(Eh)			
23(17h)	0(0h)	CPU I/Oグループ定義異常 デフォルトタスク用	出力選択に入力モジュールを設定している時ONします。	重故障
	1(1h)	CPU I/Oグループ定義異常 0レベルタスク用		
	2(2h)	CPU I/Oグループ定義異常 1レベルタスク用		
	3(3h)	CPU I/Oグループ定義異常 2レベルタスク用		
	4(4h)	CPU I/Oグループ定義異常 3レベルタスク用		
	5(5h)	直結I/O縮退定義異常	直結I/O縮退定義に異常がある時ONします。	重故障

23(17h)	6(6h)	リモートI/0マスタ0縮退定義異常	縮退定義に異常がある時ONします。	重故障
	7(7h)	リモートI/0マスタ1縮退定義異常		
	8(8h)	リモートI/0マスタ2縮退定義異常		
	9(9h)	リモートI/0マスタ3縮退定義異常		
	10(Ah)	リモートI/0マスタ4縮退定義異常		
	11(Bh)	リモートI/0マスタ5縮退定義異常		
	12(Ch)	リモートI/0マスタ6縮退定義異常		
	13(Dh)	リモートI/0マスタ7縮退定義異常		
	14(Eh)	未使用		
	15(Fh)			
24(18h)	0(0h)	直結I/0ホールド定義異常	出力モジュール以外のモジュールへホールド定義をしたり、システムDOに設定した出力モジュールへホールド定義をしている時ONします。	重故障
	1(1h)	直結I/0動作定義異常	SXバスに直結されたモジュールへの動作の設定が異常な時ONします。	重故障
	2(2h)	未使用		
	≀			
	15(Fh)			
25(19h)	0(0h)	リモートI/0マスタ0動作定義異常	リモートI/0マスタの動作定義に異常がある時ONします。	重故障
	1(1h)	リモートI/0マスタ1動作定義異常		
	2(2h)	リモートI/0マスタ2動作定義異常		
	3(3h)	リモートI/0マスタ3動作定義異常		
	4(4h)	リモートI/0マスタ4動作定義異常		
	5(5h)	リモートI/0マスタ5動作定義異常		
	6(6h)	リモートI/0マスタ6動作定義異常		
	7(7h)	リモートI/0マスタ7動作定義異常		
	8(8h)	未使用		
	≀			
	15(Fh)			
26(1Ah)	0(0h)	プロセッサリンク0　動作定義異常	Pリンク/PEリンク/FL-netの動作定義に異常がある時ONします。プロセッサリンク0は回線番号"8"、プロセッサリンク1は回線番号"9"のモジュールに対応しています。	重故障
	1(1h)	プロセッサリンク1　動作定義異常		

	2(2h)	未使用			
	～				
	15(Fh)				

(12)アプリケーションプログラム異常要因　(ワードオフセット値:38(26h)、39(27h))

W	B	名称	説明	故障レベル
38(26h)	0(0h)	アプリケーションWDT異常	デフォルトタスクの実行時間が設定したウォッチドッグタイマ値を超えた時ONします。	重故障
	1(1h)	アプリケーション実行異常	テンポラリサイズオーバなど、ユーザプログラム実行中に異常が発生した時ONします。	重故障
	2(2h)	未使用		
	～			
	10(Ah)			
	11(Bh)	FBインスタンス設定異常	指定した格納アドレスが存在しない時などONします。	重故障
	12(Ch)	初期値設定異常	設定した初期値が格納領域の範囲を超えている時などONします。	重故障
	13(Dh)	SFM境界定義設定異常	システムFB用インスタンスメモリの最大容量を超える容量が設定された時などONします。	重故障
	14(Eh)	POU命令異常	POUに異常がある時ONします。	重故障
	15(Fh)	タスク登録異常	タスクの登録に異常がある時ONします。	重故障
39(27h)	0(0h)	0レベルタスク抜け	タスクの実行が抜けた時ONします。 アプリケーションプログラムでOFFすることができます。	軽故障
	1(1h)	1レベルタスク抜け		
	2(2h)	2レベルタスク抜け		
	3(3h)	3レベルタスク抜け		
	4(4h)	0レベルタスク渋滞	プログラムの実行が渋滞し、設定した定周期時間が守られない時ONします。 アプリケーションプログラムでOFFすることができます。	軽故障
	5(5h)	1 レベルタスク渋滞		
	6(6h)	2 レベルタスク渋滞		
	7(7h)	3レベルタスク渋滞		
	8(8h)	未使用		
	～			
	14(Eh)			
	15(Fh)	タクト周期監視異常	システム定義で設定した値と異なっている時ONします。 アプリケーションプログラムでOFFすることができます。	軽故障

(13)アナウンスリレー　(ワードオフセット値:42(2Ah)、43(2Bh))

W	B	名称	説明
42(2Ah)	0(0h)	イニシャルフラグ	プログラムダウンロード後の最初の運転開始時およびイニシャル起動(コールド運転開始)時にONします。 運転中はOFFしません。
	1(1h)	電源断フラグ	前回運転中、電源断が発生していた時ONします。
	2(2h)	未使用	
	～		
	13(Dh)		
	14(Eh)	ダミーモジュールフラグ	コンフィグレーション内にダミーモジュールが1台以上実装されている時ONします。
	15(Fh)	プロセッサバスアクセス不可フラグ	プロセッサバスを使用できない時ONします。
43(2Bh)	0(0h)	0レベルスタートフラグ	0レベルタスクの1回目の実行中ONします。
	1(1h)	1レベルスタートフラグ	1レベルタスクの1回目の実行中ONします。
	2(2h)	2レベルスタートフラグ	2レベルタスクの1回目の実行中ONします。
	3(3h)	3レベルスタートフラグ	3レベルタスクの1回目の実行中ONします。
	4(4h)	未使用	
	～		
	14(Eh)		
	15(Fh)	デフォルトタスクスタートフラグ	デフォルトタスクの最初の1回目に実行中ONします。

(14)冗長化アナウンスリレー(ワードオフセット値:46(2Eh))

冗長化運転モード(ワードオフセット値:47(2Fh)) （読み出し専用）

W	B	名称	説明
46(2Eh)	0(0h)	冗長化継続起動フラグ	冗長化モードで運転している時、待機から稼動へ変わった時ONします。(待機側から稼動へ切り替わったCPU)
	1(1h)	未使用	
	～		
	15(Fh)		
47(2Fh)	0(0h)	冗長化論理CPU番号	冗長化時の論理CPU番号を4ビットで表示します。(0～7) 特に、デフォルト待機CPUが稼動となった時、どのデフォルト稼動CPUの代わりとなるかを認識できます。冗長化モード以外は不定です。
	～		
	3(3h)		
	4(4h)	未使用	
	～		
	7(7h)		
	8(8h)	冗長化連動切替モード0	1:1冗長化モードで運転している時、CPU0/1のペアが連動切替設定ありの時ONします。
	9(9h)	冗長化連動切替モード1	1:1 冗長化モードで運転している時、CPU2/3 のペアが連動切替設定ありの時 ON します。
	10(Ah)	冗長化連動切替モード2	1:1 冗長化モードで運転している時、CPU4/5 のペアが連動切替設定ありの時 ON します。
	11(Bh)	冗長化連動切替モード3	1:1冗長化モードで運転している時、CPU6/7のペアが連動切替設定ありの時ONします。
	12(Ch)	未使用	
	～		
	15(Fh)		

(15)リソース稼動/運転情報　(ワードオフセット値:48(30h)、49(31h))　(読み出し専用)

冗長化モード時およびシングルモード時のシステム(CPUモジュール)の状態をアプリケーションプログラムで認識するために使用します。リソース稼動情報は、冗長化時のみ有効です。

下表に示す状態はリソース構成/異常情報(ワードオフセット値:50、51)の該当ビットがONしている場合に有効です。

＜冗長化モード時＞

リソース稼動情報	リソース運転情報	リソース状態
OFF	OFF	待機 CPU 停止中
ON	OFF	稼動 CPU 停止中
ON	ON	稼動 CPU 運転中
OFF	ON	待機 CPU 運転中

＜リソース稼動情報＞

W	B	名称	説明
48(30h)	0(0h)	CPU0稼動中	冗長化モード時、CPUが稼動CPUの時ONします。 (冗長化モードではない場合は不定となります。)
	1(1h)	CPU1稼動中	
	2(2h)	CPU2稼動中	
	3(3h)	CPU3稼動中	
	4(4h)	CPU4稼動中	
	5(5h)	CPU5稼動中	
	6(6h)	CPU6稼動中	
	7(7h)	CPU7稼動中	
	8(8h)	未使用	
	～		
	15(Fh)		

＜リソース運転情報＞

W	B	名称	説明
49(31h)	0(0h)	CPU0運転中	SXバス上に、該当する番号のCPUモジュールが存在し、CPUが運転中の時ONします。
	1(1h)	CPU1運転中	
	2(2h)	CPU2運転中	
	3(3h)	CPU3運転中	
	4(4h)	CPU4運転中	
	5(5h)	CPU5運転中	
	6(6h)	CPU6運転中	
	7(7h)	CPU7運転中	
	8(8h)	未使用	
	～		
	15(Fh)		

(16)リソース構成/異常情報　(ワードオフセット値:50(32h)、51(33h))　(読み出し専用)

他のリソース(CPUモジュール)の状態を、アプリケーションプログラムで認識するために使用します。

＜冗長化モード時＞

リソース構成情報	リソース異常情報	リソース状態
OFF	OFF	存在なし
ON	OFF	正常　(運転中または停止中)
ON	ON	軽故障　(運転中または停止中)
OFF	ON	重故障　(停止中または脱落)

＜リソース稼動情報＞

W	B	名称	説明
50(32h)	0(0h)	CPU0構成	SXバス上に、該当する番号のCPUモジュールが存在し、リソース運転ステータスが正常か軽故障の時ONします。
	1(1h)	CPU1構成	
	2(2h)	CPU2構成	
	3(3h)	CPU3構成	
	4(4h)	CPU4構成	
	5(5h)	CPU5構成	
	6(6h)	CPU6構成	
	7(7h)	CPU7構成	
	8(8h)	未使用	
	～		
	15(Fh)		

＜リソース運転情報＞

W	B	名称	説明
51(33h)	0(0h)	CPU0異常	SXバス上に、該当する番号のCPUモジュールが存在し、リソース運転ステータスが重故障か軽故障の時ONします。
	1(1h)	CPU1異常	
	2(2h)	CPU2異常	
	3(3h)	CPU3異常	
	4(4h)	CPU4異常	
	5(5h)	CPU5異常	
	6(6h)	CPU6構成	
	7(7h)	CPU7構成	
	8(8h)	未使用	
	～		
	15(Fh)		

(17)コンフィグレーション構成情報（ワードオフセット値:52(34h)～67(43h)） （読み出し専用）

SXバス上にモジュールがあり、正常か軽故障で動作しているとき、該当するモジュールのSXバス局番のビットがONします。

正常か軽故障かは次のコンフィグレーション異常情報との組み合わせで区別します。

リソース構成情報	リソース異常情報	リソース状態
OFF	OFF	存在なし
ON	OFF	正常
ON	ON	軽故障
OFF	ON	重故障

ワードオフセット

ビットオフセット

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
52(34h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
53(35h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
54(36h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
55(37h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
56(38h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
57(39h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
58(3Ah)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
59(3Bh)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112
60(3Ch)	143	142	141	140	139	138	137	136	135	134	133	132	131	130	129	128
61(3Dh)	159	158	157	156	155	154	153	152	151	150	149	148	147	146	145	144
62(3Eh)	175	174	173	172	171	170	169	168	167	166	165	164	163	162	161	160
63(3Fh)	191	190	189	188	187	186	185	184	183	182	181	180	179	178	177	176
64(40h)	207	206	205	204	203	202	201	200	199	198	197	196	195	194	193	192
65(41h)	223	222	221	220	219	218	217	216	215	214	213	212	211	210	209	208
66(42h)	239	238	237	236	235	234	233	232	231	230	229	228	227	226	225	224
67(43h)		254	253	252	251	250	249	248	247	246	245	244	243	242	241	240

(18)コンフィグレーション異常情報　(ワードオフセット値:68(44h)～83(53h))　(読み出し専用)

SXバス上にモジュールがあり、重故障または軽故障のとき、モジュールのSXバス局番に該当するビットがONします。

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
68(44h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
69(45h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
70(46h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
71(47h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
72(48h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
73(49h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
74(4Ah)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
75(4Bh)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112
76(4Ch)	143	142	141	140	139	138	137	136	135	134	133	132	131	130	129	128
77(4Dh)	159	158	157	156	155	154	153	152	151	150	149	148	147	146	145	144
78(4Eh)	175	174	173	172	171	170	169	168	167	166	165	164	163	162	161	160
79(4Fh)	191	190	189	188	187	186	185	184	183	182	181	180	179	178	177	176
80(50h)	207	206	205	204	203	202	201	200	199	198	197	196	195	194	193	192
81(51h)	223	222	221	220	219	218	217	216	215	214	213	212	211	210	209	208
82(52h)	239	238	237	236	235	234	233	232	231	230	229	228	227	226	225	224
83(53h)		254	253	252	251	250	249	248	247	246	245	244	243	242	241	240

(19)SXバス直結モジュール縮退モード情報

(ワードオフセット値:84(54h)～99(63h))　(読み出し専用)

SXバス上に縮退、個別リセットができないモジュールがあるとき、そのモジュールのSXバス局番のビットがONします。

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
84(54h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
85(55h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
86(56h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
87(57h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
88(58h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
89(59h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
90(5Ah)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
91(5Bh)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112
92(5Ch)	143	142	141	140	139	138	137	136	135	134	133	132	131	130	129	128
93(5Dh)	159	158	157	156	155	154	153	152	151	150	149	148	147	146	145	144
94(5Eh)	175	174	173	172	171	170	169	168	167	166	165	164	163	162	161	160
95(5Fh)	191	190	189	188	187	186	185	184	183	182	181	180	179	178	177	176
96(60h)	207	206	205	204	203	202	201	200	199	198	197	196	195	194	193	192
97(61h)	223	222	221	220	219	218	217	216	215	214	213	212	211	210	209	208
98(62h)	239	238	237	236	235	234	233	232	231	230	229	228	227	226	225	224
99(63h)		254	253	252	251	250	249	248	247	246	245	244	243	242	241	240

(20)リモートI/0マスタ0-I/0モジュール構成/異常情報

(ワードオフセット値:128(80h)～143(8Fh))（読み出し専用）

リモートI/0マスタ0制御下にリモートI/0モジュールがあり、正常か軽故障のとき、該当するモジュールのリモート局番のビットがONします。

リソース構成情報	リソース異常情報	リソース状態
OFF	OFF	存在なし
ON	OFF	正常
ON	ON	軽故障
OFF	ON	重故障

<構成情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
128(80h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
129(81h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
130(82h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
131(83h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
132(84h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
133(85h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
134(86h)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
135(87h)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

リモートI/0マスタ0制御下にリモートI/0モジュールがあり、重故障または軽故障のとき、モジュールのリモート局番に該当するビットがONします。

<異常情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
136(88h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
137(89h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
138(8Ah)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
139(8Bh)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
140(8Ch)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
141(8Dh)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
142(8Eh)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
143(8Fh)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

以下、(21)～(27)の情報の見方は、(20)と同様です。

(21)リモートI/0マスタ1-I/0モジュール構成/異常情報

(ワードオフセット値:144(90h)～159(9Fh))（読み出し専用）

<構成情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
144(90h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
145(91h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
146(92h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
147(93h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
148(94h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
149(95h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
150(96h)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
151(97h)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

<異常情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
152(98h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
153(99h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
154(9Ah)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
155(9Bh)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
156(9Ch)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
157(9Dh)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
158(9Eh)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
159(9Fh)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

(22)リモートI/0マスタ2-I/0モジュール構成/異常情報

(ワードオフセット値:160(A0h)～175(AFh))（読み出し専用）

<構成情報>

	(Fh)	(Eh)	(Dh)	(Ch)	(Bh)	(Ah)	(9h)	(8h)	(7h)	(6h)	(5h)	(4h)	(3h)	(2h)	(1h)	(0h)
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
160(A0h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
161(A1h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
162(A2h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
163(A3h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
164(A4h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
165(A5h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
166(A6h)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
167(A7h)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

<異常情報>

	(Fh)	(Eh)	(Dh)	(Ch)	(Bh)	(Ah)	(9h)	(8h)	(7h)	(6h)	(5h)	(4h)	(3h)	(2h)	(1h)	(0h)
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
168(A8h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
169(A9h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
170(AAh)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
171(ABh)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
172(ACh)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
173(ADh)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
174(AEh)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
175(AFh)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

(23)リモートI/0マスタ3-I/0モジュール構成/異常情報

(ワードオフセット値:176(B0h)～191(BFh))（読み出し専用）

<構成情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
176(B0h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
177(B1h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
178(B2h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
179(B3h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
180(B4h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
181(B5h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
182(B6h)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
183(B7h)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

<異常情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
184(B8h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
185(B9h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
186(BAh)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
187(BBh)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
188(BCh)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
189(BDh)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
190(BEh)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
191(BFh)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

(24)リモートI/0マスタ4-I/0モジュール構成/異常情報

(ワードオフセット値:192(C0h)～207(CFh))（読み出し専用）

<構成情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
192(C0h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
193(C1h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
194(C2h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
195(C3h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
196(C4h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
197(C5h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
198(C6h)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
199(C7h)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

<異常情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
200(C8h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
201(C9h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
202(CAh)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
203(CBh)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
204(CCh)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
205(CDh)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
206(CEh)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
207(CFh)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

(25)リモートI/0マスタ5-I/0モジュール構成/異常情報

(ワードオフセット値:208(D0h)～223(DFh))（読み出し専用）

<構成情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
208(D0h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
209(D1h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
210(D2h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
211(D3h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
212(D4h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
213(D5h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
214(D6h)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
215(D7h)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

<異常情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
216(D8h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
217(D9h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
218(DAh)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
219(DBh)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
220(DCh)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
221(DDh)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
222(DEh)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
223(DFh)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

(26)リモートI/0マスタ6-I/0モジュール構成/異常情報

(ワードオフセット値:224(E0h)～239(EFh))（読み出し専用）

<構成情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
224(E0h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
225(E1h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
226(E2h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
227(E3h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
228(E4h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
229(E5h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
230(E6h)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
231(E7h)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

<異常情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
232(E8h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
233(E9h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
234(EAh)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
235(EBh)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
236(ECh)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
237(EDh)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
238(EEh)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
239(EFh)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

(27)リモートI/0マスタ7-I/0モジュール構成/異常情報

（ワードオフセット値:240(F0h)～255(FFh)）（読み出し専用）

<構成情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
240(F0h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
241(F1h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
242(F2h)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
243(F3h)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
244(F4h)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
245(F5h)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
246(F6h)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
247(F7h)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

<異常情報>

	(Fh) 15	(Eh) 14	(Dh) 13	(Ch) 12	(Bh) 11	(Ah) 10	(9h) 9	(8h) 8	(7h) 7	(6h) 6	(5h) 5	(4h) 4	(3h) 3	(2h) 2	(1h) 1	(0h) 0
248(F8h)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
249(F9h)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
250(FAh)	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
251(FBh)	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
252(FCh)	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
253(FDh)	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
254(FEh)	111	110	109	108	107	106	105	104	103	102	101	100	99	98	97	96
255(FFh)	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113	112

(28)SXバス伝送エラーレート情報

（ワードオフセット値:508(1FCh)～511(1FF)）　（読み出し専用）

タクトを10万回実行した中で、SXバスエラーが発生したタクトの数を百万分率（ppm）で表しています。

10万回中1回エラーがあると値は“10”となります。値の更新はタクト10万回実行毎に行われます。

アドレス	名称	説明
508(1FCh)	最大値(下位ワード)	自 CPU モジュールが検出した SX バスの伝送エラーレートの最大値がセットされます。
509(1FDh)	最大値(上位ワード)	
510(1FEh)	今回値(下位ワード)	自 CPU モジュールが検出した SX バスの伝送エラーレートの今回値がセットされます。
511(1FFh)	今回値(上位ワード)	

注1) システムメモリ領域の各種システムフラグ情報は、アプリケーションプログラムから参照できますが、アプリケーションプログラムのイベントタスクを起動する“イベント変数”には使用しないでください。（タスクが起動されない変数があります。）

(付録 4)メッセージ関数関連のエラーステータス

ステータスコード	名称	要因	対処法	M_OPEN	M_SEND	M_RECV	RREAD	RWRITE
164(A4h)	メッセージ送信先指定異常	指定 SX 局番にモジュールが存在しません。	通信先を設定する入力端子を再確認	○	○	○	○	○
165(A5h)	メッセージ受信 BUSY	SX バスでメッセージ通信相手が BUSY です。	しばらくして関数起動 メッセージ負荷減	○	○	○	○	○
170(AAh)	メッセージ送信 BUSY	CPU モジュール内でメッセージ送信資源が BUSY です。	しばらくして関数起動 自 CPU モジュール負荷減	○	○	○	○	○
197(C5h)	ネットワーク送信 BUSY	通信モジュール間において通信相手が BUSY です。	しばらくして関数起動 自 CPU モジュール負荷減	—	○	—	○	○
177(B1h)	パラメータ異常	規定の入力範囲を超えた入力です。		○	○	○	○	○
193(C1h)	チャンネルオープン異常	ステーション番号が誤りです。 通信モードが誤りです。 チャンネル番号が誤りです。		○	—	—	—	—
195(C3h)	メッセージ送信異常	メッセージ送信不可です。 通信相手より応答がありません。 ステーション番号が誤りです。 異常コード付き応答受信しました。 通信相手が未サポートです。		—	○	—	○	○
199(C7h)	チャンネルクローズ	コンフィグレーション外通信で通信相手がクローズしています。		—	○	○	—	—
200(C8h)	ポート指定異常	受信ポート番号が 1～127 の範囲外です。 リソース内で指定済みです。 通信相手未オープンです。		○	○	○	○	○

ステータスコード	名称	要因	対処法	M_OPEN	M_SEND	M_RECV	RREAD	RWRITE
201(C9h)	コネクション番号・クライアントポート番号 FULL	クライアントポート番号 FULL です。 リソース内で同時に 57 以上オープンしています。 規定を超えるポートをオープンしています。		○	—	—	○	○
206(CEh)	バッファオーバ	送信データ数が 4096 バイトを超えています。 受信データが格納変数のサイズを超えています。 モジュール種別番号に 0 以外を指定した場合、通信モジュールの制限を超えています。 RWRITE 関数において送信先で異常が検出されました。		—	○	○	○	○
207(CFh)	コネクション番号異常	未オープンのコネクション番号を使用しています。		—	○	○	—	—
05(05h)	照合エラー	メッセージ折り返しにて照合エラーを検出しました。		—	—	—	—	○
68(44h)	メモリアドレス指定異常	指定アドレスが有効範囲を超えています。		—	—	—	○	○
69(45h)	メモリサイズオーバ	アドレス読み出し・書き込みワード数が有効範囲を超えています。		—	—	—	○	○