

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

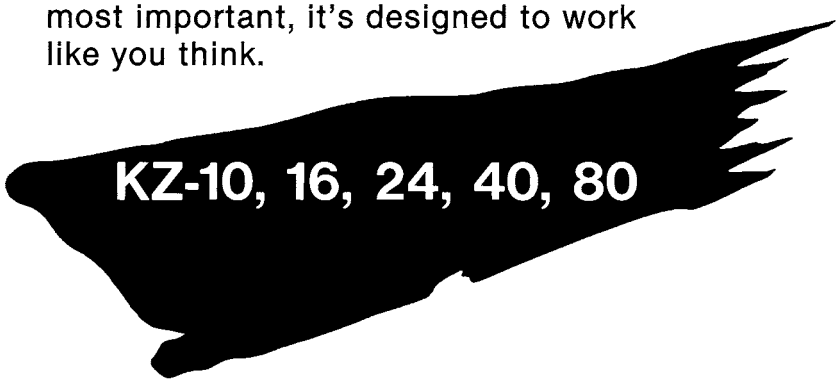
KEYENCE

079008

KZ Series

Reference Manual

The KZ Series PLC is a snap to configure, and it will fit into the tightest spots.
The KZ Series was designed with you in mind. It's logical, very small, yet very powerful and, most important, it's designed to work like you think.



KZシリーズ
リファレンスマニュアル
KZ-10, 16, 24, 40, 80

― はじめに ―

本マニュアルでは、KZ のプログラミングや操作方法についての詳細を解説いたしております。
本マニュアルをよくお読みいただき、KZ を十分に活用してください。
なお取付け、配線などにつきましては、本格添付の「取扱説明書」を合せてご覧ください。

本マニュアルは最終的にご使用いただく方のお手元に届くようご配慮ください。

《KZ シリーズの特長》

1. **超小型**
   W60×H70×D43mm (KZ-10) の世界最小サイズ (汎用リレー 2 個分のスペース)。取り付け場所を問いません。
2. **余裕のプログラム容量**
   KZ-10、16 で約 500 行、KZ-24、40、80 で約 3,000 行のプログラムが書き込めます。
3. **高速**
   パルス幅 50μs まで入力可能。10kHz までカウント可能な高速カウンタを 2 点装備。1ms タイマ装備。高速ラインにも充分対応できます。
4. **豊富な応用機能**
   割り込み機能、繰り返し機能、サブルーチン機能、ダイレクトクロックパルス発生機能、パルス幅測長機能、アナログタイマ機能など豊富な特殊機能を内蔵。
   スキャンタイムの影響を受けないリアルタイム制御なども可能です。
5. **MOS-FET 出力**
   KZ-40T、24T、16T の出力には光 MOS-FET を採用。ロジック信号、アナログ信号の制御も可能です。
6. **拡張ユニット**
   小型拡張ユニットを 4 台まで増設可能。(KZ-40、80 は 3 台まで可能)
   導入後の改善時や急な仕様変更にも対応できます。
7. **メンテナンスの省力化**
   EEPROM の採用によりメモリバックアップが不要です。
8. **メモリカード対応**
   メモリカードにプログラムを保存することができます。
   プログラムの変更時や同一のプログラムを複数の KZ で使う時に便利です。
9. **KZ ラダーサポートソフト**
   パソコンを使って、プログラムの作成、モニタが可能なソフトです。

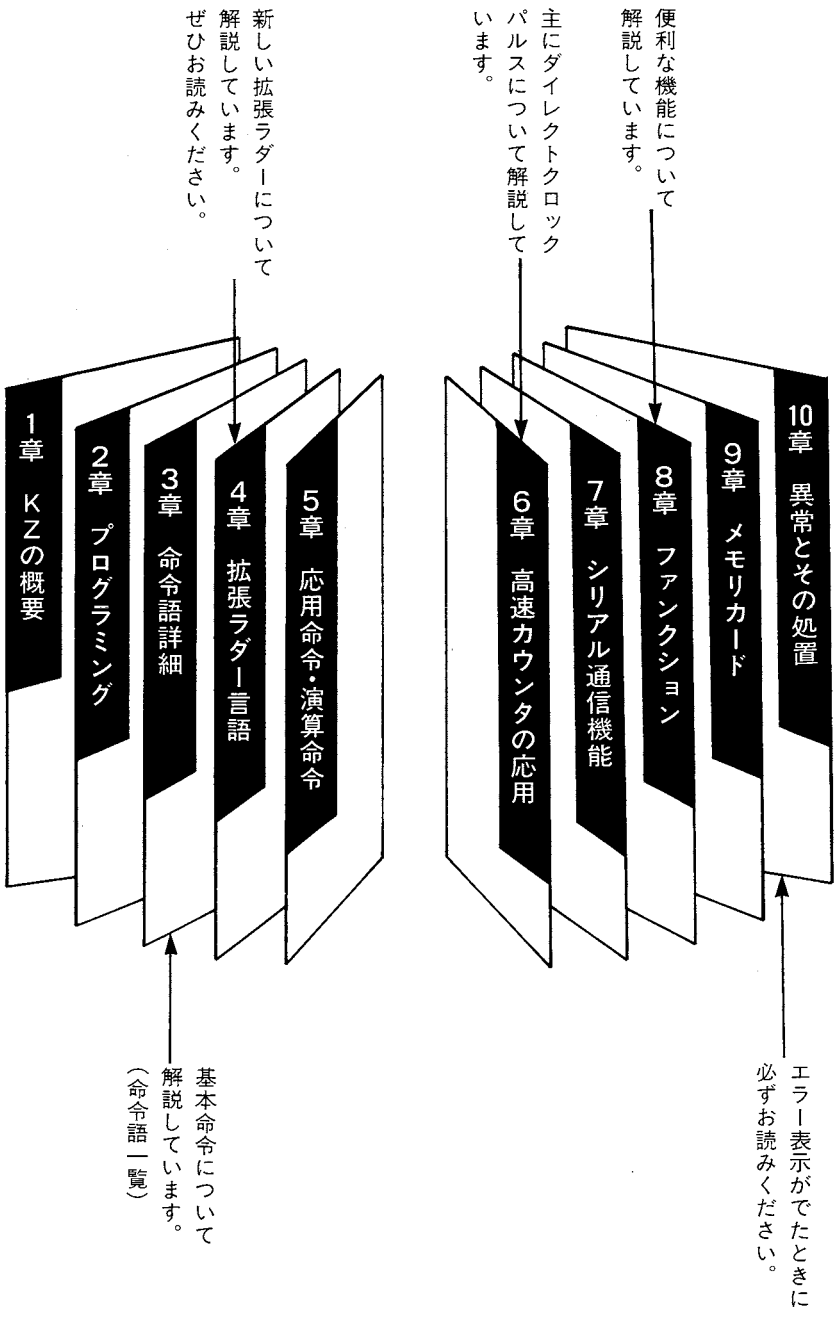
■お願い

下記に示すような条件や環境で使用する場合は定格、機能に対して余裕を持った使い方やフェールセーフなどの安全対策への配慮をいただくとともに当社営業担当までご相談ください。

- 本マニュアルに記載のない条件や環境での使用
- 原子力制御・鉄道施設・航空施設・車両・燃焼装置・医療機器・娯楽機械・安全機器などへの使用
- 人命や財産に大きな影響が予測され、特に安全性が要求される用途への使用

(1)本書の内容の一部または全部を無断転載することは禁止されています。
(2)本書の内容に関しては改良のため予告なしに変更することがありますのでご了承ください。
(3)本書の内容については万全を期して作成いたしましたが、万一ご不審な点や誤り、記載もれなどお気付きのことがありましたら巻末記載の営業所までご連絡ください。
(4)運用した結果の影響については(3)項にかかわらず責任を負いかねますのでご了承ください。
(5)乱丁・落丁はお取り替えいたします。

本紙の構成は以下のようになっています



●プログラマブルコントローラの基本についてよりわかりやすく解説したビギナーズマニュアルを別途用意しています。最寄りの営業所までお問い合せください。

目　次

1章　KZの概要

1-1. KZの取扱方法

- KZとは？ …… 1－1
- システム構成 …… 1－2
- 各部の名称とはたらき …… 1－3

1-2. 仕　　様

- 型式一覧・性能仕様 …… 1－4
- 一般仕様・入出力仕様 …… 1－5
- 入出力仕様/入出力回路構成 …… 1－6

1-3. 拡張ユニットの接続

- 拡張ユニットの接続 …… 1－7
- 拡張ユニットのリレー番号 …… 1－8

1-4. 保守と設置環境

- リレーの交換 …… 1－9
- 設置環境 …… 1－10

1-5. 入出力配線

- 入出力配線図 …… 1－11
- 外形寸法図 …… 1－15

2章　プログラミング

2-1. プログラミングの前に

- プログラムの作成手順 …… 2－1
- 入出力処理と応答遅れ …… 2－2
- プログラムの領域 …… 2－3

2-2. リレー番号の割付と機能解説

- 入力リレー・出力リレー …… 2－4
- 内部補助リレー …… 2－5
- 特殊補助リレー …… 2－6
- リレー番号一覧と特殊補助リレー一覧 …… 2－7
- タイマ・カウンタ …… 2－9
- データメモリ・テンポラリデータメモリ …… 2－10

2-3. プログラミングコンソールの取扱方法

- プログラミングコンソールの概要 …… 2－11
- プログラミングコンソール各キーの機能 …… 2－12
- プログラミングコンソール取扱上のご注意 …… 2－13

2-4. プログラミングの基本操作

- プログラミングの基本操作 …… 2－14
- クリア・カーソルの移動 …… 2－16
- 行の移動・オールクリア (FUN60) …… 2－17
- プログラムの「書込」 …… 2－18
- プログラムの「挿入」「削除」 …… 2－20
- プログラムの「検索」 …… 2－21
- モニタ・サイクルタイムモニタ …… 2－24
- 状態モニタ …… 2－25
- 導通モニタ …… 2－27

3章　命令語詳細

3-1. 命令一覧

- 基本命令 …… 3－1
- 応用命令 …… 3－3
- 演算命令 …… 3－5
- 割込み命令 …… 3－10
- 命令語 FUN NO 一覧 …… 3－10

3-2. 命令の読み方

…… 3－11

3-3. 基本命令

- ロード/ロードバー …… LD/LDB …… 3－12
- アンド/アンドバー …… AND/ANB …… 3－13
- オア/オアバー …… OR/ORB …… 3－14
- アンドロード/オアロード …… ANL/ORL …… 3－15
- アウト/アウトバー …… OUT/OUB …… 3－17
- セット/リセット …… SET/RES …… 3－18
- タイマ …… TMR …… 3－19
- 高速タイマ …… TMH (49) …… 3－20
- 高速 1ms タイマ …… TMS (51) …… 3－20
- カウンタ …… C …… 3－22
- アップダウンカウンタ …… UDC (52) …… 3－24
- 立上り微分/立下り微分 …… DIFU (10)/DIFD (09) …… 3－25
- キープ …… KEEP (22) …… 3－26
- シフト …… SFT (39) …… 3－27
- ハイスピード …… HSP (18) …… 3－29
- マスターコントロール/マスターコントロールリセット … MC (24)/MCR (25) …… 3－30
- メモリスイッチ …… MEMSW (26) …… 3－31
- ノップ …… NOP (30) …… 3－33
- エンド/エンドハイ …… END/ENDH …… 3－33

3-4. プログラム作成のポイント

…… 3－34

4章　拡張ラダー言語

4-1. 拡張ラダー言語の特長

…… 4－1

4-2. 新しい機能と変更点

…… 4－5

注：(　) 内の数字は FUN NO を表わします。

5章　応用命令・演算命令

5-1. 応用命令

- ウエイトオン/ウエイトオフ …… W-ON/W-OF …… 5－1
- ウエイトアップエッジ/ウエイトダウンエッジ …… W-UE/W-DE …… 5－2
- コネクト …… CON (06) …… 5－6
- プッシュ/リード/ポップ …… MPS/MRD/MPP …… 5－7
- ステージ/ジャンプ/エンドステージ …… STG (44)/JMP (21)/ENDS (14) …… 5－9
- ステップ/ステップエンド …… STP (45)/STE (43) …… 5－13
- インターバルタイマ …… ITVL …… 5－15
- 高速カウンタ/コンパレータ …… CTH/CTC …… 5－17
- サブルーチンコール/サブルーチンエントリ/サブルーチンリターン …… CALL (03)/SBN (38)/RET (33) …… 5－20
- 繰り返し範囲開始/終了 …… FOR (16)/NEXT (29) …… 5－22
- 16 キー入力 …… HKEY (17) …… 5－24

5-2. 演算命令

- 内部レジスタ …… 5－26
- データメモリライト …… DW …… 5－27
- トリマイン …… TMIN (50)/@ TMIN (50) …… 5－28
- ロード A …… LDA (23)/@ LDA (23) …… 5－29
- ストア A …… STA (42)/@ STA (42) …… 5－29
- コンペア …… CMP (04)/@ CMP (04) …… 5－33
- アッド (加算) …… ADD (00)/@ ADD (00) …… 5－34
- サブ (減算) …… SUB (46)/@ SUB (46) …… 5－34
- マルチ (乗算) …… MUL (28)/@ MUL (28) …… 5－34
- デバイド (除算) …… DIV (11)/@ DIV (11) …… 5－34
- アンド A …… ANDA (01)/@ ANDA (01) …… 5－37
- オア A …… ORA (31)/@ ORA (31) …… 5－38
- エクスクルーシブオア A …… EORA (15)/@ EORA (15) …… 5－39
- シフトライト A …… SRA (41)/@ SRA (41) …… 5－40
- シフトレフト A …… SLA (40)/@ SLA (40) …… 5－40
- ローテートライト A …… RRA (37)/@ RRA (37) …… 5－41
- ローテートレフト A …… RLA (35)/@ RLA (35) …… 5－41
- コンプリメント …… COM (05)/@ COM (05) …… 5－42
- インクリメントメモリ …… INC (19)/@ INC (19) …… 5－43
- デイクリメントメモリ …… DEC (07)/@ DEC (07) …… 5－43
- マルチプレクス …… MPX (27)/@ MPX (27) …… 5－44
- デマルチプレクス …… DMX (12)/@ DMX (12) …… 5－44
- トランスファ BCD …… TBCD (47)/@ TBCD (47) …… 5－45
- トランスファ BIN …… TBIN (48)/@ TBIN (48) …… 5－45
- アスキーコード変換 …… ASC (02)/@ ASC (02) …… 5－46
- アスキーコード逆変換 …… RASC (32)/@ RASC (32) …… 5－46
- 平方根 …… ROOT (36)/@ ROOT (36) …… 5－47
- 微分型命令 …… 5－48

5-3. 割込み命令

- 割込み禁止/許可 …… DI (08)/EI (13) …… 5－49
- イント/リターンイント …… INT (20)/RETI (34) …… 5－50

5-4. 演算命令による特殊補助リレーの変化 …… 5－54

5-5. データメモリ・リレーの間接指定について …… 5－57

注：(　) 内の数字は FUN NO を表わします。

6章　高速カウンタの応用

6-1. 高速カウンタの概要 …… 6－1
6-2. ダイレクトクロックパルス出力 …… 6－3
6-3. ダイレクトクロックパルス使用例 …… 6－9
6-4. ダイレクトクロックパルス応用例 …… 6－14

7章　シリアル通信機能

7-1. KZとパソコンの通信 …… 7－1
7-2. テキストデータの取り込み …… 7－9
7-3. 参考プログラム …… 7－11

8章　ファンクション

- ファンクション番号一覧 …… 8－1
- オールクリア (FUN60)
- プロコンクリア (FUN61) …… 8－2
- カウンタクリア (FUN62)
- 高速カウンタクリア (FUN63) …… 8－3
- データメモリオールクリア (FUN64)
- 保持リレーオールクリア (FUN65) … 8－4
- プログラム送受信 (FUN66) …… 8－5
- オフラインエディタ開始 (FUN67)
- オフラインエディタ終了 (FUN68) … 8－6
- タイマ・カウンタ現在値変更 (FUN69) …… 8－7
- タイマ・カウンタ設定値変更 (FUN70) …… 8－8
- リレーの ON/OFF (FUN71) …… 8－11
- データメモリ書込み (FUN72) …… 8－12
- トリマ設定値読出 (FUN73)
- 文法チェック (FUN74) …… 8－13
- プログラム容量確認 (FUN75) …… 8－14

9章　メモリカード

- メモリカードの機能と容量 …… 9－1
- メモリカードオールクリア …… 9－2
- メモリカードへの新規追加 …… 9－3
- メモリカードアクセス …… 9－4
- KZ 間のプログラム転送 …… 9－8
- メモリカード取扱上のご注意 …… 9－8

10章　異常とその処置

- エラーメッセッジ一覧 …… 10－1

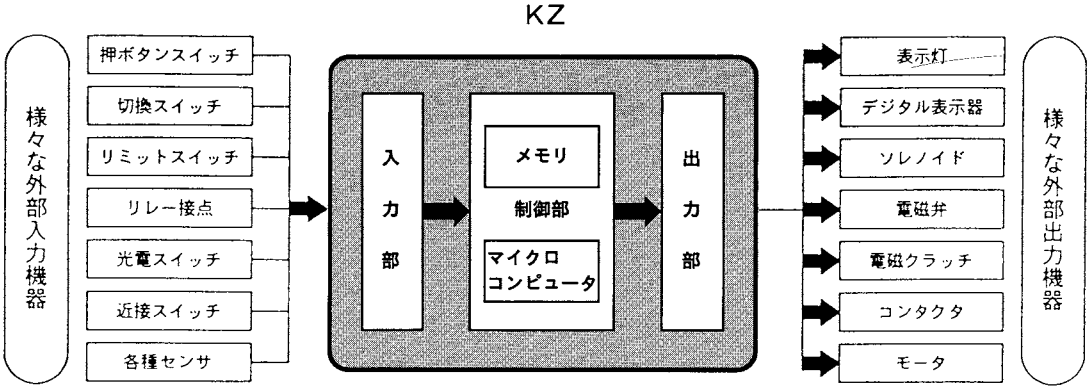
付　録
- 逆引き目次
- コーディングシート
- I/O 番号割付シート

1章　KZの概要

KZとは?

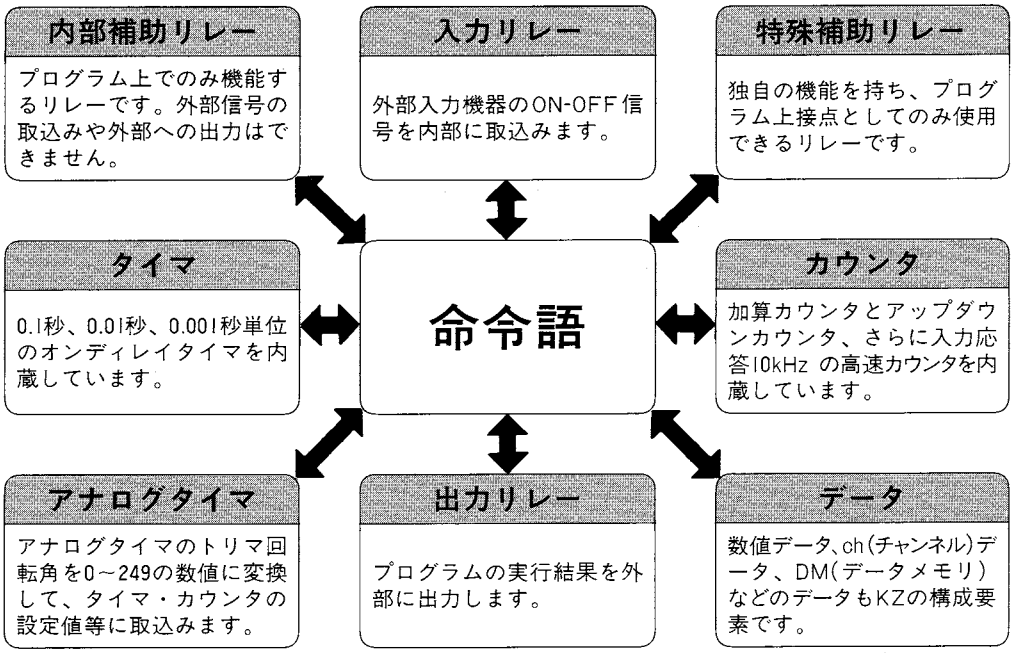
KZの機能

KZは、小型ながらマイクロコンピュータ(CPU：中央処理回路)とメモリ(記憶素子)を中心に構成されたプログラマブルコントローラです。
従来、リレーやタイマ、カウンタなどを複雑に結線しなければ組み上げられなかったシーケンス回路を、簡単なプログラミング作業だけでいともたやすく実現することができます。



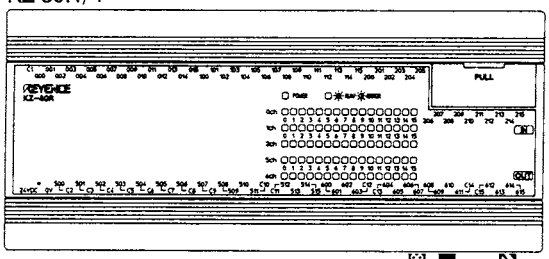
KZのしくみ

KZは、数多くの接点(a接点、b接点)を持ったリレーや、タイマ、カウンタなどの集合体である、といえます。これらの要素を様々な命令語を用いて組合せ、シーケンス回路を構成します。このシーケンス回路は、リレー回路のように、はんだゴテやドライバを使って配線するのではなく、あくまでもプログラムによって回路を組みます。

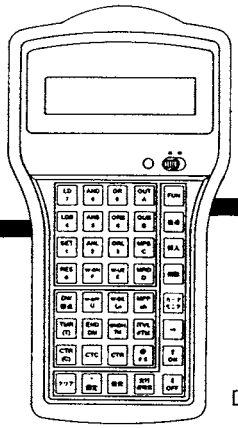


システム構成

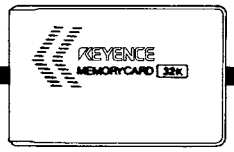
KZ-80R/T



プログラミングコンソール
KZ-P3



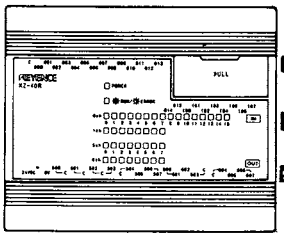
メモリカードM-2



メモリカードM-3



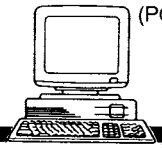
KZ-40R/T



DOS/Vパソコン
(PC/AT互換機)

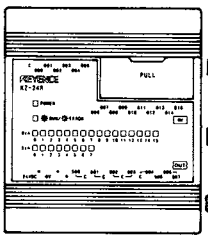
KZ-H6W
(Windows版) KZ-3H4V



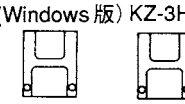


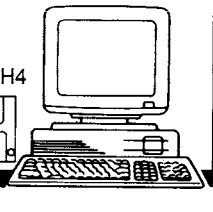
NEC PC9801シリーズ

KZ-24R/T



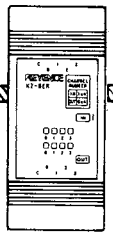
KZ-H6W
(Windows版) KZ-3H4



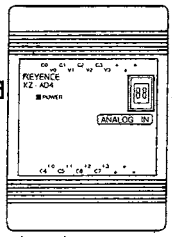




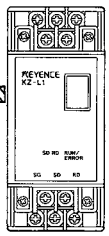
拡張ユニット（電源不要）
KZ-8ER/T、KZ-8EX、
KZ-8EYR/T



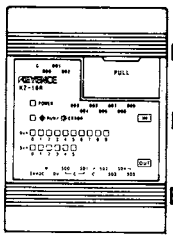
特殊ユニット
A/D, D/A 変換ユニット
KZ-AD4/DA4



カードリーダライタ※
Z-1
KZ-L1

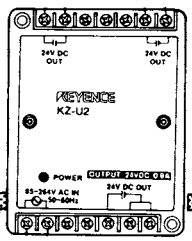


KZ-16R/T

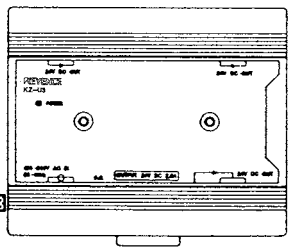


ユーザーズマニュアル参照
拡張ユニットは KZ-40、80 が 3 台まで、
他は 4 台まで接続可能。

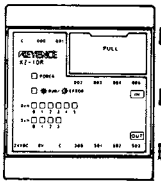
電源 KZ-U2



KZ-U3



KZ-10R/T

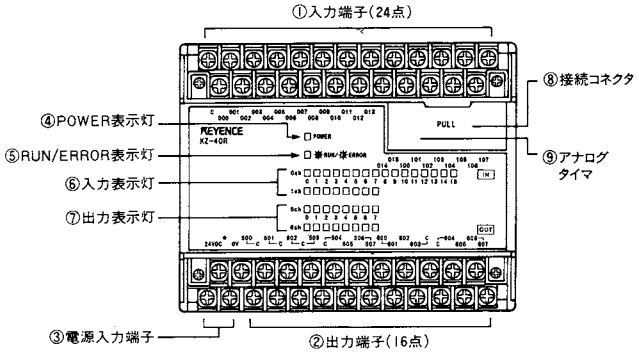


※ KZ-H6W では使用できません。

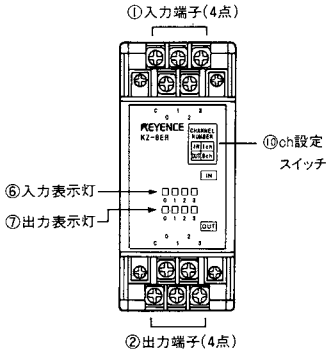
各部の名称とはたらき

各部の名称とはたらき

KZ-40R の場合



KZ-8ER の場合



①入力端子
- 入力仕様はDC24Vです。

②出力端子
- 0500、0501はダイレクトクロックパルス出力を兼ねています。

③電源入力端子
- DC24Vを供給します。

④POWER表示灯
- 電源入力時に点灯します。

⑤RUN/ERROR表示灯
- 運転時に点灯します。
- 運転時に異常が検出された時に点滅します。

⑥入力表示灯
- 入力 ON 時に点灯します。

⑦出力表示灯
- 出力ON時に点灯します。

⑧接続コネクタ
- プログラミングコンソールまたは、パソコンとつなぐ場合、カバーを開けてケーブルを接続して下さい。

⑨アナログタイマ
- カバーを開けるとトリマがあります。
- トリマを回すことにより、タイマ・カウンタの設定値や内部レジスタ値をプログラミングコンソールを使わずに簡単に変更することができます。
(詳しくは P5−28「トリマイン」をご覧下さい。)

過大な力をかけて回したり、頻繁な操作は故障の原因となりますのでご注意ください。

⑩ch設定スイッチ
- 拡張ユニットの入出力のch 番号の割り付けを設定します。
(詳しくは P1−8 をご覧下さい。)

本体ご購入時コンソールを接続せずに電源を入れますと ERROR 表示が点滅しますが問題はありません。
コンソールを接続しプログラムを行って下さい。

型式一覧・性能仕様

■型式一覧

種類	基本ユニット									
型式	KZ-10R	KZ-10T	KZ-16R	KZ-16T	KZ-24R	KZ-24T	KZ-40R	KZ-40T	KZ-80R	KZ-80T
入力点数	6点		10点		16点		24点		48点	
出力点数	4点		6点		8点		16点		32点	
出力形態	リレー	トランジスタ	リレー	MOS-FET	リレー	MOS-FET	リレー	MOS-FET	※リレー	トランジスタ

種類	拡張ユニット							
型式	KZ-8ER	KZ-8ET	KZ-8EX	KZ-8EYR	KZ-8EYT	KZ-16EX	KZ-16EYR	KZ-16EYT
入力点数	4点		8点	—		16点	—	
出力点数	4点		—	8点		—	16点	
出力形態	リレー	トランジスタ	—	リレー	トランジスタ	—	※リレー	トランジスタ

■性能仕様

型式	KZ-10	KZ-16	KZ-24	KZ-40	KZ-80
演算制御方式	ストアードプログラム方式				
プログラム言語	ラダーチャート方式+拡張ラダーチャート方式				
命令の種類	基本命令：16種　応用命令：34種　演算命令：26種　割込命令：4種(8プログラム)				
実行速度(基本接点命令)	最小 1.0μs　平均 1.92μs		最小 1.4μs　平均 3.12μs		
プログラム容量(平均)	500ステップ		3000ステップ		
基本入力	6点	10点	16点	24点	48点
基本出力	4点	6点	8点	16点	32点
拡張ユニットの増設	最大4台			※※最大3台	
拡張ユニット増設時の最大入出力	74点 入力のみI/O=70/4点 出力のみI/O=6/68点	80点 入力のみI/O=74/6点 出力のみI/O=10/70点	88点 入力のみI/O=80/8点 出力のみI/O=16/72点	88点 入力のみI/O=72/16点 出力のみI/O=24/64点	128点 入力のみI/O=80/32点 出力のみI/O=48/80点
内部補助リレー(保持可)	160点		800点		
特殊補助リレー	160点				
データメモリ(16ビット)	1000ワード		2000ワード		
一時メモリ(16ビット)	32ワード				
タイマ/カウンタ	0.1秒タイマ　(0～6553.5秒) 0.01秒タイマ　(0～655.35秒) 0.001秒タイマ　(0～65.535秒) アナログタイマ　(0～24.9秒または 0～2.49秒または 0～0.249秒) 1点 加算カウンタ アップダウンカウンタ　合計64点		0.1秒タイマ　(0～6553.5秒) 0.01秒タイマ　(0～655.35秒) 0.001秒タイマ　(0～65.535秒) アナログタイマ　(0～24.9秒または 0～2.49秒または 0～0.249秒) 2点 加算カウンタ アップダウンカウンタ　合計120点		
高速カウンタ	オートリセット加算カウンタ(入力応答最大10kHz) 2点				
高速カウンタコンパレータ	4点				
ダイレクトクロックパルス(トランジスタ、MOS-FET出力のみ)	2点　2kHz以下(出力500)　1.5kHz以下(出力501)				2点 20kHz以下(出力500) 1.5kHz以下(出力501)
メモリスイッチ	16点				
停電保持機能	プログラムメモリ：EEPROM 10年以上、5万回以上書替え可能				
	データメモリ：大容量コンデンサにより2ヵ月以上(25°C)				
自己診断機能	CPU異常、RAM異常				
周辺装置	プログラミングコンソールKZ-P3、メモリカードM-2・M-3、カードリーダライタZ-1				

※リレー交換可能

※※KZ-80では拡張ユニットの入力は2ch(0300～0315、0400～0415)のため入力のある拡張ユニットの接続は2台までです。

一般仕様・入出力仕様

■一般仕様

電源電圧		DC 24V (+10% −20%)		
内部消費電流（プログラミングコンソール、入力回路駆動用電流を除く）	基本ユニット	KZ-10R：75mA 以下 KZ-10T：65mA 以下 KZ-40R：220mA 以下 KZ-40T：115mA 以下	KZ-16R：105mA 以下 KZ-16T：70mA 以下 KZ-80R：400mA 以下 KZ-80T：300mA 以下	KZ-24R：130mA 以下 KZ-24T：75mA 以下 KZ-P3：65mA 以下
	拡張ユニット	KZ-8ER：35mA 以下 KZ-8ET：20mA 以下	KZ-8EX：15mA 以下 KZ-16EX：25mA 以下	KZ-8EYR：80mA 以下 KZ-8EYT：60mA 以下 KZ-16EYR：150mA 以下 KZ-16EYT：100mA 以下
使用周囲温度		0～50°C (氷結しないこと) 0～45°C (KZ-P3)		
使用周囲湿度		35～85% RH (結露しないこと)		
耐電圧		AC 1500V 1分間 (電源端子と入出力端子間および外部端子一括とケース間)		
耐ノイズ性		1500Vp-p 以上 パルス幅：1μs、50ns (ノイズシミュレータによる)		
耐振動		10～55Hz 複振幅 1.5mm 以下 X、Y、Z 各方向 2 時間		
絶縁抵抗		50MΩ 以上 (DC 500V メガにて電源端子と入出力端子間および外部端子一括とケース間)		
使用雰囲気		じんあい、腐食性ガスがひどくないこと		
重量	基本ユニット	KZ-10R：約 130g KZ-10T：約 120g KZ-40R：約 340g KZ-40T：約 270g	KZ-24R：約 250g KZ-24T：約 220g KZ-80R：約 600g KZ-80T：約 500g	KZ-16R：約 200g KZ-16T：約 180g KZ-P3：約 230g
	拡張ユニット	KZ-8ER：約 120g KZ-8ET：約 100g	KZ-8EX：約 130g KZ-16EX：約 180g	KZ-8EYR：約 140g KZ-8EYT：約 130g KZ-16EYR：約 240g KZ-16EYT：約 190g

電源ユニット仕様

型式		KZ-U2	KZ-U3
方式		スイッチング方式	
入力電源電圧		AC 100～240V±10% (50/60Hz)	
出力電圧		DC 24V±10%	
リップル・ノイズ		240mVp-p 以下	
出力容量		0.8A	1.4A
耐環境性	使用周囲温度	0～50℃ (氷結しないこと)	
	使用周囲湿度	35～85% RH (結露しないこと)	
耐電圧		AC 1500V 1分間 (外部端子一括とケース間)	
耐振動		10～55Hz 複振幅 1.5mm X, Y, Z 各方向2時間	
絶縁抵抗		50MΩ以上 (DC 500V メガで、外部端子一括とケース間)	
消費電力 (電流)		70VA	110VA
瞬停時間		20ms 以下	
重量 (専用金具を除く)		約170g	約300g

入出力仕様/入出力回路構成

■入力仕様

入力定格	DC24V 7mA(入力000～005) 5mA(その他)
最小ON電圧	19V
最大OFF電流	2mA
入力時定数 (プログラムにより切替え可能)	25μs/8～12ms(OFF→ON) 75μs/8～12ms(ON→OFF)
割込み入力応答 入力000～003	25μs(OFF→ON) 35μs(ON→OFF)
高速カウンタ入力応答	10kHz以下

基本ユニットのみです。

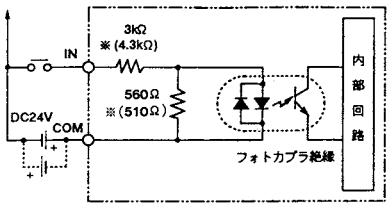
■出力仕様

出力形態	リレー	トランジスタ	MOS-FET
定格負荷	AC250V/DC30V 2A	DC30V 0.3A	DC30V 0.6A(出力500) DC30V 0.5A(その他)
ピーク負荷電流	5A	―	1A
OFF時最大電圧	―	DC30V	DC30V
OFF時漏れ電流	―	100μA以下	5μA以下
ON時残留電圧	―	1V	―
ON抵抗	50mΩ以下	―	1Ω以下
動作時間OFF→ON ON→OFF	10ms以下 10ms以下	50μs 以下 (KZ-80T の 500 は 3μs 以下) 450μs 以下 (KZ-80T の 500 は 20μs 以下)	350μs以下(出力500、501) 2ms以下(その他) 150μs以下

リレー寿命：機械的2,000万回以上
電気的 10万回以上(頻度20回/分)

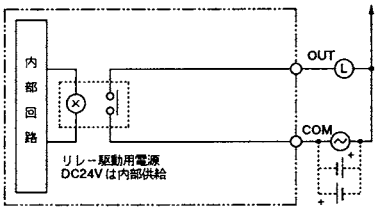
■入出力回路構成

●入力回路



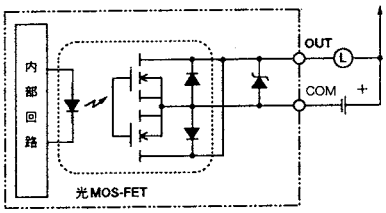
●※は0000～0005以外の入力端子。
●NPN入力(⊕コモン)、PNP入力(⊖コモン)のどちらも使用可能です。

●出力回路(リレー接点出力)



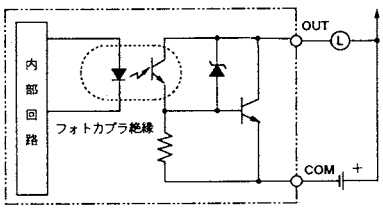
●KZ-80R、KZ-40R、KZ-24R、KZ-16R、KZ-10R、KZ-8ER、KZ-8EYR、KZ-16EYR の出力のみ。
●負荷用電源は別途ご用意下さい。

●出力回路(MOS-FET出力)



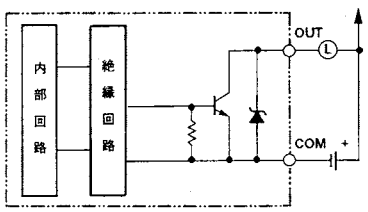
●KZ-40T、KZ-24T、KZ-16Tの出力のみ。
●負荷用電源は別途ご用意下さい。

●出力回路(トランジスタ出力)



●KZ-80T (0500 以外)、KZ-10T、KZ-8ET、KZ-8EYT、KZ-16EYT の出力のみ。
●負荷用電源は別途ご用意下さい。

●出力回路 (KZ-80T 0500 のみ)

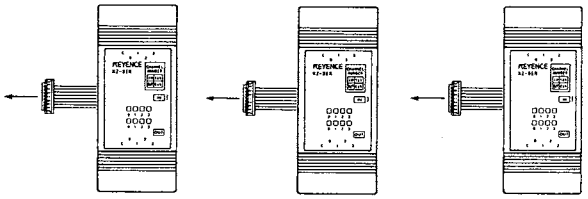


●負荷用電源は別途ご用意下さい。

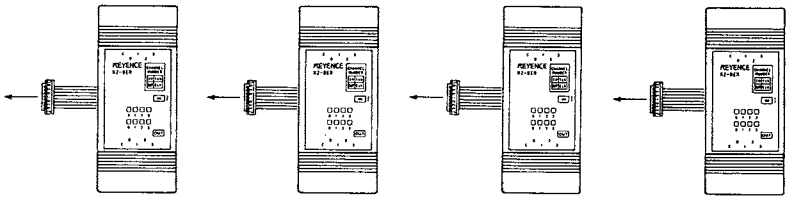
拡張ユニットの接続

拡張ユニットの接続

●KZ-80、40　拡張ユニット KZ-16EX、16EY R/T、KZ-8E R/T、8EX、8EY R/T



●KZ-24、16、10　拡張ユニット KZ-16EX、16EY R/T、KZ-8E R/T、8EX、8EY R/T

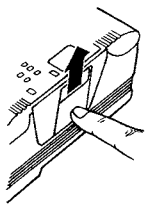


ケーブル長約 30mm

●拡張ユニットは上図のように基本ユニット 1 台に対して、KZ-80、40 は 3 台、KZ-24、16、10 は 4 台まで接続できます。(KZ-80 では入力のある拡張ユニットの接続は 2 台まで)
●拡張ユニットはすべての基本ユニットに共通です。また 8 点タイプ、16 点タイプの各ユニットの同時接続やリレー接点出力とトランジスタ出力は混在して接続することができます。
●拡張ユニットのコネクタは接続不良のないように確実にソケットに挿入してください。
●拡張ユニットの接続、取り外し ch 設定は基本ユニットの電源を OFF にして行なってください。
●基本ユニットとまた拡張ユニット同志は密着取付が可能です。
●拡張ユニットケーブルの延長はできません。
●拡張ユニットの接続可能台数に KZ-L1 は含まれません。(KZ-L1 は基本ユニット 1 台につき 1 台まで接続可能です。)

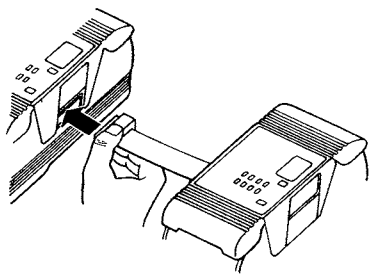
接続方法

① 基本ユニットまたは、拡張ユニットの右側面のふたを上にスライドします。



人指しゆびで上にスライドさせてください。

② コネクタを接続します。

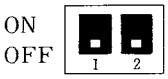


拡張ユニットのリレー番号

拡張ユニットの入出力リレー番号の割付

●拡張ユニットの入出力リレー番号はユニット表面の ch 設定スイッチで設定します。
●スイッチの設定は必ず電源を切った状態で行ってください。
●**スイッチの設定は各ユニット同士の重複設定はできません。**
(入力のみ、出力のみのユニットでもできません)
例えば KZ-10R の本体に入力ユニット (KZ-16EX)、出力ユニット (KZ-16EYT) を接続する場合、KZ-16EX を (スイッチ 1:OFF 2:OFF) の設定にした場合 KZ-16EYT を (スイッチ 1:OFF 2:OFF) と設定することはできません。この設定以外にしてください。
●スイッチの設定が終ったら、付属の ch 番号シールに番号を記入し、スイッチの上に貼ってください。

<ch設定スイッチ>



●KZ-40、24、16、10 に接続する場合

スイッチ		拡張ユニットの入出力リレー番号									
1	2	KZ-8E R/T		KZ-8EX		KZ-8EY R/T		KZ-16EX		KZ-16EY R/T	
		入力	出力	入力	出力	入力	出力	入力	出力	入力	出力
OFF	OFF	100～103	600～603	100～107			600～607	100～115			600～615
ON	OFF	200～203	700～703	200～207			700～707	200～215			700～715
OFF	ON	300～303	800～803	300～307			800～807	300～315			800～815
ON	ON	400～403	900～903	400～407			900～907	400～415			900～915

●KZ-40 では (スイッチ 1:OFF 2:OFF) の設定 (入力 100～115、出力 600～615) はできません。

●KZ-80 に接続する場合

スイッチ		拡張ユニットの入出力リレー番号									
1	2	KZ-8E R/T		KZ-8EX		KZ-8EY R/T		KZ-16EX		KZ-16EY R/T	
		入力	出力	入力	出力	入力	出力	入力	出力	入力	出力
OFF	OFF	—	—	—			—	—			—
ON	OFF	300～303	700～703	300～307			700～707	300～315			700～715
OFF	ON	400～403	800～803	400～407			800～807	400～415			800～815
ON	ON	—	900～903	—			900～907	—			900～915

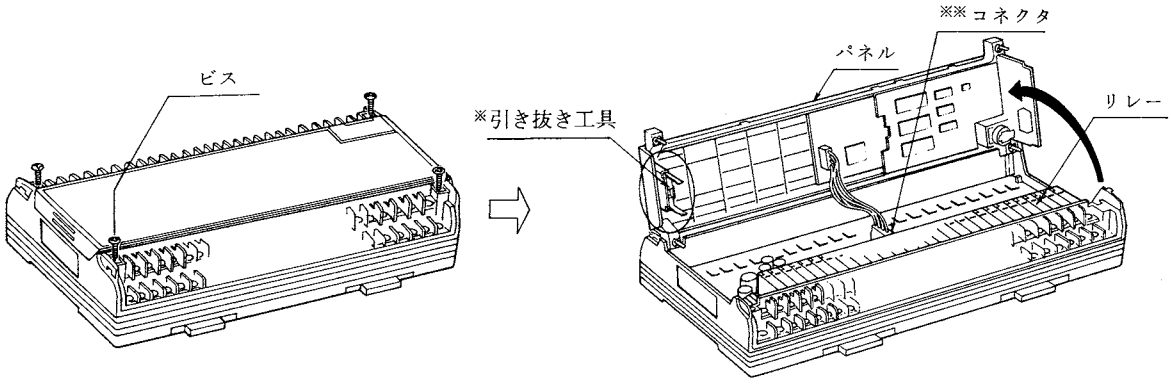
●KZ-80 では (スイッチ 1:OFF 2:OFF) の設定はできません。(入力のみの接続は 2 台まで接続可能。入力のあるユニットを 3 台接続したときは (スイッチ 1:ON 2:ON) の設定を行ったユニットは) 入力の使用ができません。)

リレーの交換

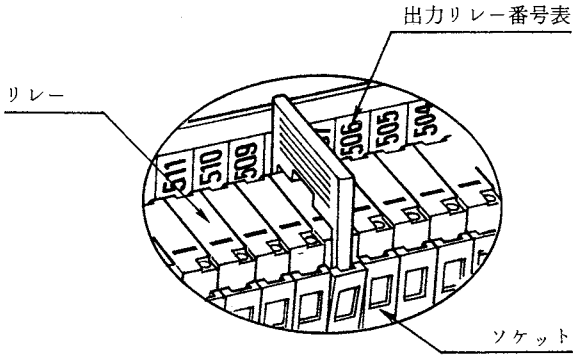
リレーの交換可能機種　　基本ユニット KZ-80R　　拡張ユニット KZ-16EYR

●以下の手順に従ってリレーの交換をしてください。
(電源を必ず OFF にしてください)

交換用リレー	OP-98592



① 4 隅のビスを均等にはずしていき、ゆっくりと真上にパネルを開けます。
② 輸送用保護テープをはがします。
(再びはる必要はありません)
③ 本体内側についている出力リレー番号表と不良リレーとをてらし合せます。
④ パネル裏側左についているリレー引き抜き工具を使用し、不良リレーとその下側にあるソケットとの間に引き抜き工具のツメを「カチッ」と音がするまで挿入し垂直に引き抜いてください。
⑤ 新しいリレーを向きに注意し(リレー上面の文字の方向を合せる)垂直に力が加わるように押し込んでください。
(注) リレーを逆挿入した場合は足が折れてしまいます。



⑥ リレーを交換したあとは逆の手順でパネルを閉め、ビスで固定してください。

●やむをえずコネクタをはずしてからリレーの交換を行う場合、コネクタ自身をラジオペンチなどを利用し垂直に引き抜いてください。ケーブルを直接引っぱったり、こじったりすると断線やコネクタ破損につながりますので注意してください。また交換したあとは必ずコネクタをさしこんでからパネルを閉めてください。
●パネルを閉めるとき、ケーブルをはさんだり、部品を曲げたりしないようにしてください。また内部に異物が入らぬように注意してください。

※ KZ-16EYR は別途付属しています。(パネルにはついていません)
※※ KZ-16EYR にはコネクタはありません。

KZ-16EYR に付属していますリレー引き抜き工具は最終的にご使用いただく方のお手元に届くようご配慮ください。

設置環境

設置環境

KZシリーズは高い信頼性をもったプログラマブルコントローラですが、システムを充分に機能させるために、以下の点を考慮のうえ設置してください。

設置場所

次のような場所は避けて設置してください。
- 直射日光があたる場所
- 周囲温度が0～50℃の範囲を超える場所
- 相対湿度が35～85%RHの範囲を超える場所
- 急激な温度変化により、結露が生じるような場所
- 腐食性ガスや可燃性ガスのある場所
- じんあい、塩分、鉄粉、油煙が多い場所
- 本体に振動や衝撃が直接加わるような場所
- 水、油、薬品などの飛沫がかかる場所
- 強磁界、強電界の発生する場所

盤内の取付位置

盤内の取付けは操作性、保守性、耐環境性を充分考慮してください。

●周囲温度に対する配慮
- 本体の使用周囲範囲は0～50℃です。
- プログラムコンソールを装着したままご使用になるときは、使用周囲範囲は0～45℃となりますのでご注意ください。

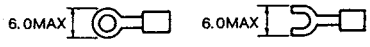
- 通風スペースを充分とってください。
- 発熱量の高い機器(ヒータ、トランス、大容量の抵抗)の真上に取付けることは避けてください。
- 周囲温度が50℃以上になるときは、強制ファンまたはクーラーを設置してください。

●耐ノイズ性能を向上させるための配慮
- 高圧機器の設置されている盤内での取付けは避けてください。
- 動力線とはなるべく離して取付けてください。
- ユニットを横並びに配置する場合、基本ユニットと拡張ユニット間にはダクトまたは配線を通さないでください。

●操作性、保守性に対する配慮
保守、操作の安全性を考え、高圧機器、動力機器からはできるかぎり離して取付けてください。

配線

基本ユニット及び拡張ユニットの端子ねじはM3.0を使用しています。
圧着端子を使用して配線される場合は次のものを使用してください。(単位はmm)

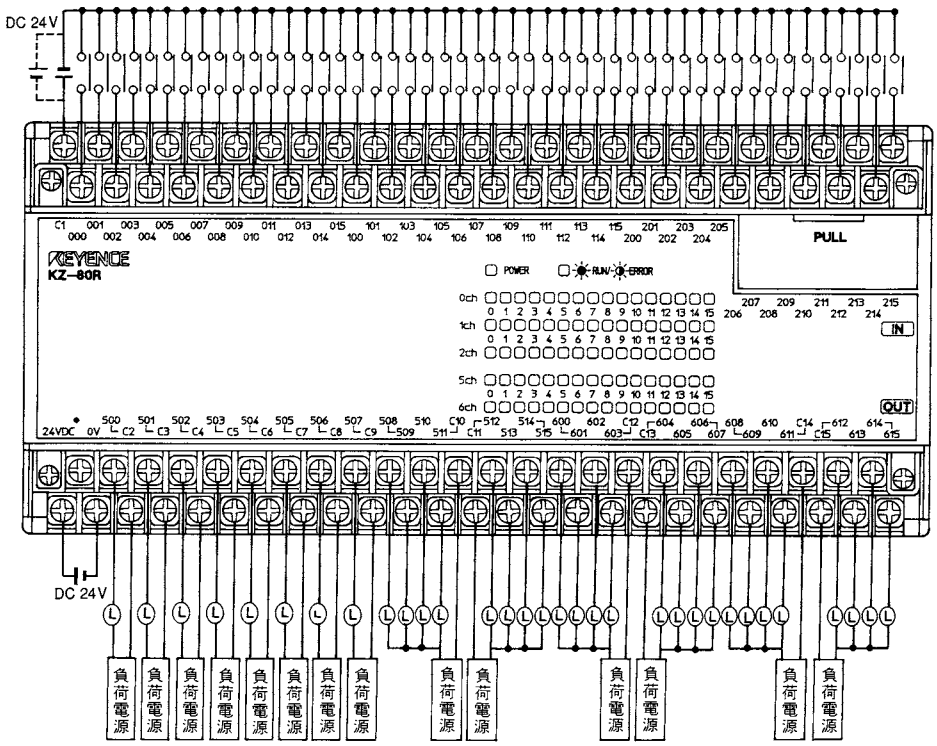


接点保護について

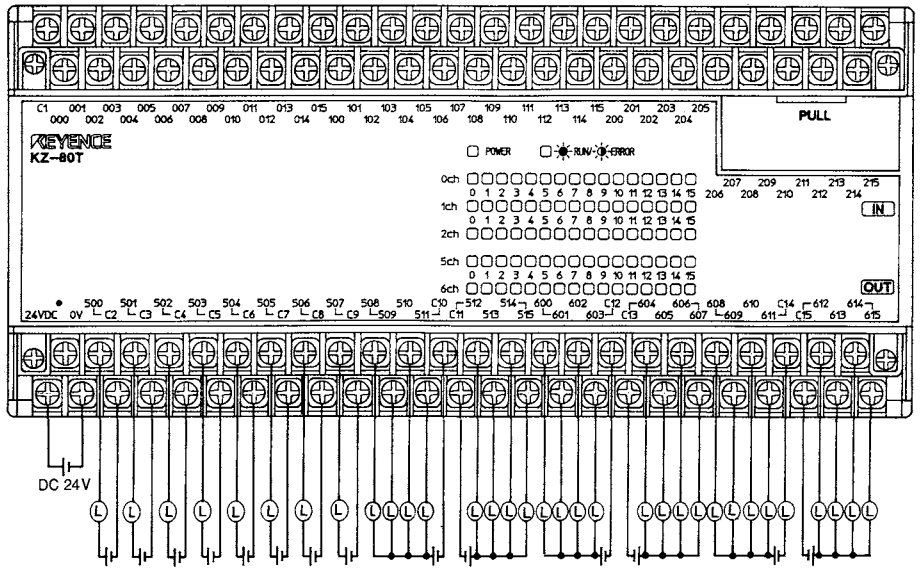
大型のリレー・電磁開閉器・モータ・ソレノイドなどの誘導性負荷や容量性負荷を開閉すると、接点定格を越えたON時の突入電流やOFF時の逆起電圧のために、接点電食や接点転移・溶着を引き起こし、接点の寿命が著しく短かくなります。
こういったことを防ぎ、KZシリーズを末永くご使用いただくために、接点保護回路を設けてください。
(詳しくはKZシリーズ取扱説明書をご参照ください)

入出力配線

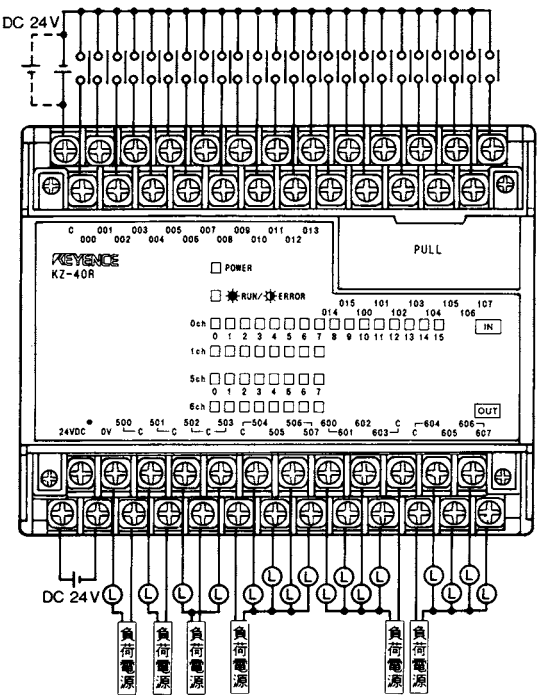
KZ-80R



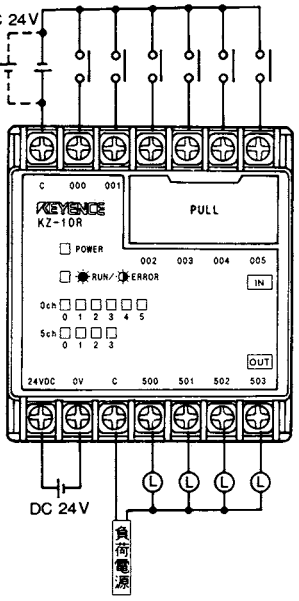
KZ-80T



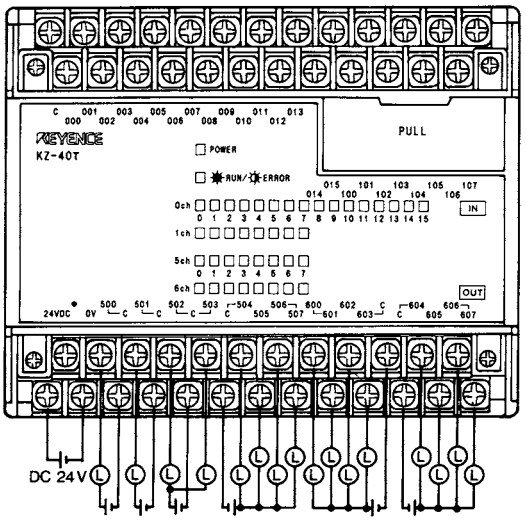
KZ-40R



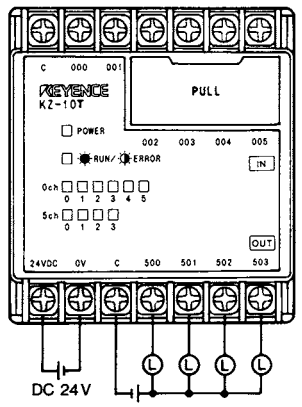
KZ-10R



KZ-40T

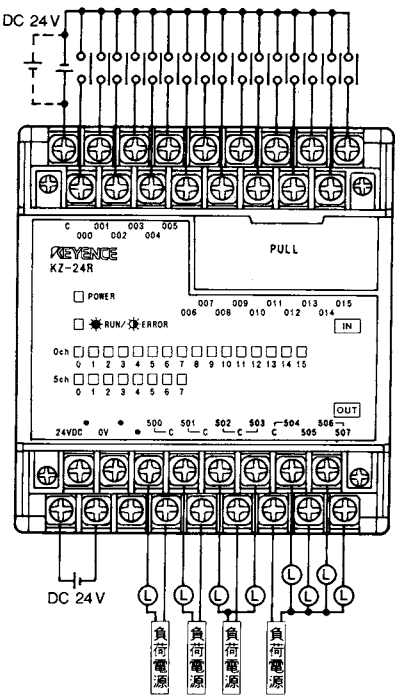


KZ-10T

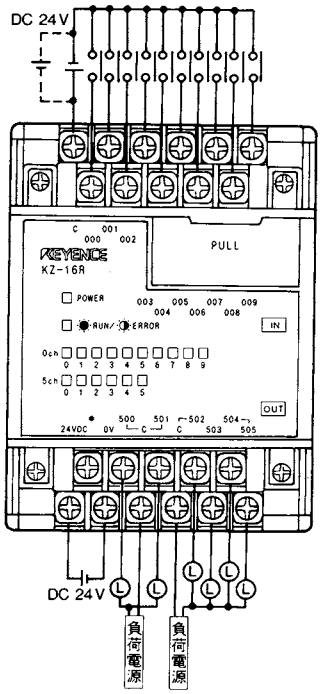


(注)●T タイプの入力配線は省略しています。
R タイプと同様に配線してください。
●基本ユニットの各 COM は独立しています。

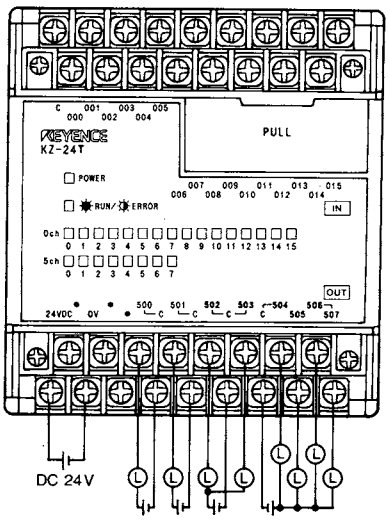
KZ-24R



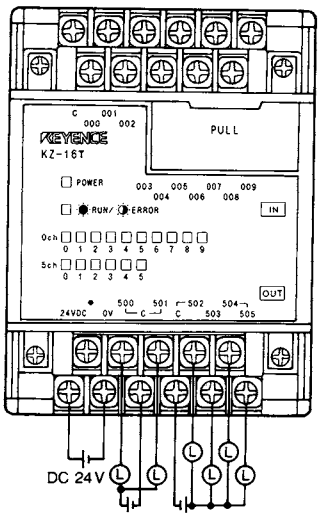
KZ-16R



KZ-24T

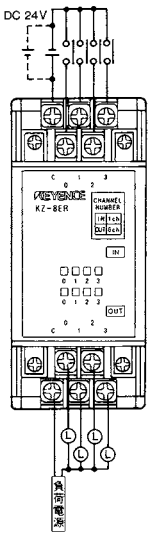


KZ-16T

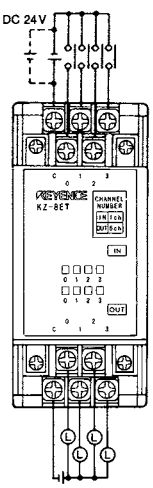


(注)●T タイプの入力配線は省略しています。
R タイプと同様に配線してください。
●基本ユニットの各 COM は独立しています。

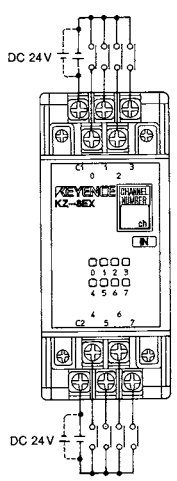
KZ-8ER



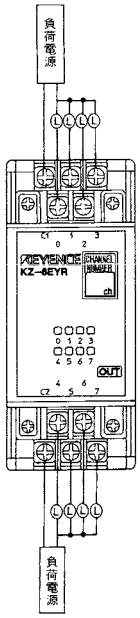
KZ-8ET



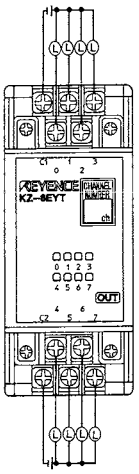
KZ-8EX



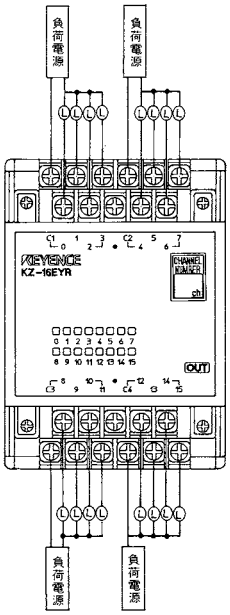
KZ-8EYR



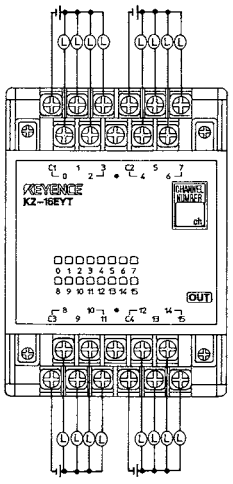
KZ-8EYT



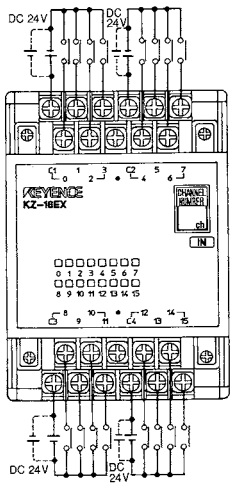
KZ-16EYR



KZ-16EYT



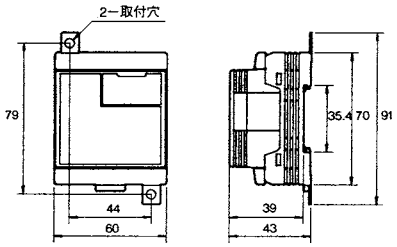
KZ-16EX



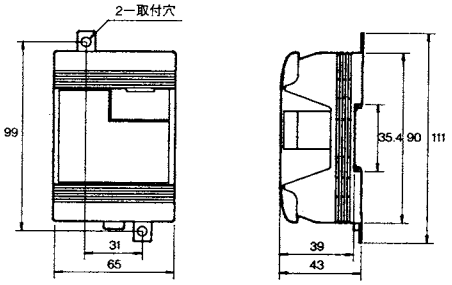
(注) 拡張ユニットの各 COM は独立しています。

外形寸法図

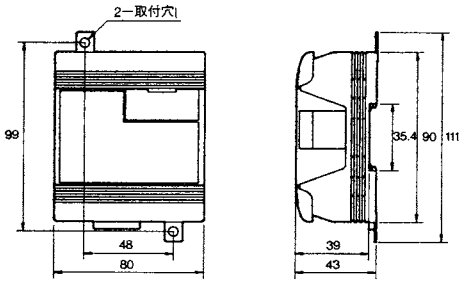
KZ-10R/T



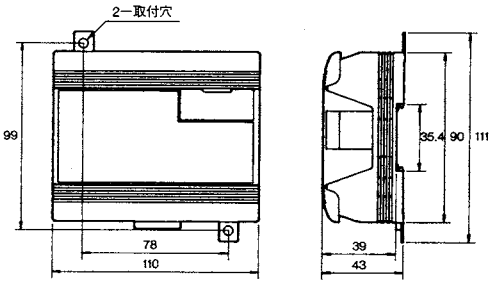
KZ-16R/T, 16EX, 16EYR/T



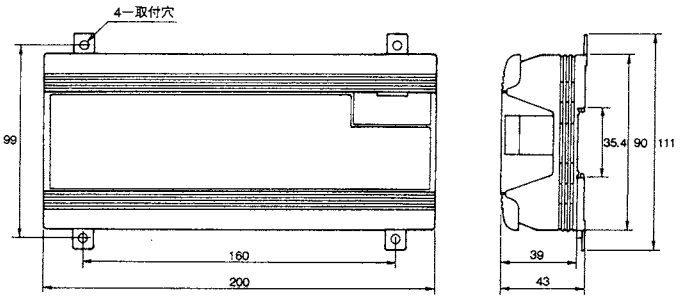
KZ-24R/T



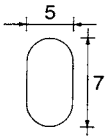
KZ-40R/T



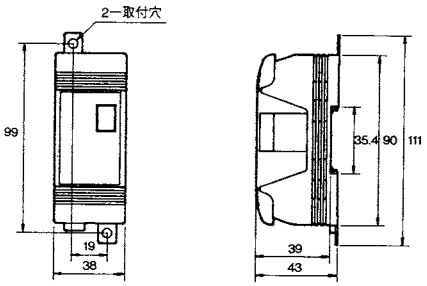
KZ-80R/T



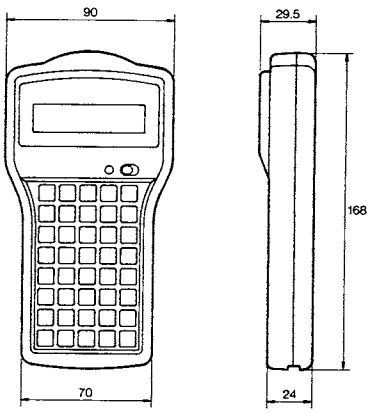
取付穴詳細



KZ-8ER/T, 8EX, 8EYR/T, L1

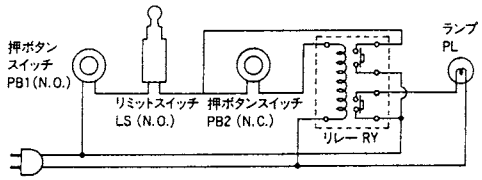


KZ-P3



2章　プログラミング

プログラムの作成手順



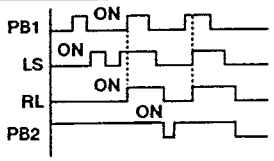
左図は、自己保持回路の一例です。押ボタンスイッチ PB1 とリミットスイッチ LS がともに ON の時ランプが点灯します。その後 PB1 と LS1 が、OFF になっても、点灯状態は保持されますが、PB2 が ON されると、ランプは消灯します。この回路を例に、プログラミングの手順をご紹介します。

シーケンス回路の検討

まず始めに、ＫＺにどのような作業をさせたいのか、ということを十分に検討してください。後々の追加・変更作業が削減できます。

タイムチャートの作成

必要に応じて作成してください。装置の動作状態や種々の条件がより明確になり、ラダー図が作成しやすくなります。上記自己保持回路をタイムチャートで表すと右図の様になります。(一例)

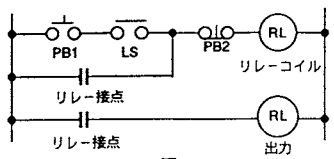


シーケンス回路の設計

ＫＺのラダー図は、リレー図と基本的に同じ考え方で作成できます。装置の動作順に大筋を設計し、後から細かな修正を加えます。

ラダー図の作成

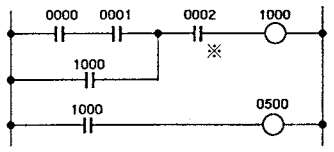
上記自己保持回路をリレーシンボルで表すと右図の回路になります。



入出力機器の割当

次に、入出力機器をＫＺ本体のどの入出力端子に割当てるかを決定します。
(通常は先に入出力機器を割当ててしまいます。)

入出力機器	要　素	番号
押ボタンスイッチ 1 (N.O.)	入力リレー	0000
リミットスイッチ (N.O.)	入力リレー	0001
押ボタンスイッチ 2 (N.C.)	入力リレー	0002
リレー	内部補助リレー	1000
ランプ	出力リレー	0500



見直し・修正

プログラミング

いよいよプログラミングです。プログラミングコンソールKZ-P3 と接続ケーブルを使用します。

コーディング

ラダー図にもとづいてコーディングリスト（命令の一覧表）を作成します。これはラダー図をＫＺが理解できる言語に翻訳する作業です。この言語をラダー言語といい、命令語とオペランドと呼ばれる要素番号から成り立っています。

コーディングリスト

行番号	命令	オペランド(要素番号)
0000	LD	0000
0001	AND	0001
0002	OR	1000
0003	AND	0002
0004	OUT	1000
0005	LD	1000
0006	OUT	0500
0007	END	
0008	ENDH	

打ち込み

修正

完　成

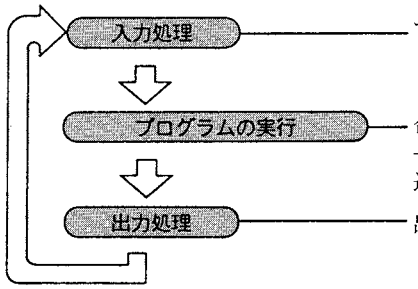
プログラミングコンソールを用い、コーディングリストに従って打ち込みます。
KZ 本体とプログラミングコンソールの接続については P2－13 を、プログラミングコンソールの操作方法・入力方法の詳細については P2－14 をご覧ください。

※0002 は外部から NC 接点を入力するため、プログラム上は
NO (AND 0002) で書きます。

入出力処理と応答遅れ

KZの入出力処理

コーディングリストにしたがってＫＺに打込まれたプログラムを「ユーザープログラム」と呼びます。ＫＺを運転すると、このユーザープログラムが繰り返し実行されます。
実際には下図の様に、「入力処理」→「プログラムの実行」→「出力処理」の三段階が繰り返し実行されています。



プログラムの実行前に全ての入力端子のＯＮ－ＯＦＦ状態を入力メモリに読込みます。

命令に応じて各要素(内部補助リレー、タイマ・カウンタなど)のＯＮ－ＯＦＦ状態を読み出し、演算します。演算結果は出力メモリに書き込まれます。

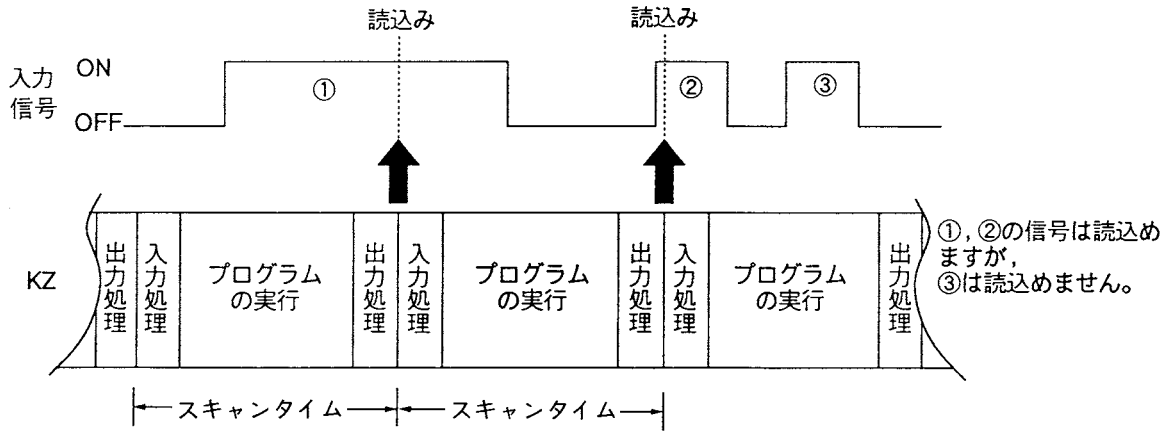
出力メモリの内容を出力端子へ出力します。

一巡するのに必要な時間を**スキャンタイム**(または **サイクルタイム**)といいます。
サイクルタイムは、ユーザープログラムの長さや、プログラム中で使用している命令の種類により変化します。

スキャンタイムと応答遅れ

KZ は、入出力処理時間の他に、スキャンタイムによる入力応答遅れが存在します。これは、入力処理時間以外に、入力状態を読込むことができない為です。入力処理後に入力状態が変化した場合、その内容を読込めるのは、次のスキャンになります。

入出力応答時間…**最短の場合**
入力応答遅れ時間＋スキャンタイム＋出力応答遅れ時間
最長の場合
入力応答遅れ時間＋スキャンタイム×2＋出力応答遅れ時間



※最小スキャンタイム（入出力処理時間）
KZ10, 16, 24, 40　300μs
KZ-80　800μs

プログラムの領域

KZ-10、KZ-16にはニモニック言語で約500行分(行番号00000～00499)、KZ-24、KZ-40、KZ-80には約3000行分(行番号00000～02999)のプログラムを書き込むことができます。しかし使用する命令によっては、これ以上書き込むことができたり、これ以内の行数でも運転時に「OBJ サイズオーバー」となり実行できないことがあります。以下の条件を越えた場合、プログラムを書き込めなかったり、実行できなくなりますのでご注意下さい。

KZ-10、KZ-16の場合

プログラム中で使用している
　命令の使用メモリの合計(バイト)≦1500バイト
プログラム中で使用している
　命令のオブジェクト使用メモリの合計(バイト)≦4000バイト

KZ-24、KZ-40、KZ-80の場合

プログラム中で使用している
　命令の使用メモリの合計(バイト)≦9000バイト
プログラム中で使用している
　命令のオブジェクト使用メモリの合計(バイト)≦21000バイト

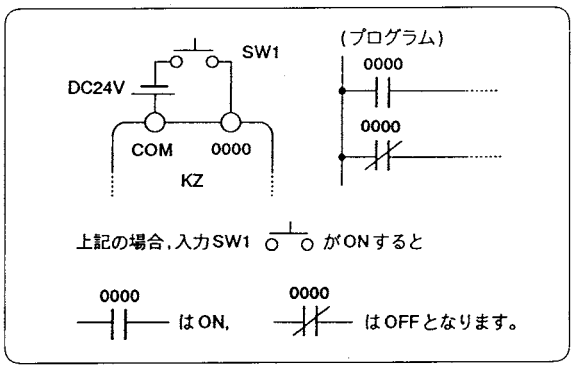
- オブジェクト使用メモリとは、運転開始時に各命令を格納しておく場所の容量(オブジェクト容量)を使用する量のことです。これは命令によって異なっています。
- 各命令の使用メモリ、オブジェクト使用メモリは命令語一覧をご覧下さい。

入力リレー

外部入力機器からのON-OFF信号を取り込みます。

- ●プログラム上、接点として機能します。リレーコイルとしての使用はできません。
- ● a 接点・b 接点の形態や、リレー番号の使用順序、使用個数に制限はありません。

- ●入力リレーの時定数は通常8～12msです。HSP 命令(FUN18)で指定すると時定数は25μs (OFF→ON)、75μs(ON→OFF)となります。(ただし拡張ユニットの入力リレーにHSP命令は使用できません。)
- ●高速カウンタ「CTH0、CTH1」のクロックソース0004、0005の入力応答は自動的に10kHzになります。
- ●INT命令(FUN20)のオペランドに指定した0000～0003の入力応答時間は、25μs(OFF→ON)、35μs(ON→OFF)になります。
  但し基本ユニットのみです。



HSP 命令 → P3－29
高速カウンタ → P5－17
INT 命令 → P5－50
をご覧下さい。

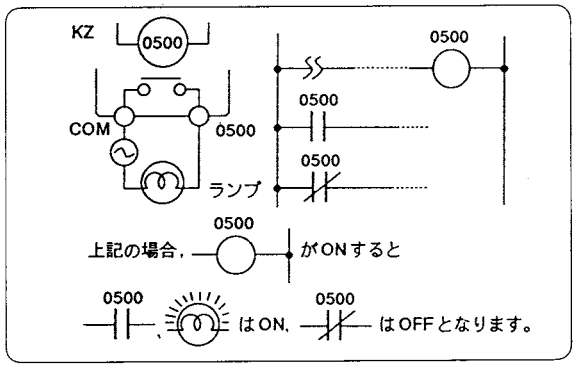
出力リレー

プログラムの実行結果を外部に出力します。出力形態にはリレーとトランジスタとMOS-FETがあります。

- ●プログラム上、接点およびリレーコイルとして機能します。
- ● a 接点・b 接点の形態や、リレー番号の順序、使用個数に制限はありません。

●出力リレーの動作時間は

リレー	OFF → ON 10ms ON → OFF 10ms	以下です
トランジスタ	OFF → ON 50μs (KZ-80T の 500 は 3μs 以下) ON → OFF 450μs (KZ-80T の 500 は 20μs 以下)	
MOS-FET	OFF → ON 350μs (500、501) 2ms (その他) ON → OFF 150μs	



●出力形態

KZ-10R	リレー
KZ-16R	リレー
KZ-24R	リレー
KZ-40R	リレー
KZ-80R	リレー

KZ-10T	トランジスタ
KZ-16T	MOS-FET
KZ-24T	MOS-FET
KZ-40T	MOS-FET
KZ-80T	トランジスタ

ダイレクトクロックパルス → 6章 高速カウンタの応用をご覧ください。

●ダイレクトクロックパルスをプログラムすると、0500、0501 がパルス出力となります。

内部補助リレー

リレー回路の場合、同一のリレー接点を回路内部で複数回使用するとき、場合によっては使用回数分の複数極リレーを用いなければならないことがあります。内部補助リレーはプログラム上でのみ機能するリレーですが、リレー回路のこういったわずらわしさが一切無くなり、回路設計が容易になります。

- プログラム上、接点およびリレーコイルとして機能します。
- a接点・b接点の形態や、リレー番号の順序、使用個数に制限はありません。

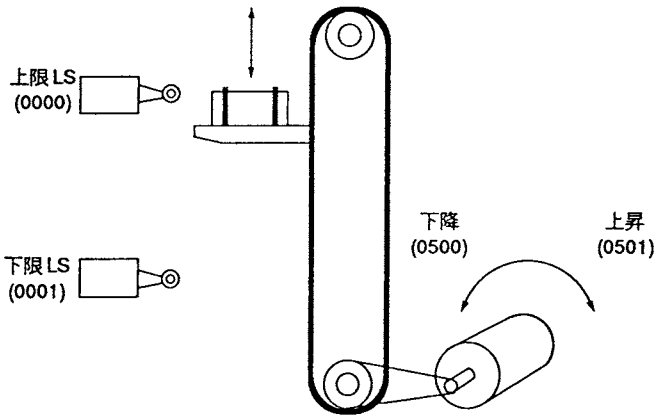
内部補助リレーの保持機能

全てのリレーは、運転モードの切り替えや電源断により運転停止状態になるとOFFし、再運転開始時に入力条件がONのものを除けばOFFのままです。しかし、内部補助リレーは、メモリスイッチにより、保持/非保持を設定することができます。
保持に設定しますと、電源OFF後も直前のON-OFF状態を記憶しています。保持リレーオールクリア(FUN 65)で、保持に設定したリレーを全てOFFにできます。
※メモリスイッチの設定については、P 3−31 をご覧下さい。

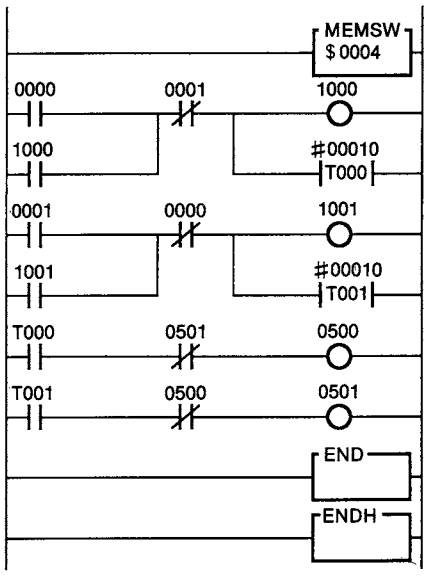
保持機能の応用例：リフトの昇降制御

※1000、1001のリレーを保持に設定します。

上下限のＬＳをたたくことで、リフトの動作方向が反転します。



(プログラム)



上昇途中で電源が OFF しても、再投入時、再びリフトは上昇 (下降) し始めます。

特殊補助リレー

特殊補助リレーは、それぞれが独自の機能をもっています。これらを効果的に活用することで、プログラムを簡略化したり、制御能力を高めることができます。

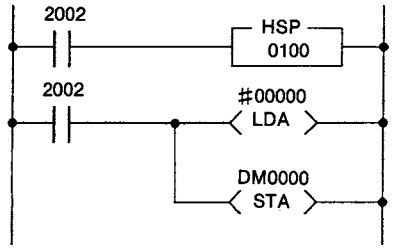
- ひとつのプログラム中で、同じ番号の特殊補助リレーは何度でも使用することができます。
- 読み出し専用の特殊補助リレーは、接点として使用することはできますが、リレーコイルとしての使用はできません。(読み出し専用のリレーは、特殊補助リレー一覧をご参照下さい。)

■解説

2002 常時 ON

2002で出力リレーをONさせると「運転中出力」として使用することができます。
他に、HSP命令の入力や、演算命令の起動用など、OFFさせる必要がない場合によく使用します。

(例)



(0100の入力時定数を25μsにします。
0000chのON/OFFデータをDM000に転送します。)

2003 常時 OFF

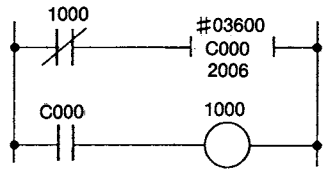
2002とは逆にＯＮさせる必要がない場所に使用します。
アップダウンカウンタを「アップ」のみで使うとき、ＤＷ(ダウン)入力に入れておきます。

2004、2005、2006 クロックパルス

それぞれ、ON：OFF＝1：1の時間比です。(従って2005のON時間は0.05秒/パルスです。)

- 出力リレーと組合せると「断続出力」が作れます。
- 2006をカウンタのカウント入力として使用すると、長時間タイマとして使用できます。

(例)

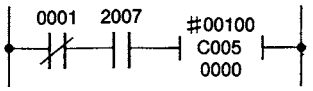


(1000は1時間ごとにONします)

2007 運転開始時１スキャンのみ OFF

運転開始後一定時間(１スキャンタイム)経過した後の信号が欲しいときに使用します。また、運転開始時のイニシャルリセット処理用として使用します。
１スキャン後はONし続けます。

(例1)

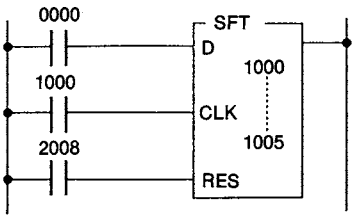


(運転開始時にＣ005の現在値を0に戻します。)

2008 運転開始時１スキャンのみ ON

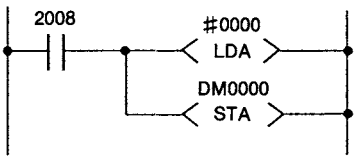
運転開始時のイニシャルリセット処理用として使用します。
１スキャン後はOFFし続けます。

(例1)



(運転開始時にシフト命令で使用している内部補助リレーをすべてOFFにします。)

(例2)



(運転開始時にDM0000の値をクリア(ゼロ)に戻します。)

その他の特殊補助リレーの使用方法については、5章応用命令、演算命令、6章高速カウンタの応用をご覧下さい。

リレー番号一覧

■ リレー番号一覧

型　式	KZ-10	KZ-16	KZ-24	KZ-40	KZ-80
基本入力リレー	0000～0005 (0ch) 6 点	0000～0009 (0ch) 10 点	0000～0015 (0ch) 16 点	0000～0015 (0ch) 0100～0107 (1ch) 合計 24 点	0000～0015 (0ch) 0100～0115 (1ch) 0200～0215 (2ch) 合計 48 点
※ 拡張入力リレー (拡張ユニット使用時)	0100～0115 (1ch) 0200～0215 (2ch) 0300～0315 (3ch) 0400～0415 (4ch) 最大拡張時合計 64 点			0200～0215 (2ch) 0300～0315 (3ch) 0400～0415 (4ch) 最大拡張時合計 48 点	0300～0315 (3ch) 0400～0415 (4ch) 最大拡張時合計 32 点
基本出力リレー	0500～0503 (5ch) 4 点	0500～0505 (5ch) 6 点	0500～0507 (5ch) 8 点	0500～0507 (5ch) 0600～0607 (6ch) 合計 16 点	0500～0515 (5ch) 0600～0615 (6ch) 合計 32 点
※ 拡張出力リレー (拡張ユニット使用時)	0600～0615 (6ch) 0700～0715 (7ch) 0800～0815 (8ch) 0900～0915 (9ch) 最大拡張時合計 64 点			0700～0715 (7ch) 0800～0815 (8ch) 0900～0915 (9ch) 最大拡張時合計 48 点	
内部リレー　内部補助リレー	1000～1015 (10ch～19ch) 1900～1915 160 点			1000～1015 (10ch～19ch) 1900～1915 3000～3015 (30ch～39ch) 3900～3915 4000～4015 (40ch～49ch) 4900～4915 5000～5015 (50ch～59ch) 5900～5915 6000～6015 (60ch～69ch) 6900～6915 800 点	
内部リレー　特殊補助リレー	2000～2015 (20ch～29ch) 2900～2915				

※使用する拡張ユニットによって使用できるリレー番号は違います。
詳しくは拡張ユニットリレー番号をご覧ください。(P1－8)

特殊補助リレー一覧

■特殊補助リレー一覧

― 特殊リレーと演算フラグ ―

リレー番号	機能
※ 2002	常時 ON
※ 2003	常時 OFF
※ 2004	0.01sec クロックパルス (duty 50%)
※ 2005	0.1sec クロックパルス (duty 50%)
※ 2006	1.0sec クロックパルス (duty 50%)
※ 2007	運転開始時 1 スキャンのみ OFF
※ 2008	運転開始時 1 スキャンのみ ON
2009	演算結果が負またはオーバーフローのとき ON
※ 2010	演算結果がゼロのとき ON
※ 2011	演算結果が正のとき ON
※ 2012	演算実行エラーのとき ON

― 8 bit 高速カウンタ用補助リレー ―

2100	CTH0 専用内部クロック (0.8μsec)
2101	CTH0 専用内部クロック (6.4μsec)
2102	CTH0 専用内部クロック (102.4μsec)
2103	コンパレータ CTC0 の ON 時、CTH0 の自動クリア　ON：する　OFF：しない
2104	コンパレータ CTC0 の ON 時 0500 へのダイレクト出力を　ON：禁止する　OFF：許可する
2105	コンパレータ CTC0 の ON 時 0500 へのダイレクト出力を OFF に　ON：する　OFF：しない
2106	コンパレータ CTC0 の ON 時 0500 へのダイレクト出力を ON に　ON：する　OFF：しない
2107	コンパレータ CTC0 が ON するごとに 0500 への出力の ON/OFF を反転 ON：する　OFF：しない
2108	コンパレータ CTC1 の ON 時 0500 へのダイレクト出力を　ON：禁止する　OFF：許可する
2109	コンパレータ CTC1 の ON 時 0500 へのダイレクト出力を OFF に　ON：する　OFF：しない
2110	コンパレータ CTC1 の ON 時 0500 へのダイレクト出力を ON に　ON：する　OFF：しない
2111	コンパレータ CTC1 が ON するごとに 0500 への出力の ON/OFF を反転 ON：する　OFF：しない

― 16 bit 高速カウンタ用補助リレー ―

2200	CTH1 専用内部クロック (0.4μsec)
2201	CTH1 専用内部クロック (0.8μsec)
2202	CTH1 専用内部クロック (3.2μsec)
2203	コンパレータ CTC2 の ON 時、CTH1 の自動クリア　ON：する　OFF：しない
2204	コンパレータ CTC2 の ON 時 0501 へのダイレクト出力を　ON：許可する　OFF：禁止する
2205	コンパレータ CTC2 の ON 時 0501 へのダイレクト出力を　ON：ON にする　OFF：OFFにする
2206	INT3 のエッジ選択　ON：ダウンエッジ　OFF：アップエッジ

― その他の特殊補助リレー ―

2804	テキストデータ受信リレー　テキストデータ受信時 1 スキャンのみ ON
2805	テキストデータ取り込みエラー 2804　ON 中にテキストデータを受信した時 1 スキャンのみ ON 2806　ON 時 1 スキャンのみ ON
2806	テキストデータ受信エラー テキストデータ受信時通信エラーが発生した時 1 スキャンのみ ON
2814	HKEY　命令　複数キー禁止
2815	HKEY　命令　スキャン終了
2900~2915	HKEY　命令のキー情報が格納される

(注意) 上記以外の特殊補助リレーは使用しないでください。
※印 読みだし専用リレー

タイマ・カウンタ

通常のタイマ・カウンタと同様に、ＫＺもプログラム上でタイマ・カウンタを設定し、その出力をプログラム上で接点として使用することができます。(a・b接点、使用順序に制限はありません。)
タイマ・カウンタは同じ番号を重複して使用することはできません。即ちT005とC005は同一プログラム内に存在できません。

■タイマ・カウンター一覧

型式		KZ-10、KZ-16			KZ-24、KZ-40、KZ-80		
	名称	接点	コイル(命令)	点数	接点	コイル(命令)	点数
タイマ	タイマ(0.1秒タイマ)	T000～T063	TMR 000～063	合計64	T000～T119	TMR 000～119	合計120
	高速タイマ(0.01秒タイマ)		TMH 000～063			TMH 000～119	
	高速1msタイマ		TMS 000～063			TMS 000～119	
カウンタ	カウンタ	C000～C063	C 000～063		C000～C119	C 000～119	
	アップダウンカウンタ		UDC 000～063			UDC 000～119	

■解説

タイマ

- 0.1秒タイマも0.01秒タイマも1msタイマも共に接点として用いるときの番号はTｎｎｎとなります。ただし、設定する際の命令は、0.1秒タイマがTMRｎｎｎ＃ｄｄｄｄｄ、0.01秒タイマがTMHｎｎｎ＃ｄｄｄｄｄ(FUN 49)、1msタイマがTMSｎｎｎ＃ｄｄｄｄｄ(FUN 51)となります。
- TMRｎｎｎとTMHｎｎｎとTMSｎｎｎに同じタイマ番号は割り付けられません。
- ＫＺの運転停止時、すべてのタイマはリセットされ現在値＝設定値となります。

TMR → P3－19
TMH → P3－20
TMS → P3－20
をご覧ください。

カウンタ

- カウンタもアップダウンカウンタも共に接点として用いるときの番号はＣｎｎｎとなります。ただし、設定する際の命令はカウンタがＣｎｎｎ＃ｄｄｄｄｄであるのに対し、アップダウンカウンタはUDCｎｎｎ＃ｄｄｄｄｕ(FUN52)となります。
- ＣｎｎｎとUDCｎｎｎに同じカウンタ番号は割付けられません。
- すべてのカウンタは運転停止時も直前の値を記憶しています。

CTR → P3－22
UDC → P3－24
をご覧ください。

高速カウンタ・高速カウンタコンパレータ

KZには入力応答10kHzの高速カウンタが2点(CTH0、CTH1)と高速カウンタコンパレータが4点(CTC0、1、2、3)装備されています。

型式	KZ-10、16、24、40、80	
高速カウンタ	CTH0	CTH1
高速カウンタコンパレータ	CTC0、1	CTC2、3

高速カウンタ、高速カウンタコンパレータについては P5－17、P5－18 をご覧下さい。

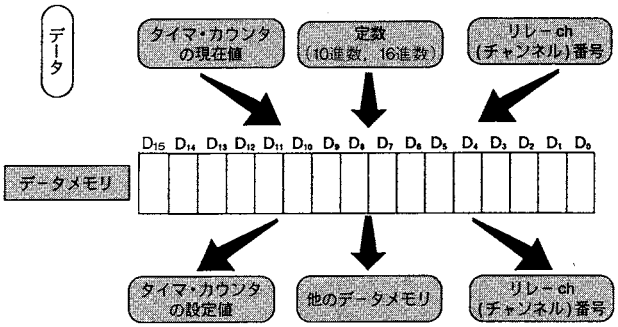
データメモリ

「データメモリ」は文字通り各種データを記憶(メモリ)しておく場所のことです。
接点・コイル・タイマ・カウンタで構成されたプログラム上では特に使用することはありませんが、演算命令を使う場合、演算を行うためのデータや、演算を行った結果のデータを格納するために使用します。

■データメモリー

型　　式	KZ-10、KZ-16		KZ-24、KZ-40、KZ-80	
名　　称	番　　号	個数	番　　号	個数
データメモリ	DM0000～DM0999	1000	DM0000～DM1999	2000

●データメモリの内部構造は、内部レジスタと同様、2 進 16 桁 (16 ビット) になっており、すべての値は 2 進数で格納されます。
●データメモリ内の記憶は、電源 OFF 時、オールクリア (FUN60) 時にも 2 ヵ月以上保持されています。
●FUN64 で「データメモリオールクリア」、FUN72 で「データメモリの書込み」が可能です。



テンポラリデータメモリ

一時記憶用のデータメモリです。演算命令を使う場合、演算を行うためのデータや演算を行った結果のデータを一時格納するために使用します。

■テンポラリデータメモリ

型　　式	KZ-10、16、24、40、80	
名　　称	番　　号	個数
テンポラリデータメモリ	TM00～TM31	32

●TM30、31 は、読み出し専用となっていますので、データを書き込むことはできません。
TM30 については P5－50、P5－51 INT 命令を、TM31 については P2－24 サイクルタイムモニタをご覧ください。
●TM00、TM01 は、MUL、DIV 命令で使用しますので、1 つのブロックで完結するように注意してご使用ください。
※TM31 ← サイクルタイムの値が格納されます。
●#TM＊＊
↓
間接指定 (P5－57)

2-3. プログラミングコンソールの取扱方法

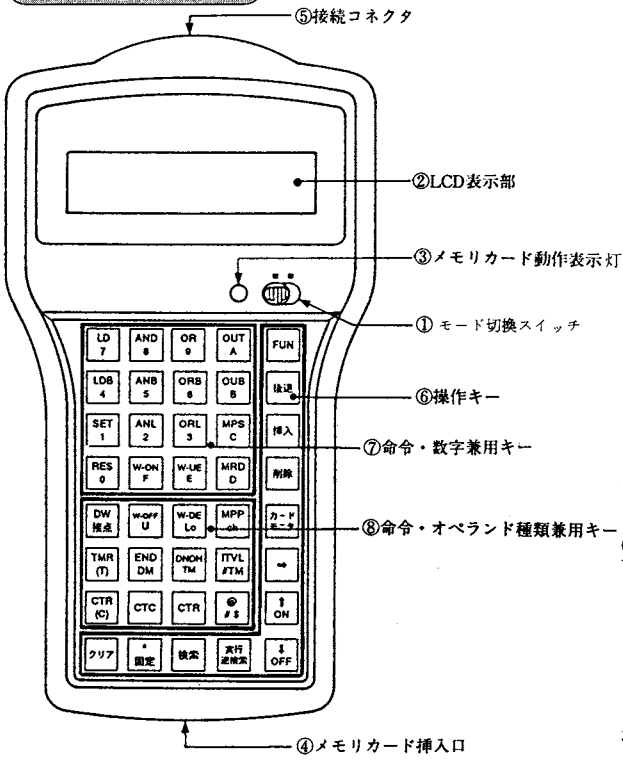
プログラミングコンソールの概要

プログラミングコンソール(以下コンソール)はKZのプログラムを行うだけでなくそのプログラムをKZ本体に「転送」したり、「運転」(RUN)↔「停止」の切換を行うことができます。
また各種モニタやメモリーカードとのアクセスもできます。

プログラミングコンソールの概要

形　式	使用可能命令語	適応機種	メモリカード
KZ-P3	ＫＺで使用可能な基本命令・応用命令・演算命令のすべて	KZ-10～80 KZ-300	M-2、M-3 使用可

各部の名称と機能



①モード切替スイッチ

KZシリーズ、プログラミングコンソールの動作モードを切り換えます。

●RUN　コンソール内のプログラムをKZ本体に転送し、運転状態にします。運転を実行しながらのモニタ、各種データの変更、オフラインエディタが使用できます。

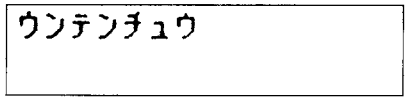
●PROGRAM　ユーザプログラムの作成、変更時に使用します。

②LCD表示部

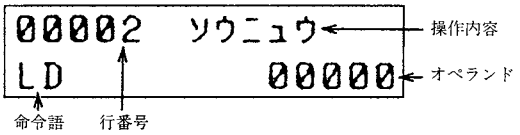
●プログラム内容の表示の他、エラーメッセージなど各種の表示を行います。
●バックライト付です。

〈表示部〉

●運転表示



●プログラミング表示



表示内容の詳細については各操作項目をご覧ください。

③メモリカード動作表示灯

メモリカードとの通信中(「読出」・「書込」・「削除」・「照合」時)に点灯します。
点灯中に、カードを抜くと通信が完全に行われませんので、ご注意ください。
カードを必要とする時、常に点灯します。クリアを押すと元に戻ります。

④メモリカード挿入口(スロット)

メモリカード「M-2またはM-3」をここに挿入します。故障の原因となりますので、異物を入れないようご注意ください。

メモリカードの取り扱い、操作方法については「9章 メモリカード」をご覧下さい。

⑤接続コネクタ

KZ本体とプログラミングコンソールをつなぐケーブルを取り付けます。

コンソールで入力されたプログラムはコンソール内のRAMに書き込まれておりその状態ではプログラムはKZ本体に入力されていません。したがって電源をOFFにするとコンソール内のプログラムは消えます。作成中のプログラムを保存したい場合はメモリカードに書き込むか一旦プログラムを完成させ本体に転送して下さい。(エラー表示がでているときは転送されていません。回路を完成させ再度転送してください。)

プログラミングコンソール各キーの機能

⑥操作キー

キー	名称	基本機能
FUN	ファンクションキー	●FUN番号を呼び出す際に使用します。
後退	後退キー	●数値入力時最後の入力を取り消します。
挿入	挿入キー	●プログラムの挿入モード、上書きモードの切り替えをします。
削除	削除キー	●プログラムの削除モードにします。
カードモニタ	カードモニタキー	●PROGモード時メモリカードへのアクセスをします。 ●RUNモード時サイクルタイムモニタ、導通モニタを行います。
➡	➡キー	●LCD表示部で表示中のカーソルを移動させます。
⬆ ON	⬆オンキー	●表示中のプログラムまたはファンクション番号をひとつ前に戻します。 ●リレーのオン/オフでは、リレーを強制的にオンします。
⬇ OFF	⬇オフキー	●表示中のプログラムまたはファンクション番号をひとつすすめます。 ●リレーのオン/オフでは、リレーを強制的にオフします。
実行 逆検索	実行逆検索キー	●LCD表示部に表示中のプログラムFUN No.他の入力状態を確定し入力実行します。 ●エディタの検索モードのときに逆検索キーとして使用します。
検索	検索キー	●指定した行番号以降の指定した条件を満足する行を検索します。
＊ 固定	＊固定キー	●検索条件を指定する時にワイルドカードとして指定します。 ●状態モニタ時、1接点常時固定表示の設定・解除をします。
クリア	クリアキー	●エラーの解除や入力ミスの取り消に使用します。

●キー操作が正しく行われるとピッと短い音が、誤った操作を行うとピピピッと音がして、確認することができます。

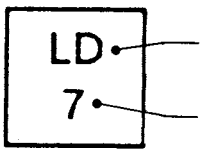
●各操作キーは他のキーとの組合せ操作を行うことで有効に働きます。機能詳細・操作詳細は各項目をご覧ください。

⑦命令・数字兼用キー

命令・数字キーは、命令語入力時には、命令入力キーとして、また数字入力時には数字入力キーとして機能します。

通常、プログラミングの際にはカーソルの位置によって、キーの機能を決定するため、特に意識する必要はありません。

〈例〉



命令：
LD(ロード)命令を入力する際に使用します。

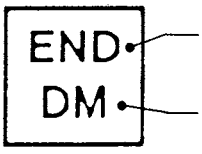
数字：
A～Fのアルファベットは、16進数入力用記号です。

⑧命令・オペランド兼用キー

命令・オペランドキーは、命令語入力時には、命令入力キーとして、またオペランド入力時にはオペランド入力キーとして機能します。

通常、プログラミングの際にはカーソルの位置によって、キーの機能を決定するため、特に意識する必要はありません。

〈例〉

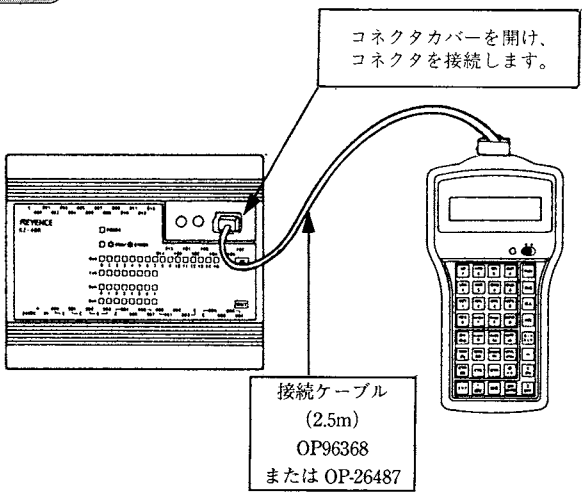


命令：
END(エンド)命令を入力する際に使用します。

オペランド：
#、$は10進数、16進数入力用記号です。

プログラミングコンソール取扱上のご注意

KZ本体への取付方法



- プログラミングコンソールの取付け・取外しは、ＫＺシリーズ本体の電源ON/OFFにかかわらずいつでも可能です。
- 運転開始後、プログラミングコンソールを取外す場合には、「RUN」モードに設定したまま取外してください。運転を継続します。

KZ本体とプログラミングコンソールの関係

状　況	電源投入時の動作モード
プログラミングコンソール未装着	電源投入後 RUN モードになります。
プログラミングコンソール装着	電源投入後 モード切換スイッチで設定されているモードで働きます。
電源投入後にプログラミングコンソールを装着	モード切換スイッチで設定されているモードに変更されます。

プログラミングコンソール接続ケーブル

- 接続ケーブルは、プログラミング時、試運転時、モニタ・調整時など、一時的な使用にとどめ通常の運転など長期の使用はさけてください。
- 接続ケーブルは、強電ラインや出力負荷ラインと束ねたり、その近くを通したりすると故障の原因となりますので、絶対に避けてください。
- 接続ケーブルに多大な力が加わらないようご配慮ください。
- 接続ケーブルは、必ず弊社指定のものをお使い下さい。他のものは誤動作のもとになります。

プログラミングの基本操作

この項ではプログラミングコンソールの基本操作を解説します。
いずれも操作頻度・重要度が高いため、覚えておくと作業が非常にスムーズに行えます。

■プログラミングコンソールの基本操作一覧

内　容		機　能	モード	キー操作	詳細ページ
クリア		入力の取り消し、エラーの解除を行ないます。	PER	[クリア]	2-16
カーソルの移動		カーソルを命令語、オペランド行番号の位置に移動します。	PE	命令 → オペランド → 行番号 ↑ → ↑	2-16
行の移動（スクロール）		プログラミングコンソールの表示を1行前、1行後または、任意の行に移動します。	PE	●1行前の行を表示……[↑ ON] ●1行後の行を表示……[↓ OFF] ●任意の行へ移動 [数字キー]（行番号の入力）→[実行 逆検索]	2-17
プログラムの作成	①オールクリア	プログラムを全て消去します。（後にはNOPが入ります。）	P	[FUN] → [ORB 6] → [RES 0] → [実行 逆検索] → [実行 逆検索] → [クリア]	2-17
	②命令語の入力	命令語を入力します。（カーソルが命令語の入力位置にない場合はカーソルを移動して下さい。）	PE	●命令キーがある場合 [命令キー] → [数字キー] （タイマ、カウンタ番号） ●命令キーがない場合 [FUN] → [数字キー] / [↑ ON] / [↓ OFF] → [実行 逆検索]	2-18
	③オペランドの入力	オペランドを入力します。（カーソルがオペランドの入力位置にない場合は、カーソルを移動して下さい。）	PE	[オペランドキー] → [数字キー] → [実行 逆検索] （オペランドの種類の入力） ↑ （オペランドが複数ある場合）	2-18
	④END ENDH	プログラムの終わりに必ず書き込みます。	PE	●END……[END DM] → [実行 逆検索] ●ENDH……[ENDH TM] → [実行 逆検索]	3-33

P：PROGRAMモード
R：RUNモード（オフラインエディタ未使用の場合）
E：オフラインエディタ起動中

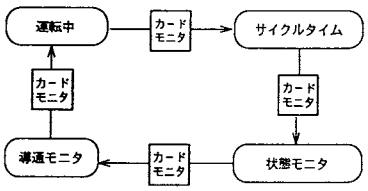
ワンポイント

ファンクションナンバーが分からない場合

[FUN]キーを押して、FUNナンバーにカーソルがある時、[↑ ON] [↓ OFF]キーで検索することができます。

プログラミングの基本操作

■プログラムコンソールの基本操作一覧

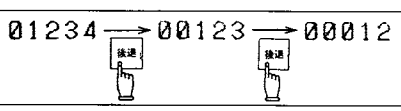
内容		機能	モード	キー操作	詳細ページ
プログラムの修正	①修正行の表示	修正したい行を表示します。	PE	前頁の「行の移動」を参照して下さい。	2-17
	②命令語の修正	命令語を修正します。（カーソルを命令語の位置に移動して下さい。） ※オペランドのみ修正する場合は③にすすんで下さい。	PE	前頁の「命令語の入力」と同様にして下さい。 命令語が上書きされます。	2-18
	③オペランドの修正	オペランドを修正します。（カーソルをオペランドの位置に移動して下さい。）	PE	前頁の「オペランドの入力」と同様にして下さい。 オペランドが上書きされます。	2-18
	行挿入	新たに1行挿入します。	PE	①「行の移動」により、挿入したい位置に行を移動します。（例えば、2行目と3行目の間に1行挿入する場合は、3行目に移動します。） ↓ ②［挿入］を押して挿入モードにします。 （画面に「ソウニュウ」と表示されます。） ↓ ③命令語の入力、オペランドの入力をします。 ※挿入モードを解除するには［クリア］キーを押します。	2-20
	行削除	1行削除します。	PE	①「行の移動」により削除したい行に移動します。 ↓ ②［削除］を押して削除モードにして［実行 逆検索］を押します。（画面に「サクジョ」と表示されます。） ※削除モードを解除するには［クリア］キーを押します。	2-20
モニタ		運転中に、サイクルタイムモニタ、状態モニタ、導通モニタを行ないます。	R		2-24

P：PROGRAMモード
R：RUNモード（オフラインエディタ未使用の場合）
E：オフラインエディタ起動中

ワンポイント

後退キー

行番号やリレー、タイマ、カウンタなどの番号を数字キーで入力する場合、後退キーを押すと最後に入力した数字から順番に取り消すことができます。



クリア

クリアキーを押すことにより、入力の取り消しやエラーの解除をすることができます。

●プログラミング中、間違った命令語やオペランドを打ち込み、入力せずに前の画面に戻したい場合。

あくまで表示画面を消去するだけですので、既に書かれているプログラムは消えません。

●発生した各種エラーメッセージを消去する場合。

エラー発生の原因が解除されていれば、再運転が可能です。詳しくは「10 章 異常とその処置」をご覧ください。

●ファンクションキー(FUN)、挿入キー(挿入)、削除キー(削除)、検索キー(検索)、カード・モニタキー(カード/モニタ)などの入力を取り消したい場合。

《操作》

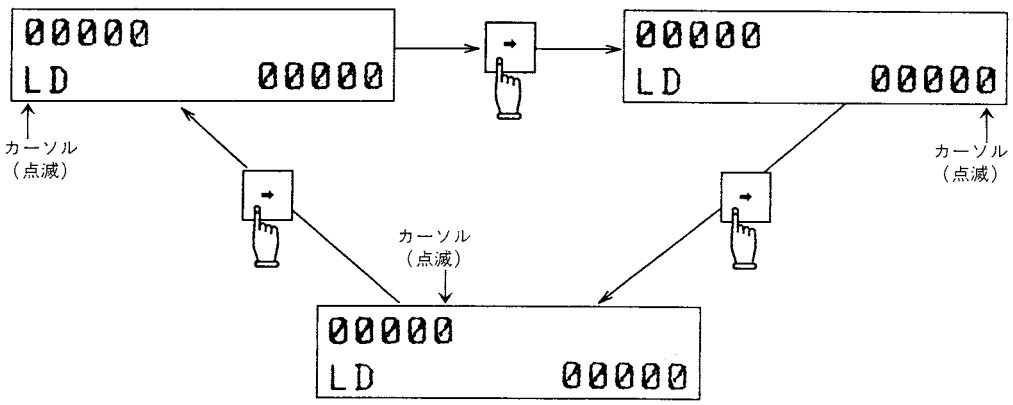


カーソルの移動

→ キーを押すことで、点滅しているカーソルを入力したい項目(命令語、オペランド、行番号)に移動することができます。

●「PROGRAM」モード、導通モニタ、オフラインエディタで機能します。

《操作》



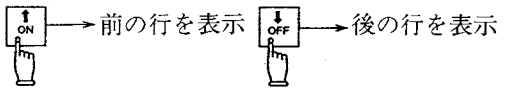
行の移動

プログラミングコンソールの表示を1行前・1行後(スクロール)、または任意の行に移動します。

1行前・後に移動

●「プログラムモード」、状態モニタ、導通モニタ、オフラインエディタで機能します。
●キーを押し続けると連続して移動します。

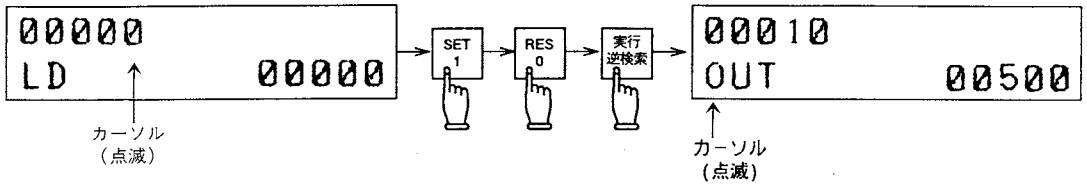
《操作》



任意の行に移動

●「プログラムモード」、導通モニタ、オフラインエディタで機能します。
●行番号に99999を設定すると最終行が表示されます。
●あらかじめ行番号にカーソルを移動して下さい。

《操作》
(例) 0行目の表示から10行目の表示に移動します。



●数字キーは命令キーも兼ねていますが、行番号設定時には自動的に数字キーとして機能します。
●上位桁の0を入力する必要はありません。

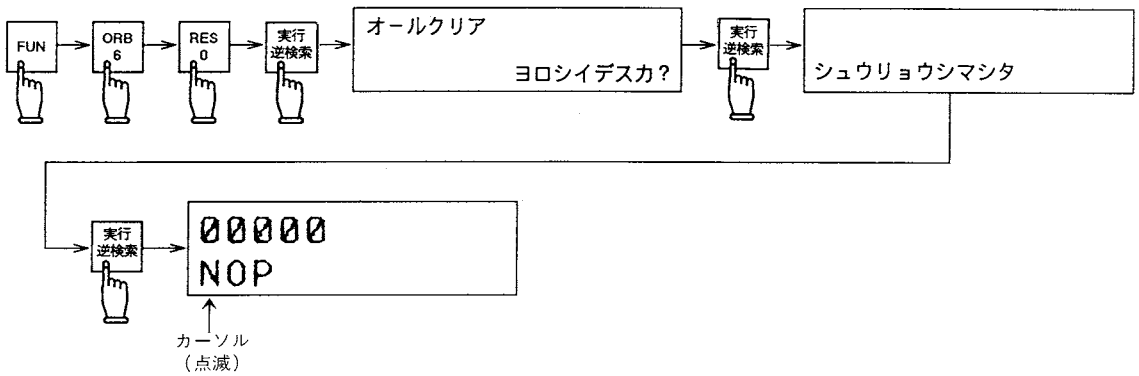
オールクリア(FUN 60)

現在KZ内に書かれているプログラムを全て消去します。(後にはNOP命令が入ります。)

新たにプログラミングをする前には、必ずこの操作を行ってください。

●「PROGRAM」モードでのみ機能します。
●「オールクリア」を行っても、「データメモリ」のデータは消去されずに保存されます。

《操作》



プログラムの「書込」

表示されている行に新しいプログラムを上書きします。
「PROGRAM」モード、オフラインエディタで機能します。

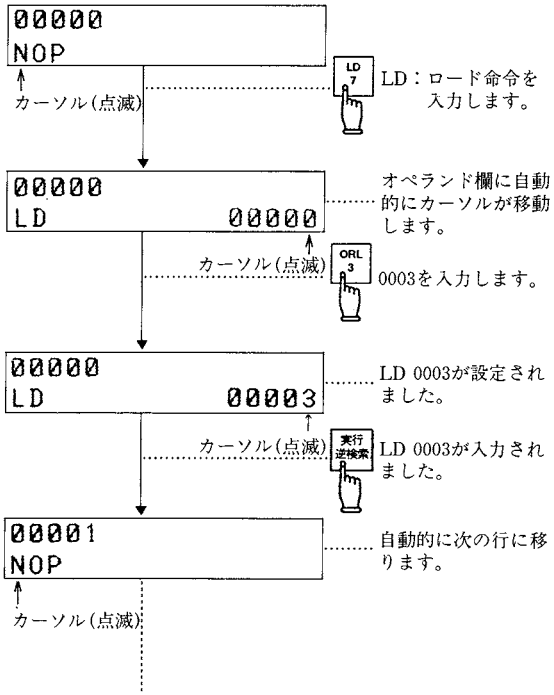
■解説

- [↑ON]・[↓OFF]キーで前後のプログラムを連続して表示できます。
- 命令・オペランドなどを入力後、[実行 逆検索]キーを押すと、表示は自動的に次の行に移ります。
- [→]キーで命令・オペランド・行番号間の「カーソル」の移動を行います。
「カーソル」のある部分が上書き・訂正可能となります。
- [削除]・[挿入]キーを押すとそれぞれのモードに移ることができます。

■操作例

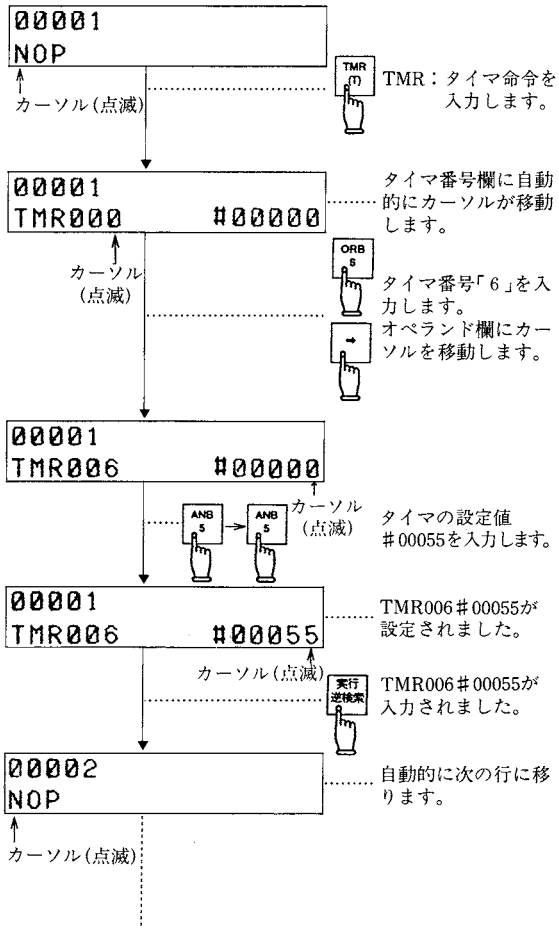
①基本接点命令の「書込」の場合

(例) LD 0003の入力



②タイマ・カウンタ命令の「書込」の場合

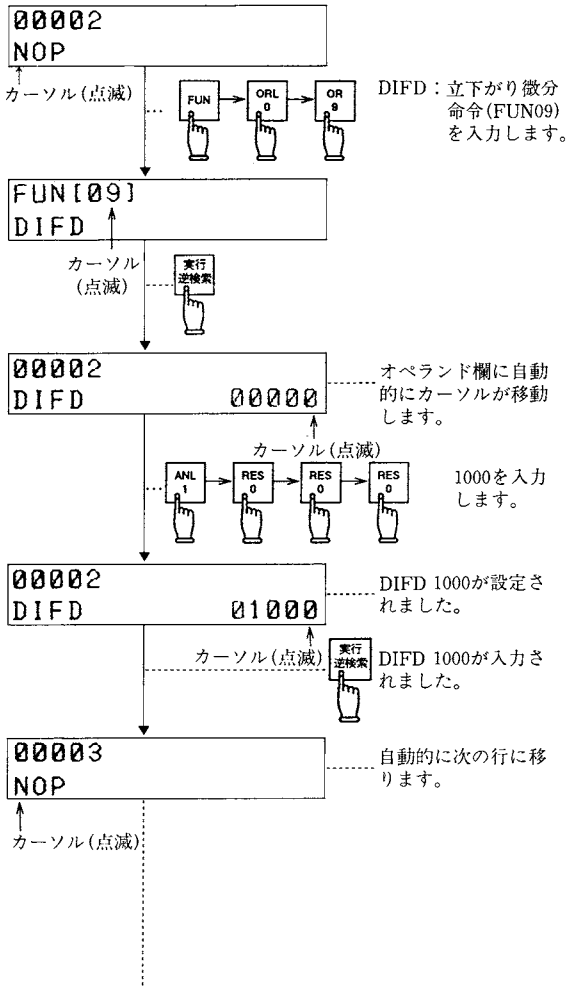
(例) TMR 006＃00055の入力



プログラムの「書込」

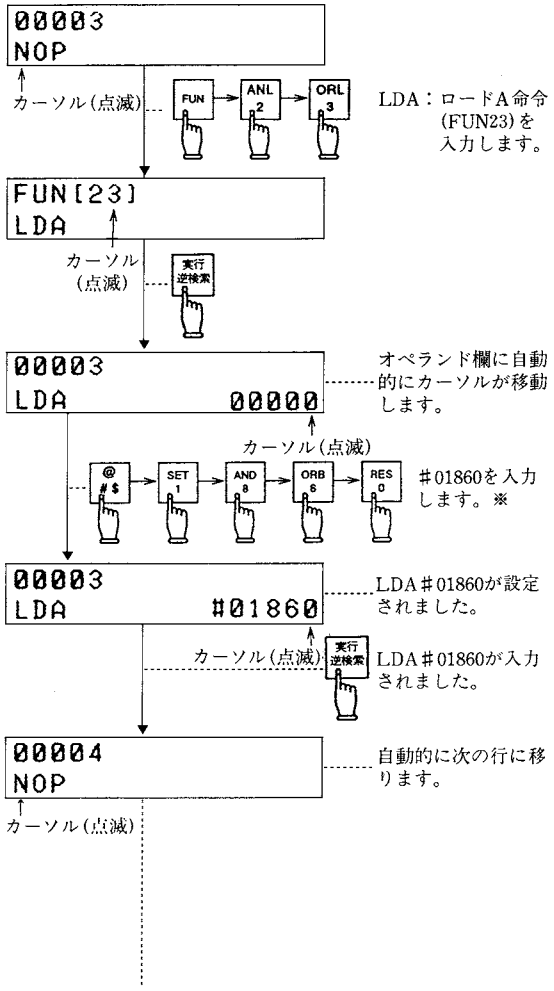
③応用命令の「書込」の場合

（例）DIFD 1000の入力



④演算命令の「書込」の場合

（例）LDA＃01860の入力



ワンポイント

接点の種類を変更する場合

タイマやカウンタの接点から、入力リレー、出力リレー、内部補助リレーなどに変更するときは、[➡] キーでカーソルを接点の所まで移動し、[接点] を押してください。

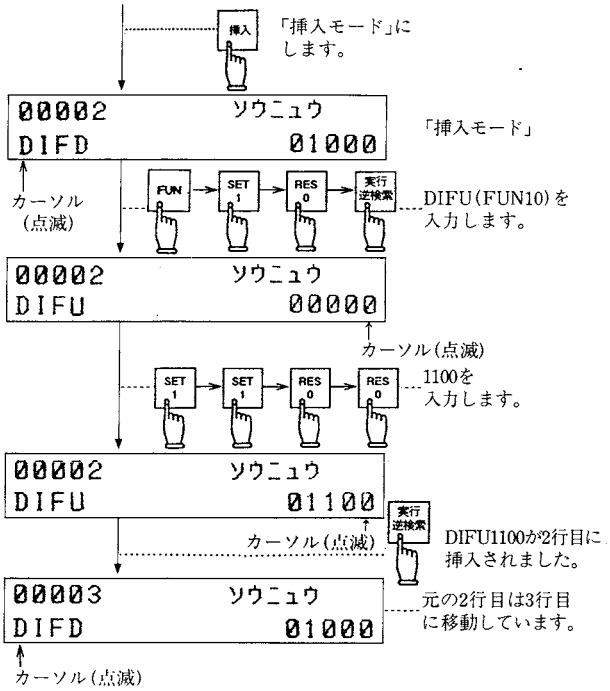
※16進数($××××)を入力するときは、[@ #$] キーをもう一度押すと「$」が表示されます。

プログラムの「挿入」

表示中の行に新しいプログラムを挿入し、その行に元あった命令も含めて、それ以降のプログラムを一行ずつ後に送ります。

《操　作》：2行目にDIFU1100を「挿入」します。(1行目と2行目の間に挿入されます。)

「行の移動」で2行目を表示します。



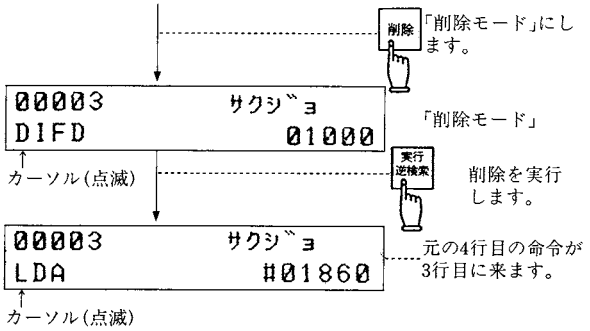
- ↑ON・↓OFFキーで前後のプログラムを続けて読出すことができます。
- 挿入が終了したらクリアキーを押して挿入モードから抜けて下さい。
- 「PROGRAM」モード、オフラインエディタで機能します。

プログラムの「削除」

表示中の行のプログラムを削除します。
削除箇所以降の行番号は、削除した行の数だけ前に送ります。

《操　作》：3行目を削除します。

「行の移動」で3行目を表示します。



- ↑ON・↓OFF キーで前後のプログラムを続けて表示することができます。
- 「削除」された行数分、最終行から「NOP」が入ります。
- 削除が終了したらクリアキーを挿して削除モードから抜けて下さい。
- 「PROGRAM」モード、オフラインエディタで機能します。

ご注意

「削除」した行のプログラムは消去されますのでご注意ください。

プログラムの「検索」

指定した行番号以降の特定のソースプログラムを検索します。

- 検索には次の3通りがあります。
  ①命令による検索
  ②命令＋リレー番号(オペランド)による検索
  ③リレー番号(オペランド)による検索
- いずれも指定した行番号以降にプログラムされているもののみ、順に検索することができます。
  (逆方向の検索も可能です。)
- 「PROGRAM」モード、オフラインエディタで機能します。

《手　順》

1 検索の開始

- 検索条件を、通常プログラムを書き込むのと同様に入力します。
- 検索キーを押すと、その時表示されている行を検索条件として検索を開始します。

プログラムを書き込む場合：[命令語]＋[オペランド]＋[実行]キー
プログラムを検索する場合：[命令語]＋[オペランド]＋[検索]キー

2 連続検索と逆検索

- 検索モードで検索キーを押すと、前回と同じ条件で検索をくり返します。
- 検索条件に一致する行がない場合(検索が終了した場合)は、「シュウリョウ」と表示します。
- 検索モードで逆検索キー([実行 逆検索])を押すと、前回と同じ条件で逆方向に検索できます。

検索モード

検索条件を入力して検索キーを押すとプログラミングコンソールに「ケンサク」又は「シュウリョウ」と表示されます。この状態を検索モードと呼びます。

3 検索モードの終了

- クリアキーを押すと、検索モードを終了できます。
- 検索モードでは、命令語、オペランド、行番号の入力はできません。
  これらの入力があると、自動的に検索モードが解除されます。

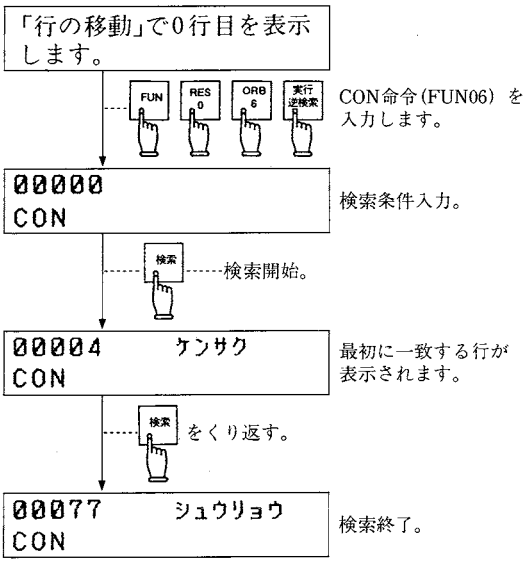
詳しくは次ページの操作例を参照して下さい。

プログラムの「検索」

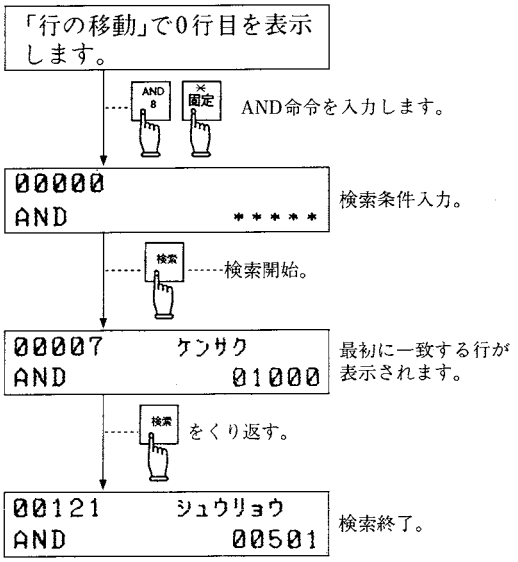
《操作例》

①命令による検索を行なう場合

(例1) すべてのCON命令を検索する

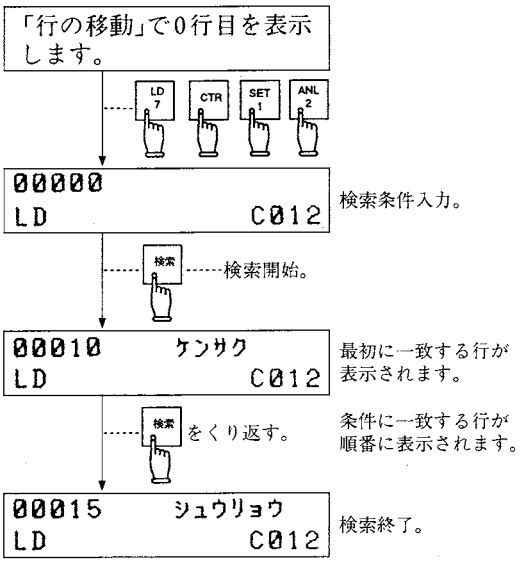


(例2) すべてのAND命令を検索する

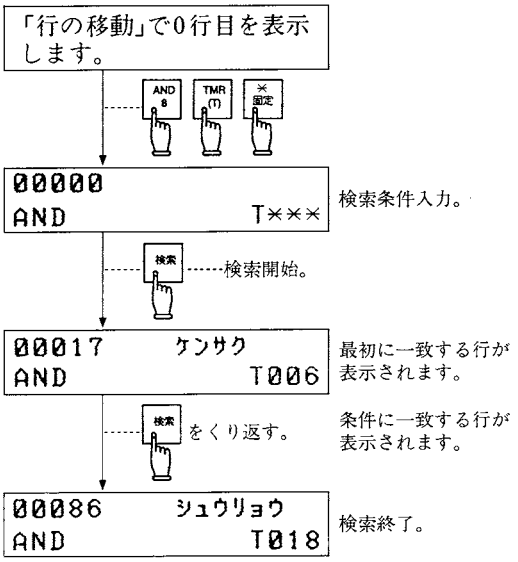


②命令＋オペランドによる検索を行なう場合

(例3)「LD C012」という命令をすべて検索する。

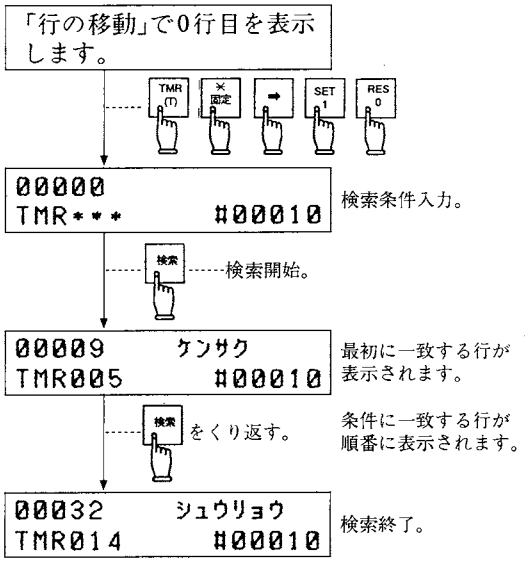


(例4) タイマをオペランドとするAND命令をすべて検索する。

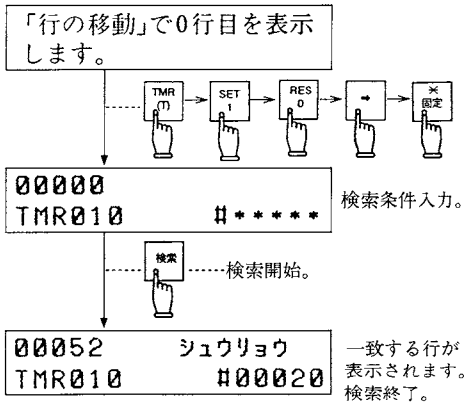


プログラムの「検索」

(例5) 設定値を10とするすべてのTMR命令を検索する。

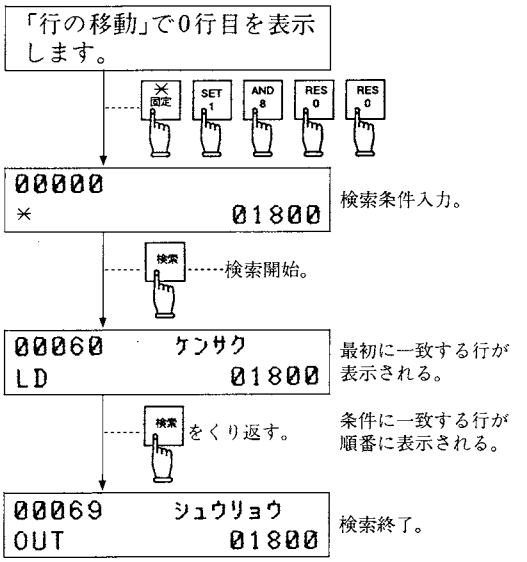


(例6) タイマ10を定義している行を検索する。

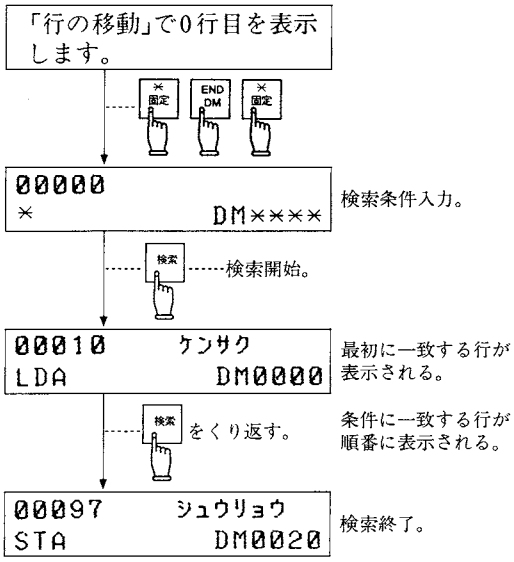


③オペランドによる検索を行なう場合

(例7) 1800番のリレーをオペランドに持つ行を検索する。

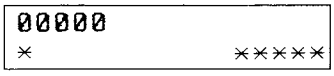


(例8) データメモリを使用している行をすべて検索する。



● 命令語、オペランドともに次のような検索条件は受け付けられません。

(例)



モニタ

運転中にKZ本体の動作状態をモニタすることができます。

- 「RUN」モードのみ機能します。(オフラインエディタ時はモニタできません。)
- サイクルタイムモニタ、状態モニタ、導通モニタが可能です。

1. サイクルタイムモニタ

現在のサイクルタイムを表示します。

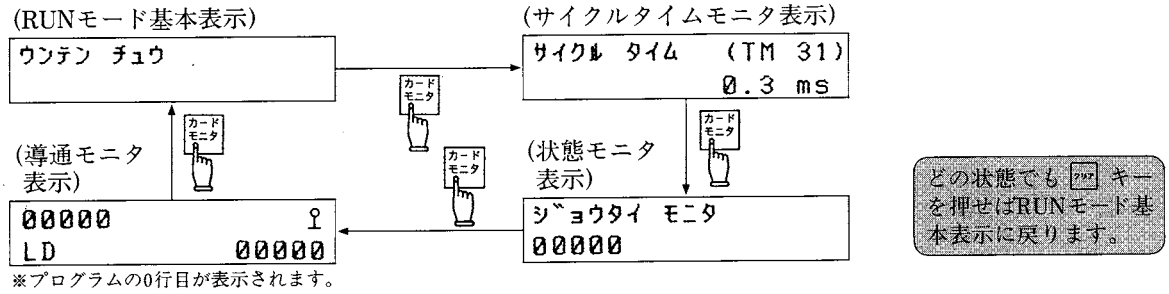
2. 状態モニタ

リレー接点のON-OFF情報、タイマ・カウンタ・データメモリの現在値や、ON-OFF接点のグループなどを16点登録することができ、そのうち2点まで同時にモニタ表示します。

3. 導通モニタ

プログラムを1行ごとに、オペランドに対応するリレー接点のON-OFF情報、タイマ・カウンタのON-OFF情報をモニタ表示します。

《基本操作》



※プログラムの0行目が表示されます。

サイクルタイムモニタ

現在のサイクルタイムを表示します。

〔操 作 例〕

キー操作	表示	説明
	ウンテン　チュウ	基本表示
カードモニタ	サイクル　タイム　(TM31) 0.3ms	サイクルタイム表示機能を呼び出します。 サイクルタイムが表示されます。
クリア	ウンテン　チュウ	基本表示にもどります。

サイクルタイムの値は、テンポラリデータメモリの31番(TM31)に格納されますのでLDA命令で読み出すことができます。

●テンポラリデータメモリ　→　P2−10
●LDA 命令　→　P5−29
} をご覧ください。

状態モニタ

リレー接点のON-OFF情報、タイマ・カウンタ・データメモリの現在値やチャンネルを16点(同時に2点まで)モニタ表示します。(チャンネル：任意の接点から、それ以降の16接点を1つにまとめたもの)

■解説

- モニタは横1行で1点（リレー接点、タイマ、カウンタ、データメモリ、チャンネル）のON-OFF又は現在値を表示します。
- 最大16点登録することができ [↑ON] [↓OFF] キーでスクロールしながらモニタできます。
- 表示画面の上下2行を使って同時に2点のモニタが可能です。
- 入力は常に下の行に対して行なわれます。
- 上の行は、常時固定表示することができます。その場合、下の行だけスクロールします。
- 一度登録されるとモニタを終了しても記憶しています。ただし、電源OFF、オールクリア、プロコンクリア、プログラムの送受信が行われると登録はクリアされます。
- 登録内容は、上書き、挿入、削除で変更できます。

〔操 作 例〕

キー操作	表示	説明
	ウンテン　チュウ	基本表示。
[カードモニタ] [カードモニタ]	ジョウタイ　モニタ 0 0 0 0 0	状態モニタ表示にします。 下の行が入力可能です。
[ANB 5] [RES 0] [RES 0] [実行 逆検索]	ジョウタイモニタ 0 0 5 0 0　♀	出力リレー500番のON-OFF状態をモニタします。(OFF)
[↓OFF]	0 0 5 0 0　♀ 0 0 0 0 0	1行下にスクロールして次の入力行を表示します。
[SET 1] [SET 1] [実行 逆検索]	0 0 5 0 0　♀ 0 0 0 1 1　●	入力リレー0011番のON-OFF状態をモニタします。(ON)
[↓OFF]	0 0 0 1 1　● 0 0 0 0 0	1行下にスクロールして次の入力行を表示します。
[TMR (T)] [ANB 5] [実行 逆検索]	0 0 0 1 1　● T 0 0 5　0 0 0 1 0　♀	タイマの5番の現在値とON-OFF状態をモニタします。(10、OFF)
[↓OFF]	T 0 0 5　0 0 0 1 0　♀ 0 0 0 0 0	1行下にスクロールして次の入力行を表示します。
[END DM] [SET 1] [実行 逆検索]	T 0 0 5　0 0 0 1 0　♀ DM 0 0 0 1　0 0 1 2 5	データメモリ1番の現在値をモニタします。(125)
[↓OFF]	DM 0 0 0 1　0 0 1 2 5 0 0 0 0 0	1行下にスクロールして次の入力行を表示します。
[ANB 5] [RES 0] [RES 0] [MPP ch] [実行 逆検索]	DM 0 0 0 1　0 0 1 2 5 0 0 5 0 0 CH　▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪▪	出力リレー500番から515番のON-OFF状態をモニタします。 (501、503、504のみON)
[クリア]	ウンテン　チュウ	基本表示に戻ります。

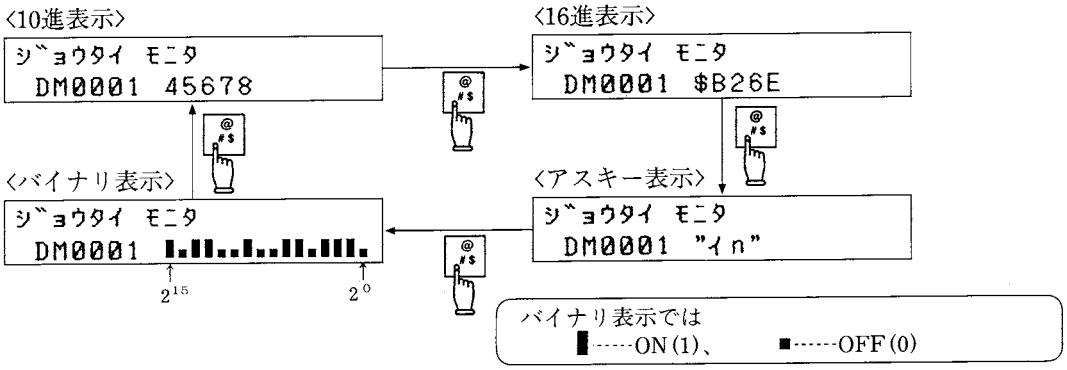
● 一度、入力した内容は自動的に登録されます。(16点まで)

状態モニタ

モニタ結果の表示

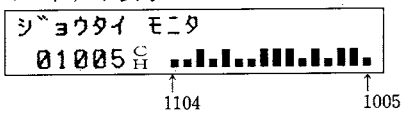
●接点のON-OFFは ⚲/⚲ の記号で表示されます。　⚲ ……OFF、 ⚲ ……ON

●タイマ・カウンタ・データメモリの現在値およびチャンネルのモニタ結果は、10進表示、16進表示、アスキー表示、バイナリ表示の4つの表示切換が可能です。
(例) データメモリ1番の現在値をモニタした場合(現在値45678)



●チャンネルは、任意の接点から16接点を1チャンネルと考えます。
(例) 1005chの場合
1005～1015、1100～1104の
16点を1005chと呼びます。

〈バイナリ表示〉



登録内容の変更

●上書き、挿入の場合

上書き又は挿入したい行に ↑ON ↓OFF キーでカーソルを移動します。 → リレー接点、タイマ、カウンタ、データメモリ、チャンネル等の入力します。 → 挿入キーを押して「上書き」「挿入」を切り替えます。 → 実行キーを押します。

●削除の場合

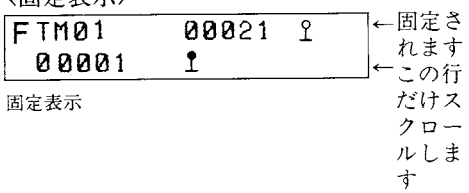
削除したい行に ↑ON ↓OFF キーでカーソルを移動します。 → 削除キーを押します。 → 実行キーを押します。

固定表示

↑ON ↓OFF キーでスクロールしながらモニタする場合、任意の1行を上の行に固定して表示し、下の行だけをスクロールすることができます。

固定表示したい行に ↑ON ↓OFF キーでカーソルを移動します。 → 固定キーを押します。

〈固定表示〉



●固定表示を解除する場合は再度、固定キーを押して下さい。

導通モニタ

プログラムの一行ごとに、オペランドに対応するリレー接点のON-OFF情報をモニタ表示します。
●導通モニタでは、検索、スクロールができます。書き込み、挿入、削除等の編集作業はできません。
●導通モニタで表示するオペランドのON-OFF情報は次の表のとおりです。

オペコード	ON-OFF情報
LD、LDB、AND、ANB、OR、ORB、OUT、OUB、SET、RES、KEEP、DIFU、DIFD、STP、HSP、STG、JMP、INT	オペランドのON-OFF情報 (オペランドがCTHの場合を除く) (LDB、ANB、ORBは、出力を逆転する)
SFT	第1オペランドのON-OFF情報
HKEY	キーの入力を検知したらオン
W-ON、W-OFF、W-UE、W-DE	イベントラッチリレー(第2オペランド)のON-OFF情報
TMR、TMH、TMS	指定されたタイマのON-OFF情報
C、UDC	指定されたカウンタのON-OFF情報
CTC	指定された高速カウンタコンパレータのON-OFF情報
その他	導通モニタしません(－を表示)

ON----------
OFF---------

〔操 作 例〕

キー操作	表示	説明
	ウンテン　チュウ	基本表示
カードモニタ カードモニタ カードモニタ	0 0 0 0 0　　LD　　0 0 0 0 0	導通モニタ機能を呼び出します。0行目が表示されます。
↓ OFF	0 0 0 0 1　　OUT　　0 0 5 0 0	行番号を＋1
↓ OFF	0 0 0 0 2　　END　　—	行番号を＋1
↑ ON	0 0 0 0 1　　OUT　　0 0 5 0 0	行番号を－1
クリア	ウンテン　チュウ	基本表示に戻ります。

3章　命令語詳細

基本命令

命令	シンボル	ニモニック	オペランド	オペランドの値 KZ-10, 16	オペランドの値 KZ-24, 40, 80	機能	実行時間	バイト数	ページ
ロード	nnnn	LD	リレー番号 タイマ・カウンタ番号	0000～2915 T000～T063 C000～C063 CTC0～CTC3	0000～6915 T000～T119 C000～C119 CTC0～CTC3	母線にN.O.接点を接続します。	1.5 ～ 2.5	3 (4)	3-12
ロードバー	nnnn	LDB				母線にN.C.接点を接続します。	2.1 ～ 3.5	3 (6)	3-12
アンド	nnnn	AND				N.O.接点を直列に接続します。	1.5 ～ 2.5	3 (4)	3-13
アンドバー	nnnn	ANB				N.C.接点を直列に接続します。	2.7 ～ 4.5	3 (8)	3-13
オア	nnnn	OR				N.O.接点を並列に接続します。	1.5 ～ 2.5	3 (4)	3-14
オアバー	nnnn	ORB				N.C.接点を並列に接続します。	2.7 ～ 4.5	3 (8)	3-14
アンドロード	—	ANL	—	—		複数あるいは単数の接点で構成されたブロック間を直列に接続します。	1.0 ～ 1.4	1 (2)	3-15
オアロード	—	ORL				複数あるいは単数の接点で構成されたブロック間を並列に接続します。	1.0 ～ 1.4	1 (2)	3-15
アウト	nnnn	OUT	リレー番号	0500～1915 2009 2300～2915	0500～1915 2009 2300～6915	入力のON-OFF状態をそのまま指定されたリレーに出力します。	5.2 ～ 8.4	3 (13)	3-17
アウトバー	nnnn	OUB				入力のON-OFF状態を反転して指定されたリレーに出力します。	6.4 ～ 10.4	3 (17)	3-17
セット	nnnn (SET)	SET		0500～1915 2009 2100～2915 T000～T063 C000～C063	0500～1915 2009 2100～6915 T000～T119 C000～C119	入力がONのとき指定されたリレーを強制的にONにし、保持します。	3.1 ～ 23.0	3 (8)	3-18
リセット	nnnn (RES)	RES		0500～1915 2009 2100～2915 T000～T063 C000～C063 CTH0～CTH1 CTC0～CTC3	0500～1915 2009 2100～6915 T000～T119 C000～C119 CTH0～CTH1 CTC0～CTC3	入力がONのとき指定されたリレー・タイマ・カウンタを強制的にOFFにします。	3.1 ～ 24.0	3 (8)	3-18
タイマ	#ddddd T xxx	TMR	タイマ番号 #設定値	000～063 #00000～#65535	000～119 #00000～#65535	最小ステップ0.1秒16ビットの減算式オンディレイタイマの動作を行ないます。	17.0 ～ 22.0	2 (5)	3-19
高速タイマ	#ddddd TH xxx	TMH (FUN49)				最小ステップ0.01秒 16ビットの減算式オンディレイタイマの動作を行ないます。	17.0 ～ 22.0	2 (5)	3-20
高速1msタイマ	#ddddd TS xxx	TMS (FUN51)				最小ステップ0.001秒16ビットの減算式オンディレイタイマの動作を行ないます。	17.0 ～ 22.0	2 (5)	3-20

(注) バイト数は命令の使用メモリのバイト数を表わします。　　(μs)
バイト数の( )内の数値は命令のオブジェクト使用メモリのバイト数を表わします。

基本命令

命令	シンボル	ニモニック	オペランド	オペランドの値 KZ-10, 16	オペランドの値 KZ-24, 40, 80	機能	実行時間	バイト数	ページ
カウンタ	#ddddd C xxx nnnn	C	#設定値 カウンタ番号 カウント入力リレー	#00000～#65535 000～063 0000～2915	#00000～#65535 000～119 0000～6915	16ビットの加算カウンタの動作をします。	19.0 ～ 21.0	4 (7)	3-22
アップダウンカウンタ	UDC xxx UP DW #ddddd RES	UDC (FUN52)	カウンタ番号 #設定値	000～063 #00000～#65535	000～119 #00000～#65535	16ビットの加減算カウンタの動作をします。	21.0 ～ 26.0	2 (5)	3-24
立上り微分	nnnn DIFU	DIFU (FUN10)	リレー番号	1000～1915	1000～1915 3000～6915	入力の立上りを検出し、指定されたリレーを1スキャンだけONします。	11.0 ～ 13.0	3 (5)	3-25
立下り微分	nnnn DIFD	DIFD (FUN09)				入力の立下りを検出し、指定されたリレーを1スキャンだけONします。	11.0 ～ 13.0	3 (5)	3-25
キープ	KEEP SET nnnn RES	KEEP (FUN22)	リレー番号	0500～1915 2009 2100～2915	0500～1915 2009 2100～6915	S(セット)入力ON時に指定されたリレーをONにして保持し、R(リセット)入力ON時に指定されたリレーをOFFします。	9.0 ～ 26.0	3 (5)	3-26
シフト	SFT D nnnn CLK RES mmmm	SFT (FUN39)	開始リレー番号 終了リレー番号	1000～1915	1000～1915 3000～6915	シフトレジスタの動作を行います。	(※)	5 (7)	3-27
ハイスピード	HSP nnnn	HSP (FUN18)	リレー番号	(KZ-10) 0000～0005 (KZ-16) 0000～0009	(KZ-24) 0000～0015 (KZ-40、80) 0000～0107	指定された入力リレーの時定数を小さくし(25μs)、入力応答性を速くします。	8.0 ～ 9.0	3 (4)	3-29
マスターコントロール	MC	MC (FUN24)	——	——		MCRまでのリレーコイル・タイマ・カウンタなどのON-OFF状態を決定します。	1.8 ～ 3.2	1 (7)	3-30
マスターコントロールリセット	MCR	MCR (FUN25)	——	——		MCと対で使用し、MC命令の終了を示します。	0.6 ～ 1.0	1 (2)	3-30
メモリスイッチ	MEMSW $hhhh	MEMSW (FUN26)	$定数	$0000～$FFFF		メモリスイッチの設定を行ないます。	—	3 (0)	3-31
ノップ	——	NOP (FUN30)	——	——		この命令を実行させても何もしません。	—	1 (0)	3-33
エンド	END	END	——	——		通常プログラムの終了を示します。	—	1 (3)	3-33
エンドハイ	ENDH	ENDH	——	——		全プログラムの終了を示します。	—	1 (0)	3-33

(μs)

(注)　バイト数は命令の使用メモリのバイト数を表わします。
バイト数の(　)内の数値は命令のオブジェクト使用メモリのバイト数を表わします。

(※)　14.0+3.0×n～17.0+3.0×n (nはシフトの段数)

応用命令

命令	シンボル	ニモニック	オペランド	オペランド値 KZ10, 16	オペランド値 KZ24, 40, 80	機能	実行時間 (μs)	バイト数	ページ
ウェイトオン	nnnn ON mmmm	W−ON	nnnn：リレー番号 タイマ・カウンタ番号 mmmm：リレー番号	nnnn：0000～2915 T000～T063 C000～C063 CTC0～CTC3 mmmm：1000～1915	nnnn：0000～6915 T000～T119 C000～C119 CTC0～CTC3 mmmm：1000～1915 3000～6915	第1オペランド(nnnn)のリレー・タイマ・カウンタON状態で第2オペランド(mmmm)のリレーをセットします。	11.0～13.0	5 (7)	5-1
ウェイトオフ	nnnn OFF mmmm	W−OFF				第1オペランド(nnnn)のリレー・タイマ・カウンタOFF状態で第2オペランド(mmmm)のリレーをセットします。	11.0～13.0	5 (7)	5-1
ウェイトアップエッジ	nnnn ↑ mmmm	W−UE				第1オペランド(nnnn)のリレー・タイマ・カウンタの立ち上がりで第2オペランド(mmmm)のリレーをセットします。	14.0～18.0	5 (7)	5-2
ウェイトダウンエッジ	nnnn ↓ mmmm	W−DE				第1オペランド(nnnn)のリレー・タイマ・カウンタの立ち下がりで第2オペランド(mmmm)のリレーをセットします。	14.0～18.0	5 (7)	5-2
コネクト	——	CON (FUN 06)	——	——		出力命令の後の直列接続を示します。	0.4～0.6	1 (1)	5-6
プッシュ	——	MPS	——	——		入力の状態と、演算フラグを記憶します。	11.0～14.0	1 (3)	5-7
リード	——	MRD	——	——		プッシュで記憶した、入力の状態と、演算フラグを読み出します。	8.0～10.0	1 (3)	5-7
ポップ	——	MPP	——	——		プッシュで記憶した、入力状態と、演算フラグを読み出し記憶をクリアします。	10.0～12.0	1 (3)	5-7
ステージ	nnnn STG	STG (FUN 44)	リレー番号	1000～1915	1000～1915 3000～6915	オペランドのリレーがONのとき、STGのブロックの命令を実行します。	11.0～14.0	3 (7)	5-9
ジャンプ	nnnn JMP	JMP (FUN 21)				入力がONのとき、現在実行しているステージのリレーをOFFにして、オペランドのステージに移行します。	10.0～13.0	3 (5)	5-9
エンドステージ	ENDS	ENDS (FUN 14)	——	——		入力がONのとき、現在実行しているステージのリレーをOFFにします。	7.0～9.0	1 (3)	5-9
ステップ	STP nnnn	STP (FUN 45)	リレー番号 タイマ・カウンタ番号 リレー番号	0000～2915 T000～T063 C000～C063 CTC0～CTC3	0000～6915 T000～T119 C000～C119 CTC0～CTC3	オペランドのリレーがONのときSTEまでのプログラムを実行します。	3.7～5.3	3 (7)	5-13
ステップエンド	STE	STE (FUN 43)	——	——		STP命令と対で使用し、ひとつのステップを形成します。	0.00	1 (0)	5-13

(注) バイト数は命令の使用メモリのバイト数を表わします。
バイト数の( )内の数値は命令のオブジェクト使用メモリのバイト数を表わします。

応用命令

命令	シンボル	ニモニック	オペランド	オペランド値 KZ10, 16	オペランド値 KZ24, 40, 80	機能	実行時間	バイト数	ページ
インターバルタイマ	ITVL PLS DMxxxx PAUSE mmmm RES	ITVL	DMxxxx：データメモリ mmmm：リレー番号	DMxxxx：DM0000～DM0985 mmmm：1000～1912	DMxxxx：DM0000～DM1985 mmmm：1000～1912 3000～6912	設定されたモードで、パルス間隔やパルス長を計測します。	29.0 ↓ 79.0	5 (7)	5-15
8ビット高速カウンタ	CTH0 nnnn	CTH	カウント入力リレー	0004 2100～2102		応答周波数10kHzの8ビット(0～255)の加算カウンタの動作を行います。	10.0 ↓ 48.0	4 (4)	5-17
8ビット高速カウンタコンパレータ	#ddddd CTCx	CTC	x：コンパレータ番号 ddddd：#設定値	x：0, 1 ddddd：#000～#255		設定値と高速カウンタ(CTH0)の現在値をハード的に比較し等しくなったときにセットします。	3.0 ↓ 4.0	2 (5)	5-17
16ビット高速カウンタ	CTH1 nnnn	CTH	カウント入力リレー	0005, 0500 2200～2202		応答周波数10kHzの16ビット(0～65535)の加算カウンタの動作を行います。	9.0 ↓ 39.0	4 (4)	5-17
16ビット高速カウンタコンパレータ	#ddddd CTCx	CTC	x：コンパレータ番号 ddddd：#設定値	x：2, 3 ddddd：#00000～#65535		設定値と高速カウンタ(CTH1)の現在値をハード的に比較し等しくなったときにセットします。	3.0 ↓ 4.0	2 (5)	5-17
サブルーチンコール	xx CALL	CALL (FUN 03)	サブルーチン番号	00～99		オペランドで指定したサブルーチンを実行します。	8.0 ↓ 10.0	2 (6)	5-20
サブルーチンエントリ	SBN xx	SBN (FUN 38)	サブルーチン番号	00～99		オペランドで指定したサブルーチンの領域の先頭を示します。	0.00	2 (0)	5-20
サブルーチンリターン	RET	RET (FUN 33)	——	——		サブルーチンを終了します。	4.6 ↓ 6.2	1 (5)	5-20
繰り返し範囲開始	FOR nnnn	FOR (FUN 16)	# 定数 データメモリ	#00000～#65535 DM0000～DM0999 TM00～TM29	#00000～#65535 DM0000～DM1999 TM00～TM29	オペランドで指定した回数だけFOR～NEXT間を繰り返し実行します。	5.0 ↓ 20.0	3 (15)	5-22
繰り返し範囲終了	NEXT	NEXT (FUN 29)	——	——		繰り返しの範囲の最後を示します。	1.7 ↓ 2.3	1 (3)	5-22
16キー入力	HKEY nnnn mmmm	HKEY (FUN 17)	nnnn：入力リレー mmmm：出力リレー	nnnn：0000～0415 mmmm：0500～0915		16キーデータを時分割で読み込み、特殊補助リレー2900～2915にキーの状態を入れます。	71 ↓ 370	5 (7)	5-24

(注) バイト数は命令の使用バイト数を表わします。
バイト数の( )内の数値は命令のオブジェクト使用メモリのバイト数を表わします。

(μs)

演　算　命　令

命　令	シンボル	ニモニック	オペランド	オペランドの値 KZ10, 16	オペランドの値 KZ24, 40, 80	機　能	実行時間	バイト数	ページ
データメモリライト	nnnn DW DMxxxx	DW	nnnn：#　定数 $　定数 DMxxxx：データメモリ	nnnn：#00000～#65535 $0000～$FFFF DMxxxx：DM0000～DM0999	nnnn：#00000～#65535 $0000～$FFFF DMxxxx：DM0000～DM1999	データメモリに直接、定数を書き込みます。	3.6～6.0	5 (6)	5-27
トリマ設定値入力	X TMIN	TMIN (FUN 50)	トリマ番号	0	0～1	トリマの回転角を0～249の数値に変換して、内部レジスタへ入力します。	13.0～15.0	1 (4)	5-28
	X TMIN	@TMIN @(FUN 50)					25.0～29.0	1 (11)	
ロードA	nnnn LDA	LDA (FUN 23)	リレー番号 カウンタ/タイマ番号 データメモリ #　定数 $　定数 #TM××　#TM××	0000～2915 T000～T063 C000～C063 CTH0～CTH1 DM0000～DM0999 TM00～TM31 #00000～#65535 $0000～$FFFF #TM00～#TM29	0000～6915 T000～T119 C000～C119 CTH0～CTH1 DM0000～DM1999 TM00～TM31 #00000～#65535 $0000～$FFFF #TM00～#TM29	オペランドで指定された値を内部レジスタに入力します。(タイマ・カウンタ番号がオペランドに指定されたとき、現在値を内部レジスタに入力します)	10.0～46.0	3 (5)	5-29
	nnnn LDA	@LDA @(FUN 23)					22.0～60.0	3 (12)	
ストアA	nnnn STA	STA (FUN 42)	リレー番号 カウンタ/タイマ番号 データメモリ #TM××	0500～1915 2100～2915 T000～T063 C000～C063 CTC0～CTC3 DM0000～DM0999 TM00～TM29 #TM00～#TM29	0500～1915 2100～6915 T000～T119 C000～C119 CTC0～CTC3 DM0000～DM1999 TM00～TM29 #TM00～#TM29	内部レジスタの値をオペランドで指定された先に転送します。(タイマ・カウンタ番号がオペランドで指定されたとき、設定値が変更されます。)	9.0～99.0	3 (5)	5-29
	nnnn STA	@STA @(FUN 42)					21.0～113.0	3 (12)	
コンペア	nnnn CMP	CMP (FUN 04)	データメモリ #　定数 $　定数 #TM××　#TM××	DM0000～DM0999 TM00～TM31 #0000～#65535 $0000～$FFFF #TM00～#TM29	DM0000～DM1999 TM00～TM31 #0000～#65535 $0000～$FFFF #TM00～#TM29	内部レジスタの値をオペランドで指定された値と比較します。	14.0～68.0	3 (5)	5-33
	nnnn CMP	@CMP @(FUN 04)					26.0～82.0	3 (12)	
アッド	nnnn ADD	ADD (FUN 00)				内部レジスタの値にオペランドで指定された値を加算し、結果を内部レジスタに再度入力します。	13.0～65.0	3 (5)	5-34
	nnnn ADD	@ADD @(FUN 00)					25.0～79.0	3 (12)	

(μs)

(注) バイト数は命令の使用メモリのバイト数を表わします。
バイト数の( )内の数値は命令のオブジェクト使用メモリのバイト数を表わします。

演　算　命　令

命　令	シンボル	ニモニック	オペランド	オペランドの値 KZ10, 16	オペランドの値 KZ24, 40, 80	機　能	実行時間	バイト数	ページ
サ　ブ	nnnn SUB	SUB (FUN 46)	データメモリ # 定数 $ 定数 #TM ×× リレー番号	DM0000 ～DM0999 TM00～TM31 #00000～#65535 $0000～$FFFF #TM00～#TM29 0000～2915 DM0000 ～DM0999 TM00～TM31 #00000～#65535 $0000～$FFFF #TM00～#TM29	DM0000 ～DM1999 TM00～TM31 #00000～#65535 $0000～$FFFF #TM00～#TM29 0000～6915 DM0000 ～DM1999 TM00～TM31 #00000～#65535 $0000～$FFFF #TM00～#TM29	内部レジスタの値からオペランドで指定された値を減算し、結果を内部レジスタに再度入力します。	15.0 ～ 68.0	3 (5)	5-34
	nnnn @SUB	@SUB @(FUN 46)					27.0 ～ 82.0	3 (12)	
マ　ル　チ	nnnn MUL	MUL (FUN 28)				内部レジスタの値とオペランドで指定された値を乗算し、結果を内部レジスタに再度入力します。	17.0 ～ 71.0	3 (5)	5-34
	nnnn @MUL	@MUL @(FUN 28)					29.0 ～ 85.0	3 (12)	
デ　バ　イ　ド	nnnn DIV	DIV (FUN 11)				内部レジスタの値とオペランドで指定された値を除算し、結果を内部レジスタに再度入力します。	22.0 ～ 79.0	3 (5)	5-34
	nnnn @DIV	@DIV @(FUN 11)					34.0 ～ 93.0	3 (12)	
ア　ン　ド　A	nnnn ANDA	ANDA (FUN 01)				内部レジスタの値とオペランドで指定された値を16ビットの各ビットごとにANDし、結果を内部レジスタに再度入力します。	11.0 ～ 64.0	3 (5)	5-37
	nnnn @ANDA	@ANDA @(FUN 01)					23.0 ～ 78.0	3 (12)	
オ　ア　A	nnnn ORA	ORA (FUN 31)				内部レジスタの値とオペランドで指定された値を16ビットの各ビットごとにORし、結果を内部レジスタに再度入力します。	11.0 ～ 63.0	3 (5)	5-38
	nnnn @ORA	@ORA @(FUN 31)					23.0 ～ 77.0	3 (12)	

(注) バイト数は命令の使用メモリのバイト数を表わします。　　(μs)
バイト数の( )内の数値は命令のオブジェクト使用メモリのバイト数を表わします。

演　算　命　令

命　令	シンボル	ニモニック	オペランド	オペランドの値 KZ10,16	オペランドの値 KZ24,40,80	機　能	実行時間	バイト数	ページ
エクスクルーシブオアA	nnnn EORA	EORA (FUN 15)	リレー番号 データメモリ # 定数 $ 定数 #TM ××	0000～2915 DM0000 ～DM0999 TM00～TM31 #00000～#65535 $0000～$FFFF #TM00～#TM29	0000～6915 DM0000 ～DM1999 TM00～TM31 #00000～#65535 $0000～$FFFF #TM00～#TM29	内部レジスタの値とオペランドを各々のビットで排他的論理和をおこない内部レジスタへ再入力します。	10.0 ↓ 63.0	3 (5)	5-39
	@nnnn EORA	@EORA @(FUN 15)					22.0 ↓ 77.0	3 (12)	
シフトライトA	#dd SRA	SRA (FUN 41)	# 定数	#01～#16		内部レジスタの内容をオペランド値の回数分右へシフトします。	11.0 ↓ 28.0	2 (4)	5-40
	@#dd SRA	@SRA @(FUN 41)					23.0 ↓ 42.0	2 (11)	
シフトレフトA	#dd SLA	SLA (FUN 40)				内部レジスタの内容をオペランド値の回数分左へシフトします。	10.0 ↓ 28.0	2 (4)	5-40
	@#dd SLA	@SLA @(FUN 40)					22.0 ↓ 42.0	2 (11)	
ローテートライトA	#dd RRA	RRA (FUN 37)				内部レジスタの内容とキャリー(2009)をオペランド値の回数分右回りに回転します。	12.0 ↓ 30.0	2 (4)	5-41
	@#dd RRA	@RRA @(FUN 37)					24.0 ↓ 44.0	2 (11)	
ローテートレフトA	#dd RLA	RLA (FUN 35)				内部レジスタの内容とキャリー(2009)をオペランド値の回数分左回りに回転します。	12.0 ↓ 29.0	2 (4)	5-41
	@#dd RLA	@RLA @(FUN 35)					24.0 ↓ 43.0	2 (11)	

(注) バイト数は命令の使用メモリのバイト数を表わします。
　　バイト数の( )内の数値はオブジェクト使用メモリのバイト数を表わします。

(μs)

演　算　命　令

命　令	シンボル	ニモニック	オペランド	オペランドの値 KZ10,16	オペランドの値 KZ24,40,80	機　能	実行時間	バイト数	ページ
コンプリメント	COM	COM (FUN 05)	―	―	―	内部レジスタの各ビットを反転します。	5.0 ～ 6.0	1 (3)	5-42
	@COM	@COM @(FUN 05)					17.0 ～ 20.0	1 (10)	
インクリメントメモリ	nnnn INC	INC (FUN 19)	データメモリ	DM0000 ～DM0999 TM00～TM29	DM0000 ～DM1999 TM00～TM29	オペランドで指定したメモリデータに1を加算します。	12.0 ～ 15.0	3 (5)	5-43
	@nnnn INC	@INC @(FUN 19)					24.0 ～ 29.0	3 (12)	
ディクリメントメモリ	nnnn DEC	DEC (FUN 07)				オペランドで指定したメモリデータから1を減算します。	14.0 ～ 17.0	3 (5)	5-43
	@nnnn DEC	@DEC @(FUN 07)					26.0 ～ 31.0	3 (12)	
マルチプレクス	#d MPX	MPX (FUN 27)	# 定数	#0～#3		オペランドで指定した内部レジスタの1桁分(4ビット)の数を16ビットデータに変換します。	24.0 ～ 25.0	1 (4)	5-44
	@#d MPX	@MPX @(FUN 27)					36.0 ～ 39.0	1 (11)	
デマルチプレクス	DMX	DMX (FUN 12)	―	―	―	内部レジスタの16ビットデータの最も上位で1になっているビットの位置を4ビットデータに変換します。	28.0 ～ 30.0	1 (3)	5-44
	@DMX	@DMX @(FUN 12)					40.0 ～ 44.0	1 (10)	

(注) バイト数は命令の使用メモリのバイト数を表わします。
　　バイト数の( )内の数値は命令のオブジェクト使用メモリのバイト数を表わします。

(μs)

演　算　命　令

命　令	シンボル	ニモニック	オペランド	オペランドの値 KZ10, 16	オペランドの値 KZ24, 40, 80	機　能	実行時間	バイト数	ページ
トランスファBCD	TBCD	TBCD (FUN 47)	——	——	——	内部レジスタの内容(16ビット2進数)を4桁BCDに変換します。	19.0 ～ 21.0	1 (3)	5-45
	@TBCD	@TBCD @(FUN 47)					31.0 ～ 35.0	1 (10)	
トランスファBIN	TBIN	TBIN (FUN 48)	——	——	——	内部レジスタの内容(4桁BCD)を16ビットの2進数に変換します。	18.0 ～ 20.0	1 (3)	5-45
	@TBIN	@TBIN @(FUN 48)					30.0 ～ 34.0	1 (10)	
アスキーコード変換	ASC	ASC (FUN 02)	——	——	——	内部レジスタの下位8バイトの数値2桁を2桁のASCII文字コードに変換します。	8.0	1 (3)	5-46
	@ASC	@ASC @(FUN 02)					20.0 ～ 22.0	1 (10)	
アスキーコード逆変換	RASC	RASC (FUN 32)	——	——	——	内部レジスタの2桁のASCII文字コードを数値2桁に変換します。	12.0 ～ 14.0	1 (3)	5-46
	@RASC	@RASC @(FUN 32)					24.0 ～ 28.0	1 (10)	
平方根	ROOT	ROOT (FUN 36)	——	——	——	TM00を上位バイト、内部レジスタを下位バイトとする32ビットの平方根を求め内部レジスタへ入力します。	102.0 ～ 103.0	1 (3)	5-47
	@ROOT	@ROOT @(FUN 36)					114.0 ～ 117.0	1 (10)	

(注) バイト数は命令の使用メモリのバイト数を表わします。　　(μs)
バイト数の( )内の数値はオブジェクト使用メモリのバイト数を表わします。

割込み命令

命令	シンボル	ニモニック	オペランド	オペランドの値 KZ10, 16	オペランドの値 KZ24, 40, 80	機能	実行時間	バイト数	ページ
割込み禁止	-|DI|-	DI (FUN 08)	——	——		割込みプログラムの実行を禁止します。	3.0 ~ 4.6	1 (4)	5-49
割込み許可	-|EI|-	EI (FUN 13)	——	——		割込みプログラムの実行を許可します。	3.0 ~ 4.6	1 (4)	5-49
インタ	INT nnn	INT (FUN 20)	リレー番号	000~003		入力 000~003 の立ち上がりで、または 003 の立ち下がりで、INT~RETI までの割込みプログラムを実行します。	35.0 ~ 48.0	1	5-50
	INT CTCx		コンパレータ番号	CTC0~CTC3		コンパレータ CTC0~CTC3 までのコンペアマッチで INT~RETI までの割込みプログラムを実行します。	35.0 ~ 48.8	(8)	
リターンインタ	RETI	RETI (FUN 34)	——	——		割込みプログラムの最後を示します。	20.0 ~ 23.0	1 (3)	5-50

(注) バイト数は命令の使用メモリのバイト数を表わします。
バイト数の( )内の数値はオブジェクト使用メモリのバイト数を表わします。

(μs)

命令語FUN NO一覧 (アルファベット順)

FUN NO	命令	FUN NO	命令	FUN NO	命令	FUN NO	命令
0	ADD	15	EORA	30	NOP	45	STP
1	ANDA	16	FOR	31	ORA	46	SUB
2	ASC	17	HKEY	32	RASC	47	TBCD
3	CALL	18	HSP	33	RET	48	TBIN
4	CMP	19	INC	34	RETI	49	TMH
5	COM	20	INT	35	RLA	50	TMIN
6	CON	21	JMP	36	ROOT	51	TMS
7	DEC	22	KEEP	37	RRA	52	UDC
8	DI	23	LDA	38	SBN		
9	DIFD	24	MC	39	SFT		
10	DIFU	25	MCR	40	SLA		
11	DIV	26	MEMSW	41	SRA		
12	DMX	27	MPX	42	STA		
13	EI	28	MUL	43	STE		
14	ENDS	29	NEXT	44	STG		

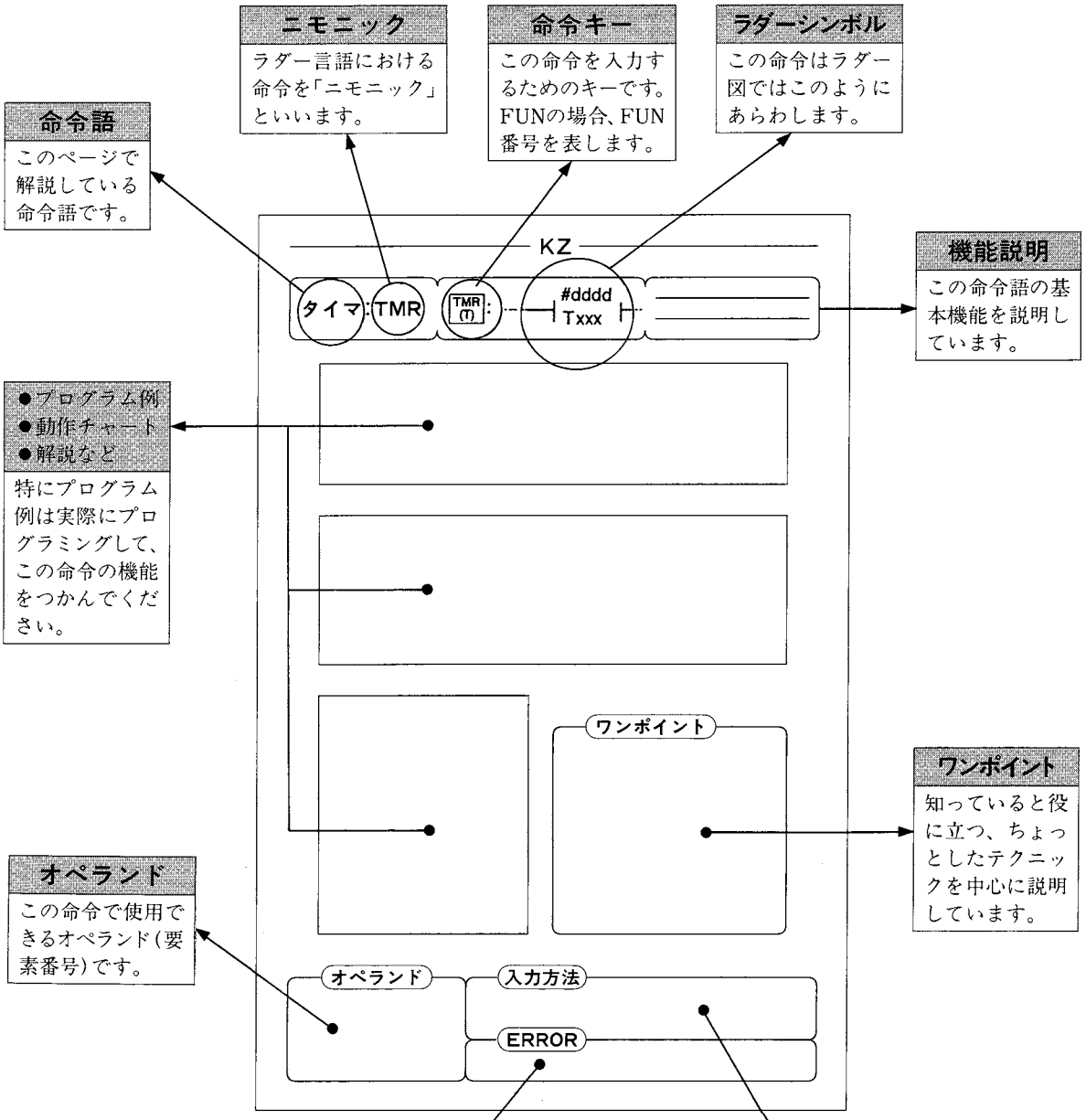
ワンポイント

ファンクションナンバーが分からない場合

FUN キーを押して、FUN ナンバーにカーソルがある時、↑ON ↓OFF キーで検索することができます。

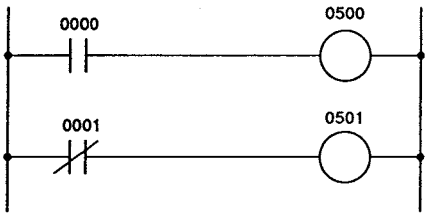
命令語の読み方

この章では、各命令語について次のように解説しています。



ロ　ー　ド：L　D	LD 7 : nnnn	母線にN.O.接点を接続します。
ロードバー：LDB	LDB 4 : nnnn	母線にN.C.接点を接続します。

（例）



コーディング

行番号	命　令	オペランド
0000	LD	0000
0001	OUT	0500
0002	LDB	0001
0003	OUT	0501

● 入力リレー0000が「ON」→出力リレー0500が「ON」
　入力リレー0001が「OFF」→出力リレー0501が「ON」 } となります

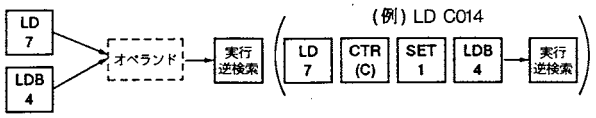
■解説

● LD、LDB命令は母線につながる接点に使用します。
● また、ANL、ORL命令使用時のブロック分け開始用としても使用します。

オペランド

KZ-10、16	KZ-24、40、80
0000-2915	0000-6915
T000-T063	T000-T119
C000-C063	C000-C119
CTC0-CTC3	CTC0-CTC3

入力方法

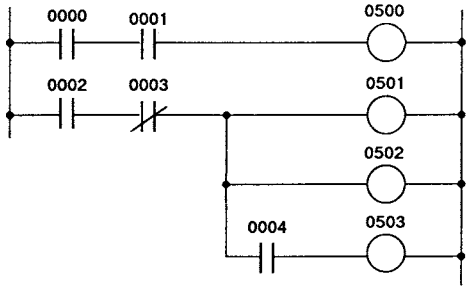


ERROR!!

● オペランドエラー：その番号のタイマ（カウンタ）は、すでにカウンタ（タイマ）として存在している。
　該当するタイマ（カウンタ）が存在しない。
● カ イ ロ エ ラ ー：ANL、ORL使用時に接続ブロックが9ヶ以上になった。
　LD、LDB命令に対する出力命令が存在しない。

アンド：AND	AND 8 : nnnn	N.O.接点を直列に接続します。
アンドバー：ANB	ANB 5 : nnnn	N.C.接点を直列に接続します。

(例)



コーディング

行番号	命　令	オペランド
0000	LD	0000
0001	AND	0001
0002	OUT	0500
0003	LD	0002
0004	ANB	0003
0005	OUT	0501
0006	OUT	0502
0007	AND	0004
0008	OUT	0503

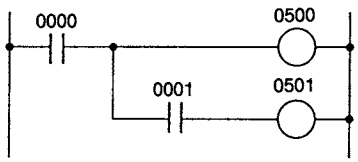
0503を「ON」させるためには、0002が「ON」かつ0003が「OFF」かつ0004が「ON」でなければなりません。この「かつ」にあたる命令がAND・ANBであるといえます。

■解説

- プログラム内であれば、直列接続する接点の数に制限はありません。
- (例)のOUT 503のように、OUT 命令のすぐ後に接点(例では0004)を介して他のリレーに接続する場合、接続方法を誤らなければ何度でも使用できます。

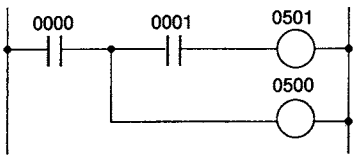
ワンポイント

- 次の回路は、どちらも同じ動作をしますが、コーディングが異なりますので注意して下さい。



コーディング

行番号	命　令	オペランド
0000	LD	0000
0001	OUT	0500
0002	AND	0001
0003	OUT	0501



コーディング

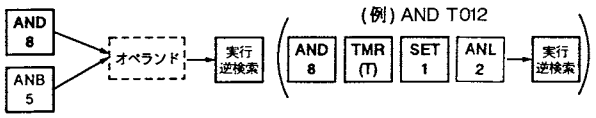
行番号	命　令	オペランド
0000	LD	0000
0001	MPS	
0002	AND	0001
0003	OUT	0501
0004	MPP	
0005	OUT	0500

MPS MPP 命令については P. 5−7 をご覧ください。

オペランド

KZ-10、16	KZ-24、40、80
0000-2915	0000-6915
T000-T063	T000-T119
C000-C063	C000-C119
CTC0-CTC3	CTC0-CTC3

入力方法

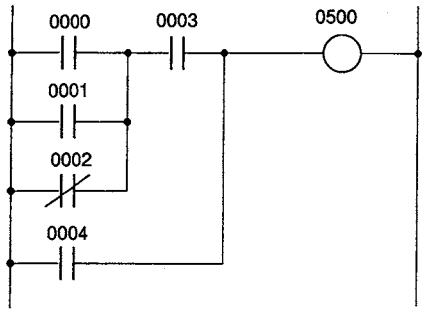


ERROR!!

- オペランドエラー：その番号のタイマ（カウンタ）は、すでにカウンタ（タイマ）として存在している。
  該当するタイマ（カウンタ）が存在しない。

オ　　ア:O　R	OR 9 : nnnn	N.O.接点を並列に接続します。
オ ア バ ー:ORB	ORB 6 : nnnn	N.C.接点を並列に接続します。

(例)



コーディング

行番号	命　令	オペランド
0000	LD	0000
0001	OR	0001
0002	ORB	0002
0003	AND	0003
0004	OR	0004
0005	OUT	0500

0500を「ON」させるためには、
- 0003が「ON」かつ
  - 0000が「ON」　または、
  - 0001が「ON」　または、
  - 0002が「OFF」
- または、
- 0004が「ON」

と、なります。

この「または」にあたる命令がOR、ORBであるといえます。

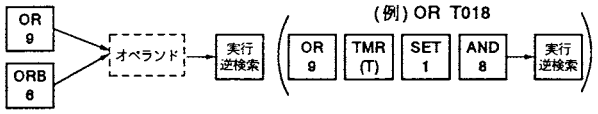
■解説

- OR、ORB命令は、この命令のある行から前行のLD、LDB命令のある行に対して並列接続されます。
- プログラム内であれば、並列接続の数に制限はありません。

オペランド

KZ-10、16	KZ-24、40、80
0000-2915	0000-6915
T000-T063	T000-T119
C000-C063	C000-C119
CTC0-CTC3	CTC0-CTC3

入力方法



ERROR!!

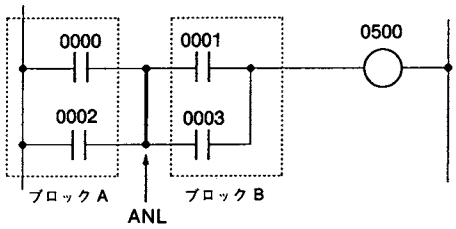
- オペランドエラー：その番号のタイマ（カウンタ）は、すでにカウンタ（タイマ）として存在している。
  該当するタイマ（カウンタ）が存在しない。

アンドロード：ANL

ANL 2 :

複数、あるいは単数の接点で構成されたブロック間を直列に接続します。

(例)



コーディング

行番号	命　令	オペランド	
0000	LD	0000	ブロックA
0001	OR	0002	
0002	LD	0001	ブロックB
0003	OR	0003	
0004	ANL		A・Bの直列接続
0005	OUT	0500	

■解説

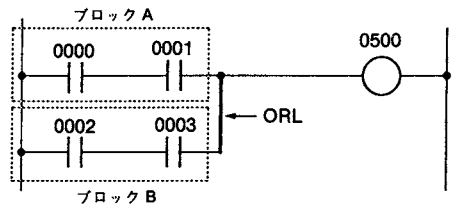
●(例)は、ブロックA(0000 / 0002)とブロックB(0001 / 0003)との直列接続である、と考えることができます。このときの接続に使用する命令がANLです。

オアロード：ORL

ORL 3 :

複数、あるいは単数の接点で構成されたブロック間を並列に接続します。

(例)



コーディング

行番号	命　令	オペランド	
0000	LD	0000	ブロックA
0001	AND	0001	
0002	LD	0002	ブロックB
0003	AND	0003	
0004	ORL		A・Bの並列接続
0005	OUT	0500	

■解説

●(例)は、ブロックA(0000 0001)とブロックB(0002 0003)との並列接続である、と考えることができます。このときの接続に使用する命令がORLです。

オペランド

―――

入力方法

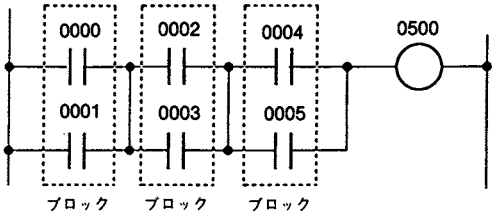
ANL 2 → 実行 逆検索 : ORL 3 → 実行 逆検索

ERROR!!

●カイロエラー：接続するブロックが存在しないのに接続しようとした。
ANL、ORLの入力をしていない。

ANL、ORL命令の例題と注意点

■ANL命令の例題



●ANL命令は何回でも続けて使用できますが、②の方法でプログラムする場合、ANLの数はその前にあるブロック数よりひとつ少ない数を入力します。
●また、②の場合ANLの数は8個以下にしてください。
①の場合、ANLの使用個数に制限はありません。

コーディング例①

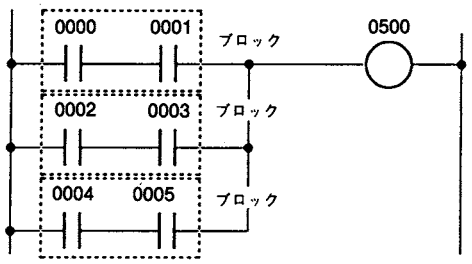
行番号	命　令	オペランド
0000	LD	0000
0001	OR	0001
0002	LD	0002
0003	OR	0003
0004	ANL	
0005	LD	0004
0006	OR	0005
0007	ANL	
0008	OUT	0500

コーディング例②

行番号	命　令	オペランド
0000	LD	0000
0001	OR	0001
0002	LD	0002
0003	OR	0003
0004	LD	0004
0005	OR	0005
0006	ANL	
0007	ANL	
0008	OUT	0500

ERROR!!
②の場合、ANLが9個以上になると「カイロエラー」となります。

■ORL命令の例題



●ORL命令は何回でも続けて使用できますが、②の方法でプログラムする場合、ORLの数はその前にあるブロック数よりひとつ少ない数を入力します。
●また、②の場合ORLの数は8個以下にしてください。
①の場合、ORLの使用個数に制限はありません。

コーディング例①

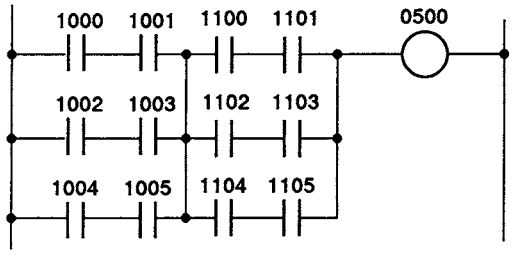
行番号	命　令	オペランド
0000	LD	0000
0001	AND	0001
0002	LD	0002
0003	AND	0003
0004	ORL	
0005	LD	0004
0006	AND	0005
0007	ORL	
0008	OUT	0500

コーディング例②

行番号	命　令	オペランド
0000	LD	0000
0001	AND	0001
0002	LD	0002
0003	AND	0003
0004	LD	0004
0005	AND	0005
0006	ORL	
0007	ORL	
0008	OUT	0500

ERROR!!
②の場合、ORLが9個以上になると「カイロエラー」となります。

ANL、ORL命令の応用例



コーディング例

行番号	命　令	オペランド
0000	LD	1000
0001	AND	1001
0002	LD	1002
0003	AND	1003
0004	ORL	
0005	LD	1004
0006	AND	1005
0007	ORL	
0008	LD	1100
0009	AND	1101
0010	LD	1102
0011	AND	1103
0012	ORL	
0013	LD	1104
0014	AND	1105
0015	ORL	
0016	ANL	
0017	OUT	0500

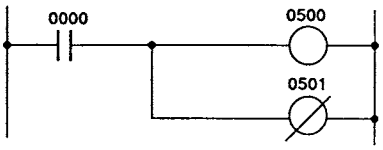
ア　ウ　ト:OUT　　OUT A : nnnn

入力のON-OFF状態をそのまま指定されたリレーに出力します。

アウトバー:OUB　　OUB B : nnnn

入力のON-OFF状態を反転して指定されたリレーに出力します。

(例)



コーディング

行番号	命　令	オペランド
0000	LD	0000
0001	OUT	0500
0002	OUB	0501

●入力リレー0000が「ON」→出力リレー0500が「ON」
　出力リレー0501が「OFF」　} となります。

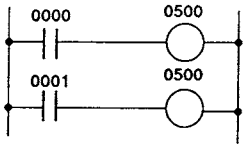
■解説

●OUT、OUB命令は、出力リレー、内部補助リレー、保持リレーに対するコイル駆動命令で、入力リレー、タイマ、カウンタ、データメモリなどには使用しません。

●OUT、OUB命令は、(例)のようにひとつの入力に対し、複数個書くことができ、使用個数に制限はありません。

ワンポイント

右図の様に同一のリレーコイルに対して、OUT、OUB命令を複数個使用することができます。(2重コイル)
しかし、プログラム上、後から書かれたものを優先出力します。(図では0001が「OFF」のとき、0000が「ON」しても0500は「ON」しません。)



オペランド

KZ-10、16	KZ-24、40、80
0500～1915	0500～1915
2009	2009
2300～2915	2300～6915

入力方法

OUT A → オペランド → 実行 逆検索 : OUB B → オペランド → 実行 逆検索

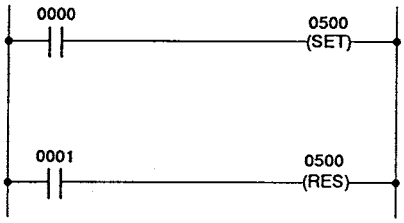
ERROR!!

●オペランドエラー：その番号のタイマ（カウンタ）は、すでにカウンタ（タイマ）として存在している。
　該当するタイマ（カウンタ）が存在しない。

●カイロエラー：入力回路に間違いがある。(例：対応する入力がない)

セット:SET	SET 1 : nnnn (SET)	入力がONのとき、指定されたリレーを強制的にONにし、保持します。
リセット:RES	RES 0 : nnnn (RES)	入力がONのとき、指定されたリレー・タイマ・カウンタを強制的にOFFにします。

(例)



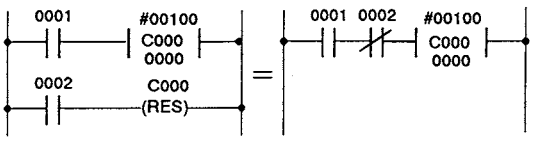
コーディング

行番号	命　令	オペランド
0000	LD	0000
0001	SET	0500
0002	LD	0001
0003	RES	0500

●出力リレー0500は入力リレー0000が「OFF」したあとも「ON」し続けます。
入力リレー0001が「ON」したとき→出力リレー0500が「OFF」します。

■解説

●SET命令は自己保持命令であり、RESが自己保持解除命令です。
●RES命令はリレーコイルをOFFさせるだけでなく、タイマ・カウンタをオペランドに指定した場合、タイマ・カウンタは「リセット信号」が入ったのと同じ動作をします。



カウンタ (C) 命令 → P3−22 をご覧ください。

保持機能について → P3−26、P3−31 をご覧ください。

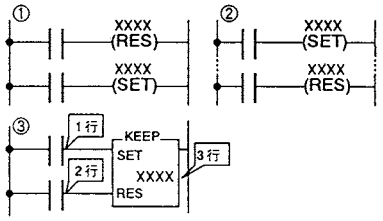
●SET・RES命令はどちらを先に書いてもかまいませんが、行番号が大きな方が優先的に実行されます。(例)の場合、0000と0001が共にONのとき、RESが優先し、0500はOFF状態となります。

ワンポイント

SET・RESとKEEP命令の違い

SET・RES と KEEP (→ P3−26) は基本的に同じ動作をします。

●SET・RES はそれぞれが独立して使用できるため、① SET・RES の順を入れ替えてプログラムができます。
②SET・RES間に他のプログラムが入っても良い。
といった利点があります。
●KEEPはプログラム行数が3行で済みますので、数多く使用するとき、行数の節約ができます。(③)



オペランド

KZ-10、16	KZ-24、40、80
0500～1915	0500～1915
2009	2009
2100～2915	2100～6915
T000～T063	T000～T119
C000～C063	C000～C119
CTH0～CTH1(RESのみ)	
CTC0～CTC3(RESのみ)	

入力方法

SET 1 → オペランド → 実行 逆検索 : RES 0 → オペランド → 実行 逆検索

ERROR!!

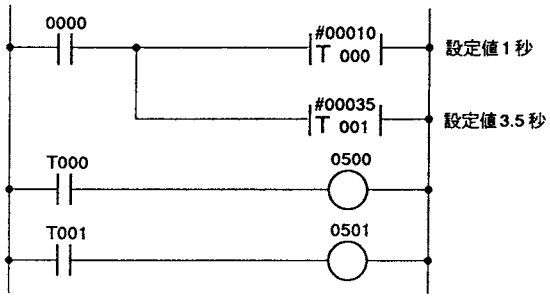
●オペランドエラー：その番号のタイマ（カウンタ）は、すでにカウンタ（タイマ）として存在している。
該当するタイマ（カウンタ）が存在しない。
●カイロエラー：入力回路に間違いがある。(例：対応する入力がない)

タ　イ　マ：TMR

TMR (T) : ……┤#ddddd T xxx├……

最小ステップ0.1秒16ビットの減算式オンディレイタイマの動作を行ないます。

（例）

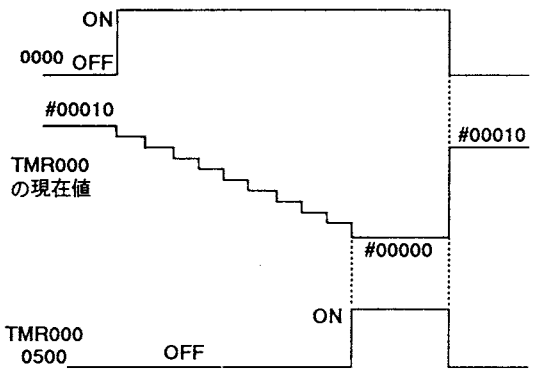


コーディング

行番号	命　令	オペランド	
0000	LD	0000	
0001	TMR	000＃00010	…0.1秒×10＝1秒
0002	TMR	001＃00035	…0.1秒×35＝3.5秒
0003	LD	T000	
0004	OUT	0500	
0005	LD	T001	
0006	OUT	0501	

0000が「ON」すると、1秒後に0500が、3.5秒後に0501が「ON」します。

■動作チャート（上例TMR 000＃0010と0500の動作）



- TMR000は現在値が「＃00000」になったとき接点T000がONし、出力リレー0500が「ON」します。
- 入力条件（ここでは入力リレー0000）がOFFになったとき、TMR000の現在値は設定値に戻り接点T000もOFFします。

■解説

- TMRで指定された番号を接点として使用する場合、使用回数に制限はありません。（N.O.接点、N.C.接点とも）
- TMRの現在値は運転停止時や電源OFF時には保持されず、リセット状態（設定値に戻る）となります。
- タイマの精度は（設定値－0.1秒）＋（1スキャンタイム）以内です。
- タイマ番号は、高速タイマ/高速1msタイマ/カウンタ/アップダウンカウンタとの二重使用はできません。

高速タイマ（0.01秒タイマ）→ P3－20
高速1msタイマ（0.001秒タイマ）→ P3－20

オペランド

KZ-10、16	KZ-24、40、80
番号000～063	番号000～119
設定値	設定値
＃00000～＃65535	＃00000～＃65535

入力方法

TMR (T) → オペランド（TMR番号） → ➡ → オペランド（設定値） → 実行 逆検索

ERROR!!

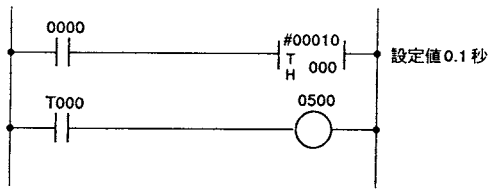
- オペランドエラー：その番号のタイマはすでにタイマ（カウンタ）として存在している。オペランドが不正である。
- カイロエラー：入力回路に間違いがある。（例：入力がない、入力が2つ以上ある。）

高速タイマ：TMH

FUN LDB 4 OR 9 : #ddddd TH xxx

最小ステップ0.01秒16ビットの減算式オンディレイタイマの動作を行ないます。

(例)



コーディング

行番号	命　令	オペランド	
0000	LD	0000	
0001	TMH	000＃00010	…0.01秒×10＝0.1秒
0002	LD	T000	
0003	OUT	0500	

●0000が「ON」すると0.1秒後に0500が「ON」します。

■解説(例の動作)

●TMHは現在値が「＃00000」になったとき指定された番号のコイルが「ON」します。
●入力条件(入力リレー0000)がOFFになったとき、現在値は設定値に戻ります。
●TMHで指定された番号を接点として使用する場合、使用回数に制限はありません。
(N.O.接点、N.C.接点とも)
●TMHの現在値は運転停止時や電源OFF時には保持されず、リセット状態(設定値に戻る)となります。
●タイマの精度は (設定値−0.01秒)＋(1スキャンタイム)以内 です。
●高速タイマの番号は、タイマ/高速1msタイマ/カウンタ/アップダウンカウンタとの二重使用はできません。

オペランド

KZ-10、16	KZ-24、40、80
番号000～063 設定値 ＃00000～＃65535	番号000～119 設定値 ＃00000～＃65535

入力方法

FUN → LDB 4 → OR 9 → 実行逆検索 → オペランド(TMS番号) → ➡ → オペランド(設定値) → 実行逆検索

ERROR!!

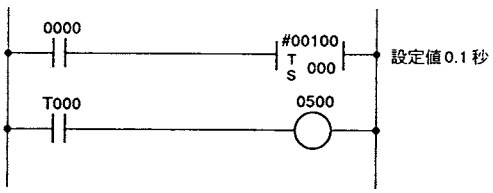
●オペランドエラー：その番号のタイマはすでにタイマ(カウンタ)として存在している。
●カ イ ロ エ ラ ー：入力回路に間違いがある。(例：入力がない、入力が多すぎる。)

高速1msタイマ：TMS

FUN ANB 5 SET 1 : #ddddd TS xxx

最小ステップ0.001秒16ビットの減算式オンディレイタイマの動作を行います。

(例)



コーディング

行番号	命　令	オペランド	
0000	LD	0000	
0001	TMS	000＃00100	…0.001秒×100＝0.1秒
0002	LD	T000	
0003	OUT	0500	

●0000が「ON」すると0.1秒後に0500がONします。

■解説(例の動作)

●TMSは現在値が「＃00000」になったとき指定された番号のコイルが「ON」します。
●入力条件(入力リレー0000)がOFFになったとき、現在値は設定値に戻ります。
●TMSで指定された番号を接点として使用する場合、使用回数に制限はありません。
(N.O.接点、N.C.接点とも)
●TMSの現在値は運転停止時や電源OFF時には保持されず、リセット状態(設定値に戻る)となります。
●タイマの精度は (設定値−0.001秒)＋(1スキャンタイム)以内 です。
●高速1msタイマの番号はタイマ/高速タイマ/カウンタ/アップダウンカウンタとの二重使用はできません。

オペランド

KZ-10、16	KZ-24、40、80
番号000～063 設定値 ＃00000～＃65535	番号000～119 設定値 ＃00000～＃65535

入力方法

FUN → ANB 5 → SET 1 → 実行逆検索 → オペランド(TMS番号) → ➡ → オペランド(設定値) → 実行逆検索

ERROR!!

●オペランドエラー：その番号のタイマはすでにタイマ(カウンタ)として存在している。
●カ イ ロ エ ラ ー：入力回路に間違いがある。(例：入力がない、入力が多すぎる。)

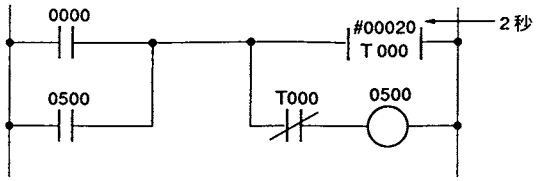
タイマ命令の応用例

ワンショット回路

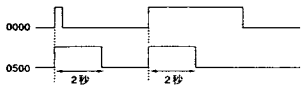
■解説

入力0000がONすると出力0500が一定時間ONします。

■ラダー図



■動作チャート



■コーディング

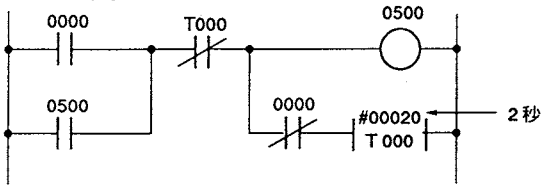
行番号	命　令	オペランド
0000	LD	0000
0001	OR	0500
0002	TMR	000#00020
0003	ANB	T000
0004	OUT	0500

オフディレイ回路

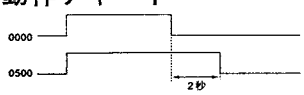
■解説

入力0000がONすると出力0500がONし、入力0000がOFFすると、一定時間遅れてOFFします。

■ラダー図



■動作チャート



■コーディング

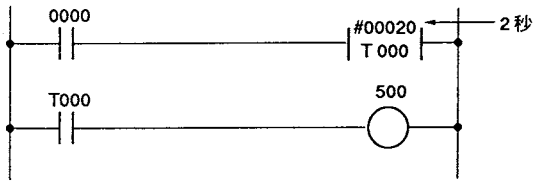
行番号	命　令	オペランド
0000	LD	0000
0001	OR	0500
0002	ANB	T000
0003	OUT	0500
0004	ANB	0000
0005	TMR	000#00020

オンディレイ回路

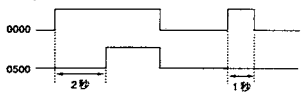
■解説

入力0000がONすると一定時間遅れて出力0500がONし、入力0000がOFFすると、出力0500もOFFします。

■ラダー図



■動作チャート



■コーディング

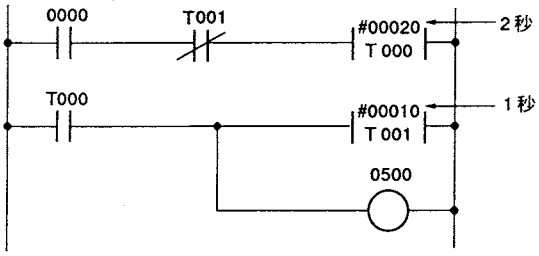
行番号	命　令	オペランド
0000	LD	0000
0001	TMR	000#00020
0002	LD	T000
0003	OUT	0500

フリッカ回路

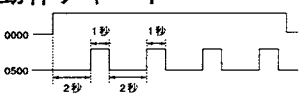
■解説

入力0000がONしている間、出力0500が一定間隔でON-OFFをくり返します。

■ラダー図



■動作チャート



■コーディング

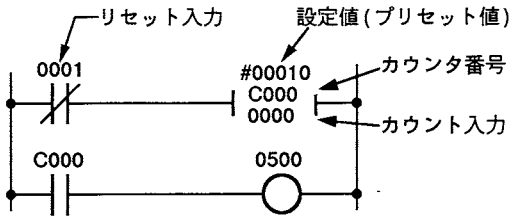
行番号	命　令	オペランド
0000	LD	0000
0001	ANB	T001
0002	TMR	000#00020
0003	LD	T000
0004	TMR	001#00010
0005	OUT	0500

カ ウ ン タ:C

CTR (C) : #d·dddd C xxx nnnn

16ビット加算カウンタの動作を行ないます。

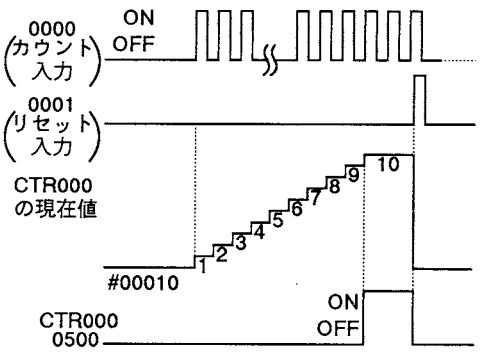
(例)



コーディング

行番号	命　令	オペランド
0000	LDB	0001
0001	C	000#00010 0000
0002	LD	C000
0003	OUT	0500

■動作チャート(上例C000＃00010と0500の動作)



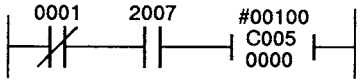
●カウンタはカウント入力の立上り(OFF→ON)時、1回カウントされ、現在値が設定値になったとき指定された番号のコイルが「ON」します。
●リセット入力(B接点)がONするとカウンタはリセットされ現在値は「00000」に戻ります。

■解説

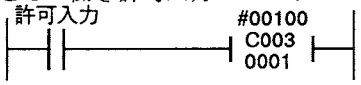
●Cで指定された番号を接点として使用する場合、使用回数に制限はありません。
(N.O. 接点、N.C. 接点とも)
●Cの現在値は運転停止時や電源OFF時にも2ヵ月以上保持されます。
●リセット入力をa接点で使用されますと、電源OFF時に現在値がリセットされます。リセットしたくない時はb接点で使用してください。
●カウンタ番号は、タイマ/高速タイマ/高速1msタイマ/アップダウンカウンタとの二重使用はできません。
●Cはソフト上のカウンタですので、応答速度はスキャンタイムに依存します。

ワンポイント

●カウンタのリセット入力に直列にa接で「2007」(運転開始時1スキャンのみOFF)のリレーを入力しておくと、毎運転開始時にカウンタをリセットすることができます。



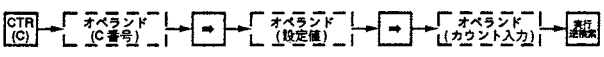
●リセット入力のb接をa接にするとリセット入力ではなく、許可入力として働き許可入力がONしている間のみカウント入力のパルスをカウントします。
許可入力がOFFするとカウンタはリセットされます。



オペランド

KZ-10、16	KZ-24、40、80
番号000～063	番号000～119
カウント入力	カウント入力
0000～2915	0000～6915
設定値	設定値
＃00000～＃65535	＃00000～＃65535

入力方法

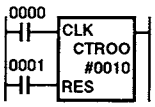
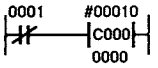


ERROR//

●オペランドエラー：その番号のカウンタはすでにカウンタ(タイマ)として存在している。
●カ イ ロ エ ラ ー：入力回路に間違いがある。(例：入力がない、入力が多すぎる。)

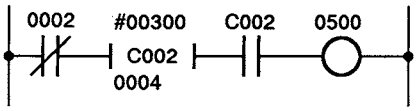
拡張ラダー

●カウンタの仕様が従来のものと異なるのは KZ の特徴の 1 つである拡張ラダーに対応できるようになっているからです。

変更点	従来のカウンタ	KZ のカウンタ
ラダーシンボル		
仕　様	ダウンカウンタ	アップカウンタ

●たとえば拡張ラダーを使用することによってカウンタの後に、接点や出力コイルをもって接続することができプログラムの行数を短くすることが可能です。

(例) カウント入力 0004 でカウントを行い現在値が 300 になると出力 500 を ON します。



拡張ラダーについては 4 章をご覧ください。

コーディング

行番号	命　令	オペランド
0000	LDB	0002
0001	C	002 #00300 0004
0002	CON	
0003	AND	C002
0004	OUT	0500

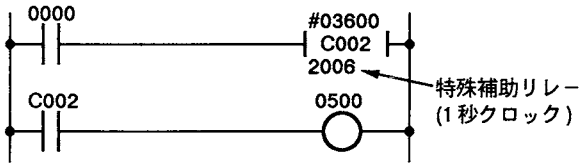
CON 命令については P5−6 をご覧ください。

長時間タイマ

タイマとカウンタを組合わせた長時間タイマ

■ 解説

1 時間後に出力 0500 を ON します。



コーディング

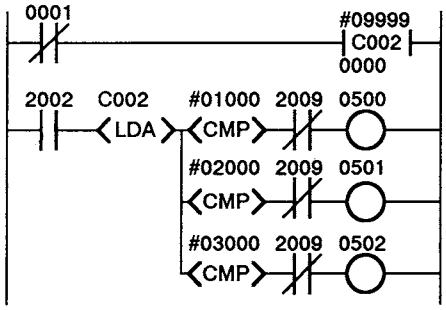
行番号	命　令	オペランド
0000	LD	0000
0001	C	002 #03600 2006
0002	LD	C002
0003	OUT	0500

特殊補助リレーについては P2−8 をご覧ください。

多段設定

■ 解説

演算命令を用いることによりカウンタの多段設定ができます。



●カウンタの現在値が 1000、2000、3000 以上になるとそれぞれ 500、501、502 が出力します。
演算命令については P5−29〜P5−33 をご覧ください。

コーディング

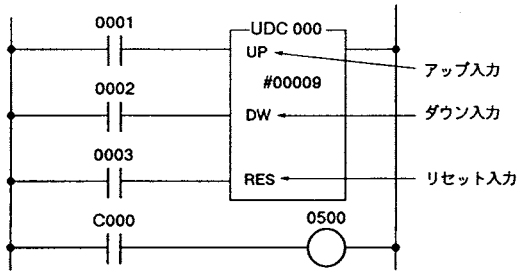
行番号	命　令	オペランド
0000	LDB	0001
0001	C	002 #9999 0000
0002	LD	2002
0003	LDA	C002
0004	CON	
0005	MPS	
0006	CMP	#01000
0007	CON	
0008	ANB	2009
0009	OUT	0500
0010	MRD	
0011	CMP	#02000
0012	CON	
0013	ANB	2009
0014	OUT	0501
0015	MPP	
0016	CMP	#03000
0017	CON	
0018	ANB	2009
0019	OUT	0502

アップダウンカウンタ：UDC

FUN　ANB 5　ANL 2　：　UDC xxx
UP　DW　RES　#ddddd

16ビット加減算カウンタの動作を行います。

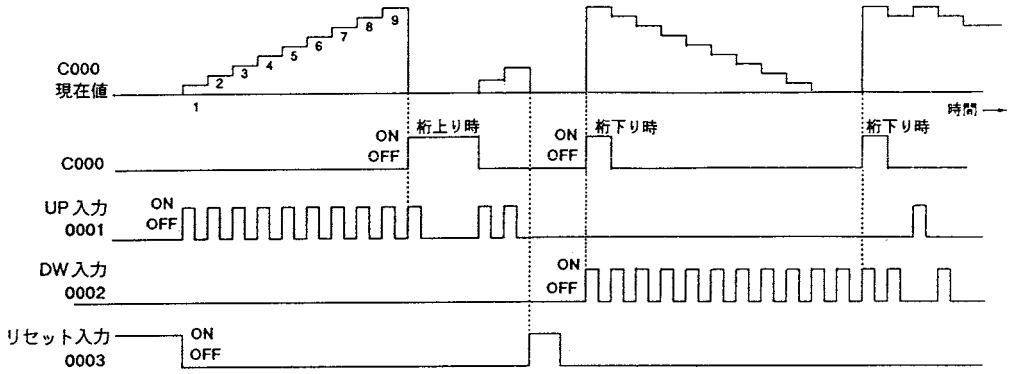
（例）



コーディング

行番号	命　令	オペランド
0000	LD	0001
0001	LD	0002
0002	LD	0003
0003	UDC	000＃00009
0004	LD	C000
0005	OUT	0500

■動作チャート



●UP 入力 (0001) が OFF → ON に変化すると、カウント値が +1 されます。DW 入力 (0002) が OFF → ON に変化すると、カウント値が −1 されます。
●UP、DW 入力が同時に変化したときは ±0 です。RES 入力 (0003) が ON すると現在値は 0 になります。
●桁上り時、桁下り時に C000 が ON します。
●接点の使用回数、電源 OFF 時の現在値保持については C (P3−22) に準じます。
●アップダウンカウンタの番号は、カウンタ/タイマ/高速タイマ/高速 1ms タイマとの二重使用はできません。

オペランド

KZ-10、16	KZ-24、40、80
番号000～063	番号000～119
設定値	設定値
＃00000～＃65535	＃00000～＃65535

入力方法

FUN → ANB 5 → ANL 2 → 実行逆検索 → オペランド(C番号) → ➡ → オペランド(設定値) → 実行逆検索

ERROR //

●オペランドエラー：その番号のカウンタはすでにカウンタ(タイマ)として存在している。
●カ イ ロ エ ラ ー：入力回路に間違いがある。(例：入力がない、入力が多すぎる。)

立上り微分：DIFU

FUN SET 1 RES 0 : nnnn DIFU

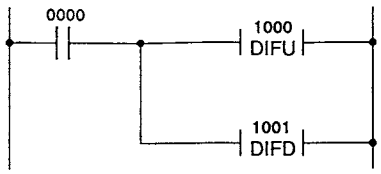
入力の立上りを検出し、指定されたリレーを1スキャンだけONします。

立下り微分：DIFD

FUN RES 0 OR 9 : nnnn DIFD

入力の立下りを検出し、指定されたリレーを1スキャンだけONします。

(例)

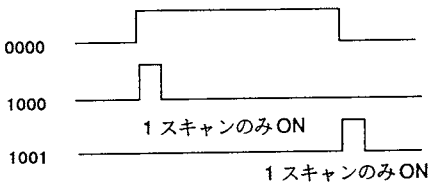


コーディング

行番号	命　令	オペランド
0000	LD	0000
0001	DIFU	1000
0002	DIFD	1001

■動作チャート(例の0000、1000、1001の動作)

(例)



- DIFU命令は、入力信号の立上がり時、1スキャンだけオペランドのリレーをONさせます。
- DIFD命令は入力信号の立下がり時、1スキャンだけオペランドのリレーをONさせます。
- オペランドの重複使用はできません。

オペランド

KZ-10、16	KZ-24、40、80
1000～1915	1000～1915 3000～6915

入力方法

FUN → SET 1 → RES 0 → 実行逆検索 → オペランド(リレー番号) → 実行逆検索 : FUN → RES 0 → OR 9 → 実行逆検索 → オペランド(リレー番号) → 実行逆検索

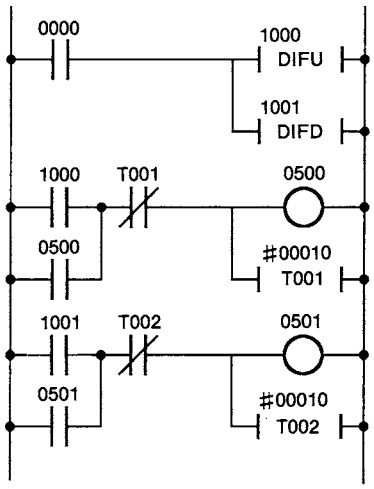
ERROR!!

- カイロエラー：入力回路に間違いがある。(例：入力がない、入力が多すぎる。)

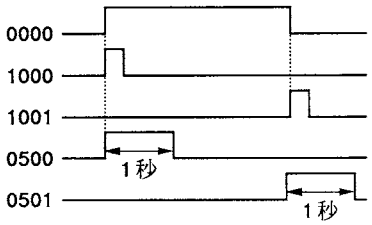
微分命令を用いたワンショット回路

■解説

入力 0000 が ON、OFF するとこにより 0500、0501 がワンショット (1 秒) で出力します。



■動作チャート



コーディング

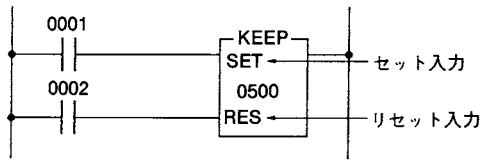
行番号	命　令	オペランド
0000	LD	0000
0001	DIFU	1000
0002	DIFD	1001
0003	LD	1000
0004	OR	0500
0005	ANB	T001
0006	OUT	0500
0007	TMR	001 #00010
0008	LD	1001
0009	OR	0501
0010	ANB	T002
0011	OUT	0501
0012	TMR	002 #00010

キープ:KEEP

FUN ANL 2 ANL 2 : KEEP SET RES nnnn

S(セット)入力ON時に指定されたリレーをONして保持し、R(リセット)入力ON時に指定されたリレーをOFFします。

(例)



コーディング

行番号	命 令	オペランド
0000	LD	0001
0001	LD	0002
0002	KEEP	0500

■解説(例の動作)

- SET入力(0001)がONでRES入力(0002)がOFFのとき0500はONします。
- SET入力(0001)がOFFでRES入力(0002)がONのとき0500はOFFします。
- SET入力(0001)とRES入力(0002)が共にOFFのとき状態保持します。
- SET入力(0001)、RES入力(0002)が共にONのときRES入力(0002)が優先され、0500はOFFします。

SET・RES 命令との使いわけ方 → P3−18

オペランド

KZ-10、16	KZ-24、40、80
0500～1915	0500～1915
2009	2009
2100～2915	2100～6915

入力方法

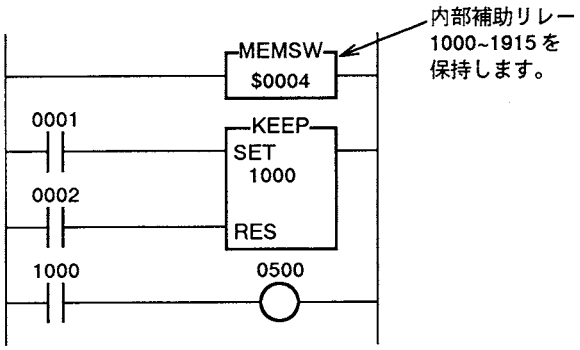
FUN → ANL 2 → ANL 2 → 実行逆検索 → オペランド(リレー番号) → 実行逆検索

ERROR!!

- カイロエラー:S、R、入力回路に間違いがある。(例:入力がない)

内部補助リレーの保持機能

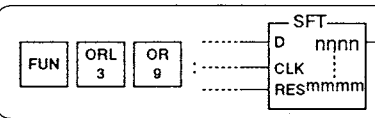
メモリースイッチ(MEMSW、☞P3−31)の設定により内部補助リレーを電源OFF時も直前のON-OFF状態を記憶することができます。
この補助リレーを保持命令など(SET、KEEP)で使用することにより、一旦セットされたリレーを電源OFF時も保持させることができます。



コーディング

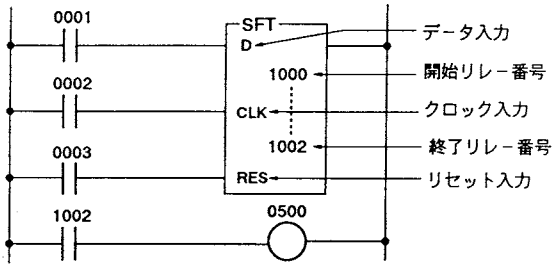
行番号	命 令	オペランド
0000	MEMSW	$ 0004
0001	LD	0001
0002	LD	0002
0003	KEEP	1000
0004	LD	1000
0005	OUT	0500

シ　フ　ト:SFT



シフトレジスタの動作を行います。

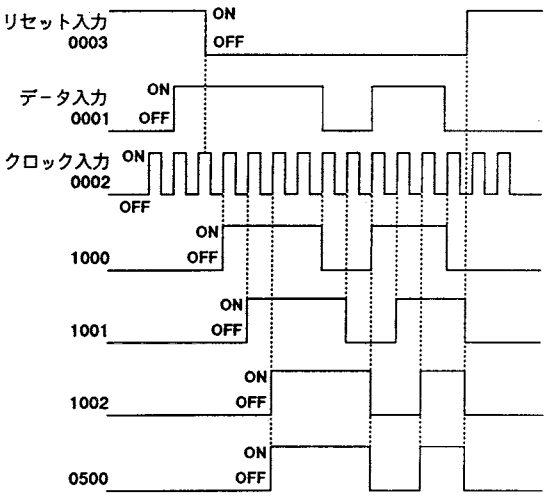
(例)



コーディング

行番号	命　令	オペランド
0000	LD	0001
0001	LD	0002
0002	LD	0003
0003	SFT	1000 1002
0004	LD	1002
0005	OUT	0500

■動作チャート(例の動作)



- 開始リレーはクロック入力の立上り時のデータ入力がONであればON、OFFであればOFFとなります。
- 開始リレー番号+1から終了リレー番号までのリレーは、クロック入力の立上り時に1段階前のリレー番号の内容が1段ずつシフトされます。
- リセット入力がONすると開始リレー番号より終了リレー番号までのすべてのリレーがOFFとなります。
- 終了リレー番号の内容は次のクロック入力の立上り時に消滅します。

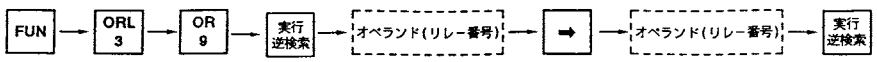
■解説

- 開始リレー番号≦終了リレー番号となるように設定して下さい。
  開始リレー番号>終了リレー番号となるような設定はできません。
- 開始リレーと終了リレーが1915から3000にまたがるような設定はできません。(1000→1915への160段、3000→6915への640段の設定は可能です。)

オペランド

KZ-10、16	KZ-24、40、80
1000～1915	1000～1915
	3000～6915

入力方法



ERROR!!

- カ イ ロ エ ラ ー：入力回路に間違いがある。(例：入力が少ない)

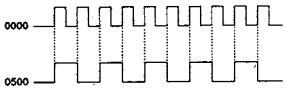
シフト命令の応用例

オルタネータ回路

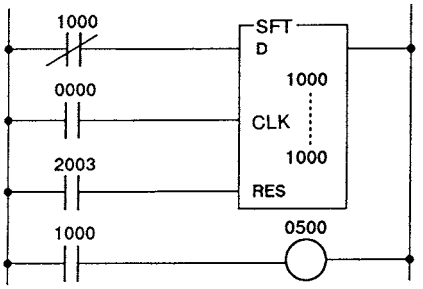
■解説

入力0000がON-OFFするごとに出力0500が交互にON-OFFします。

■動作チャート



■ラダー図

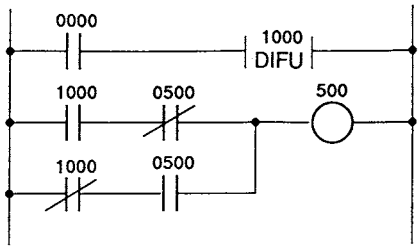


■コーディング

行番号	命　令	オペランド
0000	LDB	1000
0001	LD	0000
0002	LD	2003
0003	SFT	1000 1000
0004	LD	1000
0005	OUT	0500

参　考

オルタネータ回路はSFT命令を使用しなくても作成できます。
次の回路は上記と同じ動作をします。



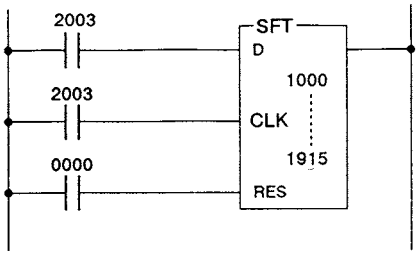
行番号	命　令	オペランド
0000	LD	0000
0001	DIFU	1000
0002	LD	1000
0003	ANB	0500
0004	LDB	1000
0005	AND	0500
0006	ORL	
0007	OUT	0500

内部補助リレーのリセット

■解説

入力0000がONすると1000から1915の全ての内部補助リレーがOFFします。
RES命令の代用として使用できます。(3000から6915を全てOFFすることもできます。)

■ラダー図



■コーディング

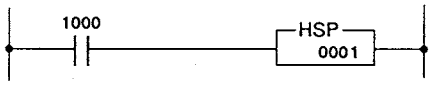
行番号	命　令	オペランド
0000	LD	2003
0001	LD	2003
0002	LD	0000
0003	SFT	1000 1915

ハイスピード：ＨＳＰ

FUN　SET 1　AND 8　：……HSP nnnn

指定された入力リレーの時定数を小さくし(25μs)、入力応答性を速くします。

(例)



コーディング

行番号	命　令	オペランド
0000	LD	1000
0001	HSP	0001

●1000が「ON」のとき、入力リレー0001の時定数は「25μs」となります。

■解説

●HSP 命令は入力リレーの時定数を小さくします。
上記の例では、入力リレー 0001 の時定数は次のようになります。

> ●1000 が ＯＮ … 時定数は高速 → 25μs
> ●1000 が OFF … 時定数は通常 → 8～12ms

●いくらスキャンタイムが短くても、入力時定数が通常 (8～12ms) だと、それ以下の幅の信号は入力することができません。
入力信号の幅が 12ms 以下の場合は HSP 命令を使用します。
例えば、
スキャンタイム＜入力信号幅＜入力時定数
の時に使用します。
●HSP 命令の入力 (例では 1000) は、プログラム上で ON-OFF させる必要がない場合、2002 (常時 ON リレー) を使用します。
●HSP 命令は基本ユニットの入力リレーでのみ使用できます。
●HSP 命令で入力時定数を小さくした場合、接点バランスを入力する時がありますので有接点入力機器 (リレー、スイッチ) を接続する場合はご注意ください。

オペランド

(KZ-10)	(KZ-24)
0000～0005	0000～0015
(KZ-16)	(KZ-40、80)
0000～0009	0000～0107

入力方法

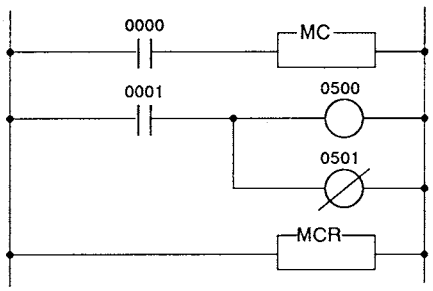
FUN → SET 1 → AND 8 → 実行逆検索 → オペランド(リレー番号) → 実行逆検索

ERROR!!

●カ イ ロ エ ラ ー：入力回路に間違いがある。(例：入力がない)

マスター:M　C コントロール	FUN ANL 2 LDB 4 : ---MC---	MCRまでのリレーコイル・タイマ・カウンタなどのON-OFF状態を決定します。
マスター:MCR コントロールリセット	FUN ANL 2 ANB 5 : ---MCR---	MCと対で使用し、MC命令の終了を示します。

(例)



コーディング

行番号	命　令	オペランド
0000	LD	0000
0001	MC	
0002	LD	0001
0003	OUT	0500
0004	OUB	0501
0005	MCR	

0000が「OFF」のとき、0001を「ON」しても0500、0501ともに「ON」しません。

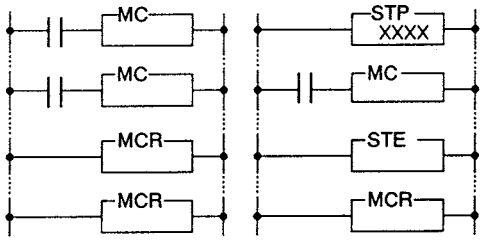
■解説

●MC-MCR命令は〝電源スイッチ〟〝メインスイッチ〟のイメージです。

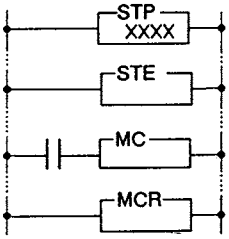
●MCの入力がOFFのとき、MC-MCR間の各命令は下表のようになります。

命　令　語	機　能
OUT・OUB命令でのリレー	OFF
TMR・TMH・TMS命令	リセット
カウンタ、シフトレジスタ、キープ命令、セット、リセット命令、高速カウンタ	状態保持
TMINなど、上記以外の命令	命令を実行しません。

●MC-MCRは次のような使用法では使えません。
①「入れ子」状態　②STP-STEとの組合せ



●STP-STEとの組合せは次のような方法であれば使えます。



オペランド

―――

入力方法

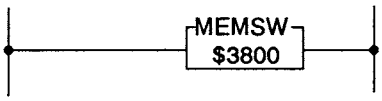
FUN → ANL 2 → LDB 4 → 実行 逆検索 → 実行 逆検索 : FUN → ANL 2 → ANB 5 → 実行 逆検索 → 実行 逆検索

ERROR//

●MC-MCRエラー：MCとMCRが一対で使用されていない。入れ子状態になっている。
●カ イ ロ エ ラ ー：入力回路に間違いがある。

メモリスイッチ：MEMSW　FUN　ANL 2　ORB 6　：MEMSW $ hhhh　メモリスイッチの設定を行います。

(例)

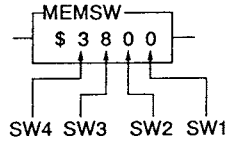


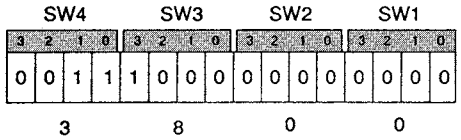
行番号	命　令	オペランド
0000	MEMSW	$ 3800

●SW3の3、SW4の0と1を設定し運転開始時にDMの値、カウンタ/アップダウンカウンタ、CTH、CTCの値をリセットします。

■解説

●メモリスイッチは、KZの動作を決めるソフトウェア上のスイッチです。
●スイッチの番号と内容は以下の通りで、内部は16ビットになっておりそのビットのON/OFFでスイッチの機能を設定します。





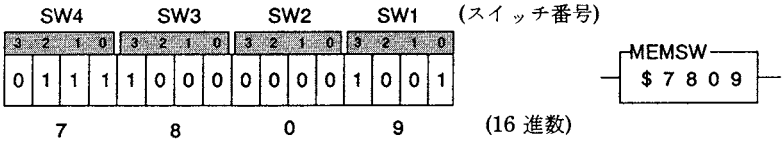
(スイッチ番号)
(1：ON
0：OFF)
(16進)

スイッチ番号		スイッチの機能	ON	OFF
SW1	0	電源ON時のエラー解除	エラーを保持します。	エラーを解除して運転します。
	1	データメモリやカウンタの値、保持機能の内容が消えた時の処理	メモリエラーを表示します。	メモリエラーを解除して運転します。
	2	内部補助リレー 1000～1915 の保持機能の選択	保持機能を使用します。	保持機能を使用しません。
	3	内部補助リレー 3000～3915 の保持機能の選択	保持機能を使用します。	保持機能を使用しません。
SW2	0	内部補助リレー 4000～4915 の保持機能の選択	保持機能を使用します。	保持機能を使用しません。
	1	内部補助リレー 5000～5915 の保持機能の選択	保持機能を使用します。	保持機能を使用しません。
	2	内部補助リレー 6000～6915 の保持機能の選択	保持機能を使用します。	保持機能を使用しません。
	3	未使用		常時OFFで使用します。
SW3	0	未使用		常時OFFで使用します。
	1	未使用		常時OFFで使用します。
	2	未使用		常時OFFで使用します。
	3	運転開始時DMをクリアします。	DMを0クリアします。	DMの0クリアをしません。
SW4	0	運転開始時カウンタ/アップダウンカウンタをクリアします。	カウンタ/アップダウンカウンタをクリアします。	カウンタ/アップダウンカウンタをクリアしません。
	1	運転開始時CTH・CTCのクリアをします。	CTH・CTCをクリアします。	CTH・CTCをクリアしません。
	2	本体プログラムのライトプロテクト。	ライトプロテクトをします。	ライトプロテクトをしません。
	3	本体プログラムのリードプロテクト。	リードプロテクトをします。	リードプロテクトをしません。

●メモリスイッチの指定はスイッチ番号ごとに 16 進数で設定します。

スイッチ番号 3	2	1	0	16進数	スイッチ番号 3	2	1	0	16進数
0	0	0	0	0	1	0	0	0	8
0	0	0	1	1	1	0	0	1	9
0	0	1	0	2	1	0	1	0	A
0	0	1	1	3	1	0	1	1	B
0	1	0	0	4	1	1	0	0	C
0	1	0	1	5	1	1	0	1	D
0	1	1	0	6	1	1	1	0	E
0	1	1	1	7	1	1	1	1	F

(例)



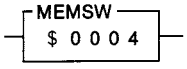
●メモリスイッチの内容は、プログラム転送時に設定されます。
●プログラム中にMEMSW命令を書かない場合は『$0000』が設定されます。
●ライトプロテクトをおこなった本体にプログラムの転送をおこなおうとすると、『カキコミホゴサレテイマス キョウコウシマスカ？』という警告が表示されます。強行する時は、[実行 逆検索] を押します。

注意

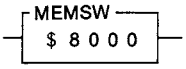
●リードプロテクトをおこなって転送したプログラムは、読み出したり、修正したりすることができません。リードプロテクトをおこなうときは、あらかじめプログラムのバックアップを取っておいてください。

ワンポイント

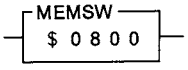
●内部補助リレー 1000～1915 を保持リレーとして使う時の設定



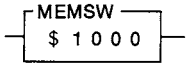
●プログラムのリードプロテクトをする時の設定



●運転開始時に DM をクリアする時の設定



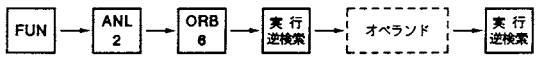
●運転開始時にカウンタをクリアする時の設定



オペランド

$0000～$FFFF

入力方法



ERROR//

ノ　ッ　プ:NOP	FUN ORL 3 RES 0 : ————	この命令を実行させても何もしません。

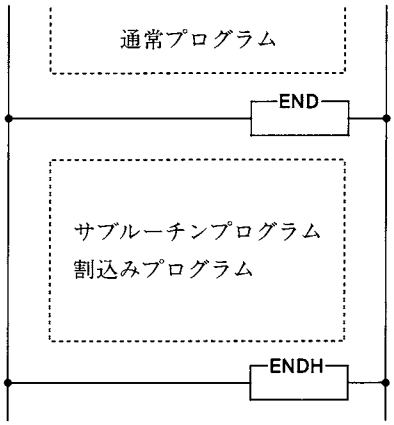
■解説

●オールクリア(FUN60)、プロコンクリア(FUN61)を実行し、プログラムを消去すると、消去された部分にNOPが入ります。

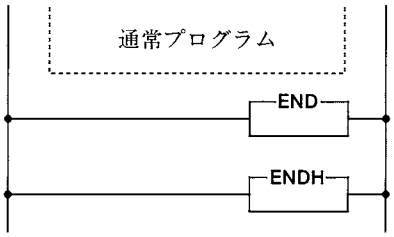
エ　ン　ド:END	END : —[END]—	通常プログラムの終了を示します。
エンドハイ:ENDH	ENDH : —[ENDH]—	全プログラムの終了を示します。

■解説

●ENDとENDH命令の間には、サブルーチンプログラムや、割込みプログラムを書きます。



●サブルーチンプログラムや、割込みプログラムを書かない場合は、END命令とENDH命令を続けて書きます。
どちらかの命令がない時には、「ENDHナシエラー」または「カイロエラー」が発生します。



オペランド	入力方法
————	END → 実行 逆検索 : ENDH → 実行 逆検索

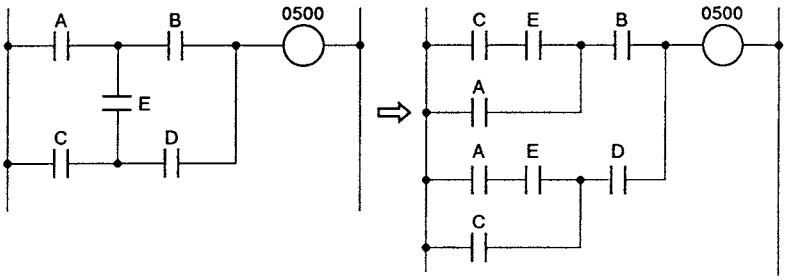
ERROR //

- ●カイロエラー　：未接続の接点(リレーコイル・タイマ・カウンタなどない接点)がある。
- ●ENDナシエラー　：END命令がない。
- ●ENDHナシエラー：ENDH命令がない。

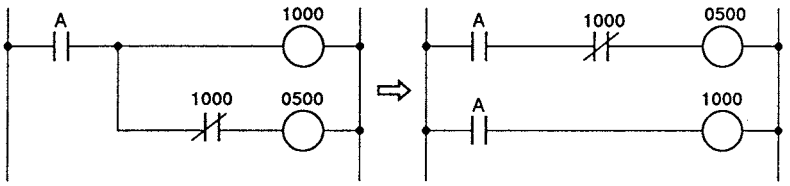
プログラム作成のポイント

■書き替えの必要な回路

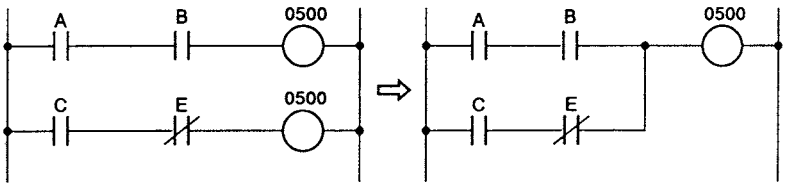
①右図のような回路はプログラムできませんので書き替えてください。



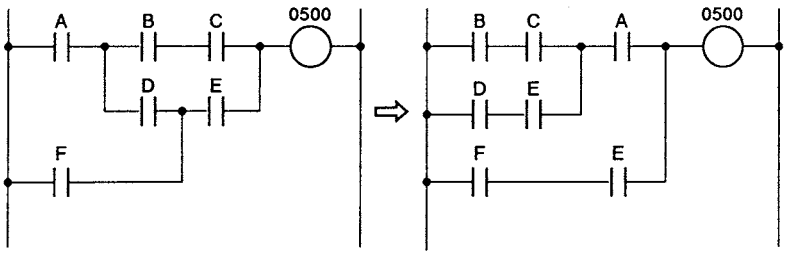
②右図の場合、0500は出力できません。書き替えにより１スキャンのみ出力できるようになります。



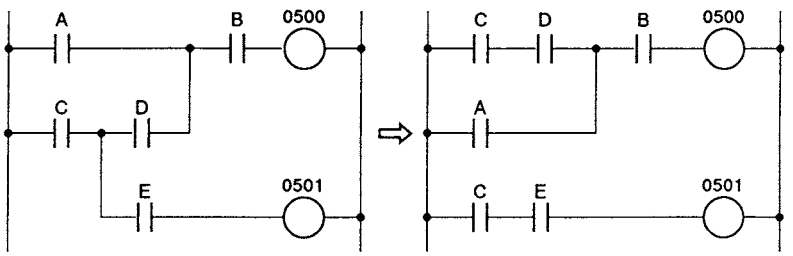
③右図のように同一コイルを２重に使用すると後ろ側のプログラムが優先され、前側は無視されます。



④右図のような回路はプログラムできませんので書き替えてください。

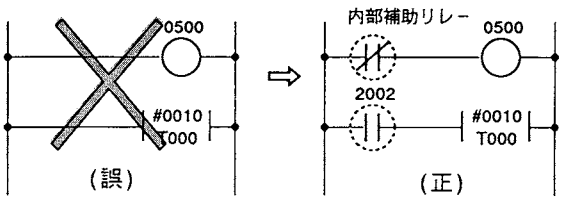


⑤右図のような回路はプログラムできませんので書き替えてください。

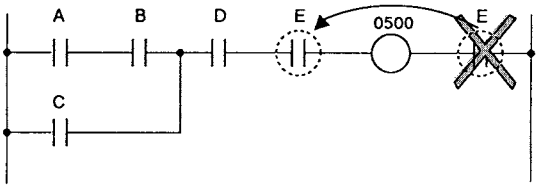


■プログラム作成時の注意点

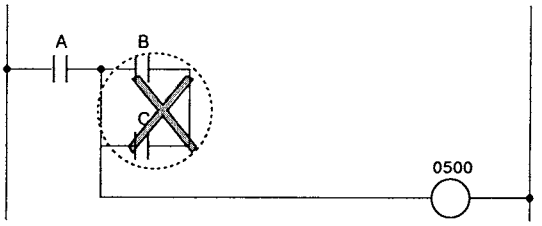
①母線から直接出力コイルやタイマ・カウンタなどを接続することはできません。
必要な場合は使用されていない内部補助リレーのb接点または特殊補助リレー2002(常時ON接点)をダミーとして挿入してください。



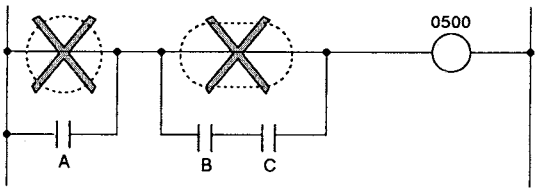
②出力線に接点を接続することはできません。



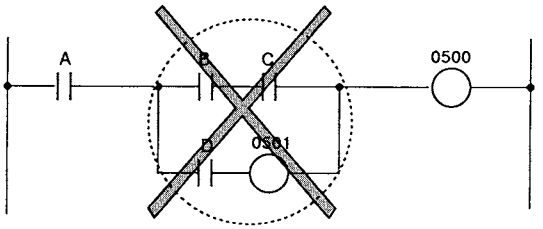
③全ての接点や出力コイルは、母線から出力線にいたる経路上にあるようにし、未接続がないようにしてください。



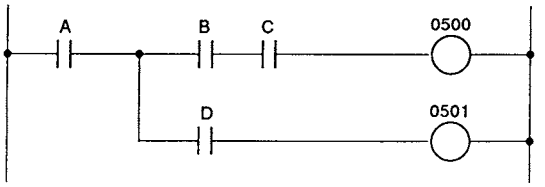
④並列接続時に不正な短絡をしないように接続してください。



⑤出力コイルを含む回路群は、並列接続することはできません。



但し右図のように並列接続の回路群が出力線にそれぞれ接続されていれば並列接続は可能です。



4章　拡張ラダー言語

4章では、KZシリーズで新しく使われているラダー言語
(拡張ラダー言語) について解説いたします。

拡張ラダー言語の特長

拡張ラダー言語とは

聞き慣れない言葉と思いますが、拡張ラダーとはPCプログラムの設計、試験、運転、保守に関わる工数を削減する目的で開発されたプログラム方式で、プログラムの流れが格段につかみやすくなっています。

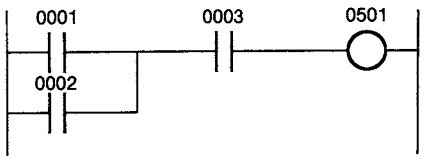
拡張ラダー開発の背景

現在FAの業界で最も幅広く用いられているプログラムはみなさんよくご存じのラダープログラムです。
ラダープログラムとは従来のリレーシーケンスの流れを組むものでメリットとして以下の点が上げられます。

従来ラダーのメリット

① リレーシーケンスをそのまま置き換え可能。
② AND、ORの処理（論理的な処理）に適している。

（論理処理例）



ということで設計から保守まで現在もっとも幅広く用いられています。
しかし一般に普及した現時点ではメリットと同時に以下のような問題点が指摘されています。

従来ラダーの問題点

① 論理的な処理には適しているが状態を順次変化させていくような処理（順次処理）には不向き

（順次処理例）

開始スイッチを入れると3秒間ランプを点灯し、さらに2秒間ブザーを鳴らした後モータを回す。

② 設計者以外には理解しにくく保守がしにくい。
③ 中継が多くなり、プログラムが縦に長く複雑になる。

そこでこれらの問題点を解決する目的で拡張ラダーが開発されました。

拡張ラダー言語の特長

拡張ラダーの特長

拡張ラダーとは従来のラダーの優れた点を残しながら、問題点となるところを改善できるように新しく以下のような特長を追加しています。

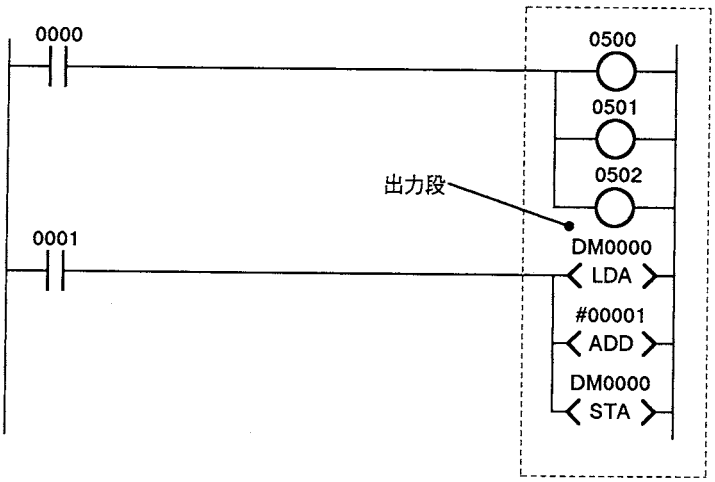
拡張ラダーのメリット 1

出力命令を横に並べて書くことができプログラムの行数を少なく記述できる(ラダー図で 1/3 程度)

●従来のラダー図では出力命令は右端の出力段にしか書けませんでした。しかし KZ のラダー図では左端の入力段以外であればどこでも自由に書くことができ、下図のように出力をならべて書くことによりラダー図の行数を短くすることができます。

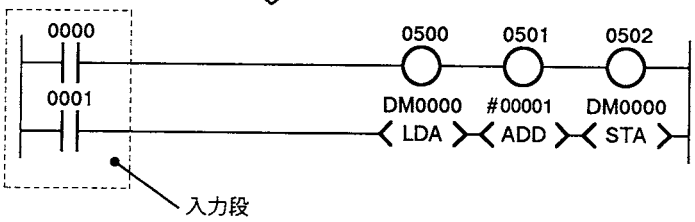
従来のラダー図と KZ のラダー図の参考例

従来のラダー図



同じ動作をします。

KZ のラダー図



拡張ラダーのメリット 2

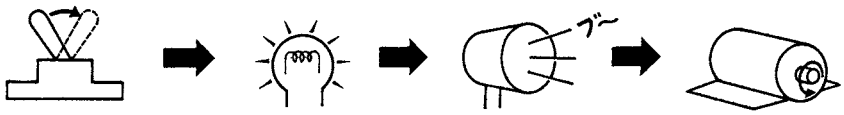
制御の流れにそってシンボルを書くことができ順次処理のプログラムがしやすい。

●拡張ラダーではラダーシンボルを処理の順番に並べてプログラムすることができます。
したがって、従来のラダーに比べてプログラムの流れがわかりやすく、メンテナンス性も大幅に向上します。

この特長について以下の例を参考に説明します。

例題 1：

開始スイッチを入れると 3 秒間ランプを点灯、次に 2 秒間ブザーを鳴らした後にモーターを回します。スイッチを OFF することによりすべてを OFF します。



トグルスィッチ (0000)　ランプ (0500)　ブザー (0501)　モーター (0502)

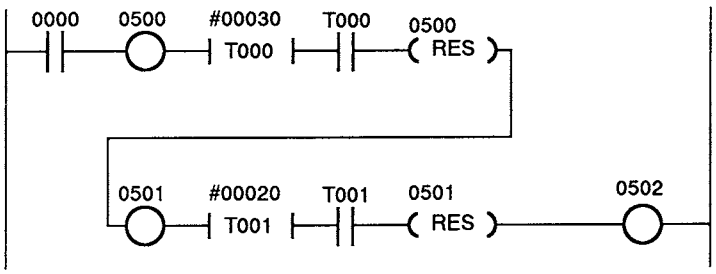


図 1　拡張ラダー図

① スタートスイッチ (0000) を ON するとランプ (0500) が点燈しタイマ 0 が動き始めます。
② タイマ 0 が 3 秒後に ON するとランプ (0500) を消してブザー (0501) を鳴らし、タイマ 1 が働き始めます。
③ タイマ 1 が 2 秒後に ON するとブザー (0501) を止めてモーター (0502) を ON にし回します。
④ スタートスイッチ (0000) を OFF にするとモーター (0502) は止まりすべてリセットします。

上記のように拡張ラダーでは行いたい処理の順番にプログラムを行っていくことができ、設計担当者以外の方が見てもプログラムの流れがわかりやすいようになっています。

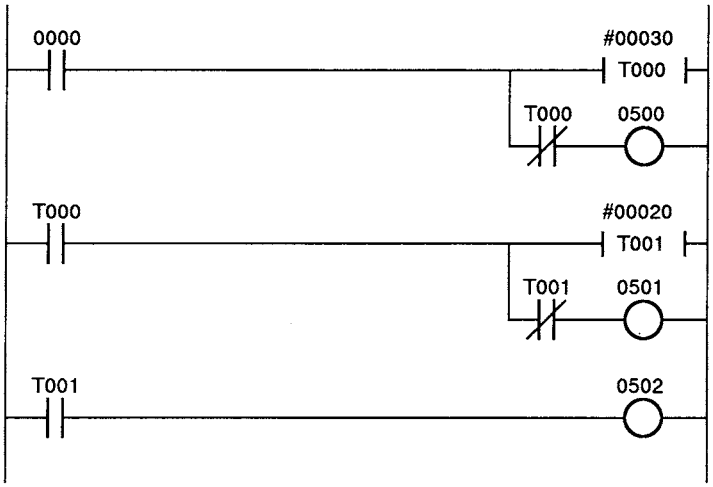


図２　従来ラダー図

上記は従来のラダー図でプログラムしており図１と同じ動きをします。
図１と比べてみると中継が多く１目では処理の流れがわかりにくくなっています。

拡張ラダーのメリット３　従来のラダープログラムと互換性がある

●いくらプログラムの仕様が優れていても、全く新しい概念を勉強しなければ使えないのでは、これまでの知識がむだになる等デメリットが大きすぎます。
拡張ラダーでは今まで述べてきた特長を持っているにもかかわらず、従来のラダーと互換性があります。
新しい機能を使用しなければ従来のラダープログラムの表現や機能をそのままでプログラムできます。
またプログラム中で新しい機能（順次処理）と従来のラダー（論理処理）の表現を混在させて使用することもできます。

新しい機能と変更点

新しい命令

拡張ラダーでは従来のラダーになかった出力命令を並べて書くことができる機能をさらに生かすために新しく加わった命令があります。
(命令の詳細については 5 章をご覧ください)

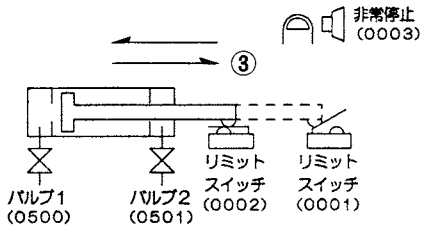
ウエイト命令　　W-ON、W-OFF、W-UE、W-DE

ウエイト命令はキーエンス独自の便利な命令で、プログラム中で前の作業が終了した条件を持って次の作業に移るときに使用します。
ウエイト命令をもちいることにより拡張ラダーをもちいた順次処理が大幅にプログラムしやすく、メンテナンスもやりやすくなります。

この命令をもちいた例をご紹介します。

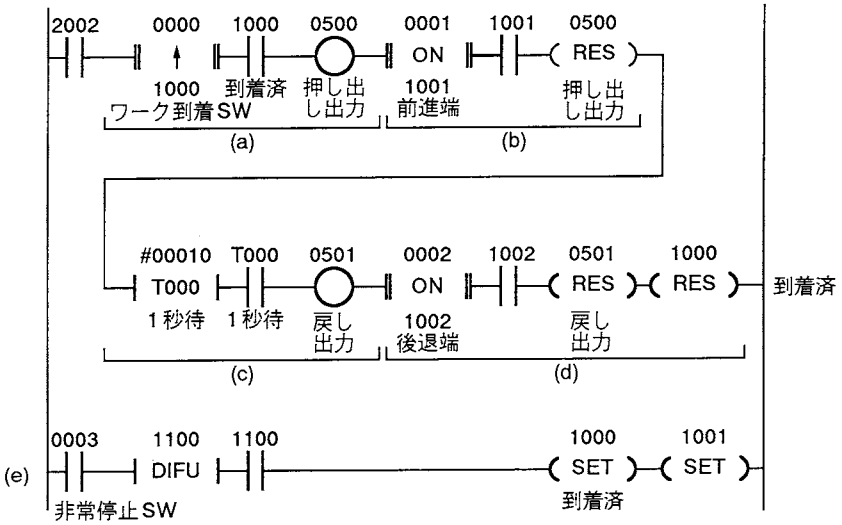
例題 2：

ワークが到着したら、シリンダでワークを押し出します。
押し出す手順は、
① 前進端に達するまで、押し出し出力を ON します。
② 前進端で 1 秒間待ちます。
③ 後退端に達するまで、戻し出力を ON します。
で行います。
また、非常停止スイッチが押されたら、シリンダを後退端まで戻して停止します。



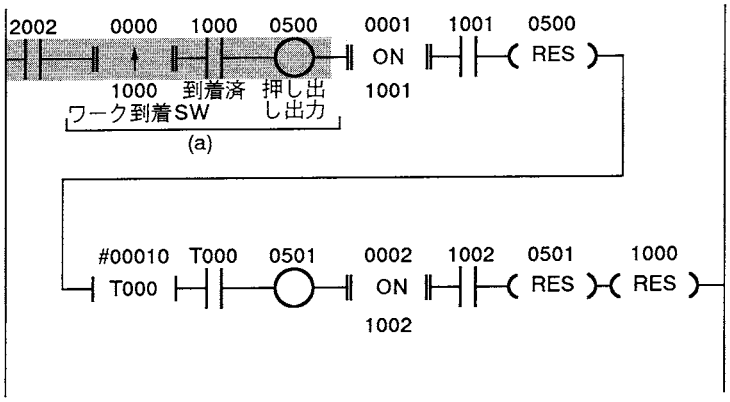
リレーの割付け

0000：ワーク到着スイッチ
0001：前進端リミットスイッチ
0002：後退端リミットスイッチ
0003：非常停止スイッチ
0500：押し出し出力
0501：戻し出力

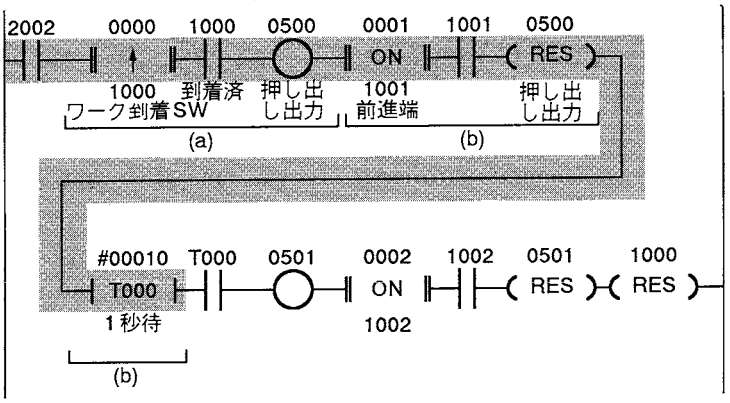


(a)：ワーク到着後、押し出し出力を ON させます。
(b)：前進端に達したら押し出し出力を OFF します。
(c)：前進端で 1 秒間待ち、1 秒後に戻し出力を ON します。
(d)：後退端に達したら、戻し出力を OFF します。
RES 1000 で次のワークを受けつけます。
(e)：非常停止SWが押されると、(b) から動作を行います。

●プログラムの動作は下図のように分けて考えれば理解しやすいでしょう。



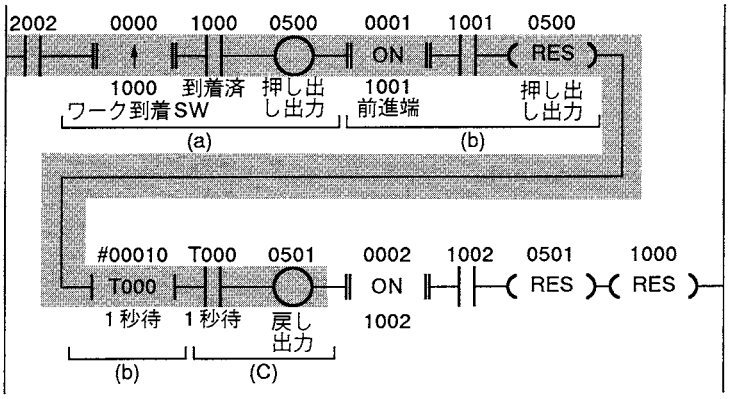
(a)：ワーク到着スイッチ 0000 が ON すると (a) の部分が ON します。
従って押し出し出力 0500 が ON します。



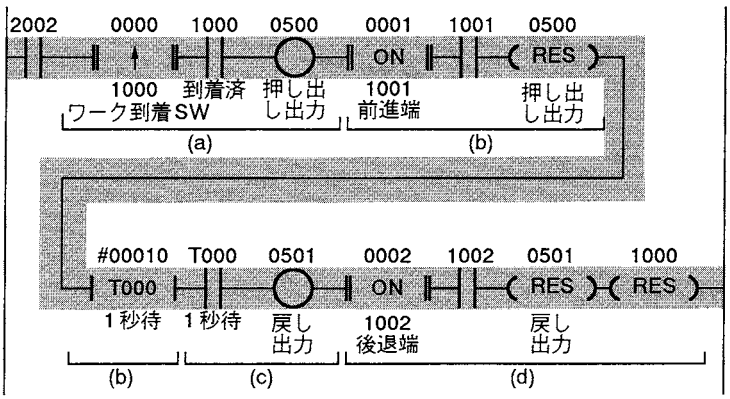
(b)：前進端リミットスイッチ 0001 が ON すると (b) の部分が ON します。
従って押し出し出力 0500 がリセットされ、タイマ 000 が働きます。

(注)：

と が同時に実行されると後が優先となるので 0500 はリセットされます。



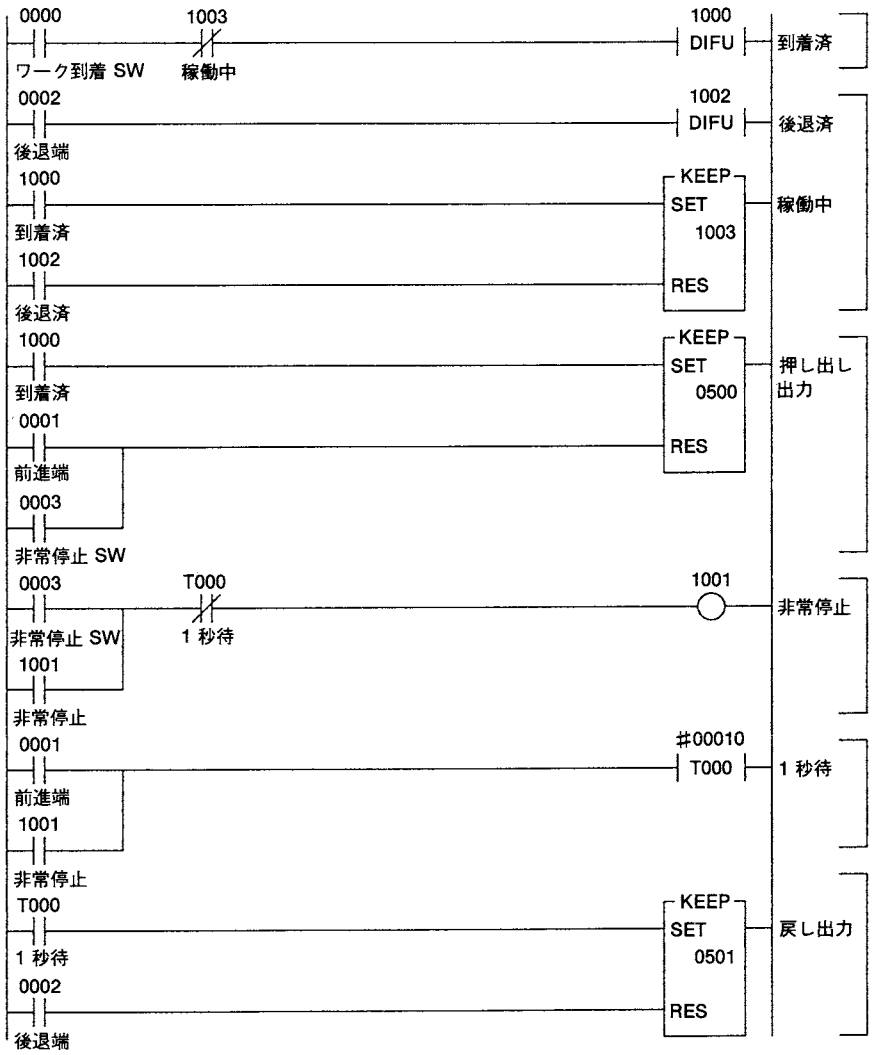
(c)：1 秒経過して T000 が ON すると (C) の部分が ON します。
従って戻し出力 0501 が ON します。



(d)：後退端リミットスイッチ 0002 が ON すると (d) の部分が ON します。
従って戻し出力 0501 がリセットされます。同時にウエイトアップエッジの 1000 もリセットされるので (a) (b) (c) (d) すべて OFF し初期状態に戻り、次のワークの到着を待ちます。

●この動作を従来のラダー図でプログラムすると以下のようになります。

従来のラダー図



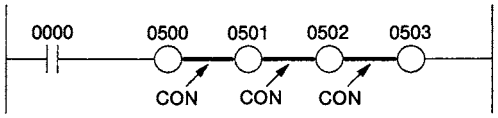
従来のラダー図と KZ のラダー図の比較

	従来のラダー図	KZ のラダー図
行　　数	13	4
処理内容の理解	理解しにくい	理解しやすい

この例でもわかるようにウエイト命令をもちいることで KZ の拡張ラダー図では大幅に行数が短縮されています。
またプログラムの内容も制御の流れにそって記述できますので従来のラダー図よりも格段に理解しやすくなります。

コネクト命令　CON

コネクト命令 (CON) は、出力命令の後に直列に他の命令を接続する場合用います。
この命令は、接続状態を表すだけで、直接動作には影響を与えません。



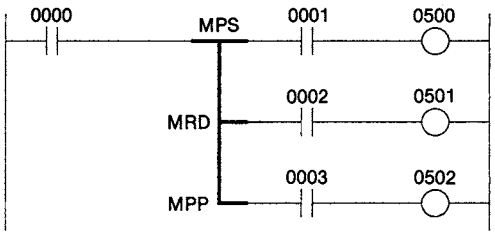
●入力リレー 0000 が「ON」のとき
0500、0501、0502、0503 が「ON」します。
●入力リレー 0000 が「OFF」のとき、
0500、0501、0502、0503 が「OFF」します。

コーディング

行番号	命令	オペランド
0000	LD	0000
0001	OUT	0500
0002	**CON**	
0003	OUT	0501
0004	**CON**	
0005	OUT	0502
0006	**CON**	
0007	OUT	0503

プッシュ、リード、ポップ命令　MPS、MRD、MPP

プッシュ (MPS)、リード (MRD)、ポップ (MPP) 命令は入力状態の記憶と読み出し、クリアをする命令です。



●入力リレー 0000 が「ON」のとき、
0001 が「ON」→ 0500 が「ON」します。
0002 が「ON」→ 0501 が「ON」します。
0003 が「ON」→ 0502 が「ON」します。

コーディング

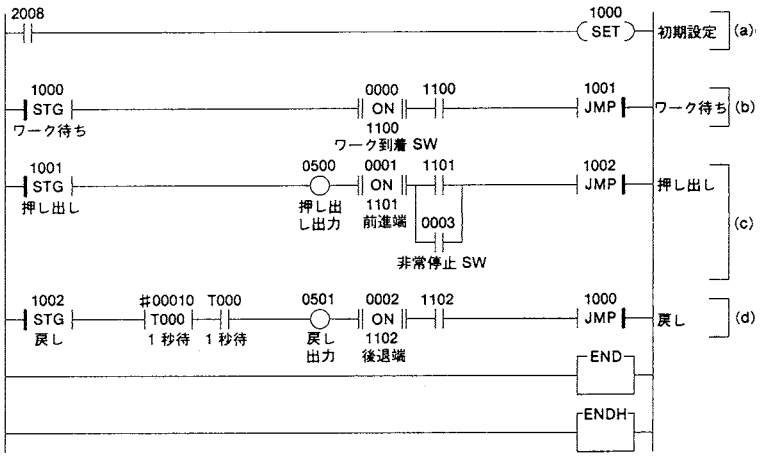
行番号	命令	オペランド
0000	LD	0000
0001	**MPS**	
0002	AND	0001
0003	OUT	0500
0004	**MRD**	
0005	AND	0002
0006	OUT	0501
0007	**MPP**	
0008	AND	0003
0009	OUT	0502

これらの各命令はニモニックでプログラムをするときには必要ですが、ラダー図でプログラムをするときには接続線のみを記入するだけでプログラムできます。

ステージ命令　STG、JMP、ENDS

ステージ命令はいわゆる「ステージ式」に近いプログラムが可能で大規模で複雑なプログラムをするときに使用することで威力を発揮します。
各動作をひとつのブロック(ステージ)としてまとめ、ブロック(ステージ)のONしている時に、そのブロック(ステージ)に書かれている動作を行いOFFしている時は動作を行いません(スキャンタイムにも影響しません)

●例題2をステージ命令を用いてプログラムすると以下のようになります。

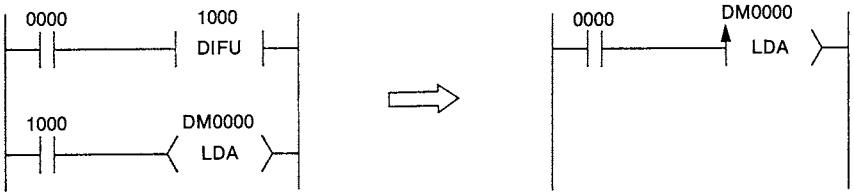


ステージ命令のプログラム例

(a)●電源投入時、ステージの1000を実行する様に指定します。
(b)●ワークの到着確認
(c)●ワークの到着確認後、押し出し出力をONします。
●前進端に達したら、押し出し出力をOFFにしてステージ1002を実行します。
●非常停止SWを押すと、ステージ1002を実行します。
(d)●前進端で1秒間待ち、1秒後に戻し出力をONします。
●後退端へ達すると、戻し出力をOFFし、ステージ1000を実行します。

微分型命令　LDA、STA、CMP、ADD、SUB、MUL、DIV、ANDA、ORA、EORA、SRA、SLA、RRA、RLA、COM、INC、DEC、MPX、DMX、TBCD、TBIN、ASC、RASC、ROOT

微分型命令は、入力リレーがOFFからONになった立ち上がりのときだけ、その命令を実行します。
従来のラダー図に比較すると行数を短く、わかりやすく記述できます。



従来のラダー図　　KZのラダー図

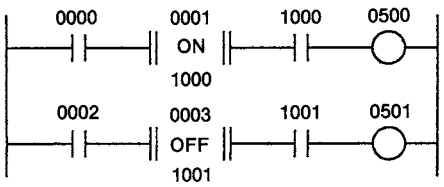
5章　応用命令・演算命令

5章ではKZシリーズで使用できる応用命令および
演算命令について解説いたします。

5-1．応用命令

ウエイトオン：W−ON	W-ON F : nnnn ON mmmm	第1オペランドのリレー接点のON状態で第2オペランドのリレーをセットします。
ウエイトオフ：W−OFF	W-OFF L: : nnnn OFF mmmm	第1オペランドのリレー接点のOFF状態で第2オペランドのリレーをセットします。

（例）

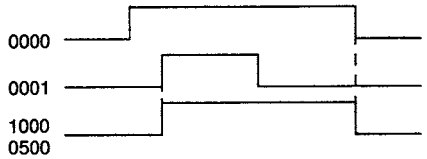


コーディング

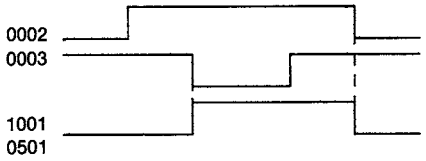
行番号	命令	オペランド
0000	LD	0000
0001	W-ON	0001 1000
0002	CON	
0003	AND	1000
0004	OUT	0500
0005	LD	0002
0006	W-OFF	0003 1001
0007	CON	
0008	AND	1001
0009	OUT	0501

タイムチャート

〔W−ON〕



〔W−OFF〕

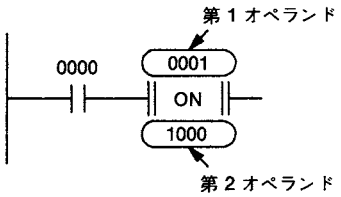


- 入力リレー0000が「ON」のときに、0001が「ON」すると、出力リレー0500が「ON」し、0001が「OFF」しても0500は「ON」したままです。
- 入力リレー0000が「OFF」すると、出力リレー0500も「OFF」します。

- 入力リレー0002が「ON」のときに、0003が「OFF」すると、出力リレー0501が「ON」し、0003が「ON」しても0501は「ON」のままです。
- 入力リレー0002が「OFF」すると、出力リレー0501も「OFF」します。

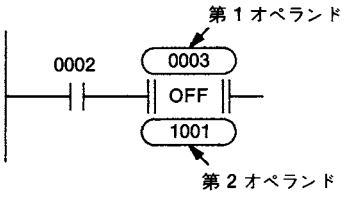
■解説

〔W−ON〕



W−ON 命令は、入力0000が「ON」で、第1オペランド0001の接点が「ON」のときに第2オペランド1000のリレーを「ON」にします。
入力0000が「OFF」のときは、第2オペランドのリレーを「OFF」にします。

〔W−OFF〕



W-OFF 命令は、入力0002が「ON」で、第1オペランド0003の接点が「OFF」のときに、第2オペランド1001のリレーを「ON」にします。
入力0002が「OFF」のときは、第2オペランドのリレーを「OFF」にします。

オペランド

KZ−10, 16	KZ−24, 40, 80
nnnn： 0000～2915 T000～T063 C000～C063 CTC0～CTC3 mmmm： 1000～1915	nnnn： 0000～6915 T000～T119 C000～C119 CTC0～CTC3 mmmm： 1000～1915 3000～6915

入力方法

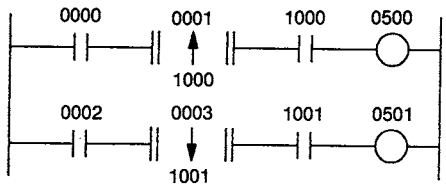


ERROR //

- カイロエラー：入力回路に間違いがある。（例：対応する入力がない）

ウエイト アップエッジ：W-UE	W-UE E : nnnn ↑ mmmm	第1オペランドのリレー接点の立ち上がりで第2オペランドのリレーをセットします。
ウエイト ダウンエッジ：W-DE	W-DE Lo : nnnn ↓ mmmm	第1オペランドのリレー接点の立ち下がりで第2オペランドのリレーをセットします。

（例）

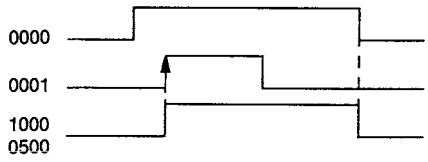


コーディング

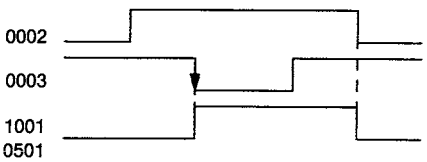
行番号	命令	オペランド
0000	LD	0000
0001	W-UE	0001 1000
0002	CON	
0003	AND	1000
0004	OUT	0500
0005	LD	0002
0006	W-DE	0003 1001
0007	CON	
0008	AND	1001
0009	OUT	0501

タイムチャート

〔W-UE〕



〔W-DE〕

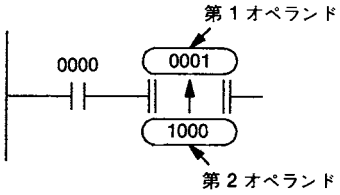


- 入力リレー0000が「ON」のとき、0001の立ち上がりで、出力リレー0500を「ON」します。
- 入力リレー0000が「OFF」すると、出力リレー0500も「OFF」します。

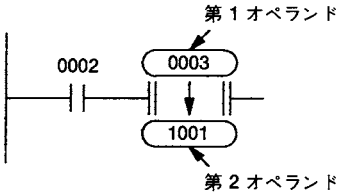
- 入力リレー0000が「ON」のとき、0003の立ち下がりで、出力リレー0501を「ON」します。
- 入力リレー0002が「OFF」すると、出力リレー0500も「OFF」します。

■解説

〔W-UE〕



〔W-DE〕



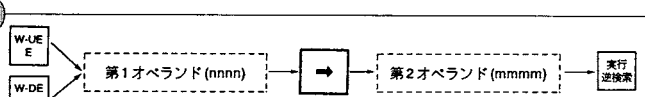
- W-UE 命令は入力0000が「ON」のとき、第１オペランドのリレーの接点の立ち上がりで、第２オペランドのリレーを「ON」にします。
  入力0000が「OFF」のとき、第２オペランドのリレーを「OFF」にします。
- サブルーチンプログラムでは毎スキャン必ず実行される場所で使用してください。

- W-DE 命令は入力0002が「ON」のとき、第１オペランドのリレー接点の立ち下がりで、第２オペランドのリレーを「ON」にします。
  入力0002が「OFF」のとき、第２オペランドのリレーを「OFF」にします。
- サブルーチンプログラムでは、毎スキャン必ず実行される場所で使用してください。

オペランド

KZ−10, 16	KZ24, 40, 80
nnnn： 0000～2915 T000～T063 C000～C063 CTC0～CTC3 mmmm： 1000～1915	nnnn： 0000～6915 T000～T119 C000～C119 CTC0～CTC3 mmmm： 1000～1915 3000～6915

入力方法



ERROR //

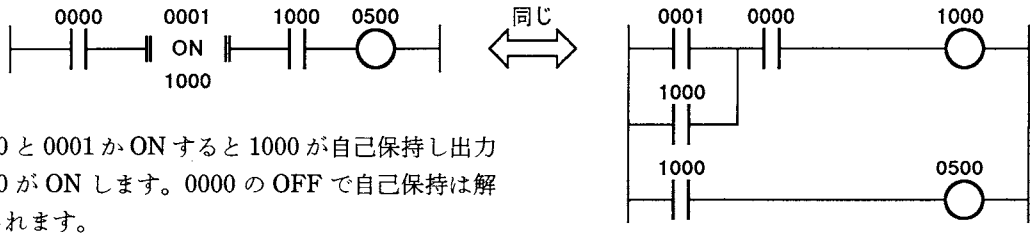
- カイロエラー：入力回路に間違いがある。（例：対応する入力がない）

W-ON, W-UEの詳細

W-ON、W-OFF、W-UE、W-DE の 4 つの命令は、シーケンスのプログラムの中で前の作業が終了した等の条件を持って次の作業に移るというようなときに使用すると便利です。
ここでは細かい動作の違いについて説明します。

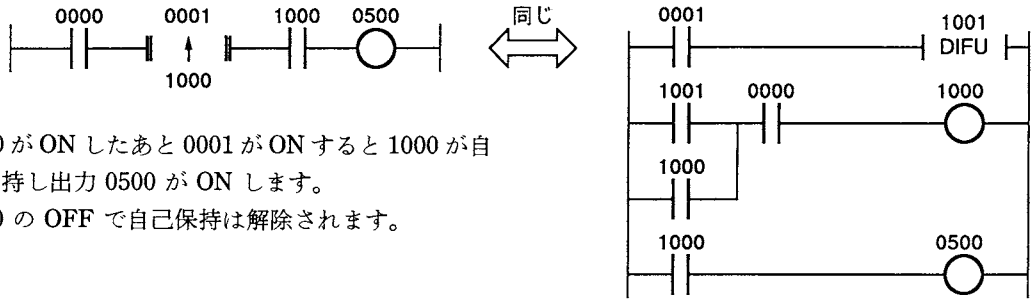
■ ウエイトオン（W-ON）とウエイトアップエッジ（W-UE）の動作と違いを通常ラダーに置き換えて説明します。

〔W-ON〕



●0000 と 0001 が ON すると 1000 が自己保持し出力 0500 が ON します。0000 の OFF で自己保持は解除されます。

〔W-UE〕

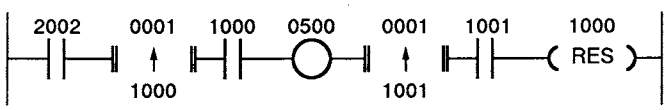


●0000 が ON したあと 0001 が ON すると 1000 が自己保持し出力 0500 が ON します。
0000 の OFF で自己保持は解除されます。

以上のように W-UE は 0000 と 0001 の ON する順番があるのに対して W-ON にはそれがありません。したがって W-UE は入力が入るたびに処理を実行していく回路に適し、W-ON は条件がととのっていれば次に進んでいく回路に適しています。
また W-OFF と W-DE は OFF する条件をもって上記と同じ動作をします。

■ ウエイトアップエッジ（W-UE）を用いた例

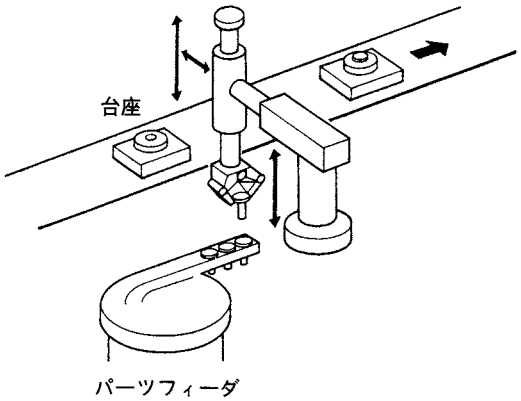
入力 0001 が ON するごとに出力 0500 が交互に ON-OFF します。



W-ON, W-UEの使用例

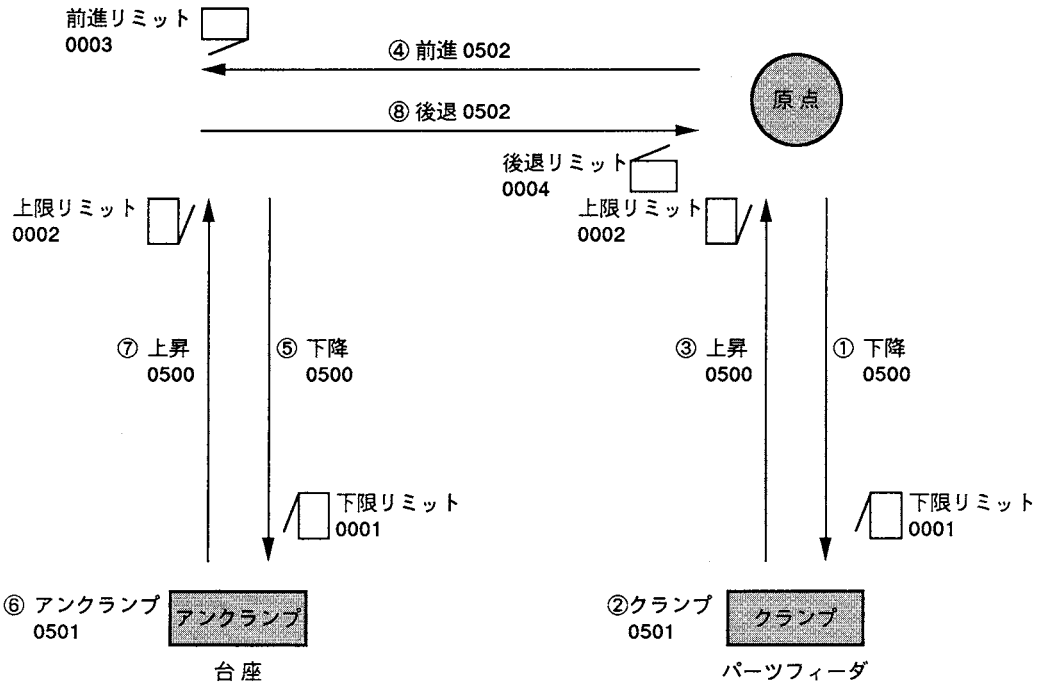
例題

原点位置からパーツフィーダにワークを取りに行き台座まで運びます。
下降、上昇、前進、後進のシリンダ制御にはシングルソレノイドを用います。
したがってソレノイドをONするとシリンダは前進し状態を保持、OFFすると原点に戻ります。
クランプは出力のON/OFFでクランプ、アンクランプとし動作後1秒間は保持し次に進むものとします。



《動作》

原点 → ① ハンド下降 → ② クランプ →
③ ハンド上昇 → ④ ハンド前進 →
⑤ ハンド下降 → ⑥ アンクランプ →
⑦ ハンド上昇 → ⑧ ハンド後退 → 原点

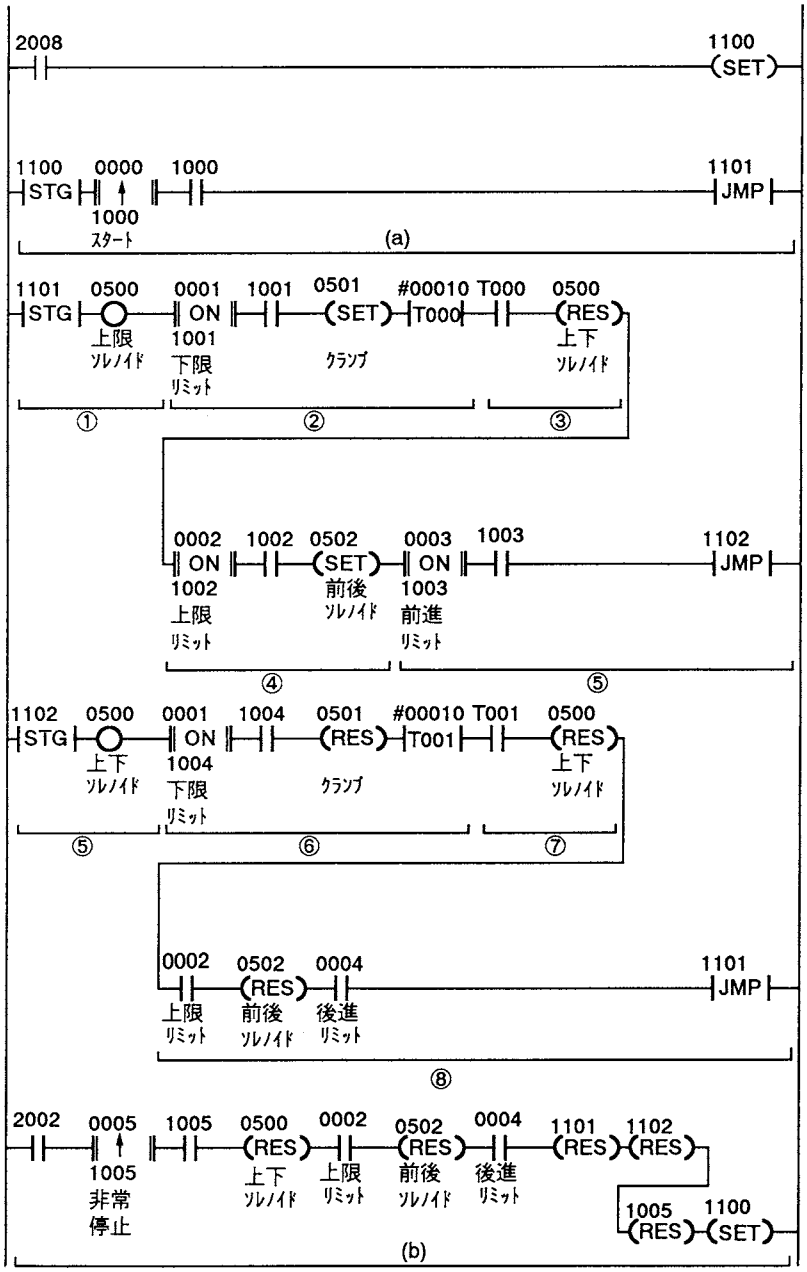


リレーの割付け

0000：スタート信号
0001：下限リミット
0002：上限リミット
0003：前進リミット
0004：後進リミット
0005：非常停止
0500：上下出力
0501：クランプ、アンクランプ出力
0502：前後出力

W-ON, W-UEの使用例

プログラム例



動作

初期設定

(a) スタート信号が ON すると

① ハンドが下降します
② 下限リミットが ON するとワークをクランプします
③ 1 秒間動作を保持しハンドが上昇します

④ 上限リミットが ON するとハンドが前進します
⑤ 前進リミットが ON するとハンドが下降します

⑥ 下限リミットが ON するとワークをアンクランプします
⑦ 1 秒間動作を保持しハンドが上昇します

⑧ 上限リミットが ON するとハンドが後退し動作をくり返します

(b) 非常停止信号が ON するとハンドを上昇させ次に後退させ停止します。
※ 非常停止はプログラムの最後に書くのがポイントです。

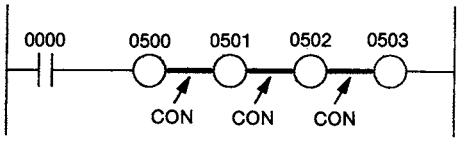
●上記例のように W-ON、W-UE 等の命令は、前の作業の終了を持って次の作業に移る動作をします。
●P4－5、5－12 にもウエイト命令をもちいた例題を掲載していますご覧ください。

コネクト：CON

FUN　RES 0　ORB 6　：　————

出力命令の後の直列接続を示します。

（例1）



●入力リレー0000が「ON」のとき、
0500、0501、0502、0503が「ON」します。
●入力リレー0000が「OFF」のとき、
0500、0501、0502、0503が「OFF」します。

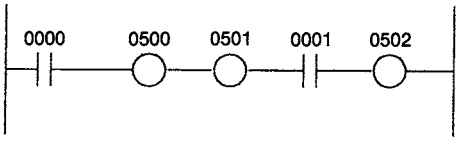
コーディング

行番号	命令	オペランド
0000	LD	0000
0001	OUT	0500
0002	CON	
0003	OUT	0501
0004	CON	
0005	OUT	0502
0006	CON	
0007	OUT	0503

■解説

●出力命令の後に、直列に他の命令を接続する場合に用いる命令です。
この命令は、接続状態を表すだけで、動作に影響は与えません。
●この命令はニモニックでプログラムをする時には必要ですが、ラダー図でプログラムする時には接続線のみを記入するだけでプログラムできます。

（例2）



●入力リレー0000が「ON」すると、0500、0501が「ON」します。さらに入力リレー0001が「ON」すると、0502が「ON」します。

コーディング

行番号	命令	オペランド
0000	LD	0000
0001	OUT	0500
0002	CON	
0003	OUT	0501
0004	CON	
0005	AND	0001
0006	OUT	0502

オペランド

————

入力方法

FUN → RES 0 → ORB 6 → 実行 逆検索 → 実行 逆検索

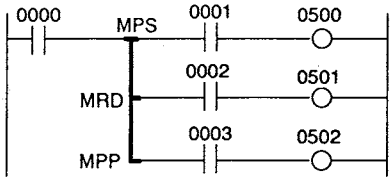
ERROR //

————

5-1. 応用命令

プッシュ：MPS	MPS C	：	入力の状態と、演算フラグを記憶します。
リ　ー　ド：MRD	MRD D	：	プッシュで記憶した入力状態と、演算フラグを読み出します。
ポ　ッ　プ：MPP	MPP Ch	：	プッシュで記憶した、入力の状態と演算フラグを読み出し記憶をクリアします。

（例）



●入力リレー0000が「ON」のとき、
0001が「ON」⟶0500が「ON」します。
0002が「ON」⟶0501が「ON」します。
0003が「ON」⟶0502が「ON」します。

コーディング

行番号	命令	オペランド
0000	LD	0000
0001	MPS	
0002	AND	0001
0003	OUT	0500
0004	MRD	
0005	AND	0002
0006	OUT	0501
0007	MPP	
0008	AND	0003
0009	OUT	0502

■解説

●MPS・MRD・MPP 命令は出力分岐の多い回路の分岐点での ON-OFF 状態の記憶と再現に使用します。
●KZ には、演算の途中結果を記憶するスタックと呼ばれる 8 個のメモリがあります。(図 1)
●MPS 命令を 1 回用いると、その時点の演算結果をスタックの 1 段目に格納します。(図 1 (a)) この状態で再び MPS 命令を実行すると、その時点の演算結果はスタックの 2 段目に格納します。(図 1 (a)) 以降の MPS 命令を実行するごとに順次上段のスタックに格納していきます。(図 1 (b)) このことを「スタックへ積む」といいます。
●MRD 命令は、最上段に格納されている最新データの読み出し専用命令で、スタック内のデータの変化はありません。(図 1 (b))
●MPP 命令は、最上位段のデータが読み出されるとともに、そのデータはスタックから消失します。(図 1 (c)) このことを「スタックからおろす」といいます。
●スタックへ積む個数は 8 個以下としてください。
●MPS の使用個数と MPP の使用個数は同一としてください。同一でない場合は、MPP または MPS 命令が入る位置で「カイロエラー」となります。
●これらの各命令はニモニックでプログラムをするときには必要ですが、ラダー図でプログラムをするときには接続線のみを記入するだけでプログラムできます。

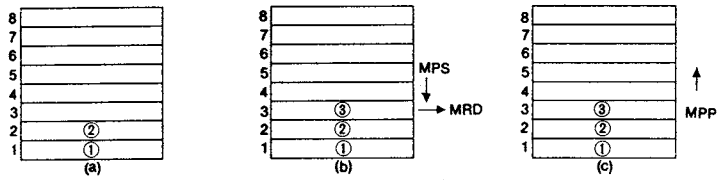
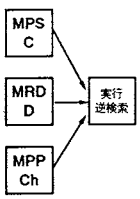


図 1　スタックの内部状態概念図

オペランド

_____

入力方法



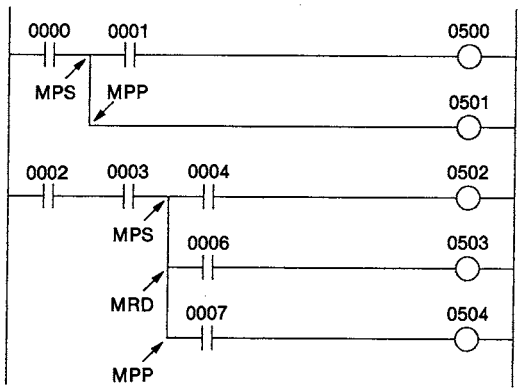
ERROR //

●カイロエラー：MPS の使用個数と MPP の使用個数が同一でない。
● MPS ネストエラー：スタックへ積む個数が 9 個以上となった。

MPS・MRD・MPP命令の使用例

MPS・MRD・MPPの命令を使用した例を以下に示します。

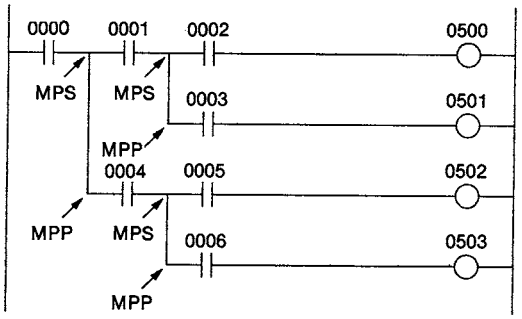
（例1）一段スタックの例



コーディング

行番号	命令	オペランド
0000	LD	0000
0001	MPS	
0002	AND	0001
0003	OUT	0500
0004	MPP	
0005	OUT	0501
0006	LD	0002
0007	AND	0003
0008	MPS	
0009	AND	0004
0010	OUT	0502
0011	MRD	
0012	AND	0006
0013	OUT	0503
0014	MPP	
0015	AND	0007
0016	OUT	0504

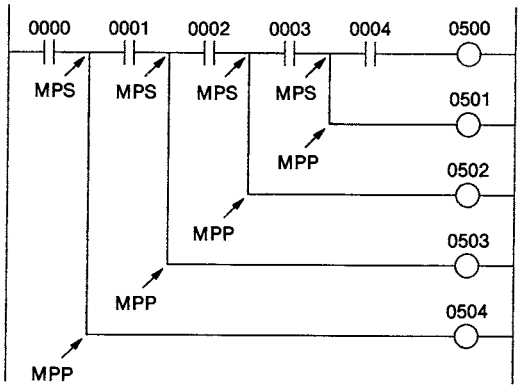
（例2）二段スタックの例



コーディング

行番号	命令	オペランド
0000	LD	0000
0001	MPS	
0002	AND	0001
0003	MPS	
0004	AND	0002
0005	OUT	0500
0006	MPP	
0007	AND	0003
0008	OUT	0501
0009	MPP	
0010	AND	0004
0011	MPS	
0012	AND	0005
0013	OUT	0502
0014	MPP	
0015	AND	0006
0016	OUT	0503

（例3）四段スタックの例



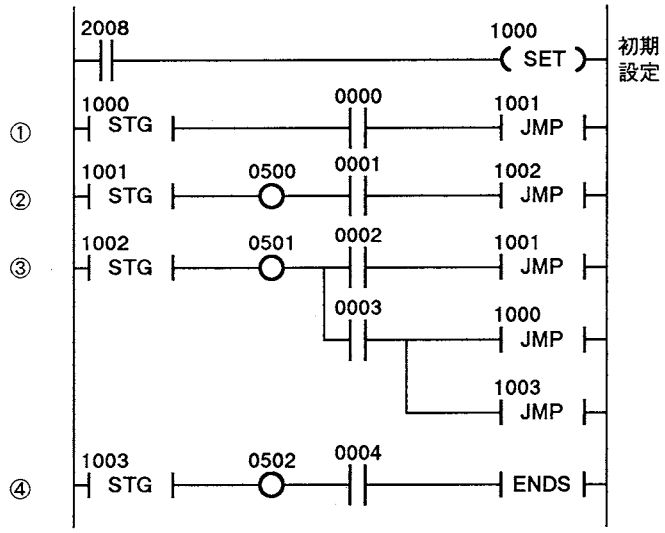
コーディング

行番号	命令	オペランド
0000	LD	0000
0001	MPS	
0002	AND	0001
0003	MPS	
0004	AND	0002
0005	MPS	
0006	AND	0003
0007	MPS	
0008	AND	0004
0009	OUT	0500
0010	MPP	
0011	OUT	0501
0012	MPP	
0013	OUT	0502
0014	MPP	
0015	OUT	0503
0016	MPP	
0017	OUT	0504

（注）MPSの個数とMPPの個数の差は8以下とし、最終的には一致させてください。

ステージ：STG	FUN LDB 4 LDB 4 : nnnn STG	オペランドのリレーがONのときSTGのブロックの命令を実行します。
ジャンプ：JMP	FUN ANL 2 SET 1 : nnnn JMP	移行条件がONのとき現在実行しているステージのリレーをOFFにしてオペランドのステージに移行します。
エンドステージ：ENDS	FUN SET 1 LDB 4 : ENDS	移行条件がONのとき、現在実行しているステージのリレーをOFFにします。

(例)



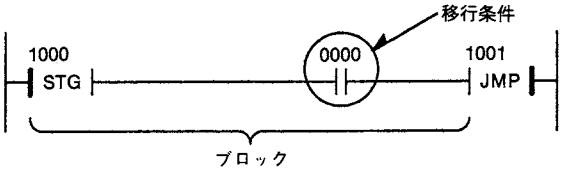
コーディング

行番号	命令	オペランド
0000	LD	2008
0001	SET	1000
0002	STG	1000
0003	AND	0000
0004	JMP	1001
0005	STG	1001
0006	OUT	0500
0007	CON	
0008	AND	0001
0009	JMP	1002
0010	STG	1002
0011	OUT	0501
0012	CON	
0013	MPS	
0014	AND	0002
0015	JMP	1001
0016	MPP	
0017	AND	0003
0018	JMP	1000
0019	JMP	1003
0020	STG	1003
0021	OUT	0502
0022	CON	
0023	AND	0004
0024	ENDS	

① 移行条件 0000 の ON で STG 1001 に移行します。
② 0500 が ON し移行条件 0001 の ON で STG 1002 に移行します。
③ 0501 が ON し
- 移行条件 0002 の ON で STG 1001 に移行し処理をくり返します。
- 移行条件 0003 の ON で STG 1000、1003 に移行します。

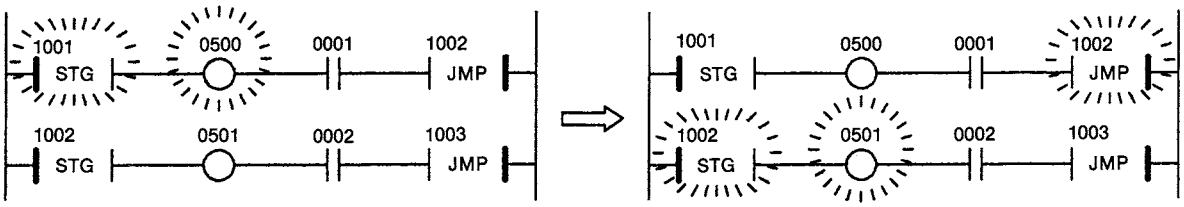
④ 0502 が ON し移行条件 0004 の ON で終了します。

■解説



●上記例で STG 命令から JMP 命令までの間をブロックと呼び、次のステージに移る条件を (この場合 0000) 移行条件と呼びます。

●STG のオペランド (この場合 1001) が ON しているときはそのブロックの命令 (この場合出力 0500) を実行します。
移行条件 (この場合 0001) が一瞬でも ON すると、そのステージを終了し次のステージに移行します。
この場合出力 0500 は OFF し 0501 が ON します。

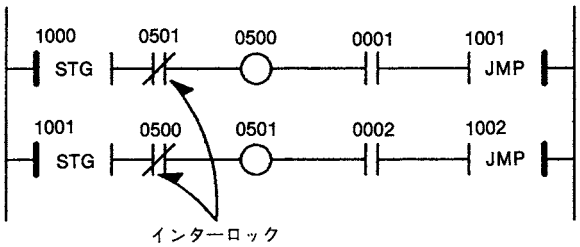


●ステージのオペランドが OFF のときもしくは OFF になったとき STG-JMP、ENDS 間の各命令は右表のようになります。

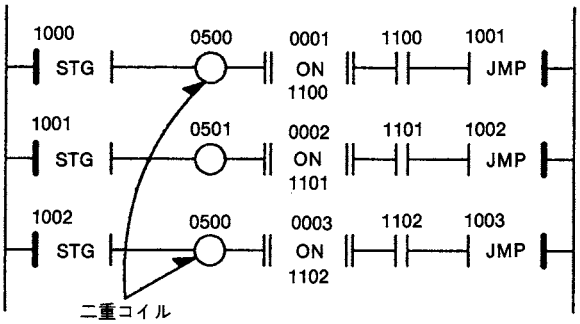
命令語	機能
OUT 命令でのリレー	OFF
OUB 命令でのリレー	ON
TMR・TMH・TMS・C (カウンタ) 命令	リセット
セット・リセット命令	状態保持
その他のリレー、演算命令	命令を実行しません

ステージ命令の詳細

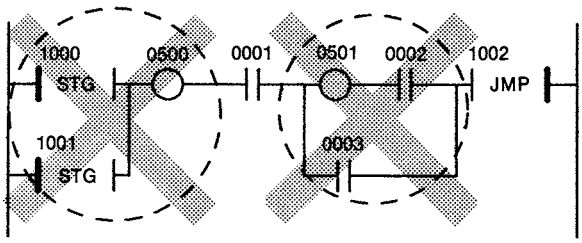
●ステージが移行するときには 1 サイクル (1 スキャン) だけ前後のステージが同時に ON します。
同時に ON すると危険があるような出力 (たとえばモーターの正転、逆転) に対してはインターロック回路にしてください。



●ステージ命令では二重コイルを使用することができます。
また出力を保持させたい場合はセット命令を使用します。
●実行してない STG、JMP、ENDS 間はスキャンタイムに影響しません。



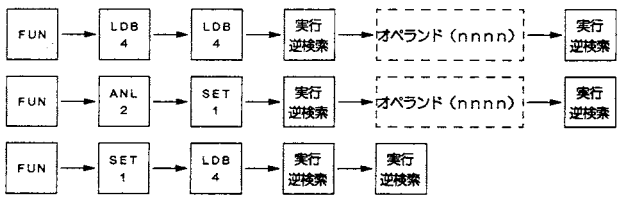
●必ず母線より始めてください。また STG のオペランドを並列に接続したり、出力回路を含む並列接続はできません。
●STG 命令のオペランドのリレー番号は重複して使用できません。
またサブルーチンプログラムの中では使用できません。



オペランド

KZ−10, 16	KZ24, 40, 80
1000～1915	1000～1915 3000～6915

入力方法



ERROR !!

●カイロエラー：入力回路に間違いがある。(例：対応する入力がない)

ステージ命令(STG)の応用について

1．ステージ命令とは

●ステージ命令とは、行程に合せて順次処理を行なっていくような(シーケンシャルな)処理の実行に適しています。
流れとしては全体を細い工程に分け、1つの工程ごとにプログラムをし処理を実行します。

●図1のように、順を追ってランプを点灯させるような場合、次の5つの処理に分けて考えることができます。

①スタートスイッチがONするのを待つ。
②ランプ1を1秒間点灯させる。
③ランプ2を3秒間点灯させる。
④ランプ3を5秒間点灯させる。
⑤ランプ1、2、3を7秒間点灯させる。

これら5つの処理を①→②→③→④→⑤と順々に行なえば、図1のフローチャートの処理を行なうことができます。
このような順次処理の行程を従来のプログラムで行うとプログラムが複雑になり、また第3者が見た場合にも、流れがわかりにくくなる欠点があります。

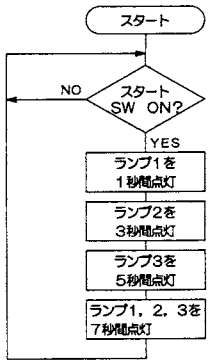


図1．シーケンシャル処理

2．入出力の割付け

ここではKZの入出力を次のように割付けます。

1．スタートスイッチ　0000　ONでスタート
2．ランプ1　0500　ONで点灯
3．ランプ2　0501　ONで点灯
4．ランプ3　0502　ONで点灯

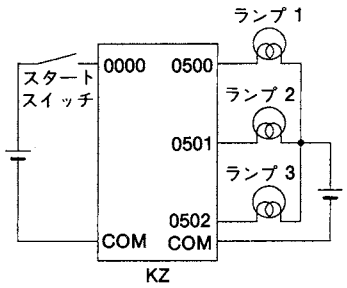


図2．入出力の割付け

3．各シーケンスのプログラム

項目1で5つに分けられた処理を駆動するリレーを次のように割り付けます。

①スタートスイッチがONするのを待つ。…1000
②ランプ1を1秒間点灯させる ……1001
③ランプ2を3秒間点灯させる ……1002
④ランプ3を5秒間点灯する ……1003
⑤ランプ1、2、3を7秒間点灯させる ……1004

上記のリレーを用いて個々の処理をステージ命令を用いて書くと図3のようになります。

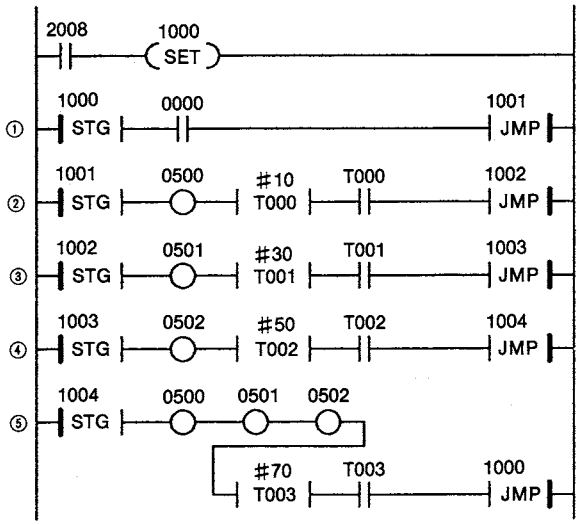


図3　ラダー図

●図3のラダー図のニモニックを右記に示します。

ワンポイント

ステージ命令に代表される拡張ラダーをもちいたプログラムを行う場合プログラムコンソール KZ-P3 をもちいてニモニックで打ち込むよりも、ラダーサポートソフトをもちいてラダー図でプログラムを行うほうが簡単に入力が行え、デバッグも早く行うことが可能です。

コーディング

行番号	命令	オペランド
0000	LD	2008
0001	SET	1000
0002	STG	1000
0003	AND	0000
0004	JMP	1001
0005	STG	1001
0006	OUT	0500
0007	CON	
0008	TMR	0000 #00010
0009	CON	
0010	AND	T000
0011	JMP	1002
0012	STG	1002
0013	OUT	0501
0014	CON	
0015	TMR	0001 #00030
0016	CON	
0017	AND	T001
0018	JMP	1003

行番号	命令	オペランド
0019	STG	1003
0020	OUT	0502
0021	CON	
0022	TMR	002 #00050
0023	CON	
0024	AND	T002
0025	JMP	1004
0026	STG	1004
0027	OUT	0500
0028	CON	
0029	OUT	0501
0030	CON	
0031	OUT	0502
0032	CON	
0033	TMR	003 #00070
0034	CON	
0035	AND	T003
0036	JMP	1000

ウエイト命令をもちいた歩進

(例)

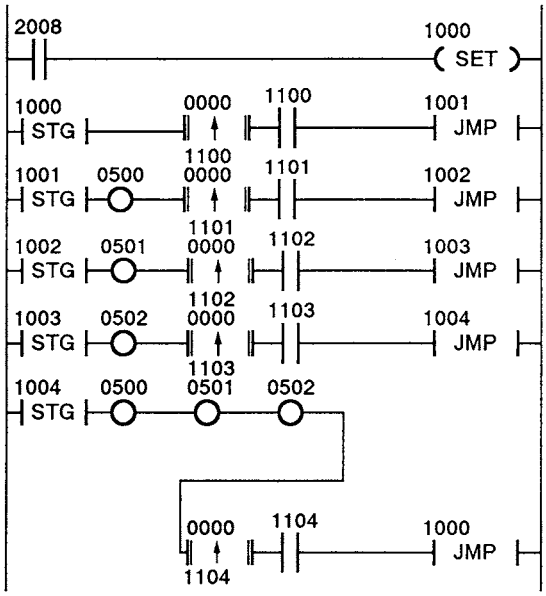


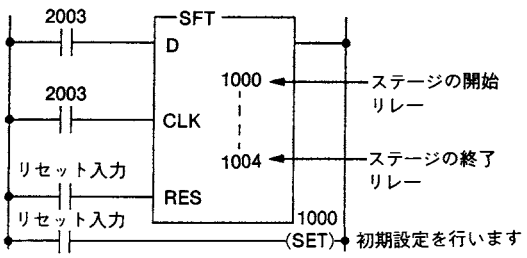
図4

■ 解説

図3のプログラムをウエイト命令をもちいてスタートスイッチ 0000 が ON するごとに (歩進) 次のステージに移行します。

ワンポイント

ステージ命令のリセット (図4の例の場合)
非常停止などですべてのステージ命令を OFF させたいときはシフト命令を用います。



(初期設定はプログラムの作成状態に応じて付加してください。図4の例では、初期設定を行なわないと次のスタートができません。)

シフト命令については P3−27 をご覧ください。

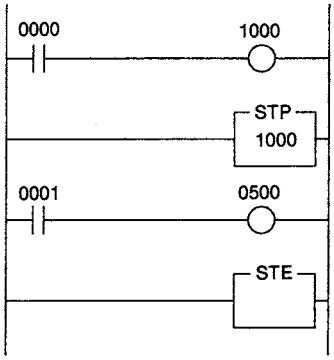
●P4−9、5−4 にもステージ命令をもちいた例題を掲載していますご覧ください。

ステップ：STP　FUN 4 LDB 5 ANB : STP nnnn

ステップエンド：STE　FUN 4 LDB 3 ORL : STE

オペランドのリレーがONのとき、STEまでの命令を実行します。
STP命令と対で使用し、ひとつのステップを形成します。

（例）



コーディング

行番号	命令	オペランド
0000	LD	0000
0001	OUT	1000
0002	STP	1000
0003	LD	0001
0004	OUT	0500
0005	STE	

1000が「ON」のとき0500は0001のON-OFFに連動しON-OFFしますが、1000が「OFF」のときは0001が「ON」しても0500は「ON」しません。また0001、0500ともに「ON」のまま1000を「OFF」すると、0001を「OFF」しても0500は「ON」のままです。

■解説

●STP-STE間のプログラムは、STP命令のオペランドのリレーがONのときのみ実行され、OFFのときは実行されずにSTEまでジャンプします。(図1)

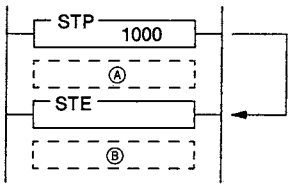
●STP、STE命令は(図2)のようなネスティング(入れ子)構造を許しています。最大8重まで可能です。1000がOFFのとき①②③は実行されず、④から実行されます。1000、1001がともにONのときは①②③の順に実行されます。1000がON、1001がOFFのときは、①③④の順で実行され、②は実行されません。

●実行されていないSTP、STE間はスキャンタイムに影響しません。

●STP−STE間のプログラムを実行中にSTPのオペランドのリレーがOFFした場合、STP−STE間の命令は右表のようになります。

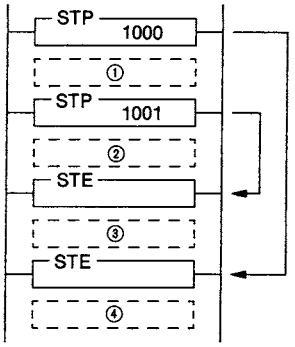
命令語	機　能
すべての命令	状態保持

●(例)では0500がON中に1000がOFFすると、0500のOFF条件が不明になるため、0500はONしたままになります。



1000がOFFのとき、Aは実行されません。

(図1)

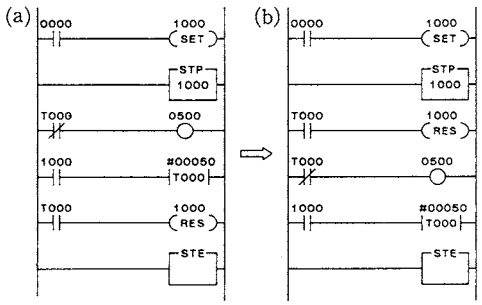


1000がOFFのとき①、②、③は実行されません。

(図2)

ワンポイント

●STP 内でのタイマの使い方
STP 内でタイマを働かせているときに、STP の実行を中断するとタイマが働き続け（状態保持）、次に STP が実行されたとき、#0000（ON 状態）から始まることがあります。（右図(a)）。このようなとき右図(b)のように書き直してください。



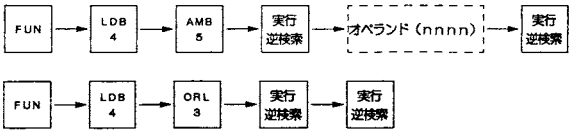
ステージ命令とステップ命令の違い

ステージ命令とステップ命令は同じような工程歩進の動作をします。
ステップ命令は従来のラダープログラムの中で工程歩進を行うための命令ですが、拡張ラダーが使用できる KZ ではより使いやすくなったステージ命令の使用をおすすめします。

オペランド

KZ−10, 16	KZ24, 40, 80
0000～2915	0000～6915
T000～T063	T000～T119
C000～C063	C000～C119
CTC0～CTC3	CTC0～CTC3

入力方法



ERROR //

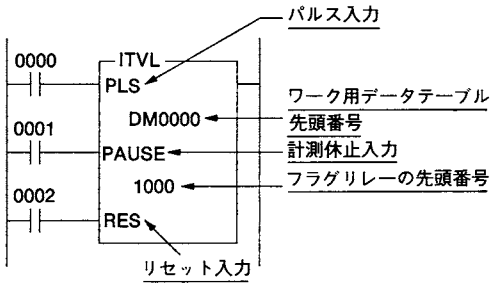
● STP−STE エラー：STP に対応する STE 命令がない。
　　入れ子の数が 9 個以上となった。
●カイロエラー　　：STE に対する STP 命令がない。
　　入力回路に間違いがある。（例：入力が直接母線に接続されていない）

インターバルタイマ　：ITVL

ITVL
#TM　：　ITVL
PLS
DMxxxx
PAUSE
mmmm
RES

設定されたモードでパルス間隔やパルス長を計測します。

（例）



コーディング

行番号	命令	オペランド
0000	LD	0000
0001	LD	0001
0002	LD	0002
0003	ITVL	DM0000 1000

●入力リレー0000のパルス間隔またはパルス長を計測します。

■動作説明

●PLS 入力へ入れるパルスを指定したモード(0～5)により計測し、計測値をデータメモリの指定の位置へ格納します。
1回計測が終了すると、1回計測終了リレーを1スキャンだけ ON します。

●計測値が MAX または、MIN 設定値を超えたときに、MAX エラーまたは、MIN エラーのリレーを1スキャンだけ ON します。
また、MAX、MIN 計測値を超えたときに、MAX、MIN 計測値を更新します。
(初期値：MAX 計測値は0、MIN 計測値は65535 です。)

●平均回数で指定した回数で平均値を計算してデータメモリの指定の位置へ格納し、平均値計測終了リレーを1スキャンだけ ON します。

●RES 入力が ON すると、データメモリの内容を初期化します。(＋4以降)
RES 入力が ON の間は計測をしません。

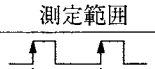
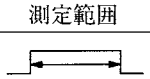
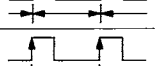
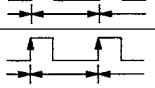
● PAUSE 入力が ON すると、計測を一時停止します。

■解説

●ITVL 命令では、指定したデータメモリ番号から後ろの15個のデータメモリと、指定したリレー番号から4個のリレーを使用します。(これらを他の用途に用いて値を変更した場合、動作は保証できません。)

●モードは0～5までの6種類あります。

モード設定

	パルス間隔測定			パルス幅測定	
モード	測定範囲	測定単位	モード	測定範囲	測定単位
0		10msec 単位	1		10msec 単位
2		100msec 単位	3		100msec 単位
4		1sec 単位	5		1sec 単位

●データメモリ、リレーの内容は次のとおりです。
● ITVL 命令を使用するときは、DM の最初から 4 個にデータを設定して使用します。

DM 内容

DM ××+0	モード設定	モード番号 0～5 を設定します。	ユーザー設定（初期設定を行います）
1	MAX設定値	計測範囲の最大値を設定します。	
2	MIN設定値	計測範囲の最小値を設定します。	
3	平均回数	平均値を求めるための回数を設定します。(1～65535 回)	
4	計測値	計測毎の計測値が入力されます。	
5	MAX計測値	計測値の最大値が入力されます。	
6	MIN 計測値	計測値の最小値が入力されます。	
7	平均値	平均回数で指定した回数の平均値が入力されます。	
8	計測回数	計測した回数が入力されます。	
9	積算値上位	計測値の積算値上位 5 桁が入力されます。（#00000～#65535）	
10	積算値下位	計測値の積算値下位 5 桁が入力されます。（$0000 ～$FFFF）	
11～15	システム用作業メモリ	計測値、MAX 計測値、MIN 計測値、平均値、積算値を計算するための作業用のメモリです。(使用しないでください)	

リレー内容

リレー番号		
+0	MAX エラー	計測値が MAX 設定値を超えた場合に 1 スキャンだけ ON します。
+1	MIN エラー	計測値が MIN 設定値を超えた場合に 1 スキャンだけ ON します。
+2	1 回計測終了	計測終了ごとに 1 スキャンだけ ON します。
+3	平均値計測終了	平均回数で指定した回数計測が終了すると 1 スキャンだけ ON します。

●平均回数に 0 を指定すると平均値の計測を行いません。このとき積算値は RES が ON するまで積算しますので、オーバーフローすることがあります。
●平均回数で指定した回数になると計測回数、積算値上位、下位はリセットされます。
●MC-MCR 中で使用すると、MC-MCR 間を実行しないときにはポーズ状態となります。
●サブルーチン、割込みプログラム、ステップ、ステージなどを使用したプログラムでは毎スキャン必ず実行される場所で使用してください。

オペランド

KZ－10, 16	KZ24, 40, 80
DM0000 ～DM0985	DM0000 ～DM1985
1000～1912	1000～1912
	3000～6912

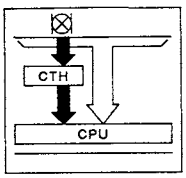
入力方法

ITVL #TM → 第1オペランド (DMxxxx) → ➡ → 第2オペランド (mmmm) → 実行 逆検索

ERROR!!

●カイロエラー：入力回路に間違いがある。(例：入力がない。少ない)
●スタックエラー：入力回路に間違いがある。(例：入力が多すぎる)

8ビット 高速カウンタ：CTH0	CTH RES 0 : CTH0 nnnn CTH SET 1 : CTH1 nnnn	応答周波数10kHzの8ビット(0～255)、16ビット(0～65535)の加算カウンタの動作を行います。 入力がONのときのみ加算し、OFFで現在値をリセットします。
16ビット 高速カウンタ：CTH1		
高速カウンタ コンパレータ：CTC	CTC : #ddddd CTCx	設定値と高速カウンタの現在値をハード的に比較し、等しくなったときにセットします。

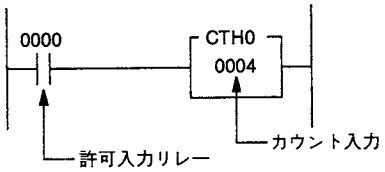


通常プログラム ⇨
高速カウンタ ➡

KZのカウンタは、ソフト上で処理されるため「サイクルタイム」の影響を受け、プログラムが長くなればなる程、短い信号が取り込みにくくなります。

「高速カウンタCTH0、CTH1」は、独立したハード上のカウンタですので、サイクルタイムの影響を受けずに10kHzの入力応答が実現できます。

高速カウンタ（CTH0、CTH1）



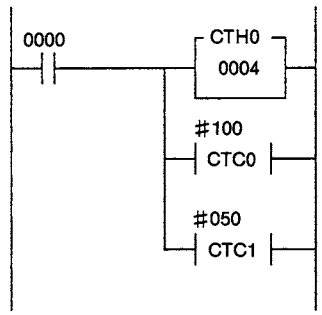
●高速カウンタは、オートリセットの加算カウンタとして動作し、現在値しか持っていません。
設定値は高速カウンタコンパレータ(CTC)により設定します。
●高速カウンタの現在値をクリアするには、許可入力リレーをOFFにするか、RES CTH命令を使用します。
●高速カウンタのオペランドに指定するリレーの入力パルスをカウントします。
●測定した現在値は電源OFF時も2ヵ月以上保持しています。

●高速カウンタのカウント入力には次のリレーが設定できます。

	計測範囲	カウント入力	
CTH0	000 ～255	0004 (外部カウント) 2100 (内部クロック 0.8μs) 2101 (内部クロック 6.4μs) 2102 (内部クロック 102.4μs)	入力0004の立ち上がりでカウントします。 内部クロックのパルスをカウントします。
CTH1	00000 ～65535	0005 (外部カウント) 0500 (外部クロック内部結線) 2200 (内部クロック 0.4μs) 2201 (内部クロック 0.8μs) 2202 (内部クロック 3.2μs)	入力0005の立ち上がりでカウントします。 出力0500の立ち上がりでカウントします。 内部クロックのパルスをカウントします。

●高速カウンタのカウント入力に内部クロックを指定すると高速タイマとして使用できます。
●CTH命令の重複はできません。
●高速カウンタはサブルーチンや割込みプログラム中では使用できません。

高速カウンタコンパレータ（CTC）



●高速カウンタコンパレータは、設定値と高速カウンタの現在値が等しくなったときに接点をセットします。
●高速カウンタコンパレータは一度ONすると、RES CTC命令を実行するまでOFFしません。(P5－19 例参照)
但し、CTCにより割込みプログラムが起動されると自動的にOFFとなります。また割込みを行うことでスキャンタイムに影響されないダイレクト出力が可能です。
(割込みプログラム→P5－50)

●高速カウンタと高速コンパレータの関係は次のようになります。

高速カウンタ	高速カウンタコンパレータ	高速カウンタコンパレータ設定範囲
CTH0	CTC0 CTC1	000～255
CTH1	CTC2 CTC3	00000～65535

CTH0 のコンパレータとして CTC2、CTC3 を使用することや、CTH1 のコンパレータとして CTC0、CTC1 を使用することはできません。

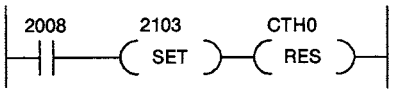
●高速コンパレータの設定値をプログラム中で変更したい場合は、STA CTC命令を使用することにより変更できます。(STA 命令 → P5－29)

高速カウンタコンパレータによる高速カウンタの現在値クリア

●高速カウンタの現在値をクリアするには、入力リレーを OFF するか、RES CTH 命令を使用します。
しかし、これとは別に特殊補助リレーをあらかじめ ON させておくことにより、高速カウンタコンパレータの設定値と高速カウンタの現在値が同値となったときに高速カウンタの現在値をクリアすることができます。
それぞれの高速カウンタ、高速カウンタコンパレータと特殊補助リレーの関係は次のようになります。

高速カウンタ	高速カウンタコンパレータ	リレー番号	動作
CTH0	CTC0	2103	特殊補助リレー2103をあらかじめ ON しておくと、CTC0 の設定値と CTH0 の現在値が同値となったときに CTH0 の現在値をクリアします。
CTH1	CTC2	2203	特殊補助リレー2203をあらかじめ ON しておくと、CTC2 の設定値と CTH1 の現在値が同値となったときに CTH1 の現在値をクリアします。

●プログラム中での設定方法（参考例）



運転開始時に2103をセットし、CTH0 の現在値を 0 クリアします。

●高速カウントと高速カウンタコンパレータの応用としてダイレクトクロックパルスの出力ができます。
詳しくは 6 章 高速カウンタの応用をご覧ください。

●INT 命令をもちいることによりスキャンタイムに影響を受けないダイレクト出力ができます。
詳しくは P5－51 例 2 をご覧ください。

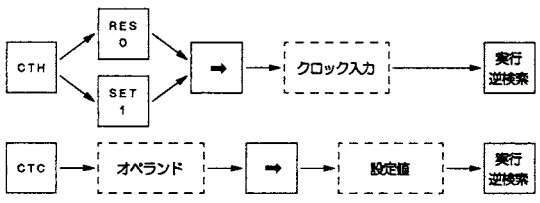
高速カウンタの現在値クリア時の注意点

●許可入力リレーを OFF にするか、RES CTH 命令を使用すると、スキャンタイムごとに現在値をクリアしますので、高速カウンタで速いパルスをカウントする時は現在値が変化することがあります。このため、コンパレータ (CTC) の設定値によっては CTC の接点が ON します。

オペランド

	オペランド
CTH0	0004 2100～2102
CTH1	0005 0500 2200～2202
CTC	設定値 0, 1 #000～#255 2, 3 #00000～#65535

入力方法

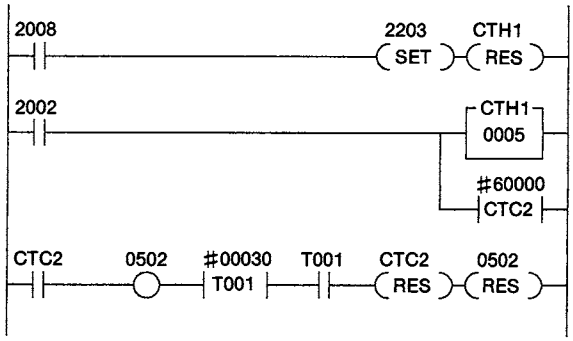


ERROR //

●カイロエラー：入力回路に間違いがある。(例：入力リレーが接続されていない)

高速カウンタの使用例

（例）：入力リレー0005のパルスをカウントし、設定値になれば出力リレー0502を3秒間ONさせます。



入力リレー0005のパルスをカウントし、60000カウントに達すると、出力リレー0502を3秒間出力します。

コーディング

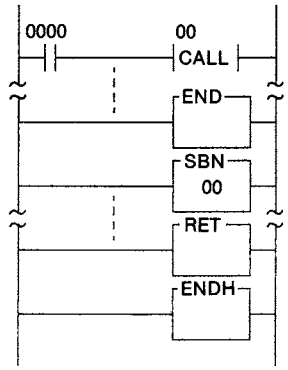
行番号	命令	オペランド
0000	LD	2008
0001	SET	2203
0002	CON	
0003	RES	CTH1
0004	LD	2002
0005	CTH	1　0005
0006	CTC	2　#60000
0007	LD	CTC2
0008	OUT	0502
0009	CON	
0010	TMR	001　#00030
0011	CON	
0012	AND	T001
0013	RES	CTC2
0014	CON	
0015	RES	0502

■解説

- 運転開始時にCTC2の接点でCTH1の現在値を0クリアするように特殊補助リレー2203をセットしておき、高速カウンタを0クリアします。
- 高速カウンタの入力時定数は「25 μs」に固定されますので、HSP命令は不要です。
- CTCの接点は一度ONすると、RES CTC命令を実行するまでOFFしません。繰り返し同じ動作をさせるときは必ずRES CTC命令を使用してください。

サブルーチンコール：CALL	FUN 0 RES 0 ORL 3 : XX CALL	オペランドで指定したサブルーチンを実行します。
サブルーチンエントリ：SBN	FUN 0 ORL 3 AND 8 : SBN XX	オペランドで指定したサブルーチンの領域の先頭を示します。
サブルーチンリターン：RET	FUN 0 ORL 3 ORL 3 : RET	指定されたサブルーチンを終了します。

（例）



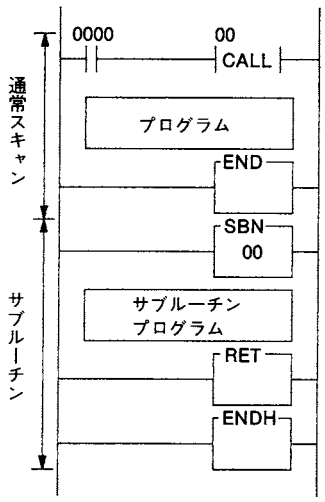
● 入力リレー0000がONすると、SBN00からRETまでのプログラムを実行し、終了すると、CALL命令の後のプログラムを実行します。

コーディング

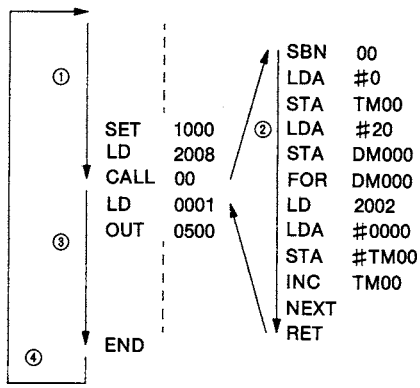
行番号	命令	オペランド
0000	⋮	⋮
⋮	⋮	⋮
⋮	LD	0000
⋮	CALL	00
⋮	⋮	⋮
⋮	END	
⋮	SBN	00
⋮	⋮	⋮
⋮	RET	
⋮	ENDH	

■解説

● CALL命令は入力リレーがONの場合、オペランドで指定した番号のサブルーチンを実行します。
● SBN命令は、サブルーチンの先頭を表します。サブルーチンの開始は必ずこの命令を使います。
● RET命令は、サブルーチンを終了する命令です。サブルーチンの最後に入れます。
●サブルーチンは、ENDとENDHの間に記述します。(図1参照)
●サブルーチン命令は通常スキャンしているプログラムより、CALL命令により、指定した番号のサブルーチン(SBN～RET間)を実行し、RET命令で再びそのサブルーチンを指定したCALL命令の次の命令に戻ります。(図2参照)



（図1）サブルーチンプログラムの記述例



通常スキャン①を実行中にCALL00を実行すると、サブルーチン②を実行します。サブルーチン②を実行し終えると、CALL00の命令の次の命令LD0001へ戻り、通常スキャン③をENDまで実行します。

（図2）サブルーチンプログラムの実行の流れ

5-1. 応用命令

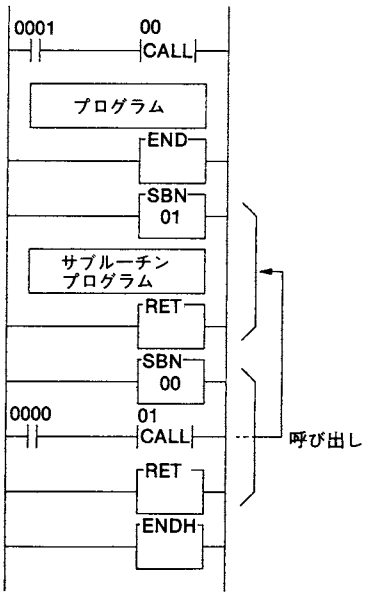
●サブルーチンを実行すると、その実行時間だけスキャンタイムが長くなります。
●サブルーチン命令は (図 3) のようなネスティング (入れ子) 構造を許しています。最大 4 重まで可能です。
●割込みプログラムよりサブルーチンを呼びだせます。(図 4 参照)
●SBN 命令のサブルーチン番号は重複使用できません。
●サブルーチンプログラム中で使用できない命令。

STG、JMP、ENDS、INT、RETI、MEMSW

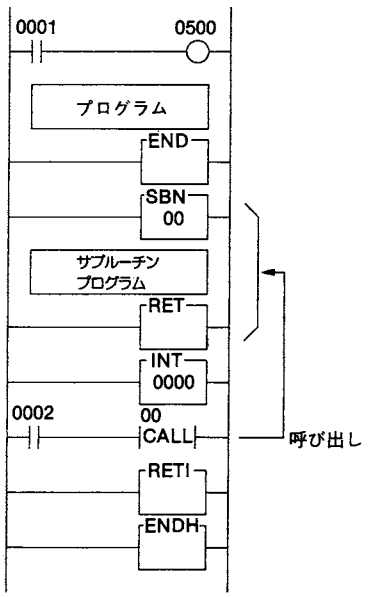
●毎スキャン必ず実行されるサブルーチンプログラムでのみ使用できる命令

TMR、TMH、TMS、C、SFT、UDC、DIFU、DIFD、W-UE、W-DE、HKEY、ITVL、CTH

スキャンについては P2−2 をご覧ください。



（図 3）サブルーチンのネスティング構造



（図 4）割り込みプログラムよりのサブルーチンの呼び出し

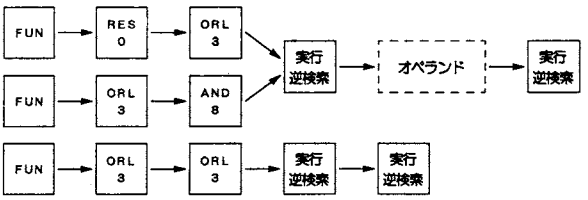
●サブルーチンの中より別のサブルーチンを呼び出すとき、呼び出されるサブルーチンのプログラムはすでに定義されている必要があります。

●割込みプログラム中よりサブルーチンを呼び出すとき、呼び出されるサブルーチンのプログラムはすでに定義されている必要があります。

オペランド

CALL	00～99
SBN	00～99
RET	——

入力方法

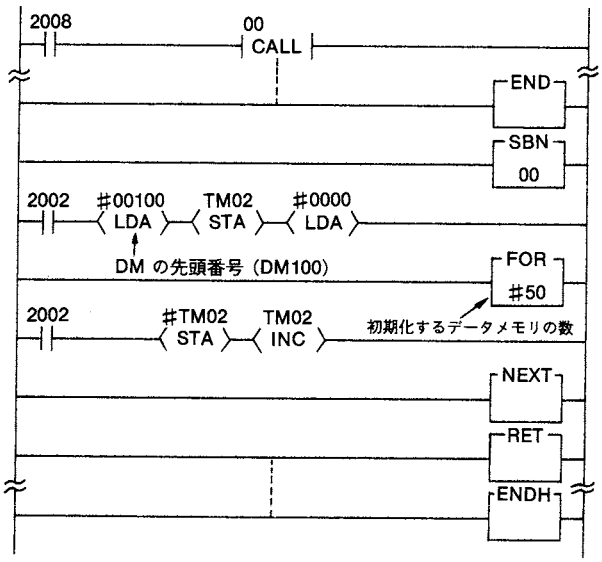


ERROR //

●カイロエラー：入力回路に間違いがある。(例：対応する入力がない)
●SBN テイギナシエラー：CALL 命令で呼ばれているサブルーチンが定義されていない。
● SBN ネストエラー：ネスティングが 5 重以上となった。
●SBN サンショウエラー：サブルーチン内の CALL 命令で呼ばれているサブルーチンが、プログラムですでに定義されていない。

繰り返し範囲開始 : FOR	FUN SET 1 ORB 6 : FOR nnnn	1スキャン中にオペランドで指定した回数だけ FOR～NEXT 間を繰り返し実行します。
繰り返し範囲終了 : NEXT	FUN ANL 2 OR 9 : NEXT	繰り返しの範囲の最後を示します。

(例 1)



コーディング

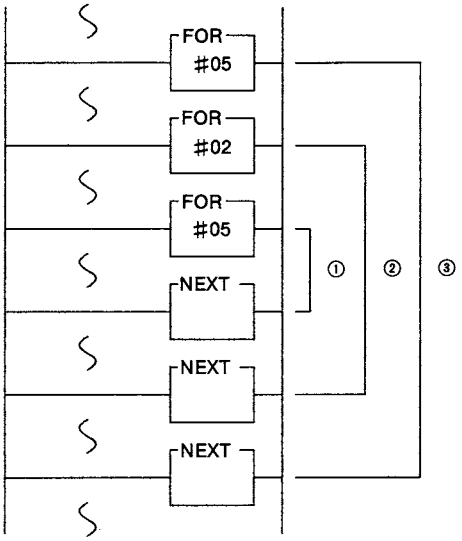
行番号	命令	オペランド
0000	LD	2008
0001	CALL	00
⋮	⋮	
0099	END	
0100	SBN	00
0101	LD	2002
0102	LDA	#00100
0103	CON	
0104	STA	TM02
0105	CON	
0106	LDA	#00000
0107	FOR	#00050
0108	LD	2002
0109	STA	#TM02
0110	CON	
0111	INC	TM02
0112	NEXT	
0113	RET	
⋮	⋮	
0199	ENDH	

●サブルーチンプログラム (SBN～RET) の中で FOR～NEXT 命令を使用し、運転開始時にデータメモリ (DM100～DM149) の初期化を行います。またデータメモリの変更には間接指定を利用し TM02 の値を毎スキャン +1 することにより初期化を完了します。
(データメモリの間接指定については P5－57 をご覧ください)

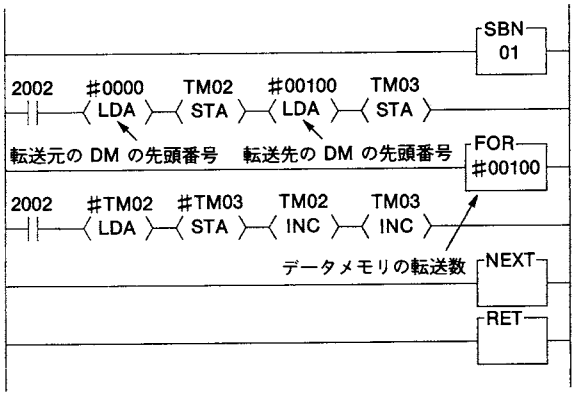
■解説

●FOR 命令は、オペランドに繰り返したい回数の値、または、その値が記憶されているメモリを指定します。
●NEXT 命令は繰り返しの最後に書きます。
●FOR 命令は毎スキャン指定回数繰り返しを行います。
使用方法が適切でない場合は、スキャンを大きく伸ばしてしまうことがあります。
このようなことを防ぐためには、FOR～NEXT 命令はステップやサブルーチンで使用します。
●上記例ではデータメモリの間接指定を利用し、DM の初期化をおこないます。
●データメモリ (DM×××)、テンポラリデータメモリ (TM××) については P2－10 をご覧ください。

●FOR～NEXT 命令は（図1）のようなネスティング（入れ子）構造を許しています。最大8重まで可能です。
ネスティングが9重以上となると「FOR～NEXT エラー」となり実行できません。
●③のプログラムを5回実行し NEXT 以降のプログラムへ移ります。
●③のプログラムを1回実行する間に②のプログラムは2回実行されます。
従って②のプログラムは計10回実行されます。
●①のプログラムは、②のプログラムが1回実行される間に5回実行され、全体で5×2×5＝50回実行されます。



(例 2) データメモリのブロック転送



●DM0000～DM0099 までの内容を間接指定を使用して DM0100～DM0199 へブロックで転送します。

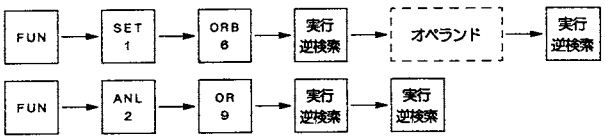
コーディング

行番号	命令	オペランド
0000	SBN	01
0001	LD	2002
0002	LDA	#0000
0003	CON	
0004	STA	TM02
0005	CON	
0006	LDA	#00100
0007	CON	
0008	STA	TM03
0009	FOR	#00100
0010	LD	2002
0011	LDA	#TM02
0012	CON	
0013	STA	#TM03
0014	CON	
0015	INC	TM02
0016	CON	
0017	INC	TM03
0018	NEXT	
0019	RET	

オペランド

KZ－10, 16	KZ24, 40, 80
#00001 ～#65535	#00001 ～#65535
DM0000 ～DM0999	DM0000 ～DM1999
TM00～TM29	TM00～TM29

入力方法



ERROR //

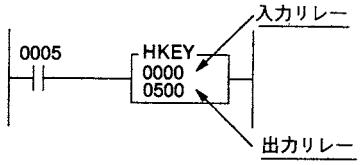
●カイロエラー：FOR の数と NEXT の数が同一でない。
● FOR～NEXT エラー：ネスティングが9重以上となった。

16キー入力：HKEY

FUN　SET 1　LD 7　:　HKEY nnnn mmmm

16キーデータを時分割で読み込み、特殊補助リレー2900～2915にキーの状態を入れます。

(例)



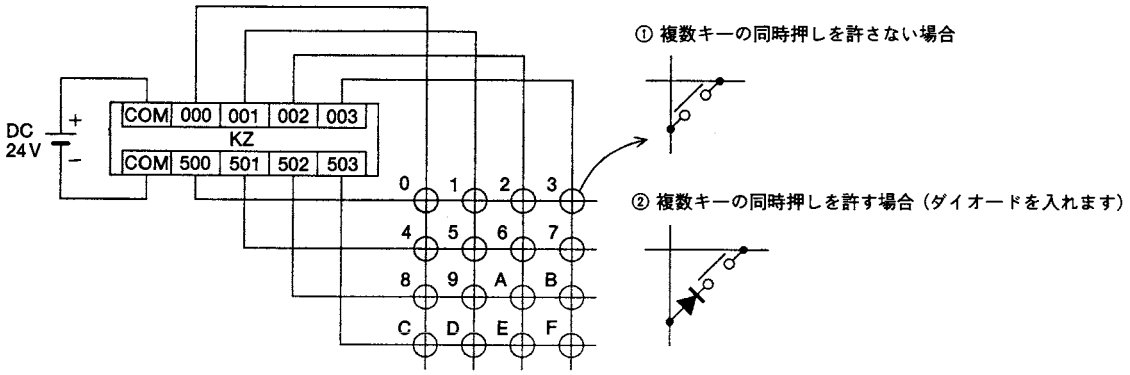
コーディング

行番号	命令	オペランド
0000	LD	0005
0001	HKEY	0000 0500

●入力リレー0005がONすると、16進データを読み込みます。

■解説

●16キー入力は、出力4点、入力4点を使用して下図のように配線します。



●入力リレーと出力リレーは命令で指定した番号から4個ずつ使用します。

●入力処理を時分割で行っていますので、入力処理を終了するまで8スキャン(ただし、スキャンタイムが13msec以下のときは約100msec)を要します。

●HKEY命令は、入力したキーと対応した特殊補助リレー(2900～2915)をONします。

リレー番号	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911	2912	2913	2914	2915
対応キー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

●特殊補助リレーの状況により、複数キー入力の許可や禁止、キー読み込み終了のタイミングがわかります。

リレー番号	2814	2815
内　容	ONのとき、複数キーの入力を禁止します。	キーの読み込み終了時にONします。

● HKEY命令はプログラム中に1回しか使用できません。

●HKEY命令はスキャンタイムごとに出力をON－OFFさせるため、トランジスタ出力または、MOS－FET出力仕様のKZをお使いください。

●2814がON時に、スキャン中同時入力が一度でもあると2815がONしたときに2900～2915はすべてOFFします。

オペランド

入力リレー	0000～0415
出力リレー	0500～0915

入力方法

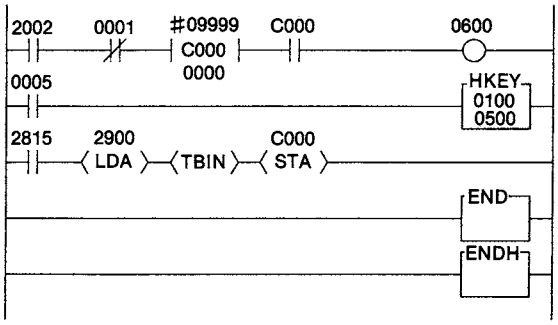
FUN → SET 1 → LD 7 → 実行逆検索 → 入力リレー → ➡ → 出力リレー → 実行逆検索

ERROR !!

●カイロエラー：入力回路に間違いがある。(例：入力リレーが接続されていない)

HKEY命令の応用例

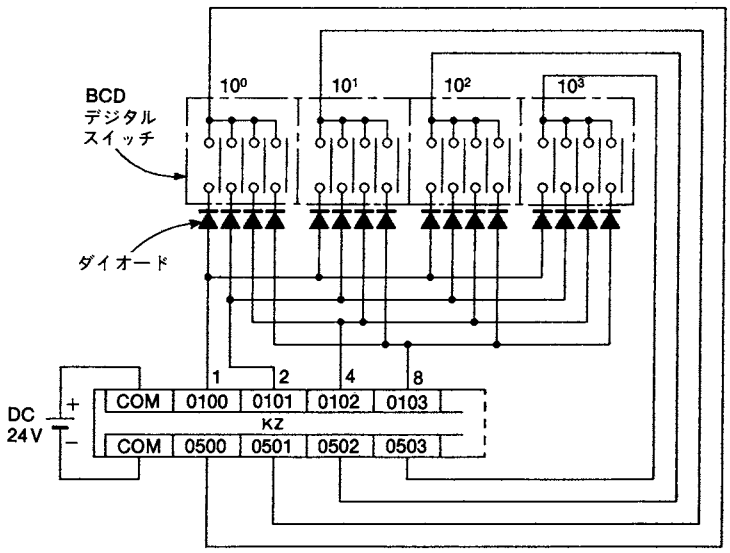
（例）BCDデジタルスイッチによる内部カウンタの設定値変更



コーディング

行番号	命令	オペランド
0000	LD	2002
0001	ANB	0001
0002	C	000 #09999 0000
0003	CON	
0004	AND	C000
0005	OUT	0600
0006	LD	0005
0007	HKEY	0100 0500
0008	LD	2815
0009	LDA	2900
0010	CON	
0011	TBIN	
0012	CON	
0013	STA	C000
0014	END	
0015	ENDH	

●0005をONすることによりカウンタの設定値をデジタルスイッチで設定した値へ変更します。
●TBINについてはP5−45をご覧ください。
●LDA、STAについてはP5−29をご覧ください。
●デジタルスイッチとKZは下図のように接続します。(ダイオードは必ず入れてください)



●設定値は#0000～#9999です。
●特殊補助リレー2815がONした時にカウンタの設定値を変更します。

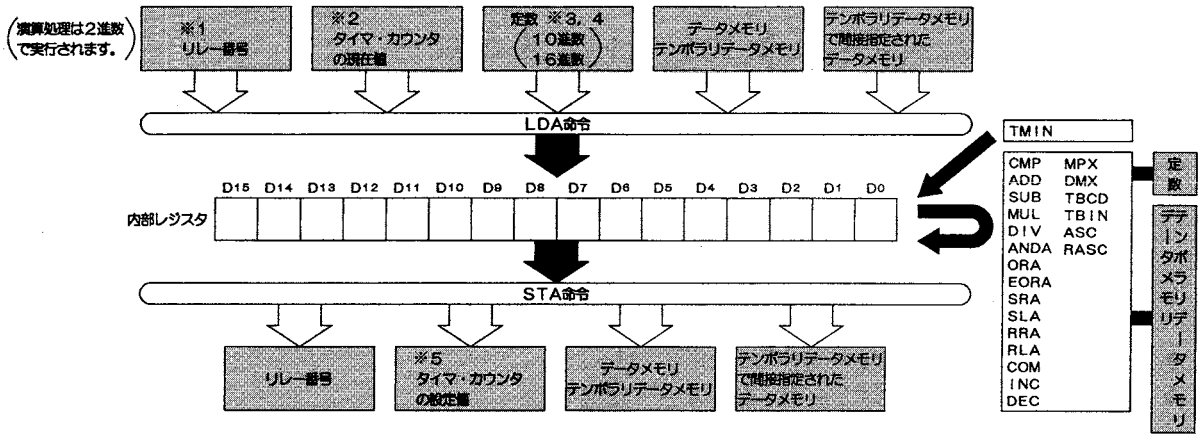
内部レジスタ

KZでは演算命令（すなわちデータのやりとり）は、演算処理速度向上のためすべて「内部レジスタ」を介して実行されます。
ここではまずこの「内部レジスタ」について解説いたします。

内部レジスタとは？

内部レジスタは2進16桁のデータ格納部です。
2進16桁、つまりON－OFF情報（1-0情報）を格納できる小部屋（ビット）が16個あるわけです。
ただしこの内部レジスタはKZが演算処理をする上で必要なだけで、内部レジスタという命令語やオペランドが存在するわけではありません。
したがってプログラミングに際して特に意識する必要はないでしょう。

内部レジスタと演算命令



※1 指定したリレーを先頭に16個分のON－OFFデータが内部レジスタに入力されます。

※2 タイマ・カウンタの現在値（10進数＝#＊＊＊＊＊）は、自動的に2進数に変換されて内部レジスタに入力されます。

※3 定数（10進数＝#＊＊＊＊＊、16進数＝$＊＊＊＊）は自動的に2進数に変換されて内部レジスタに入力されます。

※4 入力できる定数の範囲は次のとおりです。

10進数……#00000～#65535
16進数……$0000～$FFFF

定数の入力には [@ #$] キーを使用します。
（10進数は1回、16進数は2回 [@ #$] キーを押します。）

※5 内部レジスタの値(2進数)は自動的に10進数(#＊＊＊＊＊)に変換されてタイマ・カウンタの設定値に入力されます。

補足説明

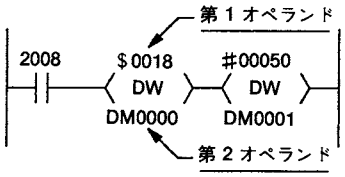
●内部レジスタとデータメモリの内部構造は同じですが、データメモリは電源ＯＦＦ時もデータを保存できるのに対し、内部レジスタは保存できません。

●内部レジスタはひとつのプログラムの中でも演算命令を実行する毎に書き替わります。

●内部レジスタを直接モニタすることはできません。

データメモリライト　:DW	DW接点 : nnnn / DW / DMxxxx	データメモリに直接定数を書き込みます。

(例)



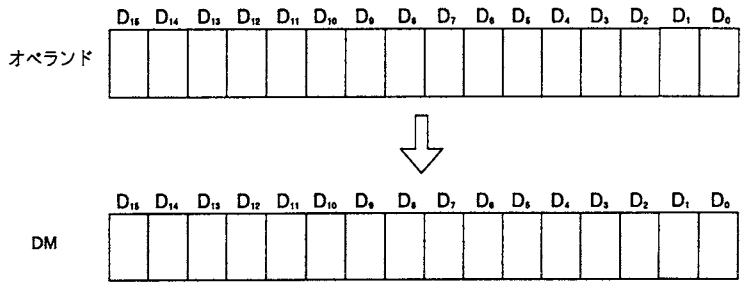
コーディング

行番号	命令	オペランド
0000	LD	2008
0001	DW	$0018 DM0000
0002	CON	
0003	DW	#00050 DM0001

●運転開始時 DM000 へ $00018 を DM0001 へ #00050 を書き込みます。

■解説

●データメモリに内部レジスタを介さずに直接値を書き込みます。



● DW 命令を実行しても内部レジスタは変化しません。
● DW 命令は DM を初期化するときに使用すると便利です。

オペランド

KZ−10, 16	KZ24, 40, 80
nnnn #00000 ～#65535 $0000～$FFFF DMxxxx DM0000 ～DM0999	nnnn #00000 ～#65535 $0000～$FFFF DMxxxx DM0000 ～DM1999

入力方法

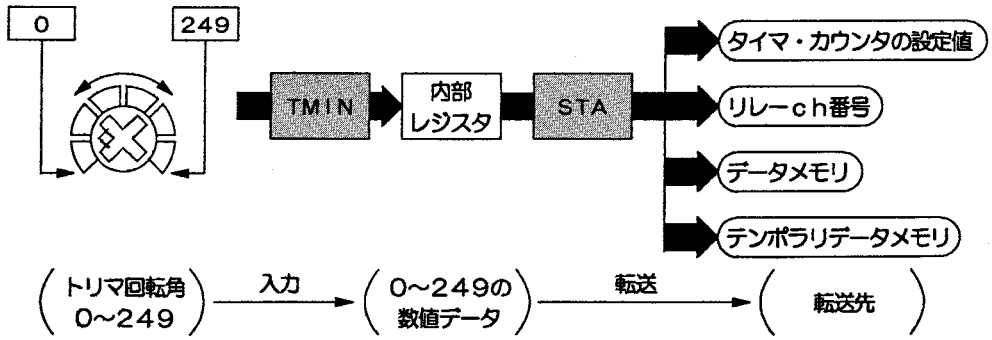
DW接点 → 第1オペランド → ➡ → 第2オペランド → 実行 逆検索

ERROR //

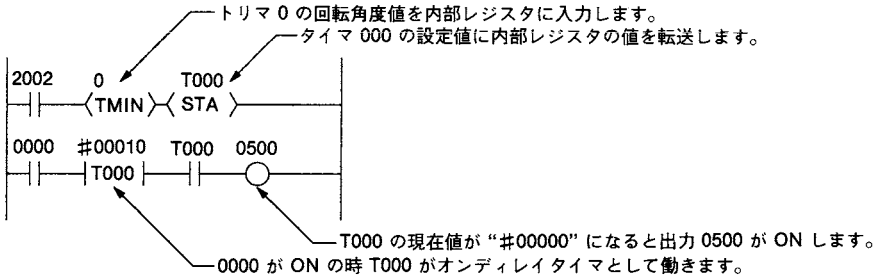
●カイロエラー：入力回路に間違いがある。(例：入力リレーが接続されていない)

トリマ・イン　TMIN (トリマ設定値入力)　@TMIN	FUN ANB 5 RES 0 : X TMIN @ # $ FUN ANB 5 RES 0 : ↑ X TMIN	トリマの回転角を0～249の数値に変換して、内部レジスタへ入力します。

TMIN は、トリマ回転角（0～249）の内部レジスタへの転送命令ですが、さらにその内部レジスタの値をタイマ・カウンタ・リレー番号・データメモリなどに転送（STA 命令を使用します）すると、転送先の数値データをプログラムの変更なしに可変することができます。



（例）T000 の設定値を可変できるようにします。



■解説

- 例では T000 の設定値は0～24.9秒で可変できます。
- 0～249以外の数値に変更するためには四則演算命令が必要です。(P5－36 例 2 参照)
- トリマ回転角（0～249）を確認する場合は、「トリマ設定値読出し」(FUN73) を使用します。
- @TMIN 命令は、入力リレーの立ち上がり時に 1 回だけ実行します。

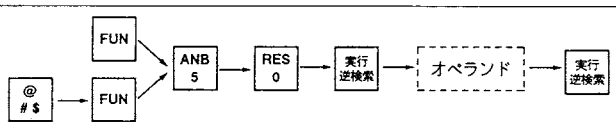
コーディング

行番号	命令	オペランド
0000	LD	2002
0001	TMIN	0
0002	CON	
0003	STA	T000
0004	LD	0000
0005	TMR	000 #00010
0006	CON	
0007	AND	T000
0008	OUT	0500

オペランド

KZ-10, 16	0
KZ-24, 40, 80	0, 1

入力方法



ERROR //

- カイロエラー：入力回路に間違いがある。(例：入力リレーがない)

ロードA： LDA / @LDA

FUN 2 3 : nnnn LDA
@ FUN 2 3 : nnnn LDA

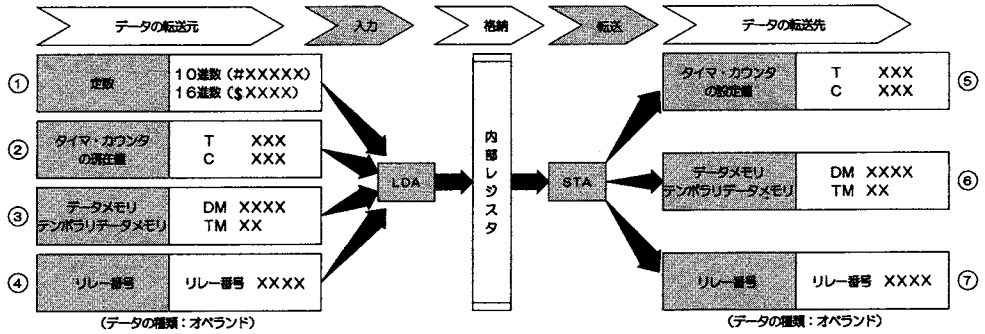
オペランドで指定された値を内部レジスタに入力します。

ストアA： STA / @STA

FUN 4 2 : nnnn STA
@ FUN 4 2 : nnnn STA

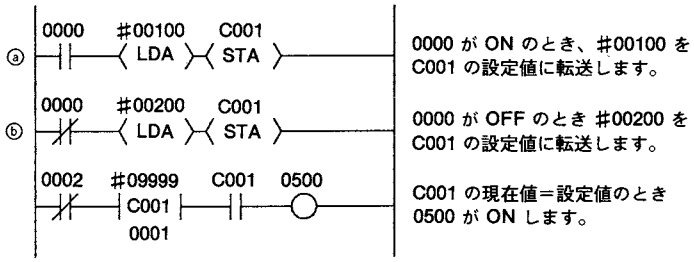
内部レジスタの値をオペランドで指定された先に転送します。

LDA は各データを内部レジスタに入力し、STA は内部レジスタの値を他のデータ格納場所に転送します。



内部レジスタについて詳しくは P5−26 をご覧ください。

(例 1)：カウンタの設定値を変更します。(①→⑤)



コーディング

行番号	命令	オペランド
0000	LD	0000
0001	LDA	#00100
0002	CON	
0003	STA	C001
0004	LDB	0000
0005	LDA	#00200
0006	CON	
0007	STA	C001
0008	LDB	0002
0009	C	001 #09999 0001
0010	CON	
0011	AND	C001
0012	OUT	0500

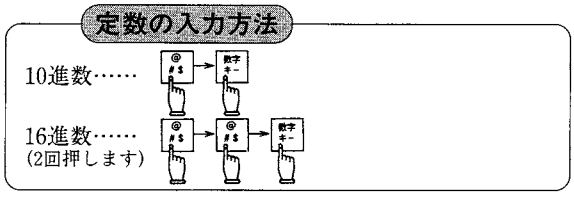
■解説

- ⓐ部では、0000がON のとき、定数 #00100 は内部レジスタを経由して C001の設定値となります。
- ⓑ部では、0000がOFF のとき、定数 #00200 は内部レジスタを経由して C001の設定値となります。
- C001 の設定値は #09999 となっていますが、例の場合、0000 の切替で設定値を変更するためどんな値でもかまいません。
  変更後は、次に変更されるまでその値が保存されます。
- 現在値＞設定値の時に STA 命令を実行した場合、現在値は設定値へ変更されます。
- 電源を切って再起動した時、タイマ・カウンタの設定値は元の値に戻ります。
- LDA 命令で入力できる定数の範囲は次のとおりです。

  10進数……#00000～#65535
  16進数……$0000～$FFFF

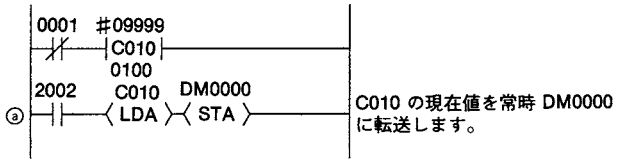
- 定数の入力には [@ #$] キーを使用します。

定数の入力方法



LDA命令：STA命令

(例 2)：カウンタの現在値をデーターメモリに転送します。(②→⑥)



コーディング

行番号	命令	オペランド
0000	LDB	0001
0001	C	010 #09999 0100
0002	LD	2002
0003	LDA	C010
0004	CON	
0005	STA	DM0000

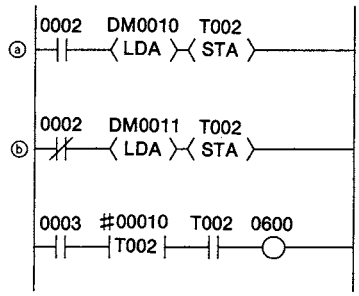
■解説

●ⓐ部ではC010の現在値が、運転中常時内部レジスタを経由して DM0000 に転送・格納されています。

ワンポイント

●データメモリ(DM ××××)の書き込みには [END DM] キーを使用します。
●データメモリの現在値は「ジョウタイモニタ」で見ることができます。
[@ #$] キーで10進、16進、アスキー文字列、2進数が表示出来ます。

(例 3)：データメモリの値をタイマの設定値に転送します。(③→⑤)



0002が ON のとき、DM0010 の値を T002 の設定値に転送します。

0002 が OFF のとき、DM0011 の値を T002 の設定値に転送します。

T002 の現在値が 0 のとき、0600 が ON します。

コーディング

行番号	命令	オペランド
0000	LD	0002
0001	LDA	DM0010
0002	CON	
0003	STA	T002
0004	LDB	0002
0005	LDA	DM0011
0006	CON	
0007	STA	T002
0008	LD	0003
0009	TMR	002 #00010
0010	CON	
0011	AND	T002
0012	OUT	0600

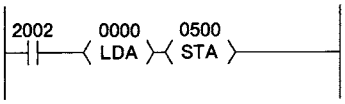
■解説

●ⓐ部では、0002が ON のとき、DM0010 の値が内部レジスタを経由して T002 の設定値となります。
●ⓑ部では、0002 が OFF のとき、DM0011 の値が内部レジスタを経由して T002 の設定値となります。
●T002 の設定値は#00010となっていますが、例の場合、0002の切替で設定値を変更するため、どんな値でもかまいません。変更後は次に変更されるまでその設定値が保存されます。
●現在値＞設定値の時に STA 命令を実行した場合、現在値は設定値へ変更されます。
●電源を切って再起動した時、タイマ・カウンタの設定値は元の値に戻ります。

ワンポイント

●データメモリへの定数の書き込みは、[FUN] [LD 7] [ANL 2] で行います。

(例 4)：入力リレーのデータ（ON－OFF 状態）を出力リレーにそのまま出力します。（④→⑦）

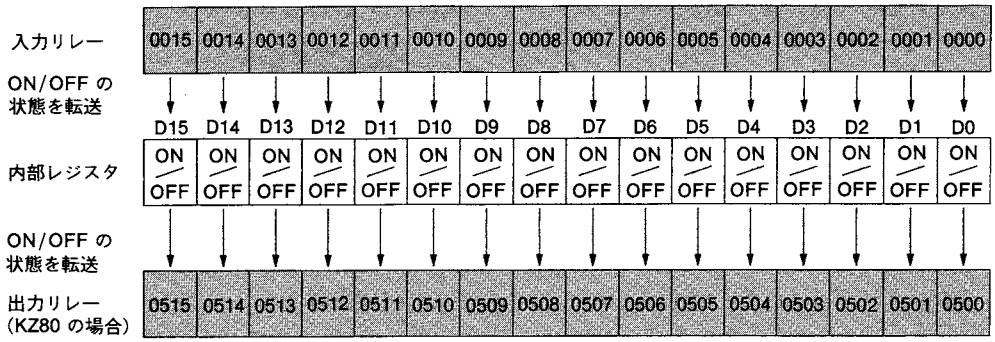


コーディング

行番号	命令	オペランド
0000	LD	2002
0001	LDA	0000
0002	CON	
0003	STA	0500

■解説

●入力リレー 0000～0015 の ON-OFF データを出力リレー 0500～0515 にそのまま転送します。
このとき内部レジスタは次のようになります。ch の入出力リレーが 16 点ないものは規定された最大の入出力を取り込みもしくは出力します。

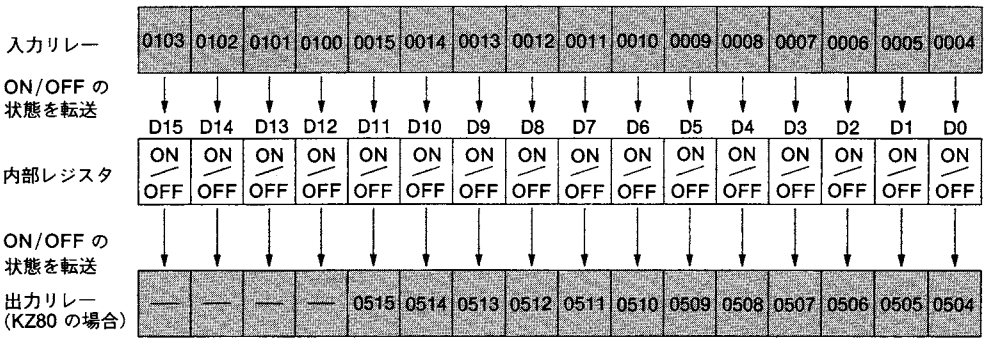


●内部レジスタではデータはすべて2進数として扱われます。(ON → 1,OFF → 0) たとえば、上記の例で入力リレー0000,0002,0003がON、その他はOFFだとすれば内部レジスタでは2進数ですから

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

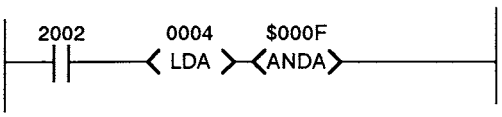
となり、10進数で表すと $2^0+2^2+2^3=1+4+8=13$ (10進数) となります。
このとき出力リレーの0500,0502,0503がONし、他は全てOFFとなります。

●上記例でLDA0004、STA0504とした場合、内部レジスタは次のようになります。



●LDA 命令では ch (チャンネル) をまたいで内部レジスタに取り込むことは可能ですが、STA 命令では ch (チャンネル) をまたいで出力をすることはできません。

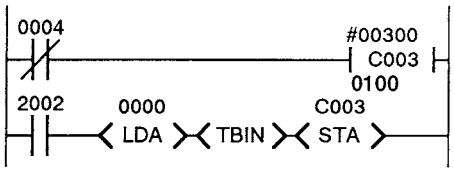
●例えば入力の 0004～0007 の ON/OFF のみを内部レジスタのビット 0～3 に入れる場合は次のようにします。
(0008～0015、0100～0103 の入力をキャンセルします) ANDA 命令 → P5－37



コーディング

行番号	命令	オペランド
0000	LD	2002
0001	LDA	0004
0002	CON	
0003	ANDA	$ 000F

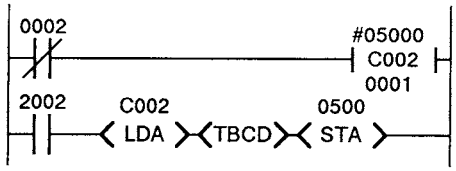
(例 5)：BCD デジタルスイッチ 4 ケタ (0000～0015) のデータをバイナリーに変換し力ウンタの設定値に転送します。(④→⑤)　TBIN 命令→P5－45



コーディング

行番号	命令	オペランド
0000	LDB	0004
0001	C	003 #00300 0002
0002	LD	2002
0003	LDA	0000
0004	CON	
0005	TBIN	
0006	CON	
0007	STA	C003

(例 6)：カウンタの現在値のデータ 4 ケタを BCD に変換し出力 0500～0515 に出力します。(②→⑦)
TBCD 命令→P5－45



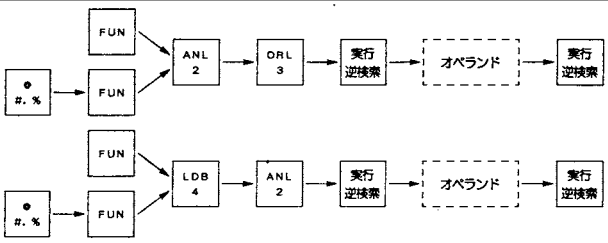
コーディング

行番号	命令	オペランド
0000	LDB	0002
0001	C	002 #05000 0001
0002	LD	2002
0003	LDA	C002
0004	CON	
0005	TBCD	
0006	CON	
0007	STA	0500

オペランド

	KZ-10, 16	KZ24, 40, 80
L D A	0000～2915 T000～T063 C000～C063 CTH0～CTH1 DM0000 ～DM0999 TM00～TM31 #00000 ～#65535 $ 0000～$ FFFF #TM00～#TM29	0000～6915 T000～T119 C000～C119 CTH0～CTH1 DM0000 ～DM1999 TM00～TM31 #00000 ～#65535 $ 0000～$ FFFF #TM00～#TM29
S T A	0500～1915 2100～2915 T000～T063 C000～C063 CTC0～CTC3 DM0000 ～DM0999 TM00～TM29 #TM00～#TM29	0500～1915 2100～6915 T000～T119 C000～C119 CTC0～CTC3 DM0000 ～DM1999 TM00～TM29 #TM00～#TM29

入力方法



ERROR //

●カイロエラー：入力回路に間違いがある。
(例：入力がない、入力が多すぎる)

コンペア（比　較）：CMP
：@CMP

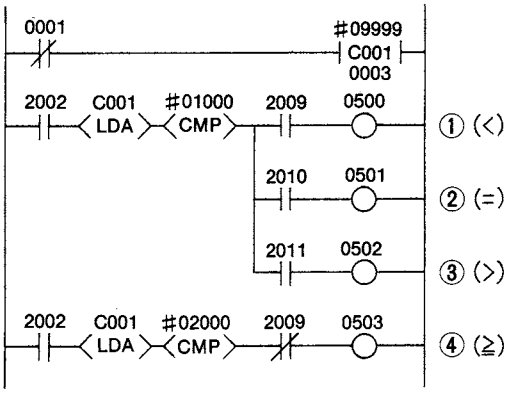
FUN RES 0 LDB 4 ： nnnn CMP
@ #$ FUN RES 0 LDB 4 ： nnnn CMP

内部レジスタの値をオペランドで指定された値と比較します。

CMP 命令は内部レジスタの値とオペランドで指定された値 (CMP の値) とを比較し、
特殊補助リレー (2009、2010、2011) を以下のように ON－OFF させます。

	2009	2010	2011
(内部レジスタ) < (オペランド)	ON	OFF	OFF
(内部レジスタ) = (オペランド)	OFF	ON	OFF
(内部レジスタ) > (オペランド)	OFF	OFF	ON

(例) カウンタの現在値と CMP の値とを比較します。



カウンタの現在値が 1000 と比べて
① 小さいとき出力 0500 が ON します。
② 等しいとき出力 0501 が ON します。
③ 大きいとき出力 0502 が ON します。
④ カウンタの現在値が 2000 以上になると出力 0503 が ON します。

コーディング

行番号	命令	オペランド
0000	LDB	0001
0001	C	001 #09999 0003
0002	LD	2002
0003	LDA	C001
0004	CON	
0005	CMP	#01000
0006	CON	
0007	MPS	
0008	AND	2009
0009	OUT	0500
0010	MRD	
0011	AND	2010
0012	OUT	0501
0013	MPP	
0014	AND	2011
0015	OUT	0502
0016	LD	2002
0017	LDA	C001
0018	CON	
0019	CMP	#02000
0020	CON	
0021	ANB	2009
0022	OUT	0503

■解説

- 例のように特殊補助リレー (2009～2011) は、CMP 命令の直後の状態を取り込むようにプログラムします。
- CMP 命令は内部レジスタの内容を変化させることはありません。
- オペランドに書かれた定数 (#＊＊＊＊＊、$＊＊＊＊)、データメモリはすべて自動的に 2 進数に変換され、内部レジスタ (2 進数) と比較されます。

オペランド

KZ－10, 16	KZ24, 40, 80
DM0000 ～DM0999	DM0000 ～DM1999
TM00～TM31	TM00～TM31
#00000～#65535	#00000～#65535
$0000 ～$FFFF	$0000 ～$FFFF
#TM00～#TM29	#TM00～#TM29

入力方法

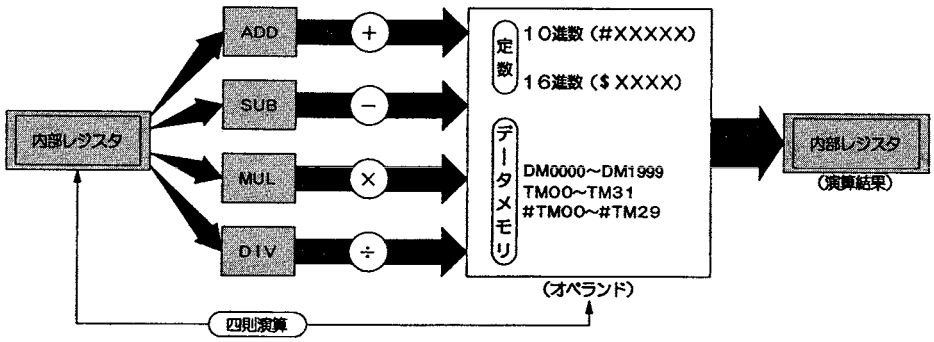
FUN → RES 0 → LDB 4 → 実行逆検索 → オペランド → 実行逆検索
@ #$ → FUN

ERROR //

- カイロエラー：入力回路に間違いがある。(例：入力がない。入力が多すぎる)

アッド（加算）	: ADD	FUN RES 0 RES 0 : nnnn ADD	内部レジスタの値にオペランドで指定された値を加算し、結果を内部レジスタに再度入力します。
	: @ADD	@ #$ FUN RES 0 RES 0 : nnnn ADD	
サブ（減算）	: SUB	FUN LDB 4 ORB 6 : nnnn SUB	内部レジスタの値からオペランドで指定された値を減算し、結果を内部レジスタに再度入力します。
	: @SUB	@ #$ FUN LDB 4 ORB 6 : nnnn SUB	
マルチ（乗算）	: MUL	FUN ANL 2 AND 8 : nnnn MUL	内部レジスタの値をオペランドで指定された値で乗算し、結果を内部レジスタに再度入力します。
	: @MUL	@ #$ FUN ANL 2 AND 8 : nnnn MUL	
デバイド（除算）	: DIV	FUN SET 1 SET 1 : nnnn DIV	内部レジスタの値をオペランドで指定された値で除算し、結果を内部レジスタに再度入力します。
	: @DIV	@ #$ FUN SET 1 SET 1 : nnnn DIV	

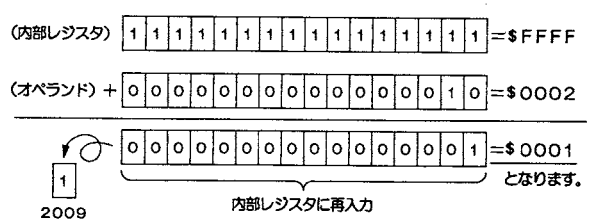
ADD・SUB・MUL・DIV の各命令は内部レジスタとオペランドの値との四則演算命令です。



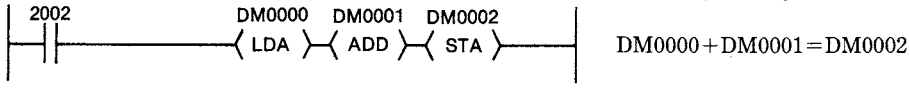
■解説

ADD（加算）

- 演算の結果、16ビットを越える分($FFFF、#65535を越える分)はオーバーフローし、下位16ビットのみ内部レジスタに再入力されます。
- オペランドに#TM00～#TM29を書いたときは、TM＊＊によって間接指定されたデータメモリの内容を内部レジスタに加算します。



（例）DM0000 に DM0001 のデータを加算しその答えを DM0002 へ入れます。



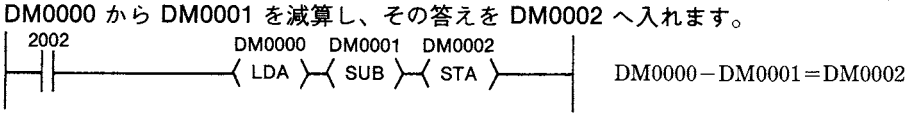
DM0000＋DM0001＝DM0002

SUB（減算）

- オペランドに#TM00～#TM29を書いたときは、TM＊＊によって間接指定されたデータメモリの内容を内部レジスタより減算し、演算結果を内部レジスタへ再入力します。

●間接指定については P5－57 をご覧ください。

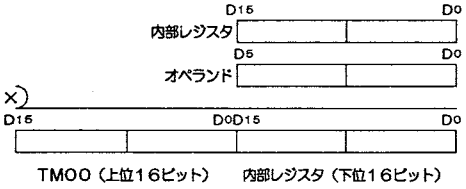
（例）DM0000 から DM0001 を減算し、その答えを DM0002 へ入れます。



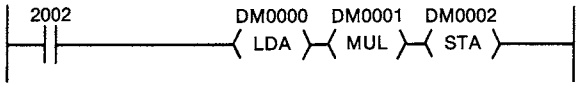
DM0000－DM0001＝DM0002

MUL（乗算）

●内部レジスタの内容とオペランドの値を乗算した結果の32ビットの上位16ビットをTM00に、下位16ビットを内部レジスタへ再入力します。



（例）DM0000とDM0001を乗算し、その答えをDM0002へ入れます。



DM0000×DM0001＝DM0002

DIV（除算）

●TM00を上位16ビットに、内部レジスタの値を下位16ビットとした32ビットの値（2進数）をオペランドで指定した値で除算し、商の上位16ビットをTM00に、下位16ビットを内部レジスタに再入力し、余りをTM01に格納します。

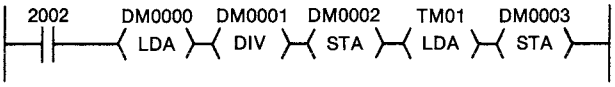
TM00（上位16ビット）　内部レジスタ（下位16ビット）　÷　オペランド

＝　TM00　内部レジスタ　----　TM01

商（上位16ビット）　商（下位16ビット）　余り

比較演算結果に対する内部補助リレーの変化→P5−54

（例）DM0000をDM0001で除算し、その答えをDM0002へ、余りをDM0003へ入れます。

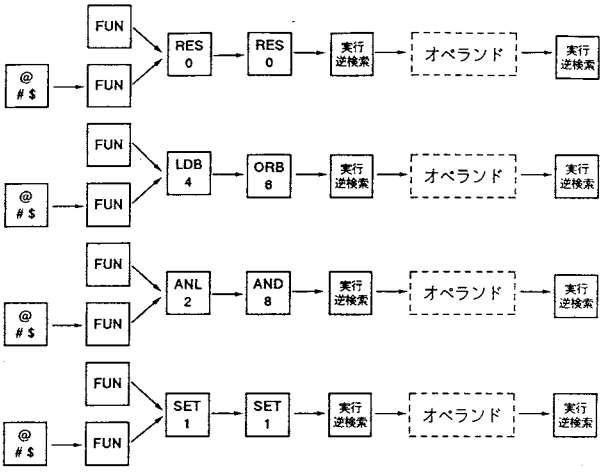


DM0000÷DM0001＝DM0002…DM0003

オペランド

KZ−10, 16	KZ24, 40, 80
DM0000 ～DM0999	DM0000 ～DM1999
TM00～TM31	TM00～TM31
#00000 ～#65535	#00000 ～#65535
$0000～$FFFF	$0000～$FFFF
#TM00～#TM29	#TM00～#TM29

入力方法

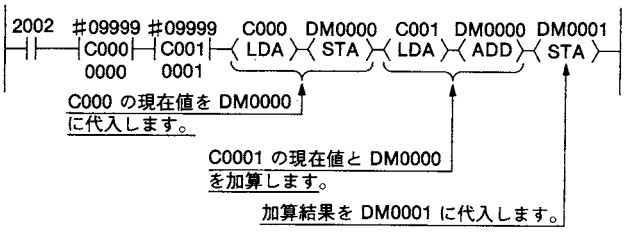


ERROR //

●カイロエラー：入力回路に間違いがある。（例：入力がない、入力が多すぎる）

ADD 命令・SUB 命令・MUL 命令・DIV 命令の応用例

（例 1）：2 つのカウンタの現在値を合計してデータメモリに転送します。



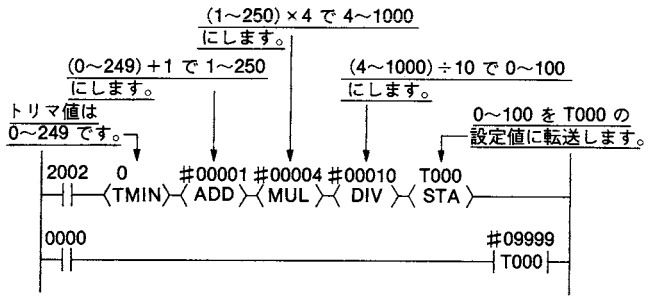
コーディング

行番号	命令	オペランド
0000	LD	2002
0001	C	000 #09999 0000
0002	CON	
0003	C	001 #09999 0001
0004	CON	
0005	LDA	C000
0006	CON	
0007	STA	DM0000
0008	CON	
0009	LDA	C001
0010	CON	
0011	ADD	DM0000
0012	CON	
0013	STA	DM0001

■ 解説

●(例 1) のように内部レジスタ値を一時的に保存しておきたい場合はデータメモリ、またはテンポラリデータメモリを使用します。

（例 2）：アナログタイマのトリマ可変幅 (0～249) を 0～100 にして T000 の設定値とします。



コーディング

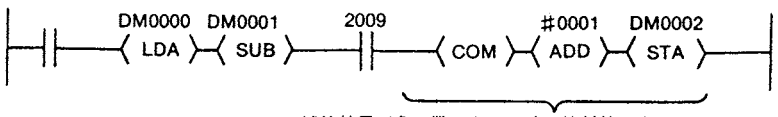
行番号	命令	オペランド
0000	LD	2002
0001	TMIN	0
0002	CON	
0003	ADD	#00001
0004	CON	
0005	MUL	#00004
0006	CON	
0007	DIV	#00010
0008	CON	
0009	STA	T000
0010	LD	0000
0011	TMR	000 #09999

■ 解説

●(例 2) のプログラムで、T000 の設定値は #00000～#00100 即ち 0 秒～10 秒の範囲をトリマで可変できます。

負の数と絶対値

KZ では、負の数は 2 の補数で表現します。
減算命令 (SUB) 実行時に答えが負の数となると、特殊補助リレー 2009 が ON し、内部レジスタへは補数が入ります。
絶対値を求めるには次のプログラムにより行ないます。



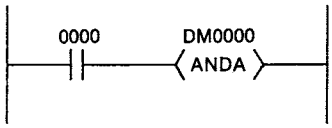
減算結果が負の数になった時に絶対値を求めるプログラムの例

アンドA（論理積）：ANDA / @ANDA

FUN RES 0 SET 1 ： nnnn ANDA

@ #$ FUN RES 0 SET 1 ： nnnn ANDA

内部レジスタの値とオペランドで指定された値を16ビットの各ビットごとにANDし、結果を内部レジスタに再度入力します。



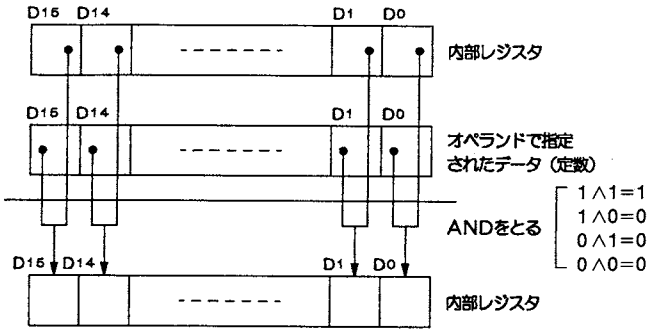
コーディング

行番号	命令	オペランド
0000	LD	0000
0001	ANDA	DM0000

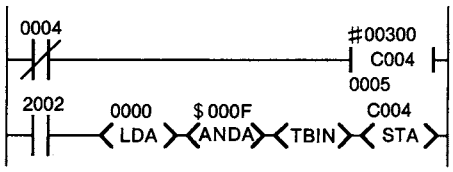
■解説

●内部レジスタの内容とオペランドで指定されたデータメモリ（16ビット）または定数（16ビット）をビットごとにANDし、内部レジスタにもどします。

●演算の結果内部レジスタの内容が「0」になった時に、特殊補助リレー 2010 が ON します。



（例）



コーディング

行番号	命令	オペランド
0000	LDB	0004
0001	C	004 #00300 0005
0002	LD	2002
0003	LDA	0000
0004	CON	
0005	ANDA	$000F
0006	CON	
0007	TBIN	
0008	CON	
0009	STA	C004

■解説

●BCDデジタルスイッチ1ケタ (0000～0003) のON/OFFデータのみを取り込みバイナリーに変換しカウンタの設定値に転送します。(0004～0015の入力は無視します)

オペランド

KZ－10, 16	KZ24, 40, 80
0000～2915	0000～6915
DM0000 ～DM0999	DM0000 ～DM1999
TM00～TM31	TM00～TM31
#00000 ～#65535	#00000 ～#65535
$0000～$FFFF	$0000～$FFFF
#TM00～#TM29	#TM00～#TM29

入力方法

FUN / @ #$ FUN → RES 0 → SET 1 → 実行 逆検索 → オペランド → 実行 逆検索

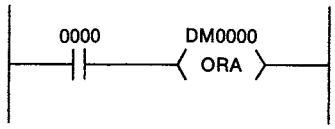
ERROR //

●カイロエラー：入力回路に間違いがある。(例：入力がない、入力が多すぎる)

オ ア A（論理和）：ORA
：@ORA

FUN ORL 3 SET 1 ： nnnn ORA
@ #$ FUN ORL 3 SET 1 ： nnnn ORA

内部レジスタの値とオペランドで指定された値を16ビットの各ビットごとにORし、結果を内部レジスタに再度入力します。



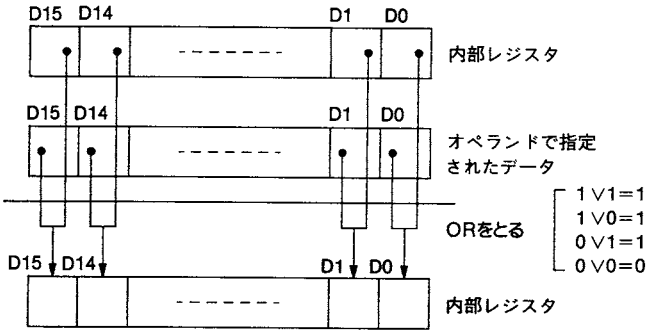
コーディング

行番号	命令	オペランド
0000	LD	0000
0001	ORA	DM0000

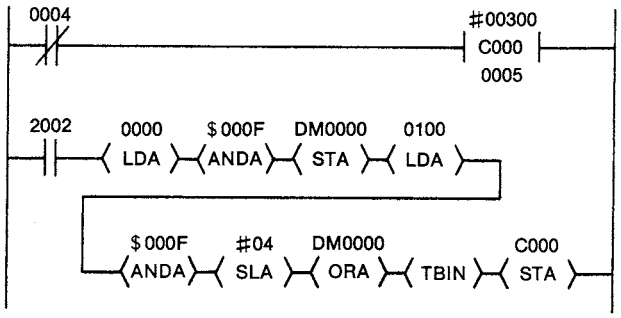
■解説

●内部レジスタの内容とオペランドで指定されたデータメモリ（16ビット）または定数（16ビット）をビットごとにORし、内部レジスタにもどします。

●演算の結果内部レジスタの内容が「0」になった時に、特殊補助リレー 2010 が ON します。



（例）



コーディング

行番号	命令	オペランド
0000	LDB	0004
0001	C	000 #00300 0005
0002	LD	2002
0003	LDA	0000
0004	CON	
0005	ANDA	$ 000F
0006	CON	
0007	STA	DM0000
0008	CON	
0009	LDA	0100
0010	CON	
0011	ANDA	$ 000F
0012	CON	
0013	SLA	#04
0014	CON	
0015	ORA	DM0000
0016	CON	
0017	TBIN	
0018	CON	
0019	STA	C000

■解説

●BCD デジタルスイッチ 2 桁 (1 桁目 0000～0003、2 桁目 0100～0103) の ON/OFF データのみを取り込み、バイナリーに変換しカウンタの設定値に転送します。離れた入力に接続された値を合成します。

オペランド

KZ－10, 16	KZ24, 40, 80
0000～2915	0000～6915
DM0000 ～DM0999	DM0000 ～DM1999
TM00～TM31	TM00～TM31
#00000 ～#65535	#00000 ～#65535
$0000～$FFFF	$0000～$FFFF
#TM00～#TM29	#TM00～#TM29

入力方法

FUN / @ #$ → ORL 3 → SET 1 → 実行逆検索 → オペランド → 実行逆検索

ERROR //

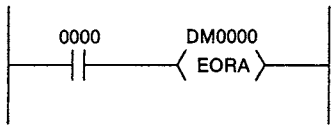
●カイロエラー：入力回路に間違いがある。(例：入力がない、入力が多すぎる)

エクスクルーシブ：EORA
オ ア A
(排他的論理和)：@EORA

FUN SET 1 ANB 5 ： nnnn EORA

@ #$ FUN SET 1 ANB 5 ： ↑ nnnn EORA

内部レジスタの値とオペランドを各々のビットで排他的論理和を行い、内部レジスタに再入力します。



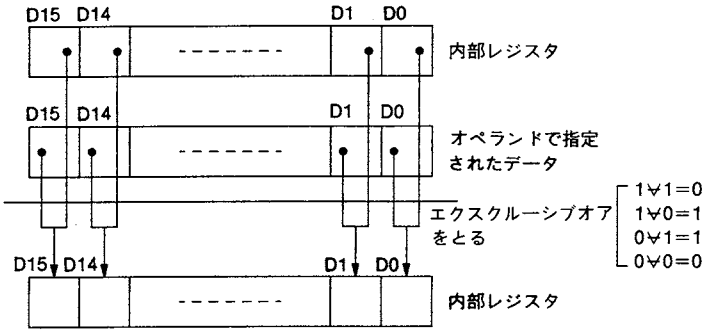
コーディング

行番号	命令	オペランド
0000	LD	0000
0001	EORA	DM0000

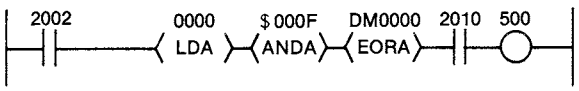
■解説

●内部レジスタの内容とオペランドで指定されたデータメモリ（16ビット）または定数（16ビット）をビットごとに排他的論理和を行い、内部レジスタへもどします。

●演算の結果内部レジスタの内容が「0」になった時に、特殊補助リレー 2010 が ON します。



(例)



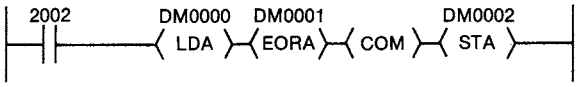
コーディング

行番号	命令	オペランド
0000	LD	2002
0001	LDA	0000
0002	CON	
0003	ANDA	$000F
0004	CON	
0005	EORA	DM0000
0006	AND	2010
0007	OUT	0500

■解説

●入力 (0000～0003) の値とあらかじめ DM0000 へ入っている値が一致すると出力 0500 が ON します。

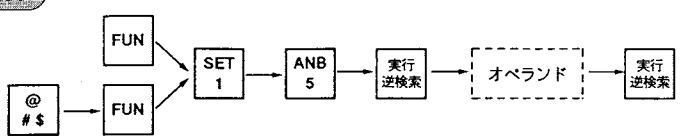
●排他的論理和否定の演算は COM 命令を使うことにより行なえます。



オペランド

KZ－10, 16	KZ24, 40, 80
0000～2915	0000～6915
DM0000 ～DM0999	DM0000 ～DM1999
TM00～TM31	TM00～TM31
#00000 ～#65535	#00000 ～#65535
$0000～$FFFF	$0000～$FFFF
#TM00～#TM29	#TM00～#TM29

入力方法



ERROR //

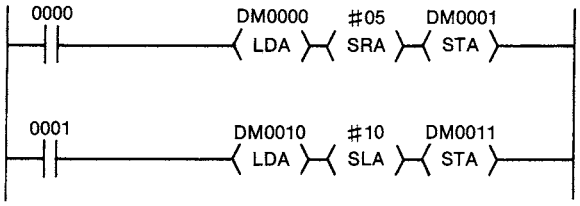
●カイロエラー：入力回路に間違いがある。(例：入力がない、入力が多すぎる)

シフトライトA：SRA　FUN LDB 4 SET 1 : #dd SRA
シフトライトA：@SRA　@ #$ FUN LDB 4 SET 1 : #dd SRA

内部レジスタの内容をオペランド値の回数分、右へシフトします。

シフトレフトA：SLA　FUN LDB 4 RES 0 : #dd SLA
シフトレフトA：@SLA　@ #$ FUN LDB 4 RES 0 : #dd SLA

内部レジスタの内容をオペランド値の回数分、左へシフトします。



コーディング

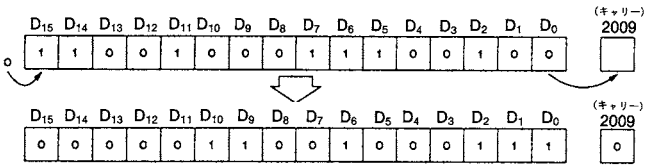
行番号	命令	オペランド
0000	LD	0000
0001	LDA	DM0000
0002	CON	
0003	SRA	#05
0004	CON	
0005	STA	DM0001
0006	LD	0001
0007	LDA	DM0010
0008	CON	
0009	SLA	#10
0010	CON	
0011	STA	DM0011

■解説

●SRA 命令の動作

入力 0000 が ON の時に DM0000 の値を内部レジスタに入れ、オペランドの回数分 (5 回) 右へシフトし、演算結果を DM0001 へ代入します。D_0 の内容はキャリー (2009) へシフトします。

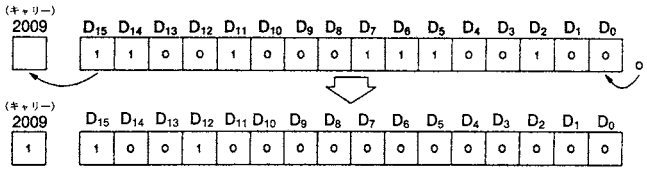
上記例では、入力 0000 が ON している間は、スキャンごとにオペランドの数をシフトします。入力 0000 が OFF → ON になった時だけ実行させるには、微分型の命令を使用します。



●SLA 命令の動作

入力 0001 が ON の時に DM0010 の値を内部レジスタに入れ、オペランドの回数分 (10 回) 左へシフトし、演算結果を DM0011 へ代入します。D_{15} の内容はキャリー (2009) へシフトします。

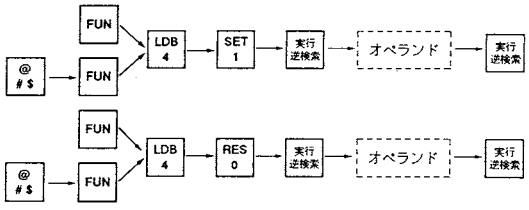
上記例では、入力 0001 が ON している間は、スキャンごとにオペランドの数をシフトします。入力 0001 が OFF → ON になった時だけ実行させるには、微分型の命令を使用します。



オペランド

#01～#16

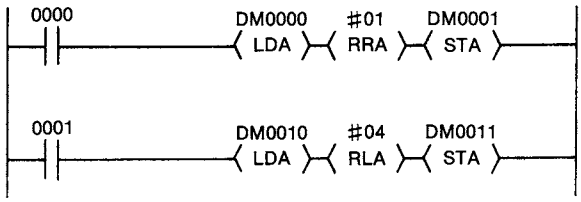
入力方法



ERROR //

●カイロエラー：入力回路に間違いがある。(例：入力がない、入力が多すぎる)

ローテートライトA：RRA ローテートライトA：@RRA	FUN ORL 3 LD 7 : #dd RRA @ #$ FUN ORL 3 LD 7 : #dd RRA	内部レジスタの内容とキャリー(2009)をオペランド値の回数分、右回りに回転します。
ローテートレフトA：RLA ローテートレフトA：@RLA	FUN ORL 3 ANB 5 : #dd RLA @ #$ FUN ORL 3 ANB 5 : #dd RLA	内部レジスタの内容とキャリー(2009)をオペランド値の回数分、左回りに回転します。



コーディング

行番号	命令	オペランド
0000	LD	0000
0001	LDA	DM0000
0002	CON	
0003	RRA	#01
0004	CON	
0005	STA	DM0001
0006	LD	0001
0007	LDA	DM0010
0008	CON	
0009	RLA	#04
0010	CON	
0011	STA	DM0011

■解説

●RRA 命令の動作

入力 0000 が ON の時に DM0000 の値を内部レジスタに入れ、内部レジスタの内容とキャリー(2009) をオペランドの回数分右回りに回転シフトします。

上記例では、入力 0000 が ON している間は、スキャンごとに回転シフトします。

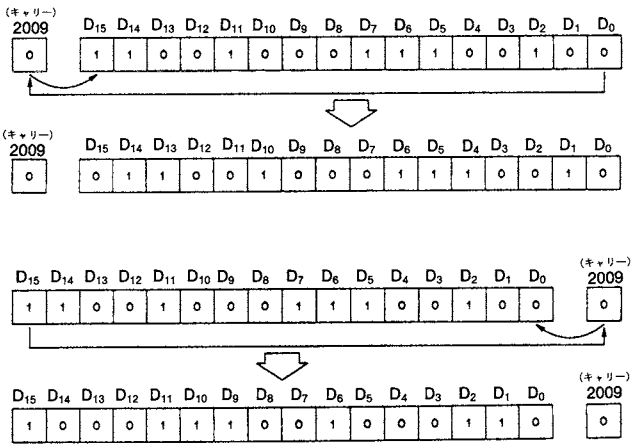
入力 0000 が OFF→ON になった時だけ実行させるには、微分型の命令を使用します。

●RLA 命令の動作・

入力 0001 が ON の時に DM0010 の値を内部レジスタに入れ、内部レジスタの値とキャリー (2009) をオペランドの回数分左回りに回転シフトします。

上記例では、入力 0000 が ON している間は、スキャンごとに回転シフトします。

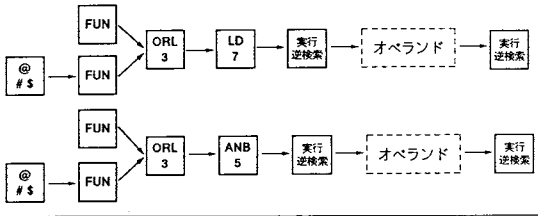
入力 0001 が OFF→ON になった時だけ実行させるには、微分型の命令を使用します。



オペランド

#01～#16

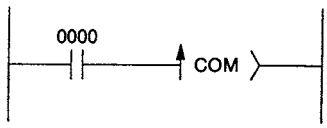
入力方法



ERROR //

●カイロエラー：入力回路に間違いがある。(例：入力がない、入力が多すぎる)

コンプリメント（ビット反転）	：COM ：@COM	FUN RES 0 ANB 5 : COM @ #$ FUN RES 0 ANB 5 : COM	内部レジスタの各ビットを反転します。



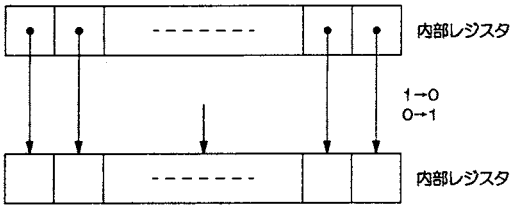
コーディング

行番号	命令	オペランド
0000	LD	0000
0001	@COM	

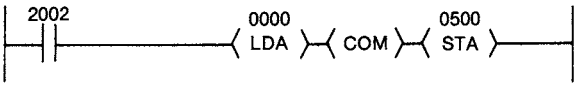
■解説

● @COM 命令の動作
入力0000がOFF → ON に変化するたびに内部レジスタの各ビットを反転し、内部レジスタへ再入力します。

●演算の結果内部レジスタの内容が「0」になった時に、特殊補助リレー 2010 が ON します。



(例)

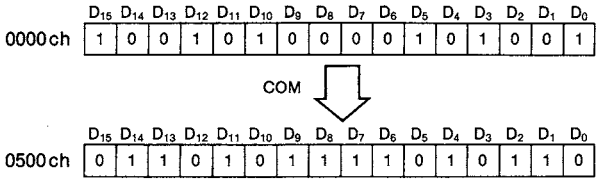


コーディング

行番号	命令	オペランド
0000	LD	2002
0001	LDA	0000
0002	CON	
0003	COM	
0004	CON	
0005	STA	0500

■解説

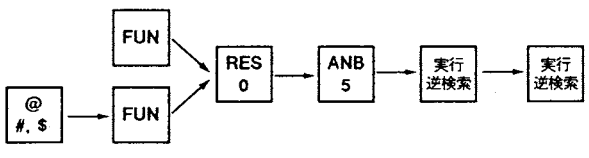
入力 0000～0015 へ入る ON/OFF 信号を反転させて 出力 0500～0515 へ出力します。



オペランド

―――

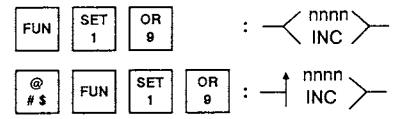
入力方法



ERROR //

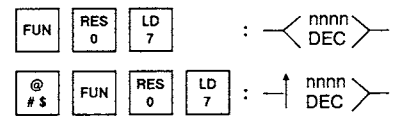
●カイロエラー：入力回路に間違いがある。(例：入力がない、入力が多すぎる)

インクリメント：INC
メモリ　：@INC

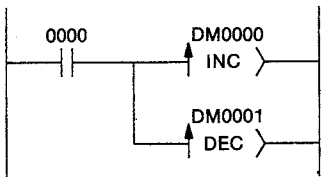


オペランドで指定したメモリデータに1を加算します。

ディクリメント：DEC
メモリ　：@DEC



オペランドで指定したメモリデータから1を減算します。



コーディング

行番号	命令	オペランド
0000	LD	0000
0001	@INC	DM0000
0002	@DEC	DM0001

■解説

● @INC命令の動作

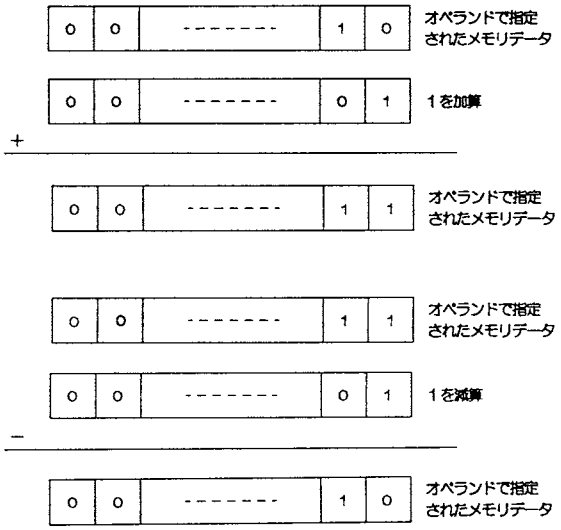
入力0000がOFF → ONに変化するたびにオペランドで指定したメモリの内容に1を加算し、オペランドで指定したメモリへ再入力します。

(上記の例ではDM0000へ1を加算します。DM0000+1 → DM0000)

オペランドで指定したメモリの値が65535のとき、加算結果は00000となり、特殊補助リレー2009がONします。

● @DEC命令の動作

入力0000がOFF → ONに変化するたびにオペランドで指定したメモリの内容から1を減算し、オペランドで指定したメモリへ再入力します。



(上記の例ではDM0001より1を減算します。DM0001−1 → DM0001)

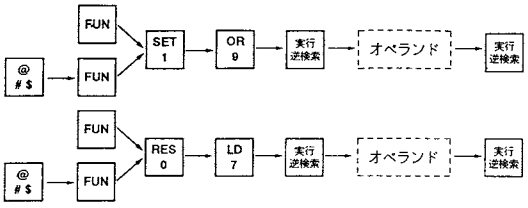
オペランドで指定したメモリの値が00001のとき、減算結果は00000となり、特殊補助リレー2010がONします。

メモリの値が00000のとき、減算結果は65535 ($FFFF)となり、特殊補助リレー2009がONします。

オペランド

KZ−10, 16	KZ24, 40, 80
DM0000 ~DM0999	DM0000 ~DM1999
TM00~TM29	TM00~TM29

入力方法



ERROR //

●カイロエラー：入力回路に間違いがある。(例：入力がない、入力が多すぎる)

マルチプレクス（4→16デコーダ）：MPX

FUN | ANL 2 | LD 7 ： #d MPX

オペランドで指定した内部レジスタの1桁分（4ビット）の数を16ビットデータに変換します。

マルチプレクス（4→16デコーダ）：@MPX

@ #$ | FUN | ANL 2 | LD 7 ： #d MPX

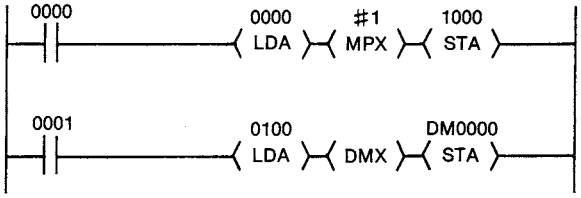
デマルチプレクス（16→4エンコーダ）：DMX

FUN | SET 1 | ANL 2 ： DMX

内部レジスタの16ビットデータの1になっている最上位ビットの数を4ビット変換します。

デマルチプレクス（16→4エンコーダ）：@DMX

@ #$ | FUN | SET 1 | ANL 2 ： DMX



コーディング

行番号	命令	オペランド
0000	LD	0000
0001	LDA	0000
0002	CON	
0003	MPX	#1
0004	CON	
0005	STA	1000
0006	LD	0001
0007	LDA	0100
0008	CON	
0009	DMX	
0010	CON	
0011	STA	DM0000

■解説

●MPX 命令の動作

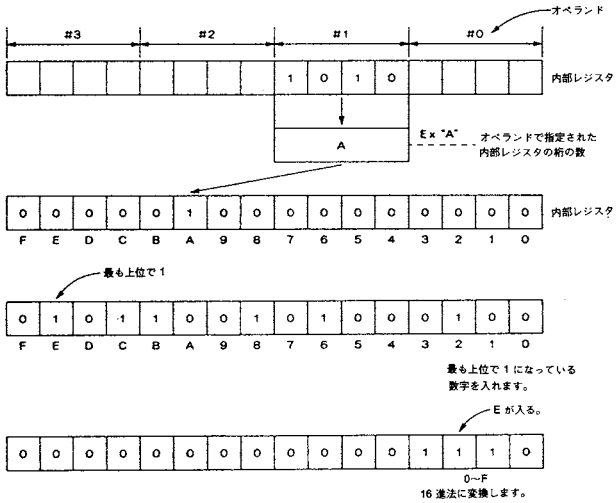
入力 0000 が ON の時に入力 0000～0015 の ON/OFF を内部レジスタに取り込み、MPX のオペランドで指定した内部レジスタの 1 桁分 (入力 0004～0007 の 4 ビット分) の数字 (0～F) をもとに、その数字が示すビット位置のみを 1 に他を 0 にして、内部レジスタへ再入力します。内部レジスタの内容を 1000～1015 へ出力します。

上記の例で入力 0004～0007 の数字が「A」だとすると、内部リレー 1010 が ON します。

●DMX 命令の動作

入力 0001 が ON の時に入力 0100～0115 の ON/OFF を内部レジスタに取り込み、内部レジスタの 16 ビットデータの最上位ビットの数を 4 ビットデータ (0～F) に変換し、内部レジスタへ再入力します。内部レジスタの内容を DM0000 へ代入します。

上記の例で、入力 0114 が ON したとすると、DM0000 の値は「E」となります。



●DMX 命令は内部レジスタのビットがすべて OFF のとき変換しません。このとき、特殊補助リレー 2012 が ON します。

オペランド

#0～#3

入力方法

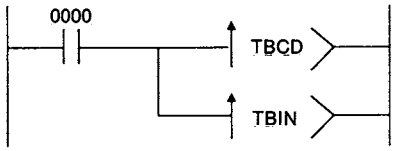
FUN / @ #$ → FUN → ANL 2 → LD 7 → 実行逆検索 → オペランド → 実行逆検索

FUN / @ #$ → FUN → SET 1 → ANL 2 → 実行逆検索 → オペランド → 実行逆検索

ERROR //

●カイロエラー：入力回路に間違いがある。(例：入力がない、入力が多すぎる)

トランスファ：TBCD BCD (BCD変換)：@TBCD	FUN LDB 4 LD 7 : TBCD @ #$ FUN LDB 4 LD 7 : TBCD	内部レジスタの内容（16ビット2進数）を4桁BCDに変換します。
トランスファ：TBIN BIN (BIN変換)：@TBIN	FUN LDB 4 AND 8 : TBIN @ #$ FUN LDB 4 AND 8 : TBIN	内部レジスタの内容(4桁BCD)を16ビット2進数に変換します。



コーディング

行番号	命令	オペランド
0000	LD	0000
0001	@TBCD	
0002	@TBIN	

●TBCD、TBINの例題はP5−32をご覧ください。

■解説

● @TBCD命令の動作

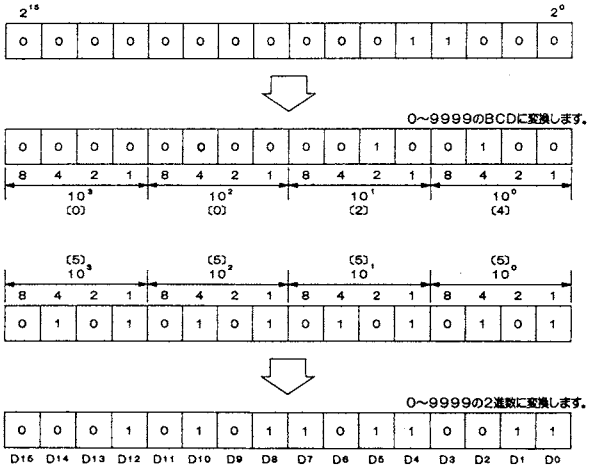
• 入力0000がOFF → ONに変化したとき、内部レジスタの内容（16ビット2進数）を4桁BCDの数字に変換し、内部レジスタへ再入力します。

• 内部レジスタの値が10000以上の場合、変換しません。このとき、特殊補助リレー2012がONします。

● @TBIN命令の動作

• 入力0000がOFF → ONに変化したとき、内部レジスタの内容(4桁BCD)を16ビット2進数に変換し、内部レジスタへ再入力します。

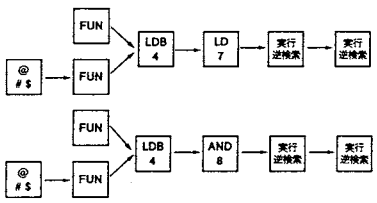
• 内部レジスタの値がBCDでないとき変換しません。このとき、特殊補助リレー2012がONします。



オペランド

—

入力方法



ERROR //

●カイロエラー：入力回路に間違いがある。(例：入力がない、入力が多すぎる)

アスキーコード：ASC
変　換　：@ASC

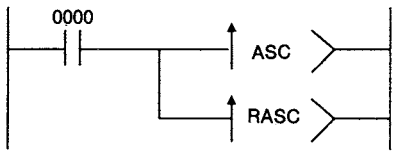
FUN RES 0 ANL 2 ： ASC
@ #$ FUN RES 0 ANL 2 ： ASC

内部レジスタの下位 8 ビットの数値 2 桁を 2 桁の ASCII 文字コードに変換します。

アスキーコード：RASC
逆変換　：@RASC

FUN ORL 3 ANL 2 ： RASC
@ #$ FUN ORL 3 ANL 2 ： RASC

内部レジスタの 2 桁の ASCII 文字コードを数値 2 桁に変換します。



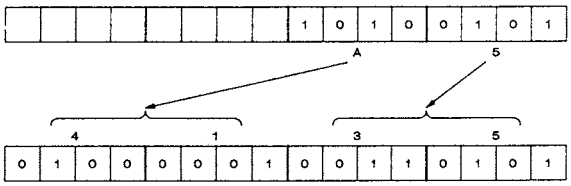
コーディング

行番号	命令	オペランド
0000	LD	0000
0001	@ASC	
0002	@RASC	

■解説

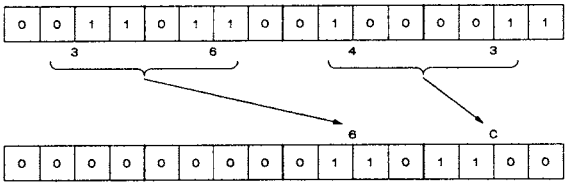
●@ASC 命令の動作

入力 0000 が OFF → ON に変化したとき、内部レジスタの下位 8 ビットの数値 2 桁を 2 桁の ASCII 文字コードに変換し、内部レジスタへ再入力します。



●@RASC 命令の動作

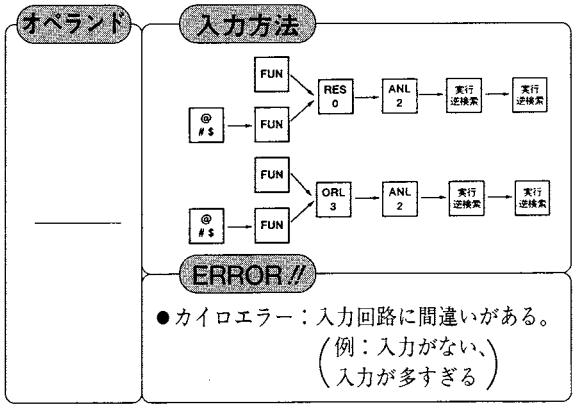
入力 0000 が OFF → ON に変化したとき、内部レジスタの 2 桁の ASCII 文字コードを下位 8 ビットの数値 2 桁に変換し、内部レジスタへ再入力します。



■ ASCII コード変換例

数値	ビット内容				コード	ビット内容							
0	0	0	0	0	$ 30	0	0	1	1	0	0	0	0
1	0	0	0	1	$ 31	0	0	1	1	0	0	0	1
2	0	0	1	0	$ 32	0	0	1	1	0	0	1	0
3	0	0	1	1	$ 33	0	0	1	1	0	0	1	1
4	0	1	0	0	$ 34	0	0	1	1	0	1	0	0
5	0	1	0	1	$ 35	0	0	1	1	0	1	0	1
6	0	1	1	0	$ 36	0	0	1	1	0	1	1	0
7	0	1	1	1	$ 37	0	0	1	1	0	1	1	1
8	1	0	0	0	$ 38	0	0	1	1	1	0	0	0
9	1	0	0	1	$ 39	0	0	1	1	1	0	0	1
A	1	0	1	0	$ 41	0	1	0	0	0	0	0	1
B	1	0	1	1	$ 42	0	1	0	0	0	0	1	0
C	1	1	0	0	$ 43	0	1	0	0	0	0	1	1
D	1	1	0	1	$ 44	0	1	0	0	0	1	0	0
E	1	1	1	0	$ 45	0	1	0	0	0	1	0	1
F	1	1	1	1	$ 46	0	1	0	0	0	1	1	0

●変換前の内部レジスタの上位 8 ビット、下位 8 ビットとも $30～$39 または $41～$46 の範囲のときに変換します。(上位 8 ビットあるいは下位 8 ビットのどちらかが、この範囲にないときは変換しません。このとき特殊補助リレー 2012 が ON します。)

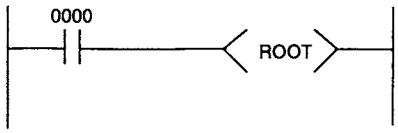


平方根

：ROOT　FUN　ORL 3　ORB 6　：ROOT

：@ROOT　@ #$　FUN　ORL 3　ORB 6　：ROOT

TM00を上位ビット、内部レジスタを下位ビットとする32ビットの平方根を求め、内部レジスタへ入力します。

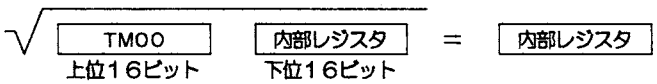


コーディング

行番号	命令	オペランド
0000	LD	0000
0001	ROOT	

■解説

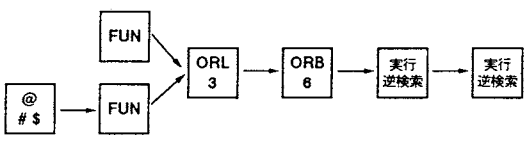
- TM00を上位16ビット、内部レジスタを下位16ビットとする32ビットの値（2進数）のデータの平方根を求め、内部レジスタに再入力します。
- 計算結果は、少数点以下は切り捨てられて内部レジスタへ入力されます。
- 計算結果が0のとき、特殊補助リレー2010をONします。



オペランド

―――

入力方法

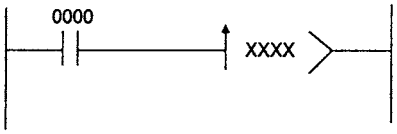


ERROR !!

- カイロエラー：入力回路に間違いがある。(例：入力がない、入力が多すぎる)

微分型命令：@××××　[@ #$] [FUN] [ ] [ ] : ↑ XXXX

入力リレーがOFFからONになったときだけ、その命令を実行します。



コーディング

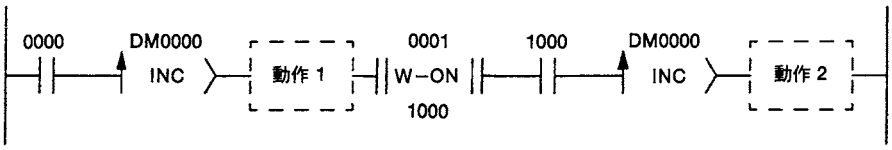
行番号	命令	オペランド
0000	LD	0000
0001	@××××	×××××

■解説

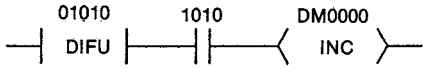
- 入力リレー0000がOFFからONになった、立ち上がりのときにだけ、その命令を実行します。
- 微分型命令のある命令

  LDA、SAT、CMP、ADD、SUB、MUL、DIV、ANDA、ORA、EORA、SRA、SLA、RRA、RLA、COM、INC、DEC、MPX、DMX、TBCD、TBIN、ASC、RASC、ROOT

（例）条件が成立するごとにDM0000をインクリメントする場合



- 入力0000がOFFからONの立ち上がり時にDM0000をインクリメントし、動作1を実行します。
- 動作1実行後、0001がONするとDM0000をインクリメントし、動作2を実行します。
- INC命令に微分型命令を使用しないと条件が成立している間、毎スキャン+1されます。微分型命令を使用せずにINC命令で実現しようとするとINC命令を



と書き替える必要があります。

オペランド

―――

入力方法

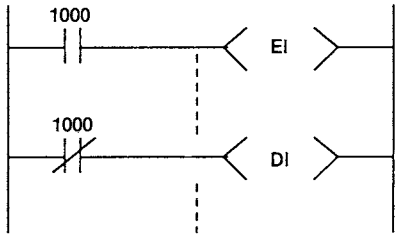
[@ #$] → [FUN] → [ファンクション番号] → [実行 逆検索]

ERROR //

- カイロエラー：入力回路に間違いがある。（例：入力がない、入力が多すぎる）

割込み禁止：DI	FUN 0 : DI	割込みプログラムの実行を禁止します。
割込み許可：EI	FUN 1 : EI	割込みプログラムの実行を許可します。

(例)



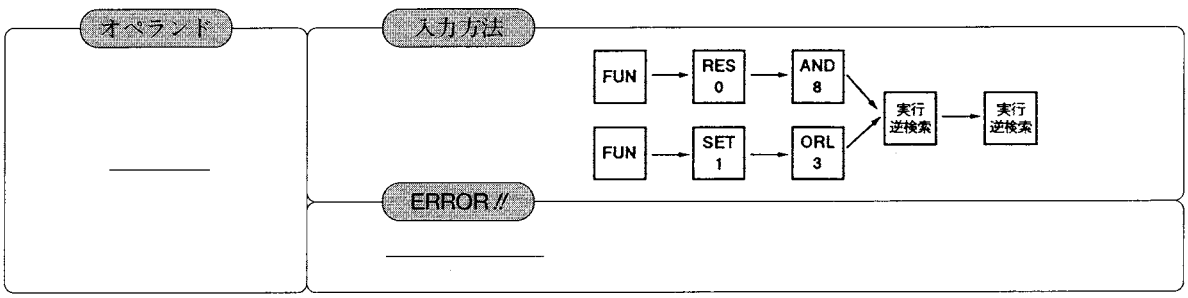
コーディング

行番号	命令	オペランド
0000	LD	0000
0001	EI	
⋮	⋮	
0021	LDB	1000
0022	DI	
	⋮	

●1000がONのとき、割込みプログラムの実行を許可します。
1000がOFFのとき、割込みプログラムの実行を禁止します。
EI命令～DI命令の間が割込み許可範囲です。

■解説

- KZは運転開始時は、常に割込み禁止状態になっています。
- DI命令は割込みプログラムの実行を禁止します。一時的に割込みを禁止したいときに使用します。
- EI命令は割込みプログラムの実行を許可します。DI命令で禁止したものを解除するのに使用します。
- DI～EI命令間（割込み禁止区間）で割込みが発生した場合は、EI命令が実行されるまで記憶しておき、EI命令が実行されるとすぐに割込みプログラムが実行されます。
- INT CTC割込みでは、割込み禁止状態で割込みが発生した場合は、これを記憶していますが、RES CTCを実行するとリセットされますので、EI命令を実行してもその割込みプログラムは実行しません。
- 割込みプログラム実行中は、他の割込みは禁止されます。
割込みプログラム実行中にさらに割込みプログラムを実行したい場合は、割込みプログラム中でEI命令を実行することにより4重まで受け付け可能となります。
速い速度で割込みが入ると、「INT ネストエラー」となります。
- 多数の割込みが順次発生した場合は先発優先です。完全同時発生の場合は、0000＞0001＞0002＞0003＞CTC2＞CTC3＞CTC0＞CTC1の順で優先されます。

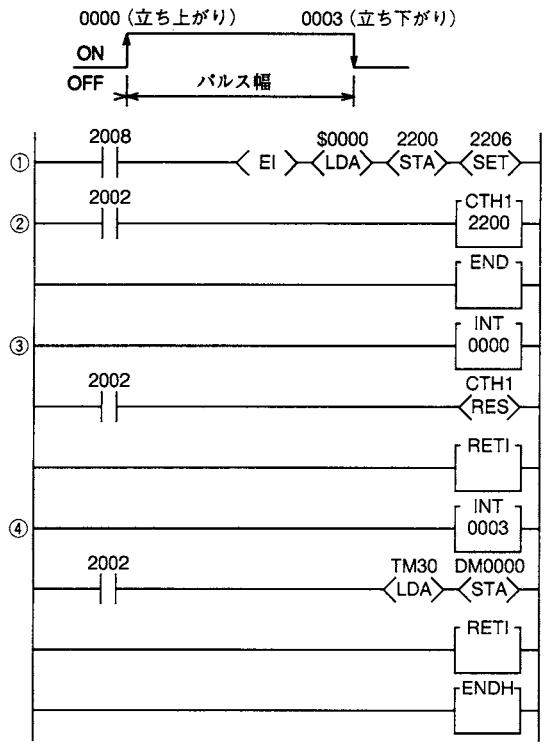


イント　：INT	FUN ANL 2 RES 0 ： INT nnnn	オペランドの立ち上がり、または入力リレー 0003 の立ち下がりで INT～RETI までの割込みプログラムを実行します。
リターン イント　：RETI	FUN ORL 3 LDB 4 ： RETI	割込みプログラムの最後を示します。

イント命令を実行することにより、スキャンタイムに影響を受けずに入力の取り込み (Min 25μs) を行うことができます。詳しくは例 1～4 をご覧ください。

(例 1)：パルス幅の計測

入力0003のパルス幅の計測を行ないます。



コーディング

行番号	命令	オペランド
00000	LD	2008
00001	EI	
00002	CON	
00003	LDA	$0000
00004	CON	
00005	STA	2200
00006	CON	
00007	CON	
00008	SET	2206
00009	LD	2202
00010	CTH	1 2200
00011	END	
00012	INT	0000
00013	LD	2002
00014	RES	CTH1
00015	RETI	
00016	INT	0003
00017	LD	2002
00018	LDA	TM30
00019	CON	
00020	STA	DM0000
00021	RETI	
00022	ENDH	

■解説

① 初期設定を行ないます。
- EI 命令により割込みを許可します。
- 特殊補助リレー 2200～2215 をすべて OFF にします。これは2203を OFF することにより、CTC2による高速カウンタの0クリアを無効にします。
  2206を ON することにより INT 00003の立下がりまちになります。

② 内部クロック 0.4 μs のパルスを高速カウンタでカウントします。

③ 入力00000の割込み設定を行ないます。
- 割込みが実行されると、高速カウンタを0クリアします。

④ 入力00003の割込み設定を行ないます。
- 結果は DM0000 に格納されます。

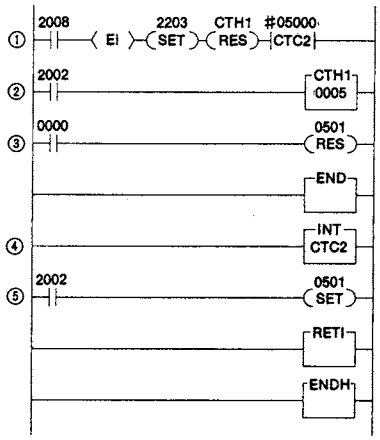
■解説

●INT0003は、入力0003の立ち上がりまたは立ち下がりで割込みプログラムを実行します。
この設定は特殊補助リレー2206のON－OFFにより行います。

リレー番号	ON時の動作	OFF時の動作
2206	入力0003の立ち下がりエッジにて割込みプログラムを実行します。	入力0003の立ち上がりエッジにて割込みプログラムを実行します。

● INT0003で割込みが入ったとき、その瞬間のCTH1の現在値が、自動的にTM30へ転送されます。
● INT0000～0002は、入力0000～0002の立ち上がりで割込みプログラムを実行します。

（例2）：カウンタによるダイレクト出力



●入力0005へ入るパルスをカウントし、5000パルスになったとき0501をONさせます。

コーディング

行番号	命令	オペランド
0000	LD	2008
0001	EI	
0002	CON	
0003	SET	2203
0004	CON	
0005	RES	CTH1
0006	CON	
0007	CTC	2　#05000
0008	LD	2002
0009	CTH	1　0005
0010	LD	0000
0011	RES	0501
0012	END	
0013	INT	CTC2
0014	LD	2002
0015	SET	0501
0016	RETI	
0017	ENDH	

■動作説明

①の部分
●運転開始時、EI命令により割込みを許可します。
●特殊補助リレー2203をONすることにより、CTC2による高速カウンタの0クリアをします。
●高速カウンタCTH1を0クリアし、CTC2の値を設定します。
②の部分
●入力0005へのパルスを高速カウンタにてカウントします。
③の部分
●0501の出力をリセットします。
④の部分
● CTC2による割込みプログラムの実行をします。
⑤の部分
●0501へのダイレクト出力をします。

■解説

●INT CTC2は、CTC2の高速カウンタコンパレータの設定値と高速カウンタの現在値が同値となったときに、割込みプログラムを実行します。
割込みプログラムが実行されると、CTC2の接点は自動的にリセットします。
他の高速カウンタコンパレータCTC0、CTC1、CTC3についても同様な動作となります。

●割込みプログラム中で出力0500と0501について、SETまたはRES命令が実行された場合、ただちに出力します。(ダイレクト出力)

●割込みプログラムはENDとENDHの間に記述します。

●割込みプログラムの開始は「INT」命令を使用します。

●割込みプログラムの終了は「RETI」命令を使用します。

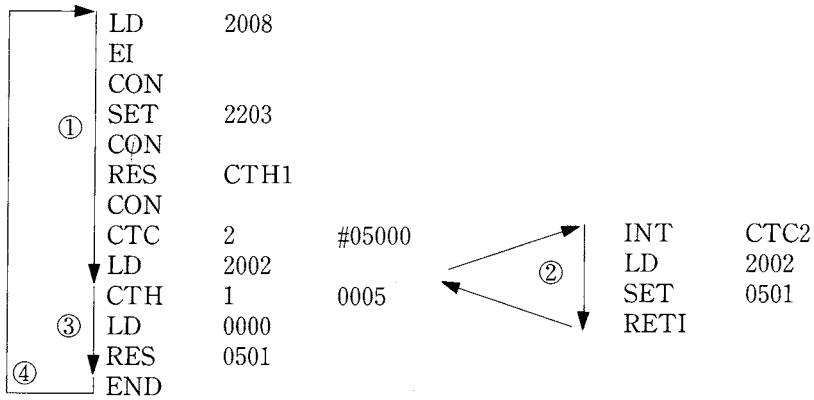
● RETI命令により、割込みがかかる前のシーケンスプログラムに戻ります。(図1)

● INT命令のオペランドの重複使用はできません。

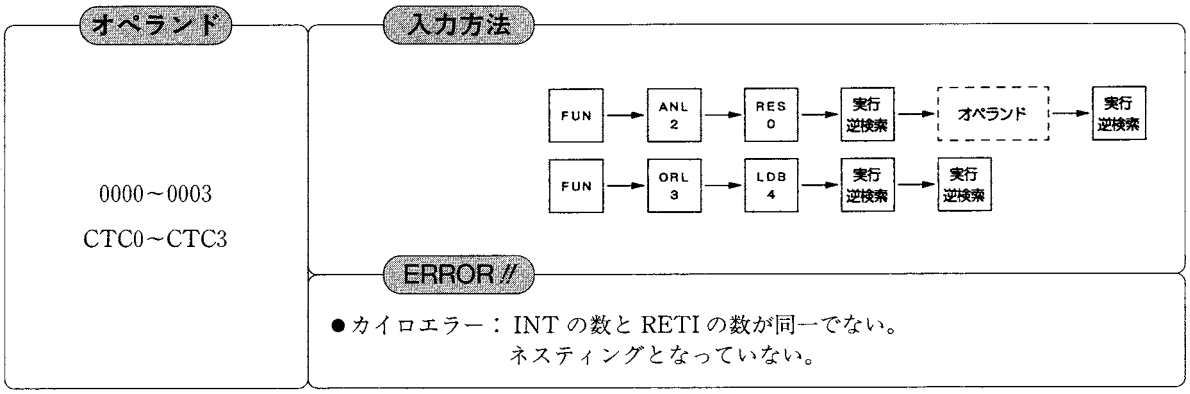
●割込みプログラム中では、以下の命令は使用できません。

TMR、TMH、TMS、DIFU、DIFD、W−ON、W−OFF、W−UE、W−DE、 STG、JMP、ENDS、HKEY、ITVL、@×××命令、C、UDC、SFT、CTH

●割込みプログラムが実行されると、その実行時間だけスキャンタイムが長くなります。

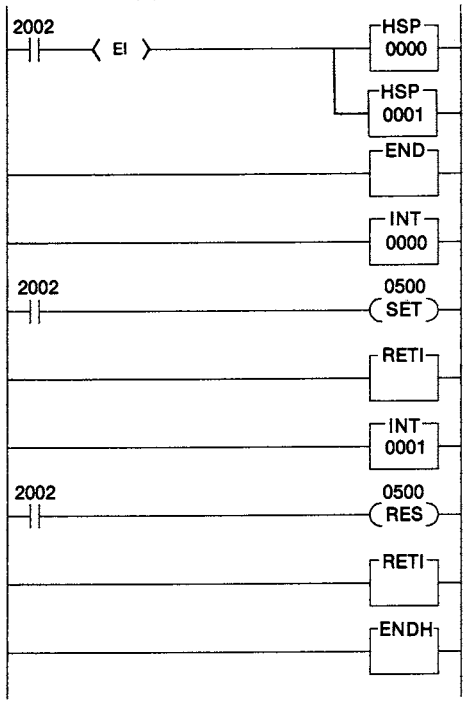


(図1) 割込みプログラム実行の流れ



割込みプログラムの例

(例 3) ダイレクト出力



● 入力0000がONすると、0500がただちに出力されます。0001をONすると0500はリセットされます。

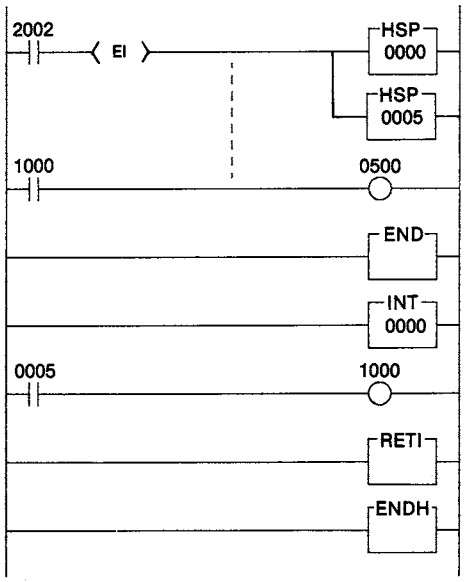
コーディング

行番号	命令	オペランド
0000	LD	2002
0001	EI	
0002	CON	
0003	HSP	0000
0004	HSP	0001
⋮	⋮	⋮
0100		
0101		
0102	END	
0103	INT	0000
0104	LD	2002
0105	SET	0500
0106	RETI	
0107	INT	0001
0108	LD	2002
0109	RES	0500
0110	RETI	
0111	ENDH	

● ダイレクト出力は割込みプログラム中で出力0500または0501についてSETまたはRES命令が実行された場合に可能です。

● ダイレクト出力はサイクルタイムに影響されません。

(例 4) ダイレクト入力



● 入力0000がONした時、入力0005のON−OFF状況をただちに入力します。

コーディング

行番号	命令	オペランド
0000	LD	2002
0001	EI	
0002	CON	
0003	HSP	0000
0004	HSP	0005
⋮	⋮	⋮
0100	LD	1000
0101	OUT	0500
0102	END	
0103	INT	0000
0104	LD	0005
0105	OUT	1000
0106	RETI	
0107	ENDH	

● ダイレクト入力は0000～0005でおこなえます。

● ダイレクト入力を使用する場合HSP命令で入力時定数を高速用にしておく必要があります。

● ダイレクト入力を使用することで、スキャン途中で入力を取り込むことが可能です。

● ダイレクト入力した接点情報は割込みプログラム中でのみ有効です。

演算命令による特殊補助リレーの変化

A：内部レジスタ

命令語	ニーモニック	オペランド	2009 (A＜0)	2010 (A＝0)	2011 (A＞0)	2012 (エラー)
データメモリライト	DW	$,#定数				
ロードA	LDA (FUN23)	リレー番号		⑤		
		タイマ/カウンタ番号		⑤		
		データメモリ番号		⑤		
		$,#定数		⑤		
		#TM		⑤		⑱
ストアA	STA (FUN42)	リレー番号				
		タイマ/カウンタ番号				
		データメモリ番号				
		#TM				⑱
比較	CMP (FUN04)	データメモリ番号	①	②	③	
		$,#定数	①	②	③	
		#TM	①	②	③	⑱
加算	ADD (FUN00)	データメモリ番号	④	⑤		
		$,#定数	④	⑤		
		#TM	④	⑤		⑱
減算	SUB (FUN46)	データメモリ番号	⑥	⑤	⑦	
		$,#定数	⑥	⑤	⑦	
		#TM	⑥	⑤	⑦	⑱
乗算	MUL (FUN28)	データメモリ番号	⑲	⑧		
		$,#定数	⑲	⑧		
		#TM	⑲	⑧		⑱
除算	DIV (FUN11)	データメモリ番号		⑧		⑨
		$,#定数		⑧		⑨
		#TM		⑧		⑱⑨
アンドA	ANDA (FUN01)	データメモリ番号		⑤		
		$,#定数		⑤		
		#TM		⑤		⑱

演算命令による特殊補助リレーの変化

命令語	ニーモニック	オペランド	2009 (A＜0)	2010 (A＝0)	2011 (A＞0)	2012 (エラー)
オア A	ORA (FUN31)	データメモリ番号		⑤		
		$, # 定数		⑤		
		#TM		⑤		⑱
エクスクルーシブオア A	EORA (FUN15)	データメモリ番号		⑤		
		$, # 定数		⑤		
		#TM		⑤		⑱
シフトライト A	SRA (FUN41)	シフト数	⑩	⑤		
シフトレフト A	SLA (FUN40)	シフト数	⑩	⑤		
ローテートライト A	RRA (FUN37)	ローテート数	⑩	⑤		
ローテートレフト A	RLA (FUN40)	ローテート数	⑩	⑤		
コンプリメント	COM (FUN05)			⑤		
インクリメントメモリ	INC (FUN19)	データメモリ番号	⑪	⑤		
ディクリメントメモリ	DEC (FUN07)	データメモリ番号	⑫	⑤		
マルチプレクス	MPX (FUN27)	桁番号				
デマルチプレクス	DMX (FUN12)					⑬
トランスファ BCD	TBCD (FUN47)					⑭
トランスファ BIN	TBIN (FUN48)					⑮
アスキーコード変換	ASC (FUN02)					
アスキーコード逆変換	RASC (FUN32)					⑯
平方根	ROOT (FUN36)			⑤		
トリマ入力	TMIN (FUN50)			⑰		

演算命令による特殊補助リレーの変化

①CMP の結果、負になった場合 ON それ以外の場合 OFF
②CMP の結果、 0 になった場合 ON それ以外の場合 OFF
③CMP の結果、正になった場合 ON それ以外の場合 OFF
④加算の結果、16ビット（$FFFF）の範囲をこえた場合 ON それ以外の場合 OFF
⑤演算の結果、内部レジスタの内容が 0 になった場合 ON それ以外の場合 OFF
⑥演算の結果、負になる場合 ON それ以外の場合 OFF
⑦演算の結果、正になる場合 ON それ以外の場合 OFF
⑧演算の結果、32ビットが 0 の場合 ON それ以外の場合 OFF
⑨除算で除数が 0 の場合 ON、その場合に除算はされない
⑩シフト・ローテートの結果、2009に 1 が入れば、ON、それ以外の場合 OFF
⑪インクリメント演算の結果 0 になった場合 ON それ以外の場合 OFF
⑫ディクリメント演算の結果、$FFFF になった場合 ON それ以外の場合 OFF
⑬デマルチプレクス演算で ON のビットがない場合 ON（この場合は結果には 0 が入る）それ以外の場合 OFF
⑭TBCD 実行時に内部レジスタの値が $270F（#09999）を越える場合 ON（この場合 TBCD は実行されない）そうでない場合 OFF
⑮TBIN 実行時に内部レジスタの値が BCD データでない場合 ON（この場合 TBIN は実行されない）そうでない場合 OFF
⑯RASC 実行時に内部レジスタの上下 8 ビットの値がそれぞれ、$30～$39または $41～$46の範囲でない場合 ON（この場合 RASC は実行されない）そうでない場合 OFF
⑰ TMIN で入力した値が 0 であった場合 ON それ以外の場合 OFF
⑱ #TM の間接指定で DM およびリレーの範囲を越えた。
⑲乗算の結果、16ビットを超えた場合 ON それ以外の場合 OFF

データメモリ・リレーの間接指定について

LDA,@LDA
STA,@STA
CMP,@CMP
ADD,@ADD
SUB,@SUB
MUL,@MUL
DIV,@DIV
ANDA,@ANDA
ORA,@ORA
EORA,@EORA

の各命令でオペランドに #TM××と書いた場合、データメモリ・リレーの間接指定を行うことができます。
(××はテンポラリデータメモリの番号で、設定できる番号はTM02～TM29)
テンポラリデータメモリについてはP2－10をご覧ください。

■テンポラリーデータメモリによるデータメモリ・リレーの間接指定

●指定方法

TM ××に値（10進数）を指定することによりおこないます。

TM ××の値が（9999以下の場合はDM番号 / 10000以上の場合はリレー番号（10000を引いた値））を示します。

指定されたDM、リレーが存在しない時は、特殊補助リレー2012がONします。

(例) LDA #TM15 を指定した場合

TM15の値が	内部レジスタへ入る内容は
20	DM0020
11000	1000～1015

上記の場合 LDA DM0020、LDA 1000 と同じ働きをします。
TM15 の値を変えることにより自由に指定先を変えることが可能です。

(例) STA #TM02 を指定した場合

TM02の値が	転送先の指定は
00	DM0000
13000	3000～3015

上記の場合 STA DM0000、STA 3000 と同じ働きをします。
TM02 の値を変えることにより自由に指定先を変えることが可能です。

TM ××で指定できる値の範囲と対応するDM番号とリレー番号

機　種	値	DM・リレー番号
KZ－10, 16	#00000～#00999 #10000～#12915	DM0000～DM0999 0000～2915
KZ－24, 40, 80	#00000～#01999 #10000～#16915	DM0000～DM1999 0000～6915

●プログラム例については P5－22～P5－23 をご覧ください。

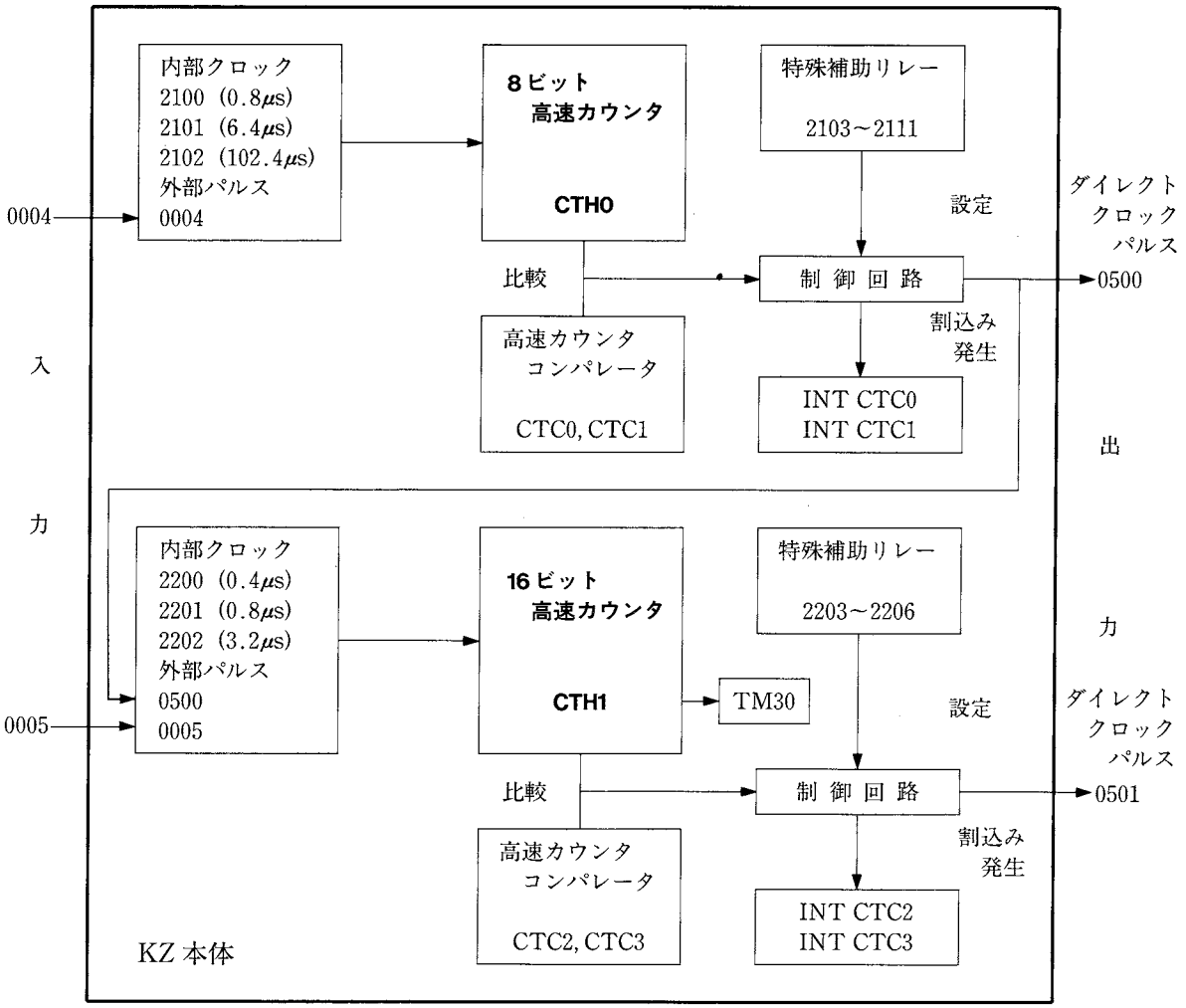
6章　高速カウンタの応用

6章では、高速カウンタを使用し、ダイレクトクロックパルスを発生させる方法について解説します。
ダイレクトクロックパルスと高速カウンタを組み合わせて用いれば、速度制御を伴った位置決め制御を行え、簡易パルスモーターコントローラとして使用できます。

高速カウンタの概要

■高速カウンタについて

KZシリーズは8ビット高速カウンタ（CTH0）と16ビット高速カウンタ（CTH1）、さらにそれぞれに2つの高速カウンタコンパレータ（CTC0、1とCTC2、3）を備えています。
これらを使用して、簡易パルスモータコントローラやパルス幅測長などをおこなうことができます。



（図1）高速カウンタのブロック図

8ビット高速カウンタ(CTH0)と高速カウンタコンパレータ(CTC0、CTC1)について

●8 ビット高速カウンタ (CTH0) は 0～255 をカウントできるハードウェアのカウンタです。
カウント入力としては、外部パルス (0004)、内部クロック (2100～2102) から 1 つ選択することができます。(外部からの変更はできません)
外部パルス (0004) を選択した場合は、応答周波数 10kHz の高速カウンタとして使用できます。また、内部クロック (2100～2102) を選択した場合は高速タイマとして使用できます。
●高速カウンタコンパレータ (CTC0、CTC1) は、高速カウンタ CTH0 用の 2 組の高速比較器です。
CTH0 の現在値が CTC0、CTC1 の設定値と等しくなる (CTC0、CTC1 が ON する) と、出力 0500 の出力状態をハード的に変更 (ダイレクト出力) したり、割込みを発生させたりすることができます。
出力 0500 の状態変更は、特殊補助リレー (2103～2111) で設定します。

16ビット高速カウンタ(CTH1)と高速カウンタコンパレータ(CTC2、CTC3)について

●16 ビット高速カウンタ (CTH1) は 0～65535 をカウントできるハードウェアのカウンタです。
カウント入力としては、外部パルス (0005、0500)、内部クロック (2200～2202) から 1 つ選択することができます。(外部からの変更はできません)
外部パルス (0005) を選択した場合は、応答周波数 10kHz の高速カウンタとして使用できます。また、外部パルス (0500) を選択した場合は、高速カウンタ (CTH0) で発生させたダイレクトクロックパルスをカウントできます。
●高速カウンタコンパレータ (CTC2、CTC3) は、高速カウンタ CTH1 用の 2 組の高速比較器です。
CTH1 の現在値が CTC2、CTC3 の設定値と等しくなる (CTC2、CTC3 が ON する) と、出力 0501 の出力状態をハード的に変更 (ダイレクト出力) したり、割込みを発生させたりすることができます。
出力 0501 の状態変更は、特殊補助リレー (2203～2206) で設定します。

(注) 高速カウンタコンパレータ (CTC0、CTC1、CTC2、CTC3) を通常プログラムの中で接点として使用する場合、次の点にご注意ください。
コンパレータの設定した値と高速カウンタの値が一致した瞬間に、この接点はスキャンタイムに依存しないで (入力処理以外でも) ラダーに反映されます。このため、入力処理時に OFF であっても出力処理では ON として処理されることがあります。

■高速カウンタ用内部クロックについて

●高速カウンタ用に専用の内部クロックがあります。
●内部クロックを使用する場合は CTH 命令のオペランドでリレー番号を指定します。
●内部クロックの周期は下表のようになります。

CTH0 専用内部クロック		CTH1 専用内部クロック	
リレー番号	周　　期	リレー番号	周　　期
2100	0.8μs	2200	0.4μs
2101	6.4μs	2201	0.8μs
2102	102.4μs	2202	3.2μs

(高速カウンタ、高速カウンタコンパレータについては、5 章 応用命令・演算命令 P5－17 をご参照ください)
●プログラム実行中に内部クロック、外部パルスの変更はできません。

ダイレクトクロックパルス出力

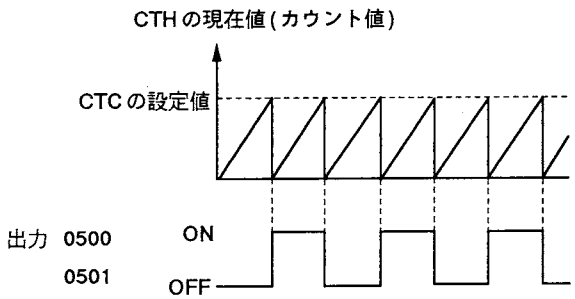
KZ シリーズは高速カウンタと高速カウンタコンパレータを使用して、サイクルタイムの影響を受けないクロックパルスを直接 (ダイレクト) 出力リレーの 0500 または 0501 に出力することができます。(トランジスタ出力タイプ、MOS-FET 出力タイプのみ)
またこの機能と高速カウンタ (CTH1) とを組み合わせることにより、速度制御を伴なう位置決め制御を行うことができます。(位置決め制御に使用する場合は一軸の制御になります)
設定できるクロック周期はプログラムにより下表の範囲で任意に選択することができます。

機　種	出 力	クロック周期(周波数)範囲
KZ-10、16、24、40	500	約 500μs (2kHz)～52.2ms (19Hz)
	501	約 666μs (1.5kHz)～419.4ms (2.4Hz)
KZ-80	500	約 50μs (20kHz)～52.2ms (19Hz)
	501	約 666μs (1.5kHz)～419.4ms (2.4Hz)

ダイレクトクロックパルス出力の概要

●ダイレクトクロックパルスは内部クロックパルスを高速カウンタ (CTH) でカウントし、その現在値が高速カウンタコンパレータ (CTC) の設定値と等しくなるごとに出力リレー 0500、0501 の ON/OFF の状態を変化し、パルスを発生させます。

(出力例)

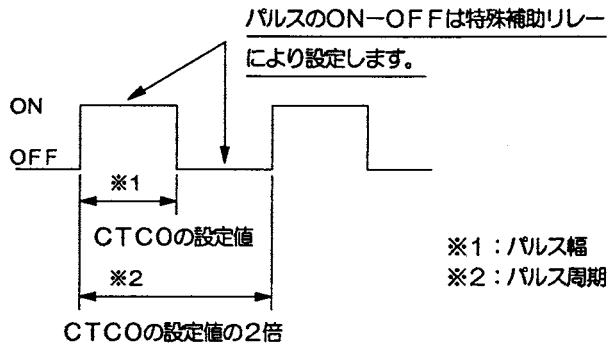


出力リレー 0500、0501 の ON/OFF の状態は特殊補助リレーの設定によって決めます。詳しくは P. 6−5 特殊補助リレーの設定による動作についてをご覧ください。

ダイレクトクロックパルスのパルス周期とパルス幅について

●パルス周期とパルス幅は高速カウンタ用 (CTH) 内部クロックの周期と高速カウンタコンパレータ (CTC) の設定値により決まります。
また高速カウンタコンパレータを 2 個 (CTC0、CTC1) 使用することによりパルス周期は一定でパルス幅だけを変化させることができます。
●高速カウンタコンパレータによるパルス周期とパルス幅は次のようになります。

(a) ON－OFF 比 1：1 のパルス



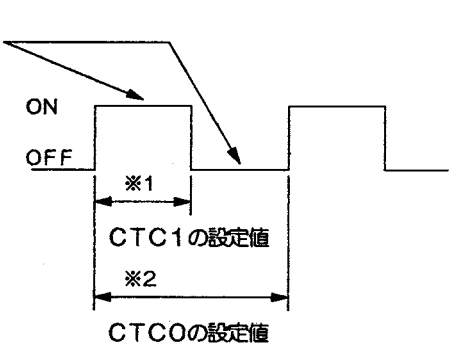
パルス幅 (μs) ＝ CTC0 の設定値×内部クロック (μs)

パルス周期 (μs) ＝ CTC0 の設定値×2×内部クロック (μs)

●パルスの ON－OFF 比は1:1となります。
●CTC 0 の設定値を可変すると、パルス周期とパルス幅が可変できます。

※1：パルス幅
※2：パルス周期

(b) パルス幅可変のパルス



パルス幅 (μs) ＝ CTC1 の設定値×内部クロック (μs)

パルス周期 (μs) ＝ CTC0 の設定値×内部クロック (μs)

●CTC 1 の設定値を可変することにより、パルス幅を可変できます。
●CTC 0 の設定値を可変すると、パルス周期が可変できます。

（図 2）高速カウンタコンパレータによるパルス周期とパルス幅

上記で決められるパルス幅は KZ の内部で出力されるパルス幅になります。
実際には出力回路を通って出力されるためパルス幅は出力回路の応答遅れや負荷により影響され変動します。
特に 2kHz 以上のパルスを使用される場合は大きく影響しますのであらかじめオシロ等で実際の出力波形を観測したうえでパルス幅の設定を行うとよいでしょう。((b)パルス幅可変のパルスを使用します)
P6－8 に応答遅れを考慮したパルス幅設定の目安を記載しています。参考にしてください。

●周波数からパルス周期を求めるには下式により計算します。

パルス周期 (μs) ＝1000÷周波数 (kHz)

6章　高速カウンタの応用

6-2. ダイレクトクロックパルス

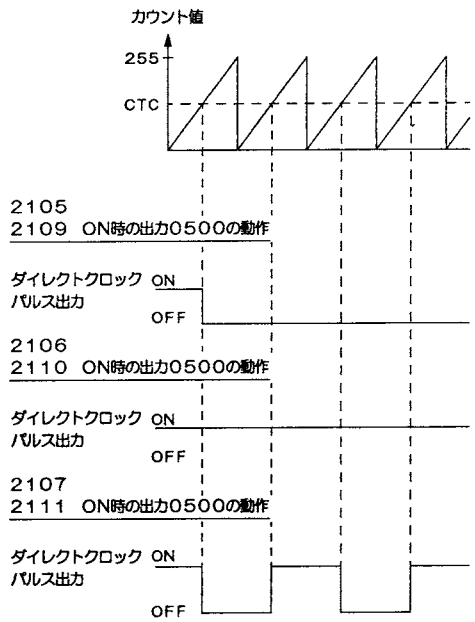
■特殊補助リレーの設定による動作について

8ビット高速カウンタ用(CTH0)特殊補助リレーについて

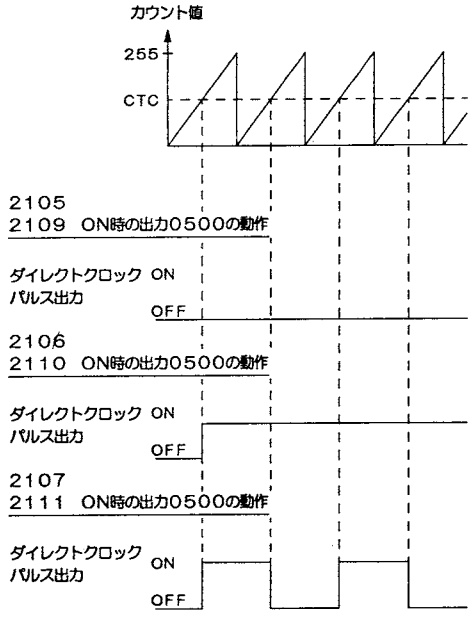
●特殊補助リレー(2104～2111)を運転開始時に設定、もしくはダイレクトクロックパルス発生中に変更することにより出力リレー 0500 の ON/OFF の状態を下記のように変えることができます。
またそれによりパルス周期とパルス幅やダイレクト出力の許可、禁止を自由に設定することが可能です。
実際の動きは CTH0 の現在値と CTC0、CTC1 の設定値が等しくなるごとに出力リレー 500 の ON/OFF を反転させパルスを発生させます。
●ダイレクトクロックパルスを発生させるには必ず特殊補助リレーの設定が必要です。

特殊補助リレー ON/OFF 時の動作		
CTC0 のリレー	CTC1 のリレー	CHT0 (現在値)=CTC (設定値) になったとき
2104	2108	CTC ON 時 0500 へのダイレクト出力を　ON：禁止する　OFF：許可する
2105	2109	CTC ON 時 0500 へのダイレクト出力を OFF に　ON：する　OFF：しない
2106	2110	CTC ON 時 0500 へのダイレクト出力を ON に　ON：する　OFF：しない
2107	2111	CTC が ON するごとに 0500 への出力の ON/OFF を反転　ON:するOFF:しない

特殊補助リレーの設定による出力 0500 の動作は下図のようになります。



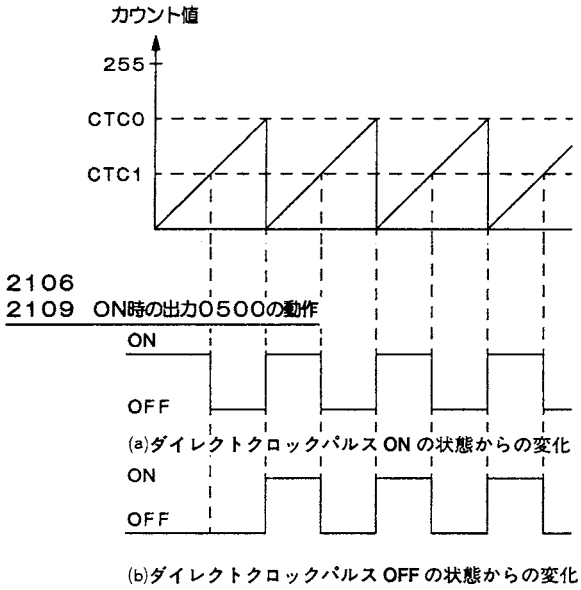
(a)ダイレクトクロックパルス ON の状態からの変化

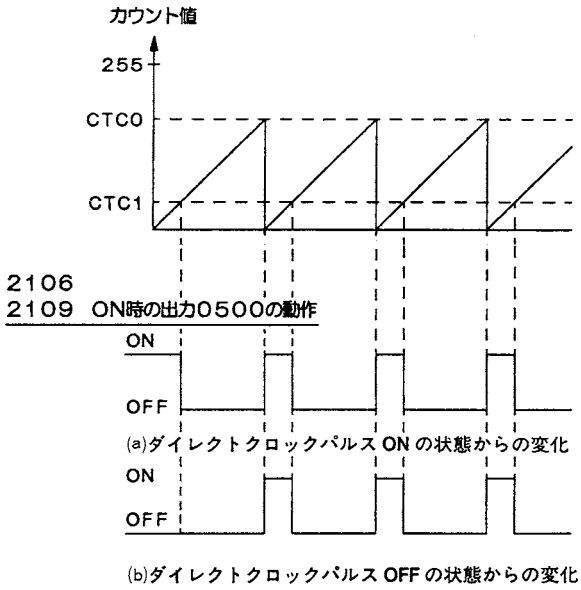


(b)ダイレクトクロックパルスOFF の状態からの変化

●2104～2107、2108～2111 の ON に設定するリレーは、それぞれ 1 個のみ有効です。
それぞれの範囲 (2104～2107、2108～2111) で複数リレーを ON した場合は、番号の若いリレーが優先となります。
●2104、2108 は OFF でダイレクト出力を許可します。初期状態は ON になっていますのでご注意ください。
●特殊補助リレー 2104～2111 の ON-OFF は、SET、RES 命令または、STA 命令にて設定します。

パルス幅可変時の特殊補助リレーの設定による出力 0500 の動作





●パルス幅可変の場合、特殊補助リレー 2103 (CTC0 ON 時、CTH の自動クリア) を ON にして使用します。CTC1 の設定値を可変することにより、パルス幅を変えることができます。
●特殊補助リレー 2104、2108 をダイレクトクロックパルス出力中に ON すると、ON する直前の出力の状態を保持します。(2104、2108 は OFF でダイレクト出力を許可します。初期状態は ON になっていますのでご注意ください)

16 ビット高速カウンタ用 (CTH1) 特殊補助リレーについて

●特殊補助リレー (2204～2205) の設定をすることにより出力リレー 0501 の ON/OFF の状態を下記のように変えることができます。

	特殊補助リレー ON/OFF 時の動作
リレー	CTH1 (現在値)=CTC2 (設定値) になったとき
2204	CTC2 ON 時 0501 へのダイレクト出力を　ON：許可する　OFF：禁止する
2205	CTC2 ON 時 0501 へのダイレクト出力を　ON：ON にする　OFF：OFF にする

ダイレクトクロックパルスを発生させるには CTH1 の現在値と CTC2 の設定値が等しくなるごとに INT 命令によって割込みを行います。(ダイレクトクロックパルス使用例 例 3 (P6－11) を参照ください)
●16 ビット高速カウンタ (CTH1) はクロック入力のオペランドに 0500 を指定することにより、CTH0 の設定によって出力 0500 から発生するダイレクトクロックパルスのパルス数をカウントすることができます。(ダイレクトクロックパルス使用例 例 4 (P6－12) を参照ください)
●上記の場合、4kHz 以上のパルスをカウントし、ある一定のパルス数になるとダイレクトクロックパルスの発生を停止させると応答遅れのため多少の誤差を生じます。
その場合、INT 命令を用いてダイレクトクロックパルスの周波数を直前で 4kHz 以下に減速して停止させるようにしてください。

6章 高速カウンタの応用

6-2. ダイレクトクロックパルス

パルス周期 500μs 以上（2kHz 以下）で使用する場合の設定値の目安

●高速カウンタ CTH0 をもちいて出力リレー 500 よりパルス同期 500μs 以上 (2kHz 以下) で ON-OFF 比 1：1 のパルスを出力する場合。
コンパレータ CTC0 の設定値の目安としてください。　P6－9 (例 1) 参照

CTC0 の設定値＝パルス周期 (μs)÷内部クロック÷2

パルス周期		CTH0 の内部クロック 0.8μs	CTH0 の内部クロック 6.4μs	CTH0 の内部クロック 102.4μs
		CTC0 の設定値	CTC0 の設定値	CTC0 の設定値
500μs	2kHz	設定できません	39	
526μs	1.9kHz		41	
566μs	1.8kHz		43	
588μs	1.7kHz		46	
625μs	1.6kHz		49	3
667μs	1.5kHz		52	設定できません
714μs	1.4kHz		56	
769μs	1.3kHz		60	
833μs	1.2kHz		65	4
1ms	1.0kHz		78	5
1.25ms	0.8kHz		98	6
1.67ms	0.6kHz		130	8
2.5ms	0.4kHz		195	12
5ms	0.2kHz		設定できません	25
10ms	0.1kHz			49
52.2ms	19Hz			255

― 8 bit 高速カウンタ用補助リレー一覧 ―

2100	CTH0 専用内部クロック (0.8μsec)
2101	CTH0 専用内部クロック (6.4μsec)
2102	CTH0 専用内部クロック (102.4μsec)
2103	コンパレータ CTC0 の ON 時、CTH0 の自動クリア　ON：する　OFF：しない
2104	コンパレータ CTC0 の ON 時 0500 へのダイレクト出力を　ON：禁止する　OFF：許可する
2105	コンパレータ CTC0 の ON 時 0500 へのダイレクト出力を OFF に　ON：する　OFF：しない
2106	コンパレータ CTC0 の ON 時 0500 へのダイレクト出力を ON に　ON：する　OFF：しない
2107	コンパレータ CTC0 が ON するごとに 0500 への出力の ON/OFF を反転 ON：する　OFF：しない
2108	コンパレータ CTC1 の ON 時 0500 へのダイレクト出力を　ON：禁止する　OFF：許可する
2109	コンパレータ CTC1 の ON 時 0500 へのダイレクト出力を OFF に　ON：する　OFF：しない
2110	コンパレータ CTC1 の ON 時 0500 へのダイレクト出力を ON に　ON：する　OFF：しない
2111	コンパレータ CTC1 が ON するごとに 0500 への出力の ON/OFF を反転 ON：する　OFF：しない

パルス周期 500μs 以下（2kHz 以上）で使用する場合の設定値の目安

●KZ-80T 使用時、高速カウンタ CTH0 をもちいて出力リレー 500 よりパルス同期 500μs 以下 (2kHz 以上) で ON-OFF 比約 1：1 のパルスを出力する場合コンパレータ CTC0、CTC1 の設定値の目安としてください。

P6－10 (例 2) 参照

(CTC1 の設定はあらかじめ、トランジスタの応答遅れを含んだ目安としています)

CTC0 の設定値＝パルス周期 (μs)÷内部クロック

CTC1 の設定値＝パルス幅 (μs)÷内部クロック

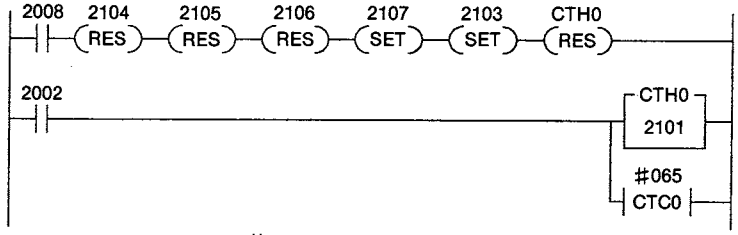
パルス周期		CTH0 の内部クロック 0.8μs		CTH0 の内部クロック 6.4μs		CTH0 の内部クロック 102.4μs	
		CTC0 の設定値	CTC1 の設定値	CTC0 の設定値	CTC1 の設定値	CTC0 の設定値	CTC1 の設定値
50μs	20kHz	62	7	8	1	設定できません	
56μs	18kHz	70	8	9	1		
60μs	16kHz	75	15	10	2		
70μs	14kHz	87	16	11	2		
80μs	12kHz	100	23	13	3		
100μs	10kHz	125	39	16	5		
130μs	8kHz	163	56	20	7		
170μs	6kHz	213	79	27	10		
200μs	5kHz	250	104	31	13		
250μs	4kHz	設定できません		39	16		
330μs	3kHz			52	23		
500μs	2kHz			79	37	5	3
670μs	1.5kHz			105	53	7	4
1ms	1kHz			156	75	10	5
1.6ms	0.6kHz			255	128	16	8

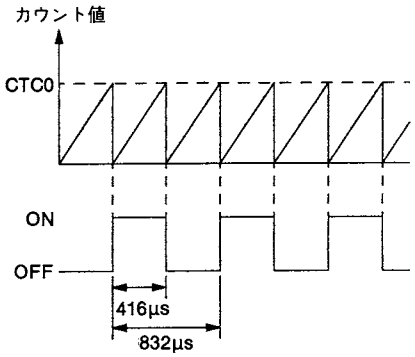
― 16 bit 高速カウンタ用補助リレー一覧 ―

2200	CTH1 専用内部クロック (0.4μsec)
2201	CTH1 専用内部クロック (0.8μsec)
2202	CTH1 専用内部クロック (3.2μsec)
2203	コンパレータ CTC2 の ON 時、CTH1 の自動クリア　ON：する　OFF：しない
2204	コンパレータ CTC2 の ON 時 0501 へのダイレクト出力を　ON：許可する　OFF：禁止する
2205	コンパレータ CTC2 の ON 時 0501 へのダイレクト出力を　ON：ON にする　OFF：OFF にする
2206	INT3 のエッジ選択　ON：ダウンエッジ　OFF：アップエッジ

ダイレクトクロックパルス使用例

(例 1)：出力リレー 0500 より周期 832μs (約 1.2kHz) のパルスを出力します。(ON－OFF 比 1：1)





コーディング

行番号	命令	オペランド
0000	LD	2008
0001	RES	2104
0002	CON	
0003	RES	2105
0004	CON	
0005	RES	2106
0006	CON	
0007	SET	2107
0008	CON	
0009	SET	2103
0010	CON	
0011	RES	CTH0
0012	LD	2002
0013	CTH	0 2101
0014	CTC	0 #065

■解説

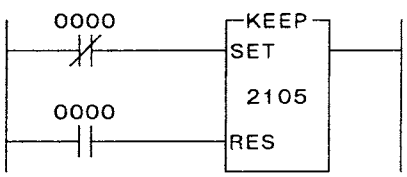
- 運転開始時に特殊補助リレー2104～2107により出力0500へのパルスの出力を設定します。
  例 1 の場合、0500へのダイレクト出力を許可し、CTH0 ＝ CTC0 となるたびに0500の出力を反転するよう設定します。
  同時に、特殊補助リレー2103を ON することで、CTC0 による CTH0 のクリアを指定します。
- 運転開始時に CTH0 の現在値をクリアします。これにより、ダイレクトクロックパルスは 1 パルス目のパルス幅より正しく出力します。
- 高速カウンタ CTH0 のクロック入力に内部クロック2101 (パルス周期6.4μs) を設定します。
- 高速カウンタコンパレータ CTC0 の設定値により、パルス周期を決めます。

CTC0 の設定値＝パルス周期÷内部クロック÷ 2 ＝832(μs)÷6.4(μs)÷ 2 ＝65

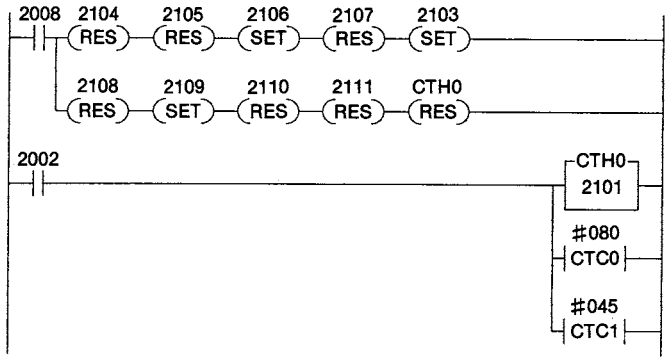
- CTH0 の入力リレーは、ダイレクトクロックパルス出力時は常時 ON (2002) を使用します。
- 出力可能なパルス周期 (周波数) は、500μs (2kHz)～52.2ms (19Hz) の範囲です。
  KZ-80 は 50μs (20kHz)～52.2ms (19Hz) の範囲です。(2kHz 以上の設定をする場合は応答遅れの影響を考え、ON－OFF 比の変れる (例 2) の使用をおすすめします。)

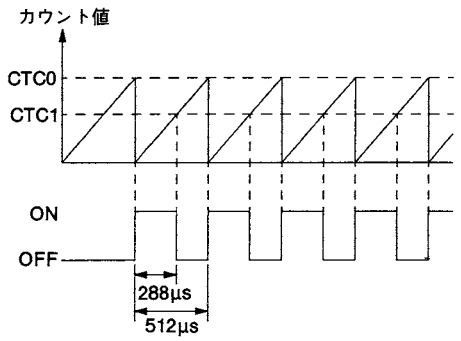
ワンポイント

- 外部入力によりダイレクトクロックパルスの出力を制御 (許可／禁止) する場合は、特殊補助リレー2105の ON－OFF により、おこないます。
  特殊補助リレー2105の ON－OFF は、SET、RES 命令または、KEEP 命令によりおこないます。
  (CTH0 の場合)
  上記の例に右図のプログラムを加えると0000が ON のときにのみ、ダイレクトクロックパルスが 0500より出力されます。



(例 2)：出力リレー 0500 より周期 512μs (約 1.95kHz)、パルス幅 288μs のパルスを出力します。





コーディング

行番号	命令	オペランド
0000	LD	2008
0001	MPS	
0002	RES	2104
0003	CON	
0004	RES	2105
0005	CON	
0006	SET	2106
0007	CON	
0008	RES	2107
0009	CON	
0010	SET	2103
0011	MPP	
0012	RES	2108
0013	CON	
0014	SET	2109
0015	CON	
0016	RES	2110
0017	CON	
0018	RES	2111
0019	CON	
0020	RES	CTH0
0021	LD	2002
0022	CTH	0 2101
0023	CTC	0 #080
0024	CTC	1 #045

■解説

- 運転開始時に特殊補助リレー2104～2107、2108～2111により、出力0500へのパルス出力を設定します。
  例 2 の場合、0500へのダイレクト出力を許可し、0500の出力を CTC0 で ON、CTC1 で OFF するように設定します。
  同時に特殊補助リレー2103を ON することで CTC0 による CTH0 のクリアを指定します。
- 運転開始時に CTH0 の現在値をクリアします。これにより、ダイレクトクロックパルスは 1 パルス目のパルス幅より正しく出力します。
- 高速カウンタ CTH0 のクロック入力に内部クロック2101 (パルス周期6.4μs) を指定します。
- 高速カウンタコンパレータ CTC0、CTC1 の設定値により、パルス周期、パルス幅を決めます。

CTC0 の設定値＝パルス周期÷内部クロック ＝512(μs)÷6.4(μs)＝80

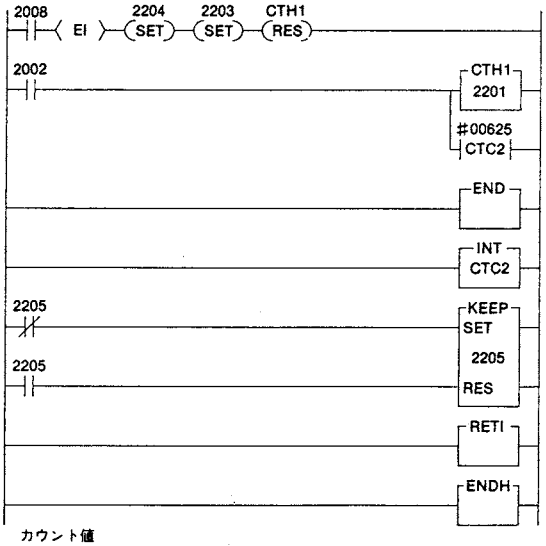
CTC1 の設定値＝パルス幅÷内部クロック ＝288(μs)÷6.4(μs)＝45

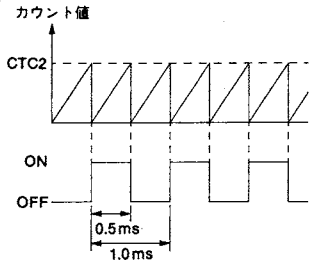
- CTH0 の入力リレーは、ダイレクトクロックパルス出力時は常時 ON(2002)を使用します。
- 出力可能なパルス周期 (周波数) は、500μs (2kHz)～26.1ms (38Hz) の範囲です。
- KZ-80 は 50μs (20kHz)～26.1ms (38Hz) の範囲です。

6章 高速カウンタの応用

6-3. ダイレクトクロックパルス使用例

(例 3)：出力リレー 0501 より周期 1ms (1kHz) のパルスを出力します。(ON−OFF 比 1：1)





コーディング

行番号	命令	オペランド
0000	LD	2008
0001	EI	
0002	CON	
0003	SET	2204
0004	CON	
0005	SET	2203
0006	CON	
0007	RES	CTH1
0008	LD	2002
0009	CTH	1 2201
0010	CTC	2 #00625
0011	END	
0012	INT	CTC2
0013	LDB	2205
0014	LD	2205
0015	KEEP	2205
0016	RETI	
0017	ENDH	

■解説

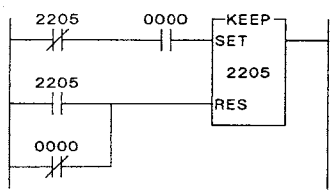
- 高速カウンタ CTH1 を使用するときは、割込みプログラム (INT～RETI) を使用します。
  割込みプログラム中で、特殊補助リレー2205を ON/OFF させることにより、0501の出力を ON−OFF させます。
- 運転開始時に EI 命令により、割込みの許可をします。
  同時に特殊補助リレー2204を ON することで0501へのダイレクト出力を許可し、2203を ON することで CTC2 による CTH1 のクリアを指定します。
- 運転開始時に CTH1 をクリアします。これにより、ダイレクトクロックパルスは 1 パルス目のパルス幅より正しく出力します。
- 高速カウンタ CTH1 のクロック入力に内部クロック2201 (パルス周期0.8μs) を指定します。
- 高速カウンタコンパレータ CTC2 の設定値により、パルス周期を決めます。

CTC2 の設定値＝パルス周期÷内部クロック÷ 2 ＝1000(μs) ÷0.8(μs) ÷ 2 ＝625

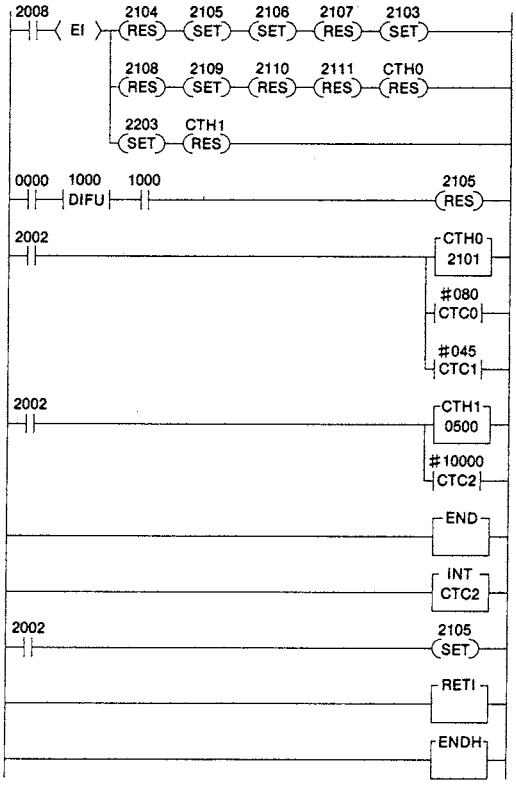
- CTH1 の入力リレーは、ダイレクトクロックパルス出力時は常時 ON(2002) を使用します。
- 出力可能なパルス周期 (周波数) は、666μs (1.5kHz)～419.4ms (2.4Hz) の範囲です。

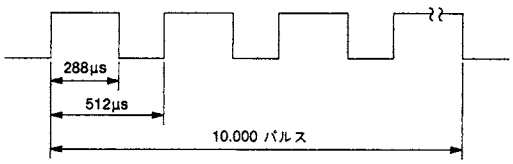
ワンポイント

- 外部入力の ON−OFF によりダイレクトクロックパルスの出力を制御する場合は、特殊補助リレー2205の ON/OFF によりおこないます。(CTH1 の場合)
  特殊補助リレー2205の ON/OFF は、SET、RES 命令または、KEEP 命令によりおこないます。上記例に右図の、プログラムを加えると、0000が ON のときに、ダイレクトクロックパルスが0501より出力されます。



(例 4)：出力リレー 0500 より周期 512μs、(約 1.95kHz)、パルス幅 288μs のパルスを 10,000 パルス出力します。





コーディング

行番号	命令	オペランド
0000	LD	2008
0001	EI	
0002	CON	
0003	MPS	
0004	RES	2104
0005	CON	
0006	SET	2105
0007	CON	
0008	SET	2106
0009	CON	
0010	RES	2107
0011	CON	
0012	SET	2103
0013	MRD	
0014	RES	2108
0015	CON	
0016	SET	2109
0017	CON	
0018	RES	2110
0019	CON	
0020	RES	2111
0021	CON	
0022	RES	CTH0
0023	MPP	
0024	SET	2203
0025	CON	
0026	RES	CTH1
0027	LD	0000
0028	DIFU	1000
0029	CON	
0030	AND	1000
0031	RES	2105
0032	LD	2002
0033	CTH	0 2101
0034	CTC	0 #080
0035	CTC	1 #045
0036	LD	2002
0037	CTH	1 0500
0038	CTC	2 #10000
0039	END	
0040	INT	CTC2
0041	LD	2002
0042	SET	2105
0043	RETI	
0044	ENDH	

■解説

- 高速カウンタ CTH0 によりダイレクトクロックパルスを0500より出力し、高速カウンタ CTH1 により、0500へ出力されるパルスをカウントします。
- 運転開始時に特殊補助リレー2104～2107、2108～2111により、出力0500へのダイレクト出力を許可し、0500への出力を CTC0 で ON、CTC1 で OFF するように設定します。
  同時に特殊補助リレー2103 を ON することで、CTC0 による CTH0 のクリアを指定します。
- 運転開始時に EI 命令により、割込みを許可します。
  割込みのプログラムにより、高速カウンタ CTH1 の現在値が高速カウンタコンパレータ CTC2 の設定値と等しくなったときに、ダイレクトクロックパルスを停止します。
- 運転開始時に、特殊補助リレー2203を ON することで、CTC2 による CTH1 のクリアを指定します。また、高速カウンタ CTH0、CTH1 の現在値をクリアします。
- ダイレクトクロックパルス出力の開始は、入力0000を ON することにより出力します。
  10,000パルス出力するとダイレクトクロックパルスは停止します。再スタートは、0000を ON させます。

6章 高速カウンタの応用

6-3. ダイレクトクロックパルス使用例

●高速カウンタ CTH0 のクロック入力に内部クロック2101（パルス周期6.4μs）を指定し、高速カウンタ CTH1 のクロック入力に0500と指定します。
これにより CTH0 はダイレクトクロックパルス出力用、CTH1 はダイレクトクロックパルスカウント用として使用します。

●高速カウンタコンパレータ CTC0、CTC1 の設定値により、パルス周期、パルス幅を決めます。

CTC0 の設定値＝パルス周期÷内部クロック ＝512(μs)÷6.4(μs)＝80

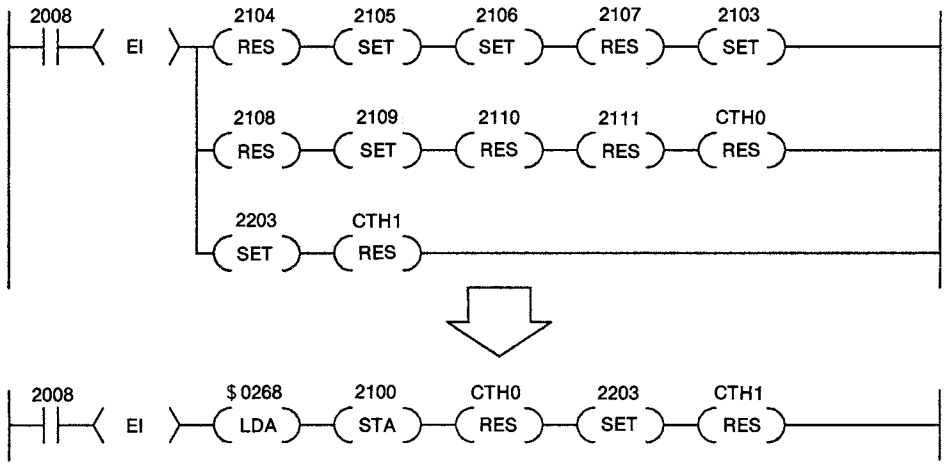
CTC1 の設定値＝パルス幅÷内部クロック ＝288(μs)÷6.4(μs)＝45

●高速カウンタコンパレータ CTC2 の設定値によりダイレクトクロックパルスのパルス数を設定します。

● CTH0 の入力リレーは、ダイレクトクロックパルス出力時は常時 ON（2002）を使用します。

ワンポイント

●特殊補助リレー2102～2111の ON－OFF は、SET、RES 命令の他に LDA、STA 命令でも設定できます。
例 4 を LDA、STA 命令で書き直すと下図のようになり、行数、ステップ数を少なくできます。



●LDA $0268について
LDA $0268を実行すると、内部レジスタへは次のように入力されます。

	〔0〕				〔2〕				〔6〕				〔8〕			
内部レジスタ	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	0

STA 2100を実行すると、内部レジスタの値が特殊補助リレー2100～2115へ割り付けられます。

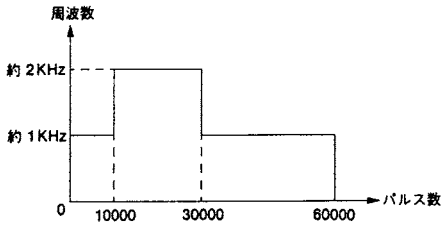
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
21XX	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	0

これより、2103,2105,2106,2109が ON となります。

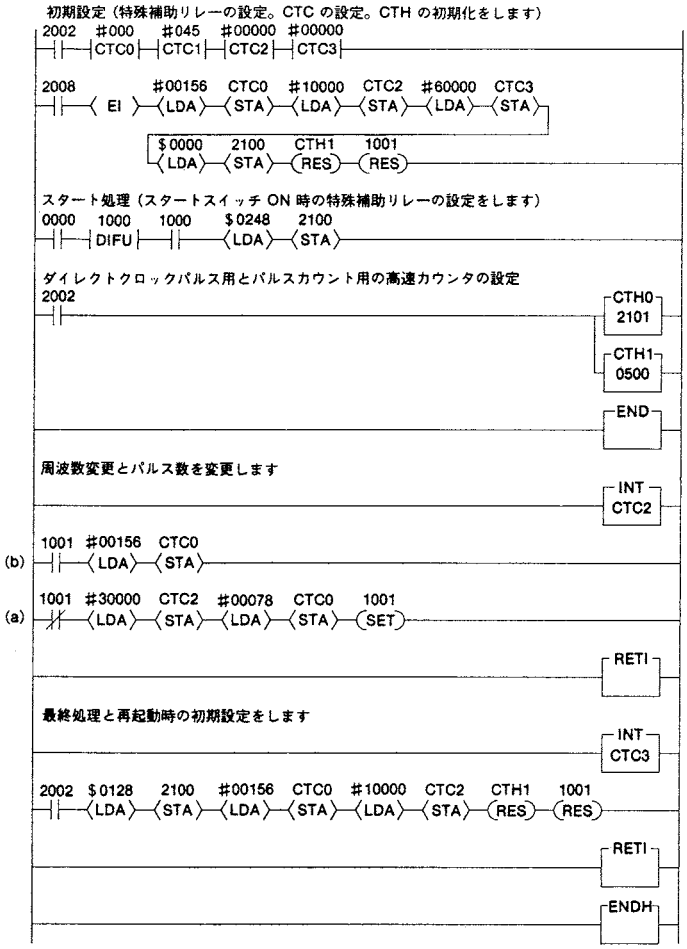
ダイレクトクロックパルスの応用例

■速度制御の例

●パルス数に合わせてダイレクトクロックパルスの周波数を変化させ、下図のような制御をします。動作は、スタートスイッチを押すとパルス数0～10000間は周波数約1kHzで、パルス数10000～30000間は周波数約2kHzで、パルス数30000～60000間は周波数約1kHzでダイレクトクロックパルスを出力します。
パルス数60000までくるとダイレクトクロックパルスは停止します。
再起動は、スタートスイッチを押すことによりおこないます。



●プログラム例



●割込みを許可します。
● CTCの設定値を設定します。
●特殊補助リレー2103～2111を設定します。
● CTH1の現在値をクリアします。

●スタートスイッチの立ち上がりで特殊補助リレー2103～2111を設定します。

●高速カウンタCTH0により、ダイレクトクロックパルスを発生させます。

●高速カウンタCTH1により、ダイレクトクロックパルスのパルス数をカウントします。

●パルス数10000の点で、30000の点の値とダイレクトクロックパルスの周波数を変更します。((a)部分)

●パルス数30000の点で、ダイレクトクロックパルスの周波数を変更します。((b)部分)

●ダイレクトクロックパルスのパルス数がCTC3の設定値と等しくなったときにダイレクトクロックパルスの出力を停止し、再起動時の初期設定をします。

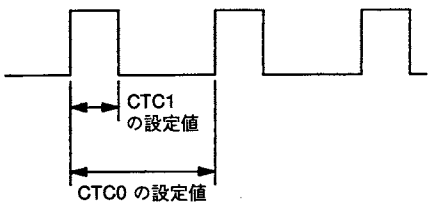
6章 高速カウンタの応用

KZ

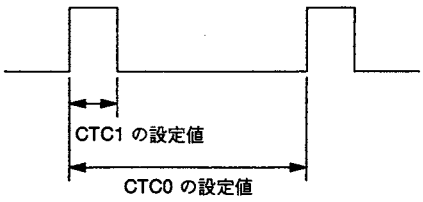
6-4. ダイレクトクロックパルスの応用例

■解説

ダイレクトクロックパルスのパルス周期はCTC0の設定値により決まります。

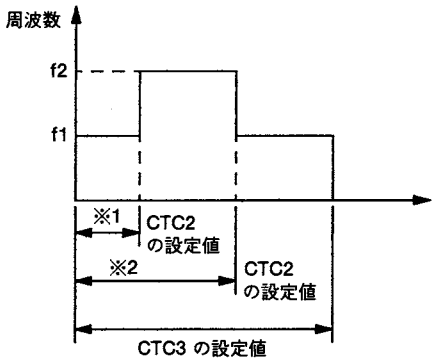


(a) 周波数約 2kHz のとき



(b) 周波数約 1kHz のとき

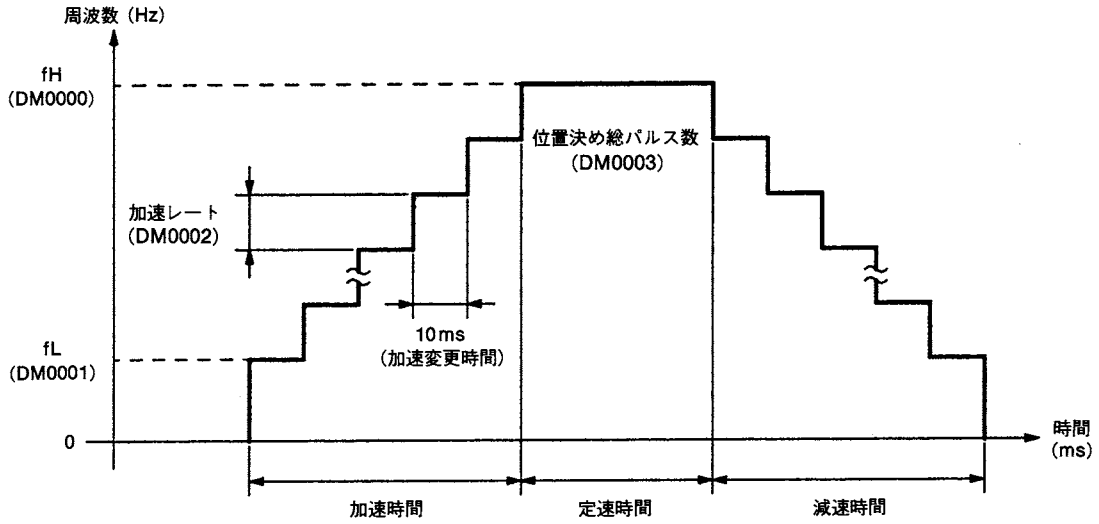
周波数の変化点はCTC2の設定値により決まります。停止点はCTC3の設定値で決まります。



- ※1の部分のCTC2の設定値は、初期設定の値が設定されます。
- ※2の部分のCTC2の設定値は、INT CTC2で指定される割込みプログラムの中で変更されます。

台形制御

●4つのパラメータ(DM0000～DM0003)を設定することにより下図のような台形制御を行います。
(例1～4とは違いCTC0～3の設定は内部のプログラムによって計算しています。)



設定するパラメータは以下の 4 つ

DM0000：fH〔Hz〕(運転周波数)
DM0001：fL〔Hz〕(起動周波数)
⇨ 612Hz～20000Hz で必ず fH＞fL (KZ-80T 以外は 612～2000Hz)

DM0002：加速レート〔Hz/10ms〕※ ⇨ $(fH-fL)\div\frac{加速時間}{10ms}$

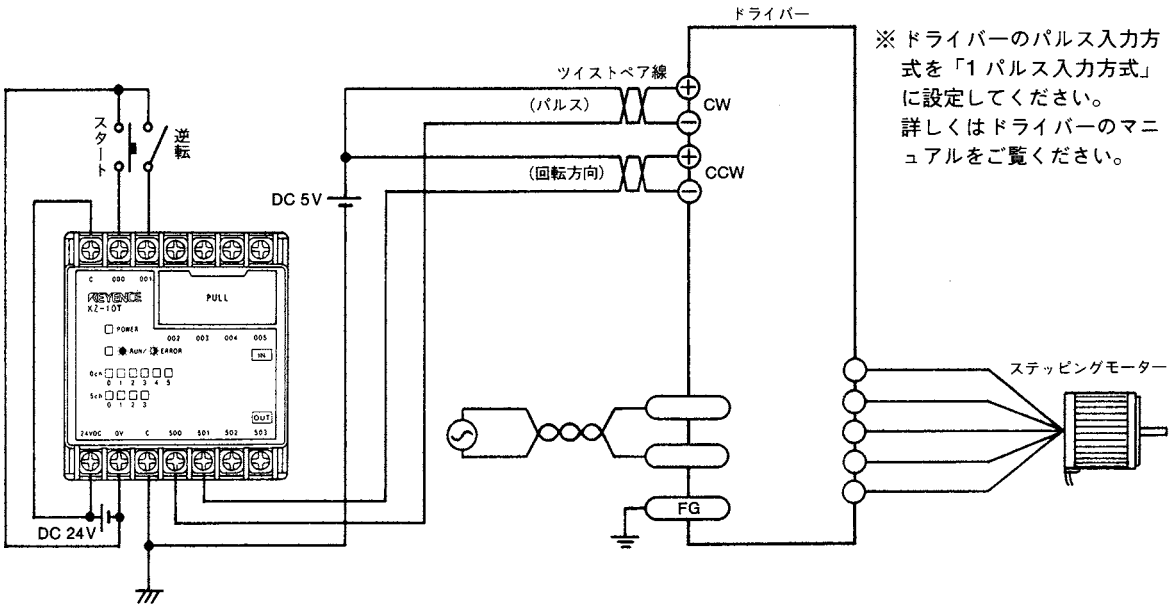
DM0003：位置決め総パルス数 ⇨ 65535 パルス以下

※ 加速レートが小さいほど滑らかな制御になりますが加速時間が長くなります。
※※ 必ず定速時間が存在するように各パラメータを設定してください。定速時間がない場合正常な台形制御を行いません。(その場合 fH の周波数を小さくしてください)

動作

プログラム例では
DM0000＝20000〔Hz〕、DM0001＝700〔Hz〕、DM0002＝483〔Hz/10ms〕(加速時間 400ms)、
DM0003＝10000 パルス
とし、スタートスイッチ 0000 が押されると上記のように台形制御を行い、10000 パルスのカウントを行うと停止します。
再起動はスタートスイッチを押すことにより行い、逆転スイッチ 0001 が ON している間は逆転動作を行います。

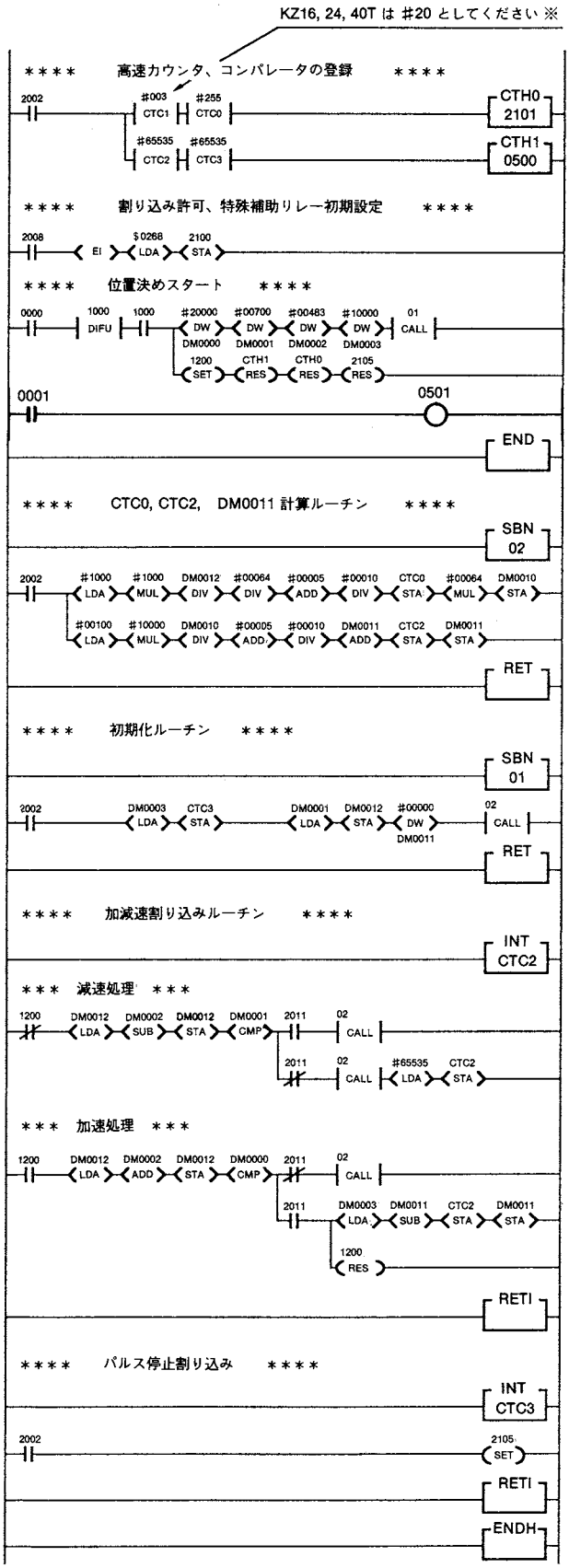
接続例



6章 高速カウンタの応用

KZ

6-4. ダイレクトクロックパルスの応用例



●高速カウンタ CTH0 により、クロックパルスを発生させます。

●高速カウンタ CTH1 によりダイレクトクロックパルスのパルス数をカウントします。

CTC0： 周期　CTC1： パルス幅
CTC2： 周波数変更点
CTC3： 位置決め総パルス数

●パラメータ (DM0000～DM0003) を設定します。

●逆転出力

●CTC0 の設定を行います。
CTC0 の設定値＝1÷DM012×10^3×10^3÷6.4

●CTC2 の設定を行います。

DM0010：周期 T (μs)×10
DM0011：累積パルス数
DM0012：現在周波数 f (Hz)
1200：　加速中フラグ

●加速中フラグ 1200 が OFF しているときに CTC2 が ON するごとに現在周波数から加速レートを減算し周波数を変更します。現在周波数が fL になると減速をやめ初期設定をします。

●加速中フラグ 1200 が ON しているときに CTC2 が ON するごとに現在周波数に加速レートを加算し周波数を変更します。現在周波数が fH になると加速をやめ、減速位置パルス数を設定します。

●ダイレクトクロックパルスのパルス数が CTC3 の設定値と等しくなったときにダイレクトクロックパルスの出力を停止します。

※パルスの ON 時間 (CTC1 の値) は #3 もしくは #20 の固定です。負荷により十分な ON 時間が得られない場合は設定値を大きくしてください。

7章　シリアル通信機能

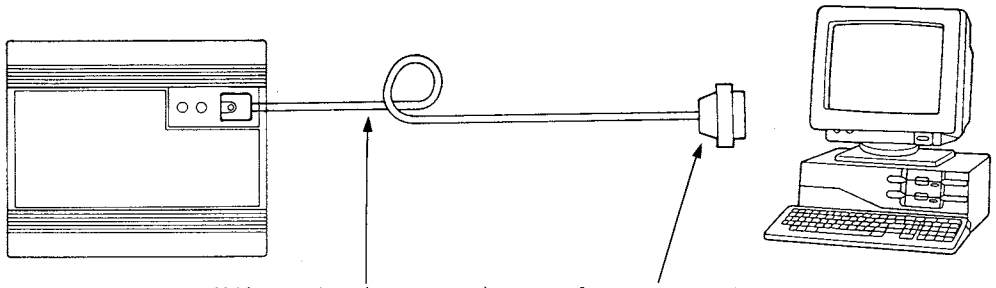
KZを、通信用インターフェイスとしてRS−232Cを持つ機器と接続して、通信するときの方法を説明します。

KZとパソコンの通信

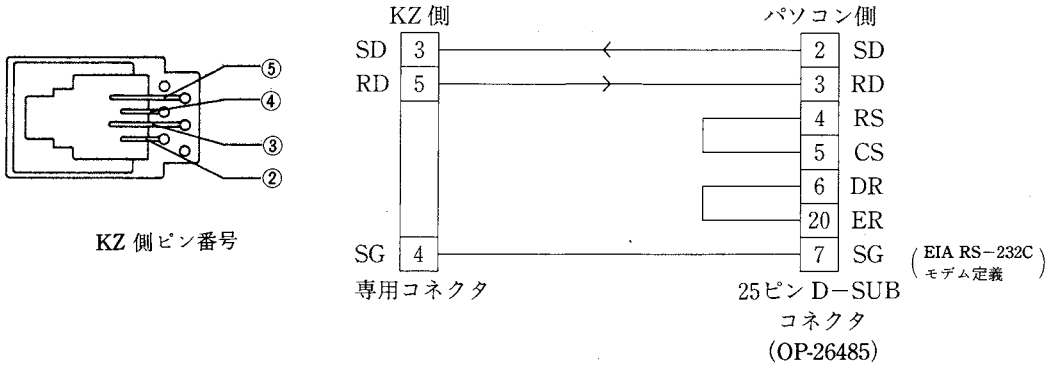
KZ とパソコンを接続して、KZ 内のカウンタ、タイマの現在値や設定値、DM の値を読み出したり、書き替えたりすることができます。

■ KZ とパソコンの接続方法

●KZ 本体とパソコンは、専用の接続ケーブル（ストレート：OP-26487）と 25 ピン D−SUB コネクター（OP-26485）を使用し、パソコンの RS−232C のコネクタへ接続します。



●RS−232C のケーブルの配線は下図のようになっています。KZ は必ず専用の接続ケーブルにて接続してください。



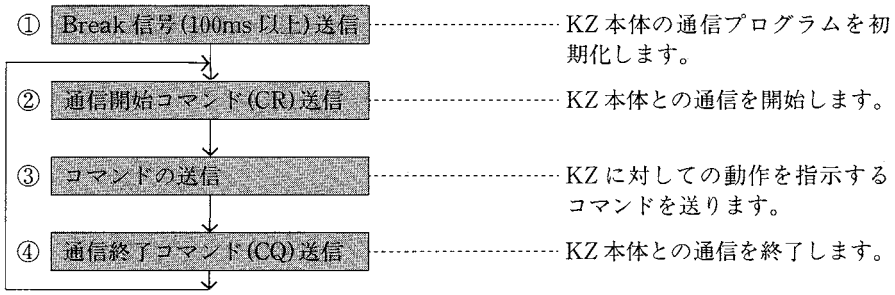
■通信仕様

●KZ の通信仕様は右表のようになっています。
右表にあわせて、パソコンの通信条件を設定してください。
※パソコンの設定に関しては、お手持ちのパソコンの取扱い説明書をご参照ください。

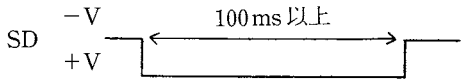
通信方式	全 2 重
同期方式	調歩同期
伝送コード	ASCII
ボーレート	9,600bit/s
データビット長	8 ビット
パリティチェック	偶 数
ストップビット長	1 ビット
データ区切	CR

■コマンド送信手順

●パソコン側からKZ本体に対してコマンドを送るときの送信手順は次のようにおこないます。



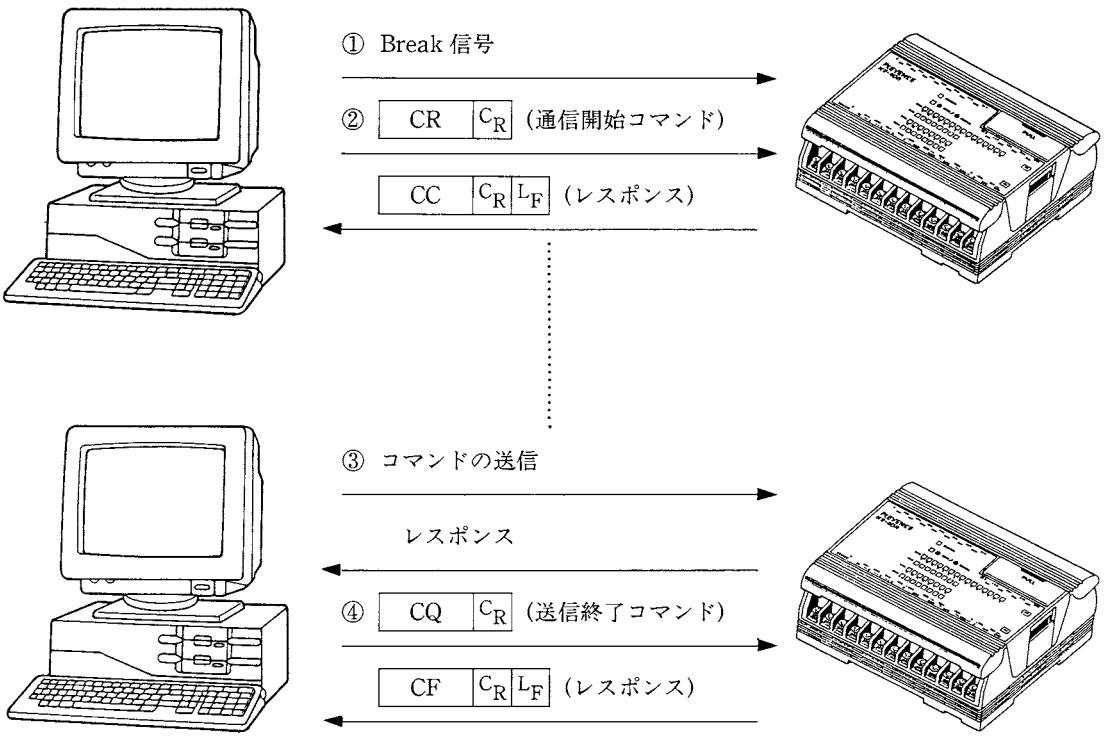
● Break信号は、下図のように送信してください。



Break信号はKZ本体の通信プログラムを初期化します。
KZ本体にKZ−P3を接続した直後にパソコンと通信する場合には必ず送信してください。

● KZ本体は、パソコンからのコマンドの処理を実行し、それに対しての返答をします。

●実際のコマンド送信手順の例を下図に示します。図中の①～④までは、送信手順の①～④に対応します。



※①、②、④は必ず実行してください。

(通信コマンドの説明 → P7−3)

7-1. KZとパソコンの通信

■コマンド/レスポンスのフォーマット

● KZ とパソコンの通信においてコマンドとレスポンスは次のように定義します。
コ マ ン ド：パソコンから KZ へのメッセージをコマンドといいます。
レスポンス：コマンドに対する応答(KZ からパソコンへのメッセージ)をレスポンスといいます。

●コマンドとレスポンスのそれぞれのフォーマットは次のようになります。

コマンドフォーマット

コマンド	C_R

又は、

コマンド	C_R	L_F

レスポンスフォーマット

レスポンス	C_R	L_F

●パソコンから KZ へコマンドを送信するときは、コマンドフォーマットに従って送ってください。
KZ 内では CR を区切り符号として識別しますので LF は無視します。
●KZ は、コマンドに対する応答をレスポンスフォーマットに従って返します。
パソコンでプログラムを作成される場合はレスポンスフォーマットに従って処理するようにプログラムしてください。(参考プログラム → P7-11)

■通信コマンドの説明

KZ とパソコンの通信に使用するコマンドと正常に処理したときのレスポンスについて説明します。

通信開始コマンド……KZ とパソコンの通信を開始します。
コ マ ン ド：CR
レスポンス：CC ……正常処理

通信終了コマンド……KZ とパソコンの通信を終了します。
コ マ ン ド：CQ
レスポンス：CF ……正常処理

モード変更コマンド……プログラムモード / 運転モードを切替えます。
コ マ ン ド：Mn　　[n=0：プログラムモード, n=1：運転モード]
レスポンス：OK ……正常処理

エラークリアコマンド……KZ 本体のエラーの解除をします。
コ マ ン ド：ER
レスポンス：OK ……正常処理

本体の異常確認……KZ 本体のエラーや異常をコードで返します。
コ マ ン ド：?E
レスポンス：dd　　[dd ：エラーコードを 10 進数で示します。
(エラーコード異常コード表 → P7−8)]

機種の問い合わせ……現在使用中のKZの機種を型式で返します。

コマンド：?K
レスポンス：KZdd　　[dd：機種型式　10/16/24/40/80]

本体動作モードの確認……本体が動作中のモードをコードで返します。

コマンド：?M
レスポンス：0 ……プログラムモード
　　1 ……運転モード

カウンタの読み出し……指定されたカウンタの接点のON/OFF状態、現在値、設定値の内容を読み出します。

コマンド：RD␣Cnnn …カウンタ、アップダウンカウンタの内容を読み出します。
　　[nnn＝カウンタ番号]
　　RD␣CTHn…高速カウンタCTHの内容を読み出します。
　　[n＝高速カウンタ番号]
　　RD␣CTCn…高速カウンタコンパレータの内容を読み出します。
　　[n＝高速カウンタコンパレータ番号]
レスポンス：f␣ddddd␣ddddd　　[f＝0,1：カウンタ接点のON/OFF状態　0：OFF,1：ON]
　　[ddddd ddddd＝現在値、設定値]

タイマの読み出し……指定されたタイマの接点のON/OFF状態、現在値、設定値の内容を読み出します。

コマンド：RD␣Tnnn　　[nnn＝タイマ番号]
レスポンス：f␣ddddd␣ddddd　　[f＝0,1：タイマ接点のON/OFF状態　0：OFF,1：ON]
　　[ddddd ddddd＝現在値、設定値]

リレー接点の読み出し……指定されたリレー番号の接点のON/OFF状態を読み出します。

コマンド：RD␣nnnnn　[nnnnn：リレー番号]
レスポンス：0 ……リレー接点のOFF状態
　　1 ……リレー接点のON状態

DMの読み出し……指定されたデータメモリの内容を読み出します。

コマンド：RD␣DMnnnn　　[nnnn：データメモリ番号]
レスポンス：ddddd　　[ddddd＝00000～65535：10進数]

DMの書き込み……指定されたデータメモリに数値を書き込みます。

コマンド：WR␣DMnnnn␣ddddd　[nnnn：データメモリ番号]
　　[ddddd＝00000～65535：10進数]
レスポンス：OK ……正常処理

TMの読み出し……指定されたテンポラリデータメモリの内容を読み出します。

コマンド：RD␣TMnn　[nn＝00～31：テンポラリデータメモリ番号]
レスポンス：ddddd　　[ddddd＝00000～65535：10進数]

TMの書き込み……指定されたテンポラリデータメモリに数値を書き込みます。

コマンド：WR␣TMnn␣ddddd　[nn = 00～29 ：テンポラリデータメモリ番号]
[ddddd = 00000～65535 ： 10 進数]

レスポンス：OK　……正常処理

アナログタイマの読み出し……アナログタイマのトリマの値を読み出します。

コマンド：RD ␣ AT

レスポンス：ddddd␣ddddd␣00000␣00000　[ddddd = 00000～00249 ：10進数]

トリマ１の値（00000～00249）

トリマ０の値（00000～00249）

カウンタ・タイマの現在値変更……指定されたカウンタ、タイマの現在値を変更します。

コマンド：WR␣Cnnn␣ddddd …カウンタ、アップダウンカウンタの現在値を変更します。
[nnn ＝カウンタ番号]
[ddddd = 00000～65535 ： 10 進数]

WR␣CTHn ␣ ddddd…高速カウンタ CTH の現在値を変更します。
[n ＝高速カウンタ番号]
[ddddd = 00000～65535 ： 10 進数
但し CTH0 は 00000～00255]

WR␣Tnnn␣ddddd …タイマの現在値を変更します。
[nnn ＝タイマ番号]
[ddddd = 00000～65535 ： 10 進数]

レスポンス：OK　……正常処理

カウンタ・タイマの設定値変更……指定されたカウンタ、タイマの設定値を変更します。

コマンド：WS␣Cnnn␣ ddddd …カウンタ、アップダウンカウンタの設定値を変更します。
[nnn ＝カウンタ番号]
[ddddd = 00000～65535 ： 10 進数]

WS␣CTCn ␣ ddddd…高速カウンタコンパレータ CTC の設定値を変更します。
[n ＝高速カウンタコンパレータ番号]
[ddddd=00000～65535：10 進数
但し CTC0, CTC1 は 00000～00255]

WS␣Tnnn ␣ ddddd …タイマの設定値を変更します。
[nnn ＝タイマ番号]
[ddddd = 00000～65535 ： 10 進数]

レスポンス：OK　……正常処理

*カウンタ・タイマの設定値変更時、EEP ROM の書き換えを行ないます。

接点の強制セット……指定されたリレー、カウンタ、タイマの接点を強制的にセットします。

コマンド：ST␣nnnnn　リレーの接点を強制的にセットします。
[nnnnn ＝リレー番号]

ST␣Cnnn …カウンタの接点を強制的にセットします。
[nnn ＝カウンタ番号]

ST␣Tnnn…タイマの接点を強制的にセットします。
[nnn ＝タイマ番号]

レスポンス：OK ……正常処理

接点の強制リセット……指定されたリレー、カウンタ、タイマの接点を強制的にリセットします。

コマンド：RS␣nnnnn…リレーの接点を強制的にリセットします。
[nnnnn＝リレー番号]

RS␣Cnnn …カウンタの接点を強制的にリセットします。
[nnn＝カウンタ番号]

RS␣CTCn ‥高速カウンタコンパレータの接点を強制的にリセットします。
[n＝高速カウンタコンパレータ番号]

RS␣Tnnn …タイマの接点を強制的にリセットします。
[nnn タイマ番号]

レスポンス：OK ……正常処理

■その他のレスポンスコードについて

KZは、コマンドに対応したレスポンスを返しますが、送られてきたコマンドが指定した以外のものや本体に異常が発生した場合に、状態に応じたレスポンスのコードを返します。

レスポンスコード	内　容	原　因	処　置
E0	リレー番号異常	●定義されていないリレー、カウンタ、タイマ、DM、CTH、CTC の番号を指定した。 ●プログラム中に使用されていないカウンタ、タイマ、CTH、CTC の番号を指定した。	●各機種で定義されている範囲の番号を指定してください。 ●プログラムを確認のうえ、プログラム中で使用している番号を指定してください。
E1	コマンド異常	●定義されていないコマンドを送った。 ●コマンドの指定がまちがっている。 ●通信確立以前に「CR」以外のコマンドを送った。	●コマンドを確認してください。 ●「CR」を送って通信を開始してからコマンドを送ってください。
E2	プログラム未登録	●本体にプログラムが入っていないのに「M1」を送った。または、カウンタ、タイマの読み出しをおこなった。	●本体にプログラムを登録してください。
E3	本体異常	●本体ハードウェア上の異常です。	●KZ 本体の電源を入れ直してください。それでも直らなければ故障が考えられます。最寄りの営業所までお問い合わせください。
E4	書き込み禁止	●ライトプロテクトのプログラムに対してカウンタ、タイマ、CTC の設定値を変更しようとした。	●ライトプロテクトを解除してから設定値を変更してください。
E5	本体エラー	●本体のエラーを解除せずに「M1」を送った。	●本体のエラーを解除し、原因を取り除いてください。

■通信コマンド表

機　　能	コマンド	レスポンス	解　　説
通信開始コマンド	CR	CC	
通信終了コマンド	CQ	CF	
モード変更コマンド	Mn	OK	n＝0：プログラムモード n＝1：運転モード
エラークリアモード	ER	OK	
本体の異常確認	?E	dd	dd：エラーコード(エラーコード表参照)
機種の問い合わせ	?K	KZdd	dd：機種型式 10/16/24/40/80
本体動作モードの確認	?M	0 1	0：プログラムモード 1：運転モード
カウンタの読み出し	RD Cnnn RD CTHn RD CTCn	f ddddd ddddd	f＝0,1　0：接点OFF 1：接点ON ddddd ddddd＝現在値、認定値
タイマの読み出し	RD Tnnn	f ddddd ddddd	f＝0,1　0：接点OFF 1：接点ON ddddd ddddd＝現在値、設定値
リレー接点の読み出し	RD nnnnn	0 1	nnnnn＝リレー番号 0：接点OFF、1：接点ON
DMの読み出し	RD DMnnnn	ddddd	nnnn＝データメモリ番号 ddddd＝00000～65535（10進数）
DMの書き込み	WR DMnnnn ddddd	OK	nnnnn＝データメモリ番号 ddddd＝00000～65535（10進数）
TMの読み出し	RD TMnn	ddddd	nn＝テンポラリデータメモリ番号 ddddd＝00000～65535（10進数）
TMの書き込み	WR TMnn ddddd	OK	nn＝テンポラリデータメモリ番号(注) ddddd＝00000～65535（10進数）
アナログタイマの読み出し	RD AT	ddddd ddddd 00000 00000	ddddd＝00000～00249（10進数） トリマ0、トリマ1

(注)TM30、TM31へは書き込めません。

■通信コマンド表

機　能	コマンド	レスポンス	解　説
カウンタ・タイマの現在値変更	WR Cnnn ddddd WR CTHn ddddd WR Tnnn ddddd	OK	n/nnn＝カウンタ・タイマ番号 ddddd＝00000～65535（10進数） 但しCTH0は00000～00255（10進数）
カウンタ・タイマの設定値変更	WS Cnnn ddddd WS CTCn ddddd WS Tnnn ddddd	OK	n/nnn＝カウンタ・タイマ番号 ddddd＝00000～65535（10進数） 但しCTC0、CTC1は00000～00255（10進数）
接点の強制セット	ST nnnnn ST Cnnn ST Tnnn	OK	nnnnn＝リレー番号 nnn＝カウンタ・タイマ番号
接点の強制リセット	RS nnnnn RS Cnnn RS CTCn RS Tnnn	OK	n＝高速カウンタコンパレータ番号 nnn＝カウンタ・タイマ番号 nnnnn＝リレー番号

■エラーコード表

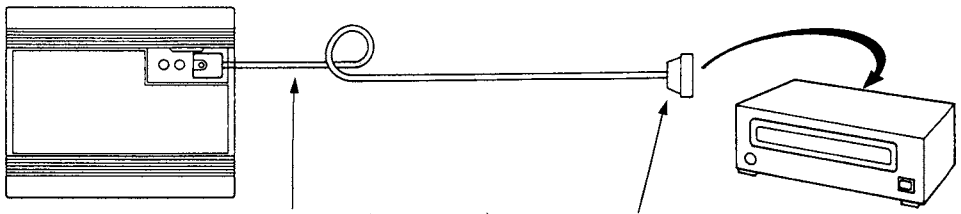
エラーコード	内　容	原　因	処　置
00	エラー無し		
10	CALLネストエラー	●サブルーチンのネスティング（入れ子）が4重をこえています。	●これらのエラーはプログラムの実行時に発生するエラーです。 サブルーチンの使い方、INT命令の使い方に特に注意してプログラムを見直してください。
11	FORネストエラー	●FOR～NEXT命令のネスティング（入れ子）が8重をこえています。	
12	INTネストエラー	●同時に5重以上の割込みが発生しました。	
13	MPSネストエラー	●MPS～MPP命令のネスティング（入れ子）構造が8重をこえています。	
20	メモリーエラー	●RAMの内容が消失したために、本体のデータメモリの値、カウンタの現在値がクリアされてしまいました。	●データメモリの現在値、カウンタの値を確認してください。
30	サイクルタイムエラー	●サイクルタイムが208ms以上になっています。	●プログラムを再度見直してください。 ※FOR～NEXT命令で回数の多い繰り返しを行っていないか確認してください。 ※割り込みのプログラムを速いレートで実行させていないか確認してください。
40	本体クリアエラー	●プログラムの登録されていないKZを動かそうとしています。	●本体にプログラムを書き込んでください。
50	本体通信エラー	●本体のハードウェア上のエラーです。	●KZ本体の電源を入れ直してください。 それでも直らなければ、故障が考えられます。最寄りの営業所までお問い合わせください。
51	0除算エラー		
52	アドレスエラー		
53	不当命令エラー		
54	ウォッチ・ドッグ・タイマエラー		

テキストデータの取り込み

KZ はRS−232C の端子を持つ機器と接続して、連続したテキストデータ(100バイト未満)を一括してデータメモリに書き込む機能があります。
これら取り込んだデータをプログラムの中で加工して使用できます。

■KZ と RS−232C の端子を持つ機器との接続

●KZ 本体と RS−232C の端子を持つ機器との接続は、専用の接続ケーブル（ストレート：OP-96368、リバース：OP-96607)と25ピン D-SUB コネクタ（OP-96369）を使用して RS−232C のコネクタへ接続します。



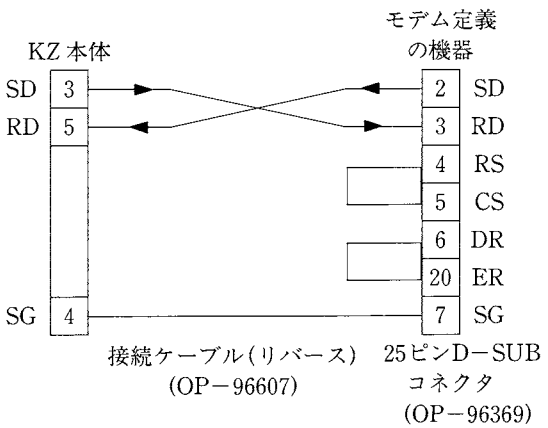
KZ 本体　接続ケーブル（ストレート）(OP-96368 または OP-26487）接続ケーブル（リバース）(OP-96607)　25ピン D−SUB コネクタ (OP−96369)　RS−232C 端子を持った機器

■通信仕様

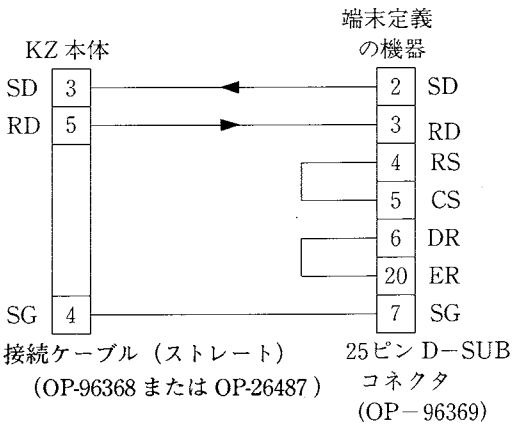
●KZ の通信仕様は、右表のようになっています。右表にあわせて、機器の通信条件を設定してください。
※機器の設定に関しましては、お手持ちの機器の取扱い説明書をご参照ください。

通信方式	全2重
同期方式	調歩同期
伝送コード	ASCII
ボーレート	9,600bit/s
データビット長	8ビット
パリティチェック	偶数
ストップ・ビット長	1ビット

●専用ケーブルで、RS−232C の端子を持つ機器との接続は下図のようにおこなってください。
(KZ 側のピン番号は P7-1 を参照してください。)



（a）モデム定義の機器との接続



（b）端末定義の機器との接続

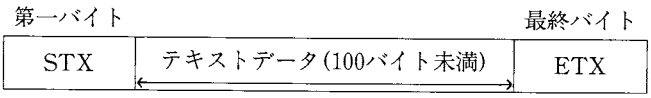
※モデム定義の機器側の RS、CS、DR、ER については必要に応じて配線してください。

■通信データフォーマットと内部データメモリについて

●KZ はSTX からETX 間の100バイト未満のテキストデータを内部のデータメモリ（DM700～DM799）に1バイトずつ書き込みます。

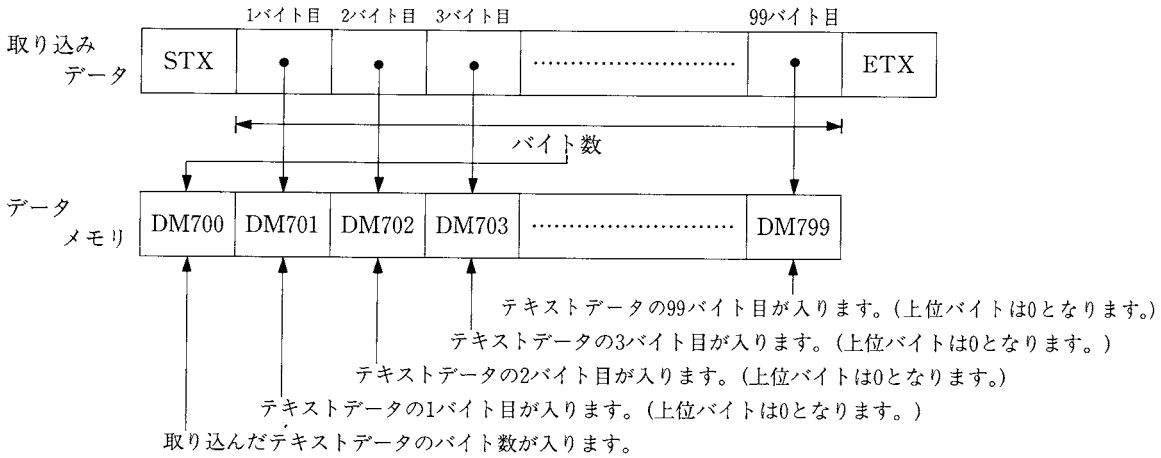
●通信データフォーマット

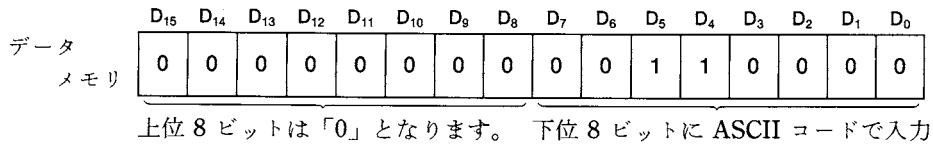
テキストデータのフォーマットは、第1バイト目をSTX（02H）、第2バイト目以降（100バイト未満）をデータとし、最終バイトをETX（03H）としてください。（下図参照）



●内部データメモリについて

KZ は、取り込んだテキストデータを内部のデータメモリDM700～DM799へ1バイトずつ書き込みます。





● KZ はテキストデータの取り込みを完了すると、特殊補助リレー2804を1スキャンだけONします。

■テキストデータを取り込む場合の注意点

●テキストデータの読み出しデータ中にETX がある場合は、その直前までのテキストデータをDM に転送します。

●テキストデータ受信中に、パリティエラー等が発生した場合は、そのテキストデータはDM へ転送されず、特殊補助リレー2805、2806を1スキャンだけONします。

●プログラミングコンソール使用直後は、KZ 本体の通信プログラムを初期化する必要があります。
(P. 7−2 参照)

● KZ 本体へのテキストデータの取り込みは、特殊補助リレー2804がOFF のときのみ行います。

●特殊補助リレー2804がON しているときにテキストデータを取り込んだ場合は、DM へ転送されず、特殊補助リレー2805が1スキャンだけONします。

参考プログラム

■ KZとパソコンの通信プログラム例（PC−9801用 N88−BASIC）

```
100 '＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊
110 '＊　入力した文字列をＫＺにコマンドとして送信しレスポンスを　＊
120 '＊　画面に表示します　　＊
130 '＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊
140 '　ＲＳ−２３２Ｃ初期化
150 ＯＰＥＮ␣"ＣＯＭ：Ｅ８１ＮＮ"ＡＳ　＃１　　'ＲＳ−２３２Ｃをオープンします
160 '
170 '　　ブレーク信号の送信（Ｐ３使用後は必要です。その他は必要ありません）
180 '　Ｉ／Ｏポートの&Ｈ３２は、ＲＳ−２３２Ｃ制御線をコントロールする
190 '　　Ｉ／Ｏアドレスです。第３ビットを１にするとブレーク信号の送信
200 '　　を開始します。また、０にするとブレーク信号の送信を止めます。
210 ＯＵＴ　&Ｈ３２，&Ｈ３Ｆ　　'ブレーク信号の送信開始
220 ＦＯＲ　Ｉ＝０　ＴＯ　１０００：　ＮＥＸＴ　Ｉ　　'待ち時間
230 ＯＵＴ　&Ｈ３２，&Ｈ３７　　'ブレーク信号送信終了
240 '
250 '　＊＊＊　　コマンド送信／レスポンス受信　　＊＊＊
260 Ａ$＝" "
270 ＷＨＩＬＥ　Ａ$＜＞"ＥＮＤ"
280　　ＰＲＩＮＴ　"Ｃｏｍｍａｎｄ＞"；：ＩＮＰＵＴ　Ａ$　　'コマンド入力待ち
290　　ＰＲＩＮＴ　＃１，Ａ$　　'コマンドをＫＺに送信
300　　ＬＩＮＥ　ＩＮＰＵＴ　＃１，Ｂ$　　'レスポンス受信待ち
310　　ＰＲＩＮＴ　"Ｒｓｐ［";Ｂ$;"]"　　'レスポンス表示
320　　ＰＲＩＮＴ
330 ＷＥＮＤ
340 ＣＬＯＳＥ　＃１
350 ＰＲＩＮＴ　"ＧＯＯＤ　ＢＹＥ！！"
360 ＥＮＤ
```

※150～230行目でBreak信号を送信します。
　パソコンのクロック周波数により、220行目のＩの値を変更して100msec以上になるようにしてください。
　Break信号はKZ−P3を使用した直後にKZ本体の通信プログラムを初期化するために必要です。
※パソコンのボーレートはあらかじめSwitchコマンド等により9600bit/sに設定しておいてください。
※ENDと入力するとプログラムは終了します。

8章　ファンクション

ファンクション番号一覧

KZには、プログラミング作業や試運転時の作業をスムーズかつ確実に行うために、16種類の付加機能が搭載されています。
これらの機能にはそれぞれ「ファンクション番号」(FUN××)が割当てられており、基本命令・応用命令・演算命令・割り込み命令の入力と同様に FUN キーで設定します。

■ファンクション番号一覧

機能	FUN番号	モード	記載ページ
オールクリア	60	P	8-2
プロコンクリア	61	P E	8-2
カウンタクリア	62	P	8-3
高速カウンタクリア	63	P	8-3
データメモリオールクリア	64	P	8-4
保持リレーオールクリア	65	P	8-4
プログラム送受信	66	P E	8-5
オフラインエディタ開始	67	R	8-6
オフラインエディタ終了	68	E	8-6

機能	FUN番号	モード	記載ページ
タイマ・カウンタ現在値変更	69	P R	8-7
タイマ・カウンタ設定値変更	70	P R	8-9
リレーのON/OFF	71	P R	8-11
データメモリ書込み	72	P R	8-12
トリマ設定値読出	73	P R E	8-13
文法チェック	74	P E	8-13
プログラム容量確認	75	P E	8-14

P：PROGRAMモード
R：RUNモード(オフラインエディタ未使用の場合)
E：オフラインエディタ起動中

ワンポイント

ファンクションナンバーが分からない場合

FUN キーを押して、FUNナンバーにカーソルがある時、

↑ON ↓OFF キーで使用可能なファンクションが順番に表示されます。

(例)

FUN[60]
オールクリア

↑ON ↓OFF

FUN[61]
プロコンクリア

オールクリア：FUN60

PROGRAMモード

■機能：現在KZ本体内およびプログラミングコンソール(KZ-P3)内に書かれているプログラムをすべて消去します。

●オールクリア後は、全行「NOP」命令が入ります。

●オールクリアを行っても、「データメモリ」のデータはクリアされずに保存されます。
カウンタ、高速カウンタ、内部補助リレー(保持)はクリアされます。

■キー操作

《基本操作》

FUN → ORB 6 → RES 0 → 実行 逆検索 → 実行 逆検索

〔操 作 例〕

キー操作	表示	説明
	0 0 0 0 0 L D　　0 0 0 0 0	プログラムの任意の行を表示します。
FUN ORB 6 RES 0 実行 逆検索	オールクリア ヨロシイデスカ？	オールクリアの機能を呼び出します。
実行 逆検索	シュウリョウシマシタ	オールクリアを実行。
クリア	0 0 0 0 0 N O P	

プロコンクリア：FUN61

PROGRAMモード
オフラインエディタ

■機能：現在プログラミングコンソール(KZ-P3)内に書かれているプログラムをすべて消去します。

●KZ本体内のプログラムは消えません。

KZ本体がリードプロテクトの場合でも、このFUNで新しいプログラムの編集ができます。

■キー操作

《基本操作》

FUN → ORB 6 → SET 1 → 実行 逆検索 → 実行 逆検索

〔操 作 例〕

キー操作	表示	説明
	0 0 0 0 0 L D　　0 0 0 0 0	プログラムの任意の行を表示します。
FUN ORB 6 SET 1 実行 逆検索	プロコン　クリア ヨロシイデスカ？	プロコンクリアの機能を呼び出します。
実行 逆検索	シュウリョウシマシタ	プロコンクリアを実行。
クリア	0 0 0 0 0 N O P	

カウンタクリア：FUN62

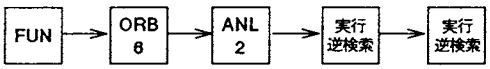
PROGRAMモード

■機能：以下の動作を行います。

通常カウンタ（C＊＊＊　＃＊＊＊＊＊）　：現在値 → 0　　接点 OFF
アップダウンカウンタ（UDC＊＊＊　＃＊＊＊＊＊）：現在値 → 0　　接点 OFF

■キー操作

《基本操作》



〔操 作 例〕

キー操作	表示	説明
	0 0 0 0 0 L D　　0 0 0 0 0	プログラムの任意の行を表示します。
FUN　ORB 6　ANL 2　実行 逆検索	カウンタ　クリア ヨロシイデスカ？	カウンタクリア機能を呼び出します。
実行 逆検索	シュウリョウシマシタ	カウンカクリアを実行。
クリア	0 0 0 0 0 L D　　0 0 0 0 0	元の表示に戻ります。

高速カウンタクリア：FUN63

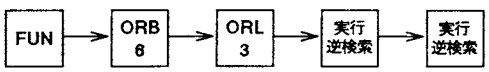
PROGRAMモード

■機能：以下の動作を行います。

高速カウンタ（CTH＊　＊＊＊＊＊）　：現在地 → 0
高速カウンタコンパレータ（CTC＊　＃＊＊＊＊＊）：接点 OFF

■キー操作

《基本操作》



〔操 作 例〕

キー操作	表示	説明
	0 0 0 0 0 L D　　0 0 0 0 0	プログラムの任意の行を表示します。
FUN　ORB 6　ORL 3　実行 逆検索	コウソク　カウンタ　クリア ヨロシイデスカ？	高速カウンタクリア機能を呼び出します。
実行 逆検索	シュウリョウシマシタ	高速カウンタクリアを実行。
クリア	0 0 0 0 0 L D　　0 0 0 0 0	元の表示に戻ります。

データメモリオールクリア：FUN64

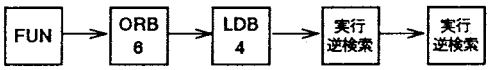
PROGRAMモード

■機能：データメモリのデータをすべてクリアします。

●クリア後、データメモリはすべて「00000」となります。

■キー操作

《基本操作》



〔操 作 例〕

キー操作	表示	説明
	00000 LD 00000	プログラムの任意の行を表示します。
FUN ORB 6 LDB 4 実行 逆検索	データメモリ オールクリア ヨロシイデスカ？	データメモリオールクリア機能を呼び出します。
実行 逆検索	シュウリョウシマシタ	データメモリのオールクリアを実行します。
クリア	00000 LD 00000	元の表示に戻ります。

保持リレーオールクリア：FUN65

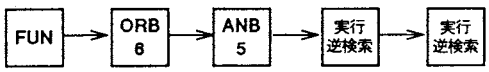
PROGRAMモード

■機能：内部補助リレーのうち保持設定のものをすべてOFFにします。

(保持設定 → P. 3−31 参照)

■キー操作

《基本操作》



〔操 作 例〕

キー操作	表示	説明
	00000 LD 00000	プログラムの任意の行を表示します。
FUN ORB 6 ANB 5 実行 逆検索	ホジリレー オールクリア ヨロシイデスカ？	保持設定のリレーのクリア機能を呼び出します。
実行 逆検索	シュウリョウシマシタ	保持設定のリレーのクリアを実行。
クリア	00000 LD 00000	元の表示に戻ります。

プログラム送受信：FUN66

PROGRAMモード
オフラインエディタ

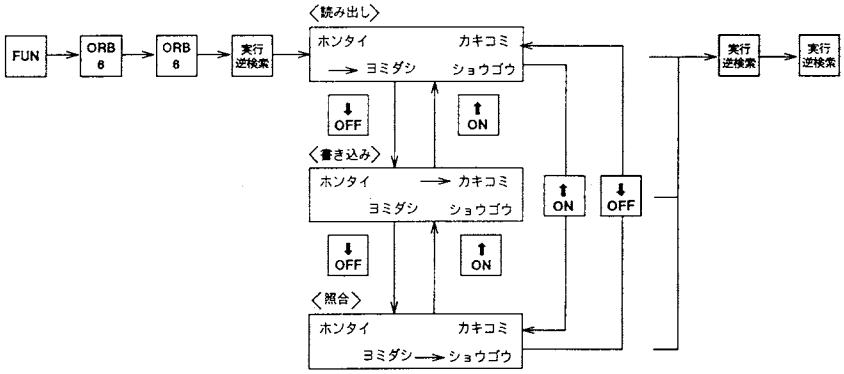
■機能：KZ本体とプログラミングコンソール(KZ-P3)の間でプログラムの「書き込み」、「読み出し」およびプログラムの「照合」ができます。

●KZ本体がリードプロテクト状態の時、「読み出し」「照合」はできません。

●書き込み：KZ-P3 ―プログラム→ KZ本体
●読み出し：KZ-P3 ←プログラム― KZ本体

■キー操作

《基本操作》



〔操 作 例〕〈照合のとき〉

キー操作	表　示	説　明
	0 0 0 0 0 L D　0 0 0 0 0	プログラムの任意の行を表示します。
FUN ORB 6 ORB 6 実行 逆検索	ホンタイ　カキコミ →ヨミダシ　ショウゴウ	プログラム送受信機能を呼び出します。
↑ ON	ホンタイ　カキコミ ヨミダシ　→ショウゴウ	「照合」をえらびます。
実行 逆検索	ホンタイト　ショウゴウシマス ヨロシイデスカ？	プログラムの照合を実行。
実行 逆検索	ショウゴウ　OK クリアデ　モドレマス	
クリア	0 0 0 0 0 L D　0 0 0 0 0	元の表示に戻ります。

●「読み出し」はプログラミングコンソールがKZ本体に接続された時および、オフラインエディタ終了時にも1回自動的に行なわれます。
●プログラミングコンソールをPROGRAMモードからRUNモードに切り換えた時にも、自動的にKZ本体へプログラムの「書き込み」、「照合」を行ないます。

オフラインエディタ開始：FUN67

RUN モード

■機能：KZ本体を運転中にプログラミングコンソールでプログラムの編集ができます。

- 「PROGRAM」モード時と同様の操作でプログラムの編集ができます。(ただし、使用できるFUNは制限されます。P8−1 参照)
- この機能はKZ本体運転中にメモリカードを対象としてプログラム編集を行なうことを前提としているため、オフラインエディタ終了時(FUN68)にはKZ本体にプログラムを書き込まず、KZ本体のプログラムを読み出します。
- オフラインエディタ起動中にKZ本体を停止させる時は、モード切替スイッチをPROGRAM側に切替えてください。編集中のプログラムをプロコン上に残した状態でPROGRAMモードに移れます。

オフラインエディタで編集したプログラムをKZ本体に書き込みたい場合は、プログラム送受信(FUN66)で書き込んでからエディタを終了して下さい。
オフラインエディタ中にKZ本体へプログラムを書き込むとき一度KZ本体を停止してから書き込を行い、書き込まれたプログラムの先頭から動作します。タイマの値がクリアされたりするので注意してください。

■キー操作

《基本操作》

FUN → ORB 6 → LD 7 → 実行 逆検索

〔操 作 例〕

キー操作	表示	説明
	ウンテンチュウ	運転中基本表示。
FUN	FUN〔67〕 オフライン エディタ カイシ	オフラインエディタの機能を呼び出します。
実行 逆検索	00000 LD 00000	0行目が表示されます。

オフラインエディタ終了：FUN68

オフラインエディタ

■機能：オフラインエディタを終了します。

- 終了時に自動的にKZ本体のプログラムと照合し、一致しない場合はKZ本体のプログラムを読み出します。また、書替えがあった場合は照合せずに読み出しを始めます。

■キー操作

《基本操作》

FUN → ORB 6 → AND 8 → 実行 逆検索

〔操 作 例〕

キー操作	表示	説明
	00000 LD 00000	プログラムの任意の行を表示します。
FUN ORB 6 AND 8 実行 逆検索	ヨミダシ シュウリョウ クリアデ モドレマス	プログラムが一致しない場合はKZ本体のプログラムを読み出します。
クリア	ウンテンチュウ	運転中基本表示に戻ります。

タイマ・カウンタ現在値変更：FUN69

PROGRAMモード
RUNモード

■機能：タイマ・カウンタの現在値を変更します。または現在値をモニタ表示します。

●現在値変更の結果

プログラミングコンソールのモード / タイマ/カウンタの種類	PROGRAMモード	RUNモード	
		●タイマ入力ONのとき ●カウンタの入力ONのとき（※1）	●タイマ入力OFFのとき ●カウンタの入力OFFのとき（※2）
タイマ(TMR) 高速タイマ(TMH) 高速1msタイマ(TMS)	現在値変更は行ないません。	変更値≧設定値のとき 設定値→現在値 変更値＜設定値のとき 変更値→現在値 ●ただし現在値＝0のとき接点出力はONします。	現在値変更は行ないません。
カウンタ(CTR)	変更値≧設定値のとき 設定値→現在値 変更値＜設定値のとき 変更値→現在値 ●ただし、現在値＝設定値のとき接点出力はONします。	←	現在値変更は行ないません。
アップダウンカウンタ(UDC)	変更値≧設定値のとき 設定値→現在値 変更値＜設定値のとき 変更値→現在値 ●接点出力はOFFします。	←	現在値変更は行ないません。
高速カウンタ(CTH)	●CTC0、2の設定値および特殊補助リレー(2103、2203)の状態で変化します。 CTH0を変更する場合 (特殊補助リレー2103 ONのとき) CTH0の変更値≧CTC0設定値のとき CTC0の設定値－1→CTH0の現在値 CTH0の変更値＜CTC0設定値のとき CTH0の変更値→CTH0の現在値 (特殊補助リレー2103 OFFのとき) CTH0の変更値→CTH0の現在値 CTH1を変更する場合 (特殊補助リレー2203 ONのとき) CTH1の変更値≧CTC2の設定値のとき CTC2の設定値－1→CTH1の現在値 CTH1の変更値＜CTC2の設定値のとき CTH1の変更値→CTH1の現在値 (特殊補助リレー2203 OFFのとき) CTH1の変更値→CTH1の現在値		

●プログラム中に存在しないタイマ・カウンタの現在値変更はできません。
●プログラムモードではタイマの現在値変更はできません。
●TMH、TMSはTMRキーで指定します。
UDCはCTRキーで指定します。

(※1)カウンタ(C)の入力ON
アップダウンカウンタ(UDC)のRES入力OFF
} の時

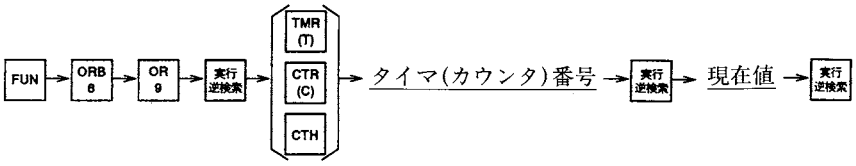
(※2)(※1)の逆の時

タイマ・カウンタ現在値変更：FUN69

PROGRAMモード
RUNモード

■キー操作

《基本操作》



〔操 作 例〕〈RUNモードのとき〉

キー操作	表示	説明
	ウンテン　チュウ	運転中基本表示。
FUN ORB 8 OR 9 実行 逆検索	T/Cゲンザイチ　＃００００１０ ＴＭＲ０００	タイマ／カウンタ現在値変更機能を呼び出します。
CTR (C) SET 1 実行 逆検索	T/Cゲンザイチ　＃００００１０ ＣＴＲ００１　＃００００００	CTR001を選択します。(現在値10)
SET 1 AND 8 RES 0	T/Cゲンザイチ　＃００００１０ ＣＴＲ００１　＃００１８０	現在値を変更します。
実行 逆検索	T /Cゲンザイチ　＃００１８０ ＣＴＲ００１	現在値変更終了。
クリア	ウンテン　チュウ	運転中基本表示に戻ります。

現在値変更機能が呼び出された後に [16進/10進] キーを押すと10進表示と16進表示を切り替えることができます。

タイマ・カウンタ設定値変更：FUN70

PROGRAMモード
RUNモード

■機能：タイマ・カウンタの設定値を変更します。または設定値をモニタ表示します。

●設定値変更の結果

タイマ/カウンタの種類 ＼ プログラミングコンソールのモード	PROGRAMモード	RUNモード	
		タイマ入力ONのとき カウンタの入力ONのとき (※1)	タイマ入力OFFのとき カウンタの入力OFFのとき(※2)
タイマ(TMR) 高速タイマ(TMH) 高速1msタイマ (TMS)	変更値→設定値 接点出力→OFF	変更値≧現在値のとき 変更値→設定値 変更値＜現在値のとき 変更値→設定値 変更値→現在値 ●ただし、現在値＝0のとき 接点出力→ON	変更値→設定値 変更値→現在値 接点出力→OFF
カウンタ(CTR)	変更値≧現在値のとき 変更値→設定値 変更値＜現在値のとき 変更値→設定値 変更値→現在値 ●ただし、現在値＝設定値のとき接点出力→ON	←	変更値→設定値 接点出力→OFF
アップダウンカウンタ (UDC)	変更値≧現在値のとき 変更値→設定値 変更値＜現在値のとき 変更値→設定値 変更値→現在値 ●接点出力→OFF	←	変更値→設定値 接点出力→OFF
高速カウンタ コンパレータ (CTC)	変更値→設定値 ●CTC0、2の設定値を変更するとCTH0、1の現在値も変化します。 CTC0を変更する場合 (特殊補助リレー2103 ONのとき) CTC0変更値≦CTH0現在値のとき CTC0変更値－1→CTH0現在値 CTC0変更値＞CTH0現在値のとき CTH0は変化なし (特殊補助リレー2103 OFFのとき) CTH0は変化なし CTC2を変更する場合 (特殊補助リレー2203 ONのとき) CTC2変更値≦CTH1現在値のとき CTC2変更値－1→CTH1現在値 CTC2変更値＞CTH1現在値のとき CTH1は変化なし (特殊補助リレー2203 OFFのとき) CTH1は変化なし ●CTC0、1、2、3の接点は変化しません。		

●本体がライトプロテクト状態のとき、設定値変更はできません。
●プログラム中に存在しないタイマ・カウンタの設定値変更はできません。
●TMH、TMSはTMRキーで指定します。
UDCはCTRキーで指定します。
●ソースプログラム中の設定値も変更されます。

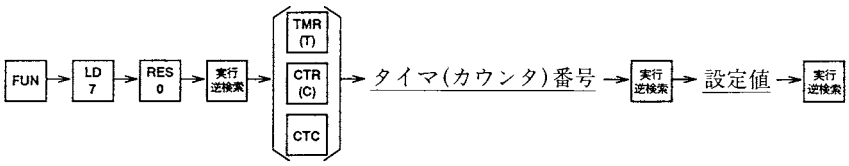
(※1)カウンタ(C)の入力ON
アップダウンカウンタ(UDC)のRES入力OFF
}の時
(※2)(※1)の逆の時

タイマ・カウンタ設定値変更：FUN70

PROGRAMモード
RUNモード

■キー操作

《基本操作》



〔操 作 例〕〈PROGRAMモードのとき〉

キー操作	表示	説明
	00000 LD　　00000	プログラムの任意の行を表示します。
FUN　LD 7　RES 0　実行 逆検索	T/Cセッテイチ　#00020 TMR000	設定値変更機能を呼び出します。 TMR000が表示されます。
CTR (C)　SET 1　実行 逆検索	T/Cセッテイチ　#00100 CTR001　#00000	CTR001の現在の設定値が表示されます。
ANL 2　AND 8　RES 0	T/Cセッテイチ　#00100 CTR001　#00280	変更する設定値を入力します。 (280)
実行 逆検索	T/Cセッテイチ　#00280 CTR001	設定値変更終了。
クリア	00000 LD　　00000	元の表示に戻ります。

設定値変更機能が呼び出された後に @ #$ キーを押すと10進表示と16進表示を切り替えることができます。

リレーのON/OFF：FUN71

PROGRAMモード
RUNモード

■機能：各リレーを強制的にON/OFFします。または、リレーのON/OFF状態をモニタ表示します。

●強制ON/OFF可能なリレー(KZ-40の場合)

リレーの種類		PROGRAMモード	RUNモード
入力リレー	0000～0107	×	×
出力リレー	0500～0607	○(※1)	○(※2)
内部補助リレー	1000～6915	○	○(※2)
特殊補助リレー	2000～2915	○(※3)	○(※3)

リレーの種類		PROGRAMモード	RUNモード
タイマ	T000～T119	○	○(※4)
カウンタ	CTR000～CTR119	○(※5)	○(※6)
高速カウンタコンパレータ	CTC0～CTC3	○(※7)	○(※7)

(※)リレー番号は機種により異なりますので、リレー番号一覧でご確認の上、ご使用下さい。

実行可能なリレー
→●左表○印のリレー

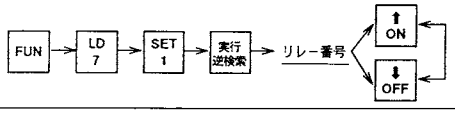
実行不可能なリレー
→●左表×印のリレー
→●プログラムに書かれていないタイマ・カウンタ。

〈補　足〉

●(※1) 出力端子も同時にON/OFFします。
●(※2) プログラム上、使用されていないリレーも強制ON/OFFが可能です。
プログラム上、使用されているリレーの場合、スキャンの始まりからプログラムによって書き換えられるまでの間の「ON/OFF」が可能です。
保持設定のものも「ON/OFF」可能です。
●(※3) 2000～2008、2010～2015はできません。
●(※4) 入力ONのとき
強制ON時：出力ON、現在値→0000
強制OFF時：出力OFF、現在値→設定値
} となります。
入力OFFのとき
スキャンの始まりから、このタイマが実行されるまでの間セットが可能です。
●(※5) 強制 ON 時：出力 ON、現在値 → 設定値
強制 OFF 時：出力 OFF、現在値 → 0000
} となります。
(UDC の場合、現在値 → 0000)
●(※6) リセット入力が OFF のとき
強制 ON 時：出力 ON、現在値 → 設定値
強制 OFF 時：出力 OFF、現在値 → 0000
} となります。
(UDC の場合、現在値 → 0000)
リセット入力が ON のとき
スキャンの始まりからこのカウンタが実行されるまでの間セットが可能です。
●(※7) 強制 OFF のみ可能です。(ON はできません。)

■キー操作

《基本操作》



〔操 作 例〕〈RUNモードのとき〉

キー操作	表示	説明
	ウンテン　チュウ	運転中基本表示。
FUN LD 7 SET 1 実行 逆検索	リレー　ノ　オン/オフ ０００００　♀	リレーON/OFF機能を呼び出します。 入力リレー0000の状態を表示します。
ANB 5 RES 0 RES 0	リレー　ノ　オン/オフ ００５００　♀	出力リレー500を選択します。 (OFF)
↑ ON	リレー　ノ　オン/オフ ００５００　♂	強制的にONさせます。
CTR (C) SET 1	リレー　ノ　オン/オフ ＣＴＲ００１　♂	CTR001を選択します。(ON)
↓ OFF	リレー　ノ　オン/オフ ＣＴＲ００１　♀	CTR001を強制的にOFFします。
クリア	ウンテン　チュウ	運転中基本表示に戻ります。

データメモリ書込み：FUN72

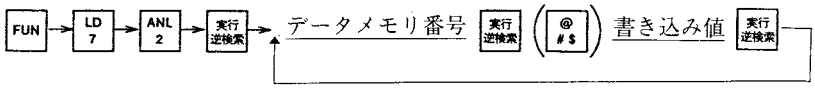
PROGRAMモード
RUNモード

■機能：データメモリの現在の内容をモニタ表示、または変更できます。

- [@ #$]キーを押すことで、「16進」↔「10進」の入力切替が可能です。
- テンポラリデータメモリも書き込み可能です。（ただし、TM30、TM31は書き込みできません。）

■キー操作

《基本操作》



〔操 作 例〕〈RUNモードのとき〉

キー操作	表示	説明
	ウンテン　チュウ	運転中基本表示。
FUN LD 7 ANL 2 実行 逆検索	メモリノ　アタイ　＃００００００ ＤＭ０００００	データメモリ書込み機能を呼び出します。DM000の現在の内容を上段に表示します。
ORL 3 実行 逆検索	メモリノ　アタイ　＃００００１０ ＤＭ０００３　＃００００００	DM003を選択表示します。
@ #$	メモリノ　アタイ　＄０００Ａ ＤＭ０００３　＄００００	10進→16進の表示切替を行います。
W-ON F W-ON F	メモリノ　アタイ　＄０００Ａ ＤＭ０００３　＄００ＦＦ	データの変更を行います。
実行 逆検索	メモリノ　アタイ　＄００ＦＦ ＤＭ０００３	書き込みを実行。
@ #$	メモリノ　アタイ　＃００２５５ ＤＭ０００３	16進→10進の表示切替を行います。
クリア	ウンテン　チュウ	運転中基本表示に戻ります。

- [ENDH TM]キーを押すことによりテンポラリーデータメモリの、[END DM]キーを押すことによりデータメモリの表示、変更ができます。

トリマ設定値読出：FUN73

PROGRAMモード
RUN　　モード
オフラインエディタ

■機能：アナログタイマのトリマ値をリアルタイムに表示します。

- 表示データの範囲は0～249です。
- KZ-10、16は1点、KZ-24、40、80は2点あります。

■キー操作

《基本操作》

FUN → LD 7 → ORL 3 → 実行 逆検索

〔操 作 例〕〈RUNモードのとき〉

キー操作	表示	説明
	ウンテン　チュウ	運転中基本表示。
FUN　LD 7　ORL 3　実行 逆検索	0：0249　　1：0000 2：0000　　3：0000	トリマ設定値読出機能を呼び出します。トリマの値を表示します。
クリア	ウンテン　チュウ	運転中基本表示に戻ります。

文法チェック：FUN74

PROGRAMモード
オフラインエディタ

■機能：プログラムの全行文法チェックを行います。

- 文法に誤まりがあればエラー表示します。
  この時、クリアキーを押すとエラーのある行を表示しますのでプログラムの修正を行なって下さい。
- 複数行のエラーがある場合、最初に発見したエラーの内容を表示します。
- PROGRAMモードから、RUNモードに切替えたときも同様のチェックを行います。

■キー操作

《基本操作》

FUN → LD 7 → LDB 4 → 実行 逆検索

〔操 作 例〕

キー操作	表示	説明
	00000 LD　　00000	プログラムの任意の行を表示します。
FUN　LD 7　LDB 4　実行 逆検索	ブンポウ　チェック ヨロシイデスカ？	文法チェック機能を呼び出します。
実行 逆検索	ブンポウ　チェック　OK クリアデ　モドレマス	文法チェックを実行します。 (エラーなしの場合)
クリア	00000 LD　　00000	元の表示に戻ります。
≈	≈	≈
実行 逆検索	ENDH　ナシ　エラー クリアデ　モドレマス	エラーがある場合、クリアキーを押すとエラー行を表示します。

プログラム容量確認：FUN75

PROGRAMモード
オフラインエディタ

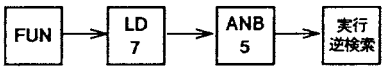
■機能：現在プログラミングコンソール(KZ-P3)で編集中のプログラムの容量をブロック単位で表示します。

●プログラムの大きさは KZ-10、16 の場合、最大でも 7 ブロック、KZ-24、40、80 の場合、最大でも 37 ブロックです。
●プログラムの先頭から最後の命令 (NOP はのぞく) までとタイマ・カウンタ設定値テーブルをプログラム容量として表示します。
●1 ブロックは 256 バイトです。3 バイト命令で考えると、1 ブロックは約 85 行となります。

タイマ・カウンタ設定値テーブルは、必ず 1 ブロック弱の容量を使用します。

■キー操作

《基本操作》



〔操 作 例〕

キー操作	表示	説明
	0 0 0 0 0 L D 0 0 0 0 0	プログラムの任意の行を表示します。
FUN LD 7 ANB 5 実行 逆検索	プログラム ヨウリョウ 0 2 ブロック	プログラムの大きさを表示します。
クリア	0 0 0 0 0 L D 0 0 0 0 0	元の表示に戻ります。

ワンポイント

●KZ-10、16 のプログラム領域は 1500 バイトです。1 ブロックは 256 バイトですから、ブロックになおすと、
1500÷256≒6 ブロック
さらにタイマ、カウンタの設定値に約 1 ブロック使用しますので、プログラム容量は最大 7 ブロックになります。
●KZ-24、40、80 の場合も同様に最大 37 ブロックになります。
9000÷256+1≒37 ブロック

●メモリカードのプログラム格納容量
M-2=125ブロック (24ファイル以内)
M-3=506ブロック (48ファイル以内)

プログラム領域 → P2−3

9章　メモリカード

メモリカード

プログラミングコンソール「KZ-P3」またはパソコン(KZ-LADDER)＋カードリーダライタ「Z-1」を使用して、KZ本体のプログラムを保存することが可能です。

■メモリカードの機能(KZ-P3使用時)

		機　能
メモリカードオールクリア		●メモリカードの初期化(フォーマット)を行ないます。(新しいメモリカードを使用するときは、必ずこの操作を実行してください) ●メモリカードに書込まれているすべてのプログラムを消去します。
メモリカード新規追加		●メモリカードに新たにプログラムを書込みます。 ●それぞれのプログラムは、名前(0～9、A～Fより3文字)と日付をつけて管理できます。
メモリカードアクセス	書込	●メモリカード内の任意のプログラムに、プログラミングコンソール上にあるプログラムを上書きします。
	読出	●メモリカードから任意のプログラムをプログラミングコンソールに読み出します。
	照合	●メモリカード内の任意のプログラムとプログラミングコンソール上のプログラムを照合します。
	削除	●メモリカードから任意のプログラムを削除します。

●メモリカードに保存されるプログラムはENDH以降に書かれているものもすべて含まれます。(ただしNOP命令は含みません。)
●1枚のメモリカードにKZ-10、16、24、40、80のプログラムを一緒に保存できます。
●プログラミングコンソールでメモリカードを使用できるのは、PROGRAMモード時または、オフラインエディタ起動時のみです。
●メモリカード内には、同じ名前のプログラムは存在できません。

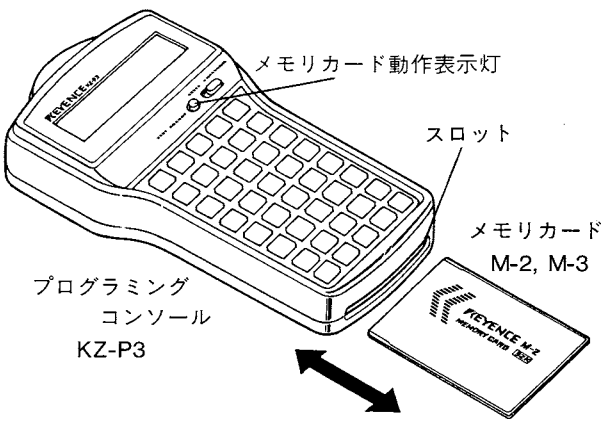
■メモリカードの容量

メモリカードに保存できるプログラム数は、下表のとおりです。

カード	容　量	保存可能プログラム数
M-2	32kbyte	最大24プログラム。ただし、プログラムの総ブロック数が125ブロック以内。
M-3	128kbyte	最大48プログラム。ただし、プログラムの総ブロック数が506ブロック以内。

※1ブロックは256バイト。

■メモリカードの抜き差し



①プログラミングコンソール(KZ-P3)をプログラムモードまたは、オフラインエディタモードにします。
②メモリカードの社名・形式名などのデザイン面を上側にして、プログラミングコンソールのスロットに挿入してください。
●逆差しをしないように注意して下さい。
●メモリカードは、プログラミングコンソールの電源がON状態でも抜き差しが可能です。
●メモリカード動作表示灯が点灯している時は、抜き差ししないで下さい。
このような時は [クリア] キーを押すと動作表示灯が消えますので、それから抜き差しして下さい。

メモリカードオールクリア

PROGRAMモード
オフラインエディタ

■機能：メモリカードの初期化(フォーマット)を行ないます。

●新しいメモリカードを使用するときには、必ずオールクリアを実行して下さい。

●オールクリアを実行すると、メモリカードに書き込まれているすべてのプログラムを消去します。

■操作例1(新しいメモリカードM-2<32kbyte>をフォーマットする場合)

キー操作	表示	説明
	00000 LD　　00000	プログラムの任意の行を表示します。
カードモニタ	アタラシイ　カードデス フォーマットシテ　イイデスカ？	確認画面が表示されます。
実行 逆検索	32KByteノカードトシテ アラタニ　フォーマットシマス	カードの容量を選択します。
実行 逆検索	ノコリ　　24プログラム 125ブロック	カードの容量が表示されます。
クリア　(＊)	00000 LD　　00000	元の表示に戻ります。

(＊)ここで 実行 逆検索 を押すと続けて操作できます。

■操作例2(すでにプログラムの書き込まれているメモリカードM-3<128kbyte>をフォーマットする場合)

キー操作	表示	説明
	00000 LD　　00000	プログラムの任意の行を表示します。
カードモニタ	ノコリ　　44プログラム 486ブロック	メモリカードの残り容量が表示されます。
実行 逆検索	メモリカード　　シンキツイカ →アクセス　　オールクリア	使用可能な機能が表示されます。
↑ ON (↓ OFF ↓ OFF)	メモリカード　　シンキツイカ アクセス　　→オールクリア	オールクリアを選択します。
実行 逆検索	32KByteノカードトシテ アラタニ　フォーマットシマス	
↓ OFF ↓ OFF	128KByteノカードトシテ アラタニ　フォーマットシマス	カードの容量を選択します。
実行 逆検索	ノコリ　　48プログラム 506ブロック	フォーマットを実行します。
クリア　(＊)	00000 LD　　00000	元の表示に戻ります。

(＊)ここで 実行 逆検索 を押すと続けて操作できます。

●メモリカードの容量とフォーマットする容量は必ず一致させて下さい。
- M-2：32kbyteでフォーマットします。
- M-3：128kbyteでフォーマットします。

※1ブロックは256バイトです。

メモリカードへの新規追加

PROGRAMモード
オフラインエディタ

■機能：メモリカードに新たにプログラムを書き込みます。

- 現在、プログラミングコンソール(KZ-P3)で作成中のプログラムをメモリカードに保存します。
- 名前は英数字(0～9、A～F)で3文字まで、日付は数字で6桁まで入力できます。

■操作例（名前：ABC、日付：92-05-19で新規に書き込む場合）

キー操作	表示	説明
	00000 LD　　00000	プログラムの任意の行を表示します。
カードモニタ	ノコリ　　48プログラム 506ブロック	メモリカードの残り容量が表示されます。
実行 逆検索 ↓OFF	メモリカード　　→シンキツイカ アクセス　　オールクリア	シンキツイカを選択します。
実行 逆検索	ナマエ：000 ヒヅケ：00－00－00	名前の入力表示になります。
OUT A　AUB B　MPS C	ナマエ：ABC ヒヅケ：00－00－00	名前を入力します。
↓OFF	ナマエ：ABC ヒヅケ：00－00－00	日付の入力表示にカーソルが移動します。
OR 9　ANL 2　RES 0　ANB 5　SET 1　OR 8	ナマエ：ABC ヒヅケ：92－05－19	日付を入力します。
実行 逆検索	ABCニ　カキコミマス ヨロシイデスカ？	書き込み実行。
実行 逆検索	ノコリ　　47プログラム 504ブロック	書き込み終了。 残り容量が表示されます。
クリア	00000 LD　　00000	元の表示に戻ります。

- 名前と日付の入力は↑ON ↓OFFキーで切り換えられます。
- カードに記録できる容量が残っていない場合は、新規追加できません。
- メモリカード内にすでにある名前と同じ名前では書き込めません。

ワンポイント

日付は、数字(0～9)を任意に入力できますので、バージョンNO.等として使用することもできます。

メモリカードアクセス

PROGRAMモード
オフラインエディタ

■機能：メモリカード内のプログラムに対して、書き込み、読み出し、削除、照合を行ないます。

●メモリカード内にプログラムがない場合(オールクリア直後など)はアクセスできません。

1. 書き込み

すでに、メモリカード内にある任意のプログラムに、現在プログラミングコンソール上のプログラムを上書きします。

2. 読み出し

すでに、メモリカード内にある任意のプログラムを、プログラミングコンソールに読み出します。
プログラミングコンソール上のプログラムは失われます。

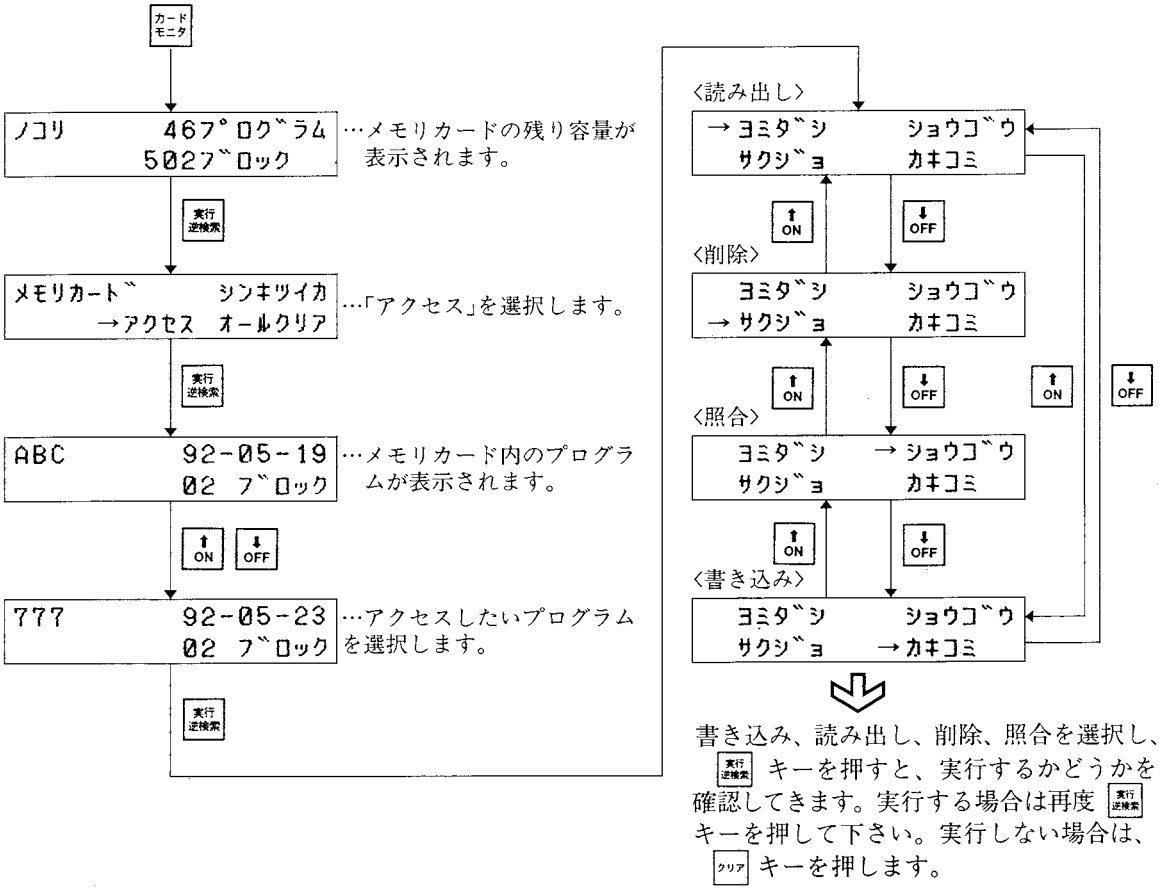
3. 削　除

すでに、メモリカード内にある任意のプログラムを削除します。

4. 照　合

すでに、メモリカード内にある任意のプログラムと現在、プログラミングコンソール上にあるプログラムが同一のものかどうか調べます。

■基本操作



書き込み、読み出し、削除、照合を選択し、実行 逆検索 キーを押すと、実行するかどうかを確認してきます。実行する場合は再度 実行 逆検索 キーを押して下さい。実行しない場合は、クリア キーを押します。

メモリカードアクセス：「書き込み」

PROGRAMモード
オフラインエディタ

■**操作例**(メモリカード内の「777」というプログラムに、現在プログラミングコンソール上にあるプログラムを上書きする場合)

キー操作	表示	説明
	00000 LD　　00000	プログラムの任意の行を表示します。
カード モニタ	ノコリ　　46プログラム 502ブロック	メモリカードの残り容量が表示されます。
実行 逆検索	メモリカード　　シンキツイカ →アクセス　　オールクリア	メモリカードアクセスを選択します。
実行 逆検索	ABC　　92－05－19 02ブロック	メモリカード内の任意のプログラムが表示されます。
↓ OFF	777　　92－05－23 02ブロック	アクセスするプログラムを選択します。
実行 逆検索	→ヨミダシ　　ショウゴウ サクジョ　　カキコミ	
↓ OFF　↓ OFF　↓ OFF　(↑ ON)	ヨミダシ　　ショウゴウ サクジョ　　→カキコミ	カキコミを選択します。
実行 逆検索	ナマエ：777 ヒヅケ：00－00－00	
OR 9　ANL 2　RES 0　ANB 5　ORL 3　RES 0	ナマエ：777 ヒヅケ：92－05－30	日付を入力します。
実行 逆検索	777ニカキコミマス ヨロシイデスカ？	書き込み実行を確認してきます。
実行 逆検索	ノコリ　　46プログラム 501ブロック	書き込み終了。 残り容量が表示されます。
実行 逆検索 (クリア)	00000 LD　　00000	元の表示に戻ります。

- 書き込みする場合、名前の変更はできません。違う名前で書き込みたい場合は、新規追加で行なってください。
- プログラミングコンソール上のプログラムの容量が、カードの残り容量＋アクセス対象のプログラムの容量をこえる場合、書き込みできません。
- 日付をまちがえて入力した場合は、 後退 キーで取り消すことができます。

ワンポイント

日付は、数字(0～9)を任意に入力できますので、バージョンNO.等として使用することもできます。

メモリカードアクセス：「読み出し」

PROGRAMモード
オフラインエディタ

■**操作例**（メモリカード内の「ABC」という名前のプログラムをプログラミングコンソール上に読み出す場合。）

キー操作	表示	説明
	00000 LD　　00000	プログラムの任意の行を表示します。
カードモニタ	ノコリ　　46プログラム 501ブロック	メモリカードの残り容量が表示されます。
実行 逆検索	メモリカード　　シンキツイカ →アクセス　　オールクリア	メモリカードアクセスを選択します。
実行 逆検索	ABC　　92－05－19 02ブロック	アクセスするプログラムを選択します。
実行 逆検索	→ヨミダシ　　ショウゴウ サクジョ　　カキコミ	ヨミダシを選択します。
実行 逆検索	ABCヲ　ヨミダシマス ヨロシイデスカ？	読み出しの実行を確認してきます。
実行 逆検索	ヨミダシ　シュウリョウ	読み出し終了。
実行 逆検索 (クリア)	00000 LD　　00000	元の表示に戻ります。

メモリカードアクセス：「照合」

PROGRAMモード
オフラインエディタ

■**操作例**（メモリカード内の「ABC」という名前のプログラムとプログラミングコンソール上のプログラムを照合する場合。）

キー操作	表示	説明
	00000 LD　　00000	プログラムの任意の行を表示します。
カードモニタ	ノコリ　　46プログラム 501ブロック	メモリカードの残り容量が表示されます。
実行 逆検索	メモリカード　　シンキツイカ →アクセス　　オールクリア	メモリカードアクセスを選択します。
実行 逆検索	ABC　　92－05－19 02ブロック	アクセスするプログラムを選択します。
実行 逆検索	→ヨミダシ　　ショウゴウ サクジョ　　カキコミ	
↓ OFF ↓ OFF	ヨミダシ　　→ショウゴウ サクジョ　　カキコミ	ショウゴウを選択します。
実行 逆検索	ABCト　ショウゴウシマス ヨロシイデスカ？	照合の実行を確認してきます。
実行 逆検索	ショウゴウ　OK	照合終了。
実行 逆検索 (クリア)	00000 LD　　00000	元の表示に戻ります。

●プログラムが同一でない場合は「ショウゴウエラー」となります。
実行 逆検索 キー（ クリア キー ）を押すともとに戻れます。

メモリカードアクセス：「削除」

PROGRAMモード
オフラインエディタ

■**操作例**(メモリカード内の「777」というプログラムを削除する場合。)

キー操作	表示	説明
	００００００ ＬＤ　　００００００	プログラムの任意の行を表示します。
カードモニタ	ノコリ　　４６プログラム ５０１ブロック	メモリカードの残り容量が表示されます。
実行 逆検索	メモリカード　　シンキツイカ →アクセス　　オールクリア	メモリカードアクセスを選択します。
実行 逆検索	ＡＢＣ　　９２－０５－１９ ０２ブロック	メモリカード内の任意のプログラムが表示されます。
↓ OFF	７７７　　９２－０５－３０ ０３ブロック	アクセスするプログラムを選択します。
実行 逆検索	→ヨミダシ　　ショウゴウ サクジョ　　カキコミ	
↓ OFF	ヨミダシ　　ショウゴウ →サクジョ　　カキコミ	サクジョを選択します。
実行 逆検索	７７７ヲ　サクジョシマス ヨロシイデスカ？	削除の実行を確認してきます。
実行 逆検索	ノコリ　　４７プログラム ５０４ブロック	削除終了。 残り容量が表示されます。
実行 逆検索 (クリア)	００００００ ＬＤ　　００００００	元の表示に戻ります。

KZ間のプログラム転送

KZは、メモリカードを介して、KZ-10、16、24、40、80の間でプログラムを転送することができます。ただし、機種間でプログラムの容量、入出力点数、タイマ・カウンタの数が異なりますので若干の変更作業が必要となります。

1.同じプログラム容量の機種間でプログラムを転送する場合

●何も問題はありません。　(例)KZ-40⇄KZ-40、KZ-24⇄KZ-40

2.プログラム容量の小さいKZ本体から大きいKZ本体へプログラムを転送する場合

●何も問題はありません。　(例)KZ-10→KZ-40、KZ-16→KZ-24

3.プログラム容量の大きいKZ本体から小さいKZ本体へプログラムを転送する場合

●プログラムの大きさがプログラム領域を越えている場合は転送できません。
(例)KZ-40の3000行のプログラムをKZ-10へ転送することはできません。
(「読み出し」「照合」を選択できません。)

●プログラムの大きさがプログラム領域を越えていない場合は転送できます。
ただし、プログラム中の入出力リレー番号、内部補助リレー番号、タイマ・カウンタ番号が、使用できる範囲を越えている場合は、番号の修正が必要です。(範囲内であれば問題ありません。)

※転送を実行すると「T/Cヲイチブヘンコウシマシタ」と表示されますのでクリアキーを押して、プログラムを変更して下さい。

文法チェック (FUN74) → P8−13
プログラムの領域 → P2−3

●プログラムを転送すると、ブロック数が変わることがあります。
●文法チェック(FUN74)を行なえばプログラムの修正が必要な箇所を調べることができます。

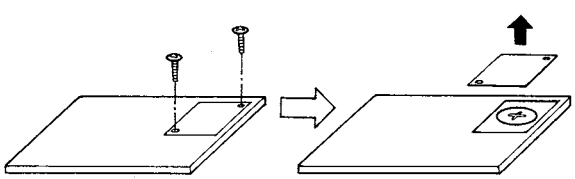
メモリカード取扱上のご注意

■メモリカードの仕様

型式	M-2	M-3
メモリ容量	32 kbyte	128 kbyte
電池寿命	5年(0～40℃)	
使用周囲温度	動作時：0～40℃　保存時：−20～60℃(氷結しないこと)	
使用周囲湿度	10～90% RH(結露しないこと)	

●電池の寿命は5年です。(0～40℃のとき)40℃以上の温度で使用する場合は、寿命が短くなりますのでご注意ください。
●高温、直射日光に長時間さらさないでください。
●水にぬらしたり、多湿場所への保管はしないでください。

●電池の交換方法



●曲げたり、落下させたり、強いショックを与えたりしないでください。
●静電気防止と防じんのため、付属のカードケースに入れて保存してください。
●コネクタ部にホコリやゴミが詰まらないよう、ご配慮ください。(コネクタ部にピンなどの異物を入れないでください。)

①カード表面にある2本のビスを精密ドライバで外してカバーを取り外します。
②古い電池を取り外し、新しい電池を入れて下さい。このとき、⊕マークのある側をカバー側としてください。

電池型式(市販品)
CR2016

③元通りカバーを取り付けてください。
※電池を外すと書き込まれていた内容がすべて失われます。あらかじめバックアップを取ってから電池交換してください。

10章　異常とその処置

異常とその処置

〈プログラミング上のエラーメッセージ一覧〉

●「オワリノギョウヲキリステマス」「ヨウリョウオーバー」は、命令の書込み時に発生します。
　[クリア]キーを押すと続けて編集できます。
●その他のメッセージは、PROGRAMモードからRUNモードにした時、または、文法チェック(FUN74)を行なった時に発生します。
　PROGRAMモードに戻す(文法チェックの時は [クリア] キーを押す)とエラーのある行を表示しますので、プログラムを修正して下さい。

プログラミングコンソールのエラーメッセージ	原　　因	処　　置
カイロエラー	プログラムに文法上の間違いがあります。	プログラムを確認して下さい。 命令語詳細(3章、5章)には各命令のカイロエラーについて解説しています。
オペランドエラー	不正なオペランドを使用しています。	オペランドを確認して下さい。 命令語詳細(3章、5章)には各命令のオペランドエラーについて解説しています。
ENDナシエラー	通常プログラムの最後にEND命令がありません。	END命令を書き込んで下さい。
ENDHナシエラー	プログラムの最後にENDH命令がありません。	ENDH命令を書き込んで下さい。
MC-MCRエラー	MC、MCRが正しく使用されていません。	MC、MCRを参照してプログラムを見直して下さい。
STP-STEエラー	STP、STEが正しく使用されていません。	STP、STEに関する部分のプログラムを見直して下さい。
FOR-NEXTエラー	FOR、NEXTが正しく使用されていません。	FOR、NEXTに関する部分のプログラムを見直して下さい。
SBNテイギナシエラー	CALL命令で呼ばれているサブルーチンがありません。	CALL命令に対応するサブルーチンがあるかプログラムを見直して下さい。
SBNサンショウエラー	任意のサブルーチンの中で、そのサブルーチンより後で定義されているサブルーチンを呼ぼうとしています。	呼びたいサブルーチンを前で定義するようにプログラムを見直して下さい。
SBNネストエラー	サブルーチンのネスティング(入れ子)構造が5重以上となっています。	サブルーチンのネスティング(入れ子)構造を4重以下にして下さい。
スタックエラー	スタックがオーバーフローしました。	MPS、MPPに関する部分のプログラムを見直して下さい。
オワリノギョウヲキリステマス	新しい行を書き込んだために、最終行がプログラム領域からあふれています。	プログラムの最終行を確認して下さい。 プログラムを短くする必要があります。
ヨウリョウオーバー	新しい行を書き込むプログラム領域が残っていません。	プログラムを短くして下さい。

MC、MCR　→P3−30
STP、STE　→P5−13
FOR、NEXT　→P5−22
MPS、MPP　→P5−7
CALL　→P5−20
SBN　→P5−20
プログラム領域　→P2−3

異常とその処置

<KZ本体に関するエラーメッセージ一覧>

エラーメッセージは「PROGRAMモード」で [クリア] キーを押すと消去できます。
メッセージを消去した後に下表に従って異常の原因を排除して下さい。

プログラミングコンソールのエラーメッセージ	原　　因	処　　置
ホンタイシステムエラー	KZ本体ハードウェア上のエラーです。	KZ本体の電源を入れ直して下さい。それでも直らなければ、故障が考えられます。最寄りの営業所までお問い合せください。
ホンタイクリアエラー	プログラムの登録されていないKZを動かそうとしています。	KZ本体にプログラムを書き込んでください。
ホンタイサイクルタイムエラー	サイクルタイムが208ms 以上になっています。	プログラムを再度見直しして下さい。 ※FOR～NEXT命令で回数の多い繰り返しを行なっていないか確認して下さい。 ※割り込みのプログラムを速いレートで実行させていないか確認して下さい。
ホンタイメモリエラー	RAMの内容が消失したために、KZ本体のデータメモリの値、カウンタの現在値がクリアされてしまいました。	データメモリの値、カウンタの現在値を確認して下さい。
ホンタイCALLネストエラー	サブルーチンのネスティング(入れ子)が4重をこえています。	これらのエラーはプログラムの**実行時**に発生するエラーです。 サブルーチンの使い方、INT命令の使い方に特に注意してプログラムを見直して下さい。
ホンタイFORネストエラー	FOR～NEXT命令のネスティング(入れ子)が8重をこえています。	
ホンタイINTネストエラー	同時に5重以上の割込みが発生しました。	
ホンタイMPSネストエラー	MPS～MPP命令のネスティング(入れ子)で8重をこえています。	

●運転中、KZ 本体にエラーが発生した場合、運転は停止します。
(出力リレーはすべて OFF します。)

異常とその処置

〈メモリカードに関するエラーメッセージ一覧〉

エラーメッセージは「PROGRAMモード」で [クリア] キーを押すと消去できます。
メッセージを消去した後に下表に従って処置して下さい。

プログラミングコンソールのエラーメッセージ	原　　因	処　　置
アタラシイカードデス フォーマットシテ イイデスカ？	挿入されたカードが初期化されていません。	初期化して下さい。 P9−2 参照
オナジナマエガ アリマス ベツノナマエニシテ クダサイ	新規追加で指定したプログラムの名前がカード内にあります。	名前は重複できません。別の名前を入力して下さい。
T/Cヲイチフ ヘンコウシマシタ	容量の大きなKZ本体で作成したプログラムを容量の小さなKZ本体で読み出したためです。	タイマ・カウンタの番号が使用できる範囲を越えていないか確認して下さい。 越えている場合は、番号を変更して下さい。 この場合、設定値が0となりますので再設定して下さい。
カードヨミダシエラー	カードからデータを読み出せませんでした。	カードが正確に挿入されているか確認して下さい。 カードの電池を確認して下さい。
カードフリョウ	カードが正常な動作をしていません。	

※9章メモリカードもあわせてお読みください。

〈その他のメッセージ〉

エラーメッセージは「PROGRAMモード」で [クリア] キーを押すと消去できます。

プログラミングコンソールのエラーメッセージ	原　　因	処　　置
ドウツウモニタ デキマセン	プログラムがリードプロテクトされているので導通モニタできません。	―――
ツウシンフリョウ	通信ができませんでした。	プログラミングコンソールと本体をつなぐ接続ケーブルを確認して下さい。
カキコミ ホゴサレテイマス キョウコウシマスカ？	書き込み保護されているKZ本体にプログラムを書き込もうとしています。	クリアキーを押すと取り消されます。強行する場合は実行キーを押して下さい。

逆引き目次

	検索内容	ユーザーズマニュアルでの説明ページ
あ	アナログトリマの値をモニタしたい	8－13　トリマ設定値の読みだし
	EEPROM の寿命はどれくらい？	1－4　性能仕様
	運転中のプログラムの編集や変更について	8－6　オンラインエディタ開始
	エラーが発生したら	10－1　異常とその処理
か	カウンタの打ち込み方法	3－22　カウンタ
	カウンタの現在値を見たい (モニタしたい)	2－25　状態モニタ
	カウンタの現在値をデータメモリに取り込む方法	5－29　LDA 命令：STA 命令
	カウンタの桁数の考え方	3－22　カウンタ
	カウンタ設定値の変更 (アナログトリマ使用)	5－28　トリマイン
	カウンタ設定値の変更 (パソコンより)	7－1　7章 シリアル通信機能
	カウンタ設定値の変更 (外部機器より)	5－24　HKEY 命令
	カウンタ設定値の変更 (BCD 入力)	5－24　HKEY 命令
	拡張ユニットの接続について	1－7　拡張ユニットの接続
	拡張ユニットの入・出力リレー番号について	1－7　拡張ユニットの入出力リレー番号の割り付け
	拡張ラダーとは	4－1　1 拡張ラダー言語
	外形寸法図	1－14　外形寸法図
	行数を指定してモニタする方法	2－27　導通モニタ
	KZ 間でのプログラム転送について	9－8　KZ 間のプログラム転送
	KZ-**R に KZ-□ET は接続できるか	1－7　拡張ユニットの接続
	KZ-**T に KZ-□ER は接続できるか	1－7　拡張ユニットの接続
	高速カウンタでのコンパレータの設定値の範囲	5－18　高速カウンタコンパレータ
	高速カウンタの使い方	5－17　応用命令
	高速カウンタの現在値を見たい	2－25　状態モニタ
	高速入力したい	3－29　ハイスピード (HSP)
さ	サイクルタイムの表示を見たい	2－24　サイクルタイムモニタ
	出力定格を知りたい	1－6　出力仕様
	出力保護回路はどうなっているのか	1－6　入出力回路構成
	シフト (SFT) レジスタのシフト段数について	3－27　SFT 命令
	消費電流 (電力) はどれくらい？	1－5　一般仕様
	使用できる温度範囲は	1－5　一般仕様
	ステッピングモータの制御について	6－1　6章 高速カウンタの応用
	設定値の変更 (アナログトリマ使用)	5－28　トリマイン
	設定値の変更 (外部機器より)	5－24　HKEY 命令
	設定値の変更 (パソコンより)	7－1　7章 シリアル通信機能
	設定値の変更 (BCD 入力)	5－24　HKEY 命令
た	耐振動について	1－5　一般仕様
	耐ノイズ性について	1－5　一般仕様
	タイマ精度は	3－19　タイマ
	タイマの最小ステップについて	3－19　タイマ
	タイマの誤差はどれくらい？	3－19　タイマ
	タイマの現在値を見たい (モニタしたい)	2－25　状態モニタ
	タイマの打ち込み方法	3－19　タイマ
	タイマ設定値の変更 (アナログトリマ使用)	5－28　トリマイン
	タイマ設定値の変更 (パソコンより)	7－1　7章 シリアル通信機能
	タイマ設定値の変更 (外部機器より)	5－24　HKEY 命令
	タイマ設定値の変更 (BCD 入力)	5－24　HKEY 命令
	ダイレクトクロックパルスの出力可能な周波数は	6－1　6章 高速カウンタの応用
	ダイレクトクロックパルスの周波数変更方法	6－1　6章 高速カウンタの応用

逆引き目次

	検索内容	ユーザーズマニュアルでの説明ページ
た	使えるリレー番号	2－7　リレー番号一覧
	作ったプログラムにプロテクトをかけたい	3－31　メモリスイッチ
	データメモリの使い方	2－10　データメモリ
	トリマの値をモニタしたい	8－13　トリマ設定値の読み出し
な	内部のリレーを電源オフ後も保持できるか	3－31　メモリスイッチ
	内部補助リレーの保持やり方	3－31　メモリスイッチ
	内部レジスタについて	5－26　内部レジスタ
	入力応答性の考え方	2－2　入出力処理と応答遅れ
	入力回路のコモンの極性は？→NPN 入力 (⊕ コモン)、PNP 入力 (⊖ コモン)、両方可能です	
	入力仕様について	1－6　入力仕様
	入力時定数	1－6　入力仕様
	入力時定数を短くできるか	3－29　ハイスピード (HSP)
	入力の立ち上がり時ワンパルスだけ取り込みたい	3－25　立上り微分 (DIFU)
は	パソコンからのコマンドで設定値を変更したい	7－1　7章 シリアル通信機能
	プログラム処理と時間	2－2　入出力処理と応答遅れ
	プログラム入力中に命令語・オペランドの検索をしたい	2－21　プログラムの「検索」
	速い信号を取り込めるか	3－29　ハイスピード (HSP)
	プログラムのモニタ	2－24　モニタ
	プロコンから KZ へのプログラム転送	8－1　8章 ファンクション
	プロコン上のプログラムをメモリカードに上書きしたい	9－5　メモリカードアクセス「書き込み」
	プロテクトをかけたい	3－31　メモリスイッチ
	BCD デジスイッチを使って、数値を入力したい	5－24　HKEY 命令
	BCD に変換したい	5－45　TBCD 命令
	保持リレーの使用について	3－31　メモリスイッチ
ま	メモリカードからのプログラムの読み出し方	9－6　メモリカードアクセス「読み出し」
	メモリカードの使い方	9－1　9章 メモリカード
	メモリカードの電池寿命	9－8　メモリカード取扱上の注意
	メモリカードへのプログラムの書き込み	9－5　メモリカードアクセス「書き込み」
	メモリカードへのプログラムの保存、呼び出しについて	9－1　9章 メモリカード
	メモリカードへ保存できるプログラム数は	9－1　メモリカードの容量
	モニタ (アナログトリマ設定値)	8－13　トリマ設定値の読み出し
	モニタ (行数を指定して)	2－27　導通モニタ
	モニタ (サイクルタイム)	2－24　サイクルタイムモニタ
	モニタ (プログラム)	2－24　モニタ
	モニタ (任意の複数ポイントを同時に)	2－25　状態モニタ
	モニタ中に命令語・オペランドの検索をしたい	2－22　プログラムの検索
ら	ラダーソフトについて　KZ ラダーユーザーズマニュアルをご覧ください	
	リレー寿命は→　機械的 2000 万回以上　電気的 10 万回以上 (頻度 20 回/分)	
	リレー割付について	2－7　リレー番号一覧
	リレーを強制的にリセットしたい	8－11　リレーの ON/OFF
わ	割り込みプログラム実行中に割り込みが発生したら	5－50　割り込み命令
	割り込み命令で使える入力端子	5－51　2－4　割り込み命令
	割り込み命令について	5－50　割り込み命令

	ニモニック	命令	ページ
@	@ADD	アッド	5－34
	@ANDA	アンドA	5－37
	@ASC	ｱｽｷｰｺｰﾄﾞ変換	5－46
	@CMP	コンペア	5－33
	@COM	コンプリメント	5－42
	@DEC	ﾃﾞｨｸﾒﾝﾄﾒﾓﾘ	5－43
	@DIV	デバイド	5－34
	@DMX	デマルチプレクス	5－44
	@EORA	ｴｸｽｸﾙｰｼﾌﾞｵｱA	5－39
	@INC	ｲﾝｸﾘﾒﾝﾄﾒﾓﾘ	5－43
	@LDA	ロードA	5－29
	@MPX	マルチプレクス	5－44
	@MUL	マルチ	5－34
	@ORA	オアA	5－38
	@RASC	ｱｽｷｰｺｰﾄﾞ逆変換	5－46
	@RLA	ﾛｰﾃｰﾄﾚﾌﾄA	5－41
	@ROOT	平方根	5－47
	@RRA	ﾛｰﾃｰﾄﾗｲﾄA	5－41
	@SLA	シフトレフトA	5－40
	@SRA	シフトライトA	5－40
	@STA	ストアA	5－29
	@SUB	サブ	5－34
	@TBCD	トランスファBCD	5－45
	@TBIN	トランスファBIN	5－45
	@TMIN	ﾄﾘﾏ設定値入力	5－28
A	ADD	アッド	5－34
	ANB	アンドバー	3－13
	AND	アンド	3－13
	ANDA	アンドA	5－37
	ANL	アンドロード	3－15
	ASC	ｱｽｷｰｺｰﾄﾞ変換	5－46
C	C	カウンタ	3－22
	CALL	ｻﾌﾞﾙｰﾁﾝｺｰﾙ	5－20
	CMP	コンペア	5－33
	COM	コンプリメント	5－42
	CON	コネクト	5－6
	CTC	高速ｶｳﾝﾀｺﾝﾊﾞｰﾚｰﾀ	5－17
	CTH	高速カウンタ	5－17
D	DEC	ﾃﾞｨｸﾒﾝﾄﾒﾓﾘ	5－43
	DI	割り込み禁止	5－49
	DIFD	立下り微分	3－25
	DIFU	立上り微分	3－25
	DIV	デバイド	5－34
	DMX	デマルチプレクス	5－44
	DW	ﾃﾞｰﾀﾒﾓﾘﾗｲﾄ	5－27
E	EI	割り込み許可	5－49
	END	エンド	3－33
	ENDH	エンドハイ	3－33
	ENDS	エンドステージ	5－9
	EORA	ｴｸｽｸﾙｰｼﾌﾞｵｱA	5－39
F	FOR	繰り返し範囲開始	5－22
H	HKEY	16キー入力	5－24
	HSP	ハイスピード	3－29
I	INC	ｲﾝｸﾘﾒﾝﾄﾒﾓﾘ	5－43
	INT	イント	5－50
	ITVL	ｲﾝﾀｰﾊﾞﾙﾀｲﾏ	5－15
J	JMP	ジャンプ	5－9
K	KEEP	キープ	3－26

	ニモニック	命令	ページ
L	LD	ロード	3－12
	LDA	ロードA	5－29
	LDB	ロードバー	3－12
M	MC	ﾏｽﾀｰｺﾝﾄﾛｰﾙ	3－30
	MCR	ﾏｽﾀｰｺﾝﾄﾛｰﾙﾘｾｯﾄ	3－30
	MEMSW	メモリスイッチ	3－31
	MPD	リード	5－7
	MPP	ポップ	5－7
	MPS	プッシュ	5－7
	MPX	マルチプレクス	5－44
	MUL	マルチ	5－34
N	NEXT	繰り返し範囲終了	5－22
	NOP	ノップ	3－33
O	OR	オア	3－14
	ORA	オアA	5－38
	ORB	オアバー	3－14
	ORL	オアロード	3－15
	OUB	アウトバー	3－17
	OUT	アウト	3－17
R	RASC	ｱｽｷｰｺｰﾄﾞ逆変換	5－46
	RES	リセット	3－18
	RET	ｻﾌﾞﾙｰﾁﾝﾘﾀｰﾝ	5－20
	RETI	リターンイント	5－50
	RLA	ﾛｰﾃｰﾄﾚﾌﾄA	5－41
	ROOT	平方根	5－47
	RRA	ﾛｰﾃｰﾄﾗｲﾄA	5－41
S	SBN	ｻﾌﾞﾙｰﾁﾝｴﾝﾄﾘ	5－20
	SET	セット	3－18
	SFT	シフト	3－27
	SLA	シフトレフトA	5－40
	SRA	シフトライトA	5－40
	STA	ストアA	5－29
	STE	ステップエンド	5－13
	STG	ステージ	5－9
	STP	ステップ	5－13
	SUB	サブ	5－34
T	TBCD	トランスファBCD	5－45
	TBIN	トランスファBIN	5－45
	TMH	高速タイマ	3－20
	TMIN	ﾄﾘﾏ設定値入力	5－28
	TMR	タイマ	3－19
	TMS	高速1msタイマ	3－20
U	UDC	ｱｯﾌﾟﾀﾞｳﾝｶｳﾝﾀ	3－24
W	W－DE	ｳｴｲﾄﾀﾞｳﾝｴｯｼﾞ	5－2
	W－OFF	ウエイトオフ	5－1
	W－ON	ウエイトオン	5－1
	W－UE	ｳｴｲﾄｱｯﾌﾟｴｯｼﾞ	5－2

■MEMO

KZ-

コーディングシート

/

システム			作成
形式	図番		

行番号	命令語	FUN No.	オペランド	行番号	命令語	FUN No.	オペランド
00				50			
01				51			
02				52			
03				53			
04				54			
05				55			
06				56			
07				57			
08				58			
09				59			
10				60			
11				61			
12				62			
13				63			
14				64			
15				65			
16				66			
17				67			
18				68			
19				69			
20				70			
21				71			
22				72			
23				73			
24				74			
25				75			
26				76			
27				77			
28				78			
29				79			
30				80			
31				81			
32				82			
33				83			
34				84			
35				85			
36				86			
37				87			
38				88			
39				89			
40				90			
41				91			
42				92			
43				93			
44				94			
45				95			
46				96			
47				97			
48				98			
49				99			

KZ-

I/O番号割付シート

/

システム				作成
形式	図番			

CH	コメント	CH	コメント
00		00	
01		01	
02		02	
03		03	
04		04	
05		05	
06		06	
07		07	
08		08	
09		09	
10		10	
11		11	
12		12	
13		13	
14		14	
15		15	

CH	コメント	CH	コメント
00		00	
01		01	
02		02	
03		03	
04		04	
05		05	
06		06	
07		07	
08		08	
09		09	
10		10	
11		11	
12		12	
13		13	
14		14	
15		15	

CH	コメント	CH	コメント
00		00	
01		01	
02		02	
03		03	
04		04	
05		05	
06		06	
07		07	
08		08	
09		09	
10		10	
11		11	
12		12	
13		13	
14		14	
15		15	

改訂履歴

印刷年月日	版　　数	改　訂　内　容
1998年 7 月	14版	1-1. KZの取扱い方法 　システム構成にKZ-AD4/DA4を追加 6-1. 高速カウンタの概要 　注意事項の追加
1998年 9 月	15版	
1998年10月	16版	1-2. 仕様 　電源ユニットの仕様追加 5-3. 割り込み命令 　(例 1) プログラムの変更 7-1. KZ とパソコンの通信 　注意事項の記入
1999年10月	17版	
2001年11月	18版	

保証について

1　保証期間

製品の保証期間は、ご指定の場所に納入後１年間といたします。

2　保証範囲

(1)上記保証期間中に当社の責任による故障が発生した場合は、無償で修理させていただきます。
ただし、次に該当する場合は、保証の範囲から除外させていただきます。

① 取扱説明書、ユーザーズマニュアル、別途取り交わした仕様書などに記載された以外の不適当な条件・環境・取り扱い・使用方法に起因した故障。

② お客様の装置または、ソフトウエアの設計内容など、当社製品以外に起因した故障。

③ 当社以外による改造、修理に起因した故障。

④ 取扱説明書、ユーザーズマニュアルなどに記載している消耗部品が正しく保守、交換されていれば、防止できたと確認できる故障。

⑤ 当社出荷時の科学・技術水準では、予見が不可能だった事由による故障。

⑥ その他、火災、地震、水害などの災害および電圧異常など当社側の責任ではない外部要因による故障。

(2)保証範囲は上記（1）を上限とし、当社製品の故障に起因するお客様での二次損害（装置の損傷、機会損失、逸失利益等）及びいかなる損害も保証の対象外とさせていただきます。

3　製品の適用範囲

当社製品は、一般工業向けの汎用品として設計・製造されております。
従いまして、下記のような用途での使用は意図しておりませんので適用外とさせていただきます。
ただし、事前に当社までご相談いただき、お客様の責任において製品の仕様をご確認のうえ、定格・性能に対してご了承いただき、必要な安全対策を講じていただく場合は適用可能とさせていただきます。
ただし、この場合においても保証範囲は上記と同様といたします。

① 原子力発電、航空、鉄道、船舶、車両、医療機器等の人命や財産に多大な影響が予想される設備

② 電気、ガス、水道等の公共設備

③ 屋外での使用および、それに準ずる条件・環境での使用

全商品、送料無料で
当日出荷

必要な時に、必要な量だけ
在庫不要でトータルコストを削減

■ **制御システムに関するお問合せ** 0120-423-723
最寄りの担当営業所に直接つながります。

■ **情報サービス** www.keyence.co.jp
カタログ、取扱説明書、マニュアル、CADデータ等をダウンロードできます。

■ **輸出書類サービス** www.keyence.co.jp/yushutsu
輸出に必要な書類をその場でダウンロードできます。

株式会社 キーエンス
本社・研究所／制御システム事業部
〒533-8555 大阪市東淀川区東中島1-3-14

仕様は改良のため予告なく変更することがあります。

制御2-0129

Copyrightⓒ 2009 KEYENCE CORPORATION.
All rights reserved.

1030-21a 079008

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP