



SYSMAC C200H ASCIIユニット

形C200H-ASC02



ユーザーズマニュアル

SYSMAC C200H ASCIIユニットは
当社の高度な制御技術と豊富な経験により開発された
プログラマブルコントローラ SYSMAC C200Hの高機能ユニットです。
このユーザズマニュアルは、
SYSMAC C200H ASCIIユニットの
仕様、取扱方法について主に解説したものです。
機能、性能などを十分ご理解のうえ、正しくご使用ください。
なお、ご使用の際はC200Hユーザズマニュアルを併せてご覧ください。

— おことわり —

- (1) 本マニュアルの内容の一部または全部を無断で複写、複製、転載することを禁じます。
- (2) 本マニュアルの内容に関しては、改良のため予告なしに仕様などを変更することがありますので、予めご了承ください。
- (3) 本マニュアルの内容に関しては万全を期しておりますが、万一ご不審な点や誤りなどお気づきのことがありましたら、お手数ですが巻末記載の当社支店または営業所までご連絡ください。その際、巻末記載のマニュアルNo.も併せてお知らせください。

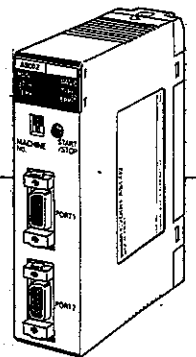
— 著作権・商標について —

PC-98シリーズは日本電気株式会社の製品です。

SYSMAC C200H ASCIIユニット

形C200H-ASC02

ユーザーズマニュアル



はじめに

■本マニュアルの対象の方

本マニュアルでは、ASCIIユニット形C200-ASC02（以下、ASCIIユニットといいます）の仕様および設置、取り扱い方法について次の方を対象に記述しています。

電気、制御、BASICプログラムの知識をお持ちの方で、

- ・制御機器の導入を担当される方
- ・制御システムの設計をされる方
- ・現場を管理される方

■お願い

- ・この製品は必ず本書に記載の一般使用及び性能使用の範囲内で使用してください。
- ・本マニュアルに記載のない条件や環境での使用、および原子力制御・鉄道・航空施設・車両・燃焼装置・医療機器・娯楽機器・安全機器・その他人命や財産に大きな影響が予測されるなど、特に安全性が要求される用途への使用をご検討の場合は、定格・性能に対し余裕を持った使い方やフェールセーフ等の安全対策へのご配慮をいただくとともに、当社営業担当者までご相談いただき仕様書等による確認をお願いします。
- ・このマニュアルは、ASCIIユニットを使用するうえで、必要な情報を記載しています。使用される前に、このマニュアルをよく読んで十分に理解してください。また、お読みになった後も本マニュアルは大切に保管し、いつも手元においてお使いください。

■海外でのご使用について

本製品の内、外国為替及び外国貿易管理法に定める輸出許可、承認対象貨物（又は技術）に該当するものを輸出（又は非居住者に提供）する場合は同法に基づく輸出許可、承認（又は役務取引許可）が必要です

安全上のご注意

このユーザーズマニュアルでは、製品をより安全に使用していただくために、注意事項を次のような表示と図記号で示しています。ここで示した注意事項は、安全に関する重大な内容を記載しています。必ず守ってください。
表示と意味は次のとおりです。

●表示の説明



誤った取り扱いをすると、人が死亡または重傷を負う可能性が想定される内容を示します。



誤った取り扱いをすると、人が傷害を負ったり、物的損害の発生が想定される内容を示します。

●図記号の説明



⊘記号は、禁止を意味しています。
具体的な内容は、⊘の中と文章で示します。
左図の場合は、「分解禁止」を表します。



△記号は、注意（警告を含む）を意味しています。
具体的な内容は、△の中と文章で示します。
左図の場合は、「感電注意」を表します。



△記号は、注意（警告を含む）を意味しています。
具体的な内容は、△の中と文章で示します。
左図の場合は、「一般的な注意」を表します。



●記号は、強制を意味しています。
具体的な内容は、●の中と文章で示します。
左図の場合は、「一般的な強制事項」を表します。

警告

通電中は、製品を分解しないでください。
感電の恐れがあります。



通電中は、端子に触れないでください。
感電の恐れがあります。



プログラマブルコントローラ(CPUユニットおよび各ユニットを含む—以下、PCといいます)の故障やPCの外部要因による異常が発生した場合も、システム全体が安全側に働くように、PCの外部で安全対策を施してください。
異常動作により、重大な事故につながる恐れがあります。



- (1) 非常停止回路、インターロック回路、リミット回路など、安全保護に関する回路は、必ずPC外部の制御回路で構成してください。
- (2) PCは、自己診断機能で異常を検出したときや、運転停止故障診断(FALS)命令を実行したとき、運転を停止して全出力をOFFします。
このとき、システムが安全側に動作するよう、PC外部で対策を施してください。
- (3) 出力リレーの溶着や焼損、出力トランジスタの破壊などによって、PCの出力がONまたはOFFになったままになることがあります。
このとき、システムが安全側に動作するよう、PC外部で対策を施してください。
- (4) PCのDC24V出力(サービス電源)が過負荷の状態または短絡されると、電圧が低下し、出力はOFFとなることがあります。
このとき、システムが安全側に動作するよう、PC外部で対策を施してください。

使用上のお願い

- 次のことを行うときは、P C 本体の電源をOFFにしてください。
 - ・ 電源ユニット、I / O ユニットなどの各種ユニットやC P U ユニット、メモ리카セットの着脱をするとき
 - ・ 装置を組み立てるとき
 - ・ ディップスイッチやロータリスイッチを設定するとき
 - ・ ケーブルを接続、配線するとき
 - ・ コネクタを取り付けたり、取り外したりするとき
- 本製品を分解して修理や改造はしないでください。
- P C のベース取り付けねじ、端子台のねじ、ケーブルのねじは規定のトルクでしめてください。
- 製品出荷時に通気孔にラベルが貼ってある製品は、ワイヤくず混入防止のため、ラベルを付けたまま配線してください。
- 配線後は、放熱のため、必ずラベルを外してください。
- 端子台やコネクタを接続する場合は、方向や極性を確認して取りつけてください。
- 端子台、メモリユニット、接続ケーブルなどロック機構のあるものは、必ずロックしていることを確認してからご使用ください。
- 作成したユーザプログラムは、十分な動作確認を行った後、本運転に移行してください。
- 次の操作は、設備に影響がないことを確認した上で行ってください。
 - ・ P C 本体の動作モード切替え
 - ・ 接点の強制セット／リセット
 - ・ 現在値や設定値の変更
- ケーブルやコードは、無理に引っ張ったり曲げたり、重いものを載せたりしないでください。
- 部品を交換する際は、必ず定格が正しいことを確認した上で交換してください。
- 接地された金属に触るなどして人体の静電気を放電させてから、ユニットに触れてください。

正しい使い方

- 本マニュアルに示す通り、正しく設置してください。
- 次のような環境には設置しないでください。
 - ・日光が直接当たる場所
 - ・周囲温度や相対湿度が仕様値の範囲を超える場所
 - ・温度変化が急激で結露するような場所
 - ・腐食性ガス、可燃性ガスのある場所
 - ・ちり、ほこり、塩分、鉄粉が多い場所
 - ・製品本体に直接振動や衝撃が伝わる場所
 - ・水、油、薬品の飛沫がかかる場所
- 次のような場所で使用する際は、遮蔽対策を十分に行ってください。
 - ・静電気などによるノイズが発生する場所
 - ・強い電界や磁界が生じる場所
 - ・放射能を被曝する恐れのある場所
 - ・電源線が近くを通る場所

マニュアル改訂履歴

マニュアル改訂記号は、表紙および裏表紙の左下に記載されている Man.No. の後尾に付記されます。

Man. No. SBCC-442H

改訂記号

改訂記号	改定日	改訂ページ・内容
—	1989年2月	初版印刷
B	1989年11月	説明追加 2・10、5・1
C	1994年12月	説明追加、誤り修正 2・9～2・12、5・9、6・7、6・27～6・30、7・41、7・61、8・15、9・7
D	1995年10月	説明追加 2・3、6・30、7・29、7・33 安全に関するご注意の追記
E	1997年4月	説明追加、誤り修正 2・9、2・10、7・6～7・11、7・15、7・60、8・1、8・12、8・13 安全に関するご注意の追記
F	1997年9月	説明追加 3・1、4・1、5・1、5・14、付・1～付・4 安全に関するご注意の追記
G	1998年3月	説明追加、誤り修正 1・4、2・1 安全に関するご注意の追記
H	2001年6月	説明追加、誤り修正 2・1、7・25、7・50、8・12

目次

第1章 特長とシステム構成

1-1 特長	1・1
1-2 システム構成	1・4

第2章 仕様と各部の機能

2-1 仕様	2・1
2-2 各部の名称と機能	2・3
2-3 スイッチの設定方法	2・5
2-4 インターフェース	2・7

第3章 リレー番号の割付方法

3-1 チャンネルの設定	3・1
3-2 リレー番号の割付け	3・2

第4章 動作説明

4-1 ASCIIユニットを動作させるにあたって	4・1
4-2 プログラムについて	4・2
4-3 PCとのデータの授受について	4・3
4-4 PC命令実行とリフレッシュタイミングの関係	4・5

第5章 プログラミング手順

5-1 プログラムの作成について	5・1
5-2 BASICプログラムの作成手順	5・2
5-3 BASICプログラムの起動/停止方法	5・8
5-4 BASICプログラムの転送	5・9
5-5 機械語プログラムの転送	5・14

第6章 プログラム例

6-1 プログラム基本例	6・1
6-2 プログラム実施例	6・7

第7章 BASIC命令

7-1 BASIC命令の構成	7・1
7-2 BASIC命令一覧	7・6
7-3 BASIC命令の説明	7・12

第8章 機械語プログラム

8-1 機械語プログラムの作り方	8・1
8-2 モニタモード	8・2
8-3 メモリマップ	8・12

第9章 異常とその処置

9-1 エラーメッセージ一覧	9・1
9-2 異常とその処置	9・4

■形式一覧表

■付録

- ・「Windows 95ハイパーターミナルでASCIIユニットのプログラミングをする方法」
- ・「ターミナルを使用したASCIIユニットの機械語プログラミング方法」

■生産中止(予定)機種一覧

第1章 特長とシステム構成

1-1 特長

周辺ツールの制御をインテリジェントに行う ASCIIユニット

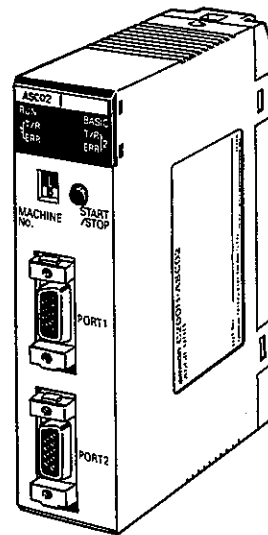
形 C 2 0 0 H — A S C 0 2 (EEP-ROMタイプ)

BASICプログラムをEEP-ROMに保存することができます。

SYSMAC C200Hの高機能I/Oユニットとして使用できます。

〈応用例〉

- 入力装置からの入力
 - ・パーソナルコンピュータからの数値入力、生産指示
 - ・バーコードリーダからのデータ入力
- 出力装置への出力
 - ・プリンタへの生産状況の印字
 - ・パーソナルコンピュータ、表示器へのPCの動作、データの表示
- その他
 - ・時刻のPCへの報知、プリンタへの印字
 - ・PC側では困難な複雑な演算処理の代行



- RS-232Cインターフェースを2ポート内蔵
生産現場でのマンマシンインターフェースとして、入出力可能なRS-232Cポートを2ポート内蔵しています。
従って、パーソナルコンピュータ、バーコードリーダなどの入力機器からの入力データの取込み、パーソナルコンピュータ、表示器、プリンタなどの出力機器へのデータ出力などすべてオンラインで行うことができます。
- BASICによるプログラミング
PCとのデータの授受、演算および周辺機器のコントロールがBASICプログラムにより容易に行えます。
BASICプログラムの入力、修正はパーソナルコンピュータなどを接続して行います。
- リアルタイムクロック内蔵
リアルタイムクロックを内蔵していますので本ユニットを時計ユニットとして使用することもできます。(西暦の下2桁、月、日、曜日、時、分、秒)
- バッテリ内蔵
バッテリによりBASICプログラム、機械語プログラム、リアルタイムクロックのバックアップが可能です。
また、EEP-ROMにBASICプログラム、機械語プログラムを保存できバッテリなしでも動作させることができます。
ただし、この場合はリアルタイムクロックは電源OFFで不定となります。
- 機械語モニタ内蔵
BASICプログラムでは処理しきれない演算を行う場合、モニタ機能によりアセンブラ(6303)にてプログラムが可能です。
- 三角関数内蔵
BASICにて三角関数、逆三角関数が使えます。

■形C200H-ASC01と形C200H-ASC02との相違点

形C200H-ASC02は、形C200H-ASC01に対して、下記の点が機能追加・機能向上になりました。

ただし、ASC01で作成したBASICプログラムをASC02にて動作させるときには、*印の項目について、BASICプログラムの変更が必要です。

1. BASICプログラム・リアルタイムクロックのバックアップ方式

項 目	ASC01	ASC02
バックアップ方式	コンデンサバックアップ	バッテリーバックアップ
バックアップ時間	7日間 (25℃)	5年間 (25℃)
診断機能	な し	バッテリー電圧LOW検出

バッテリー電圧LOW検出として、バッテリーの寿命がなくなると、「1ERR」LEDが点滅し、バッテリー異常フラグ (n+3CHの06ビット) がONします。

(n=100+10×号機No)

2. BASIC命令の追加

詳細は、「7-3 BASIC命令の説明」をご参照ください。

命 令	機 能	使用例
WAIT	次の命令語の実行終了までの時間を設定する。	実行時間の長い命令の実行時間の監視やPC READ, PC WRITE命令で、PCからの起動がかからなかったために生じるデッドロック防止などに使用します。
ON COM GOSUB	通信回線からの割込みのサブルーチン分岐先を定義する。	ポート1、ポート2の通信回線からの割込みにより処理を行うときに使用します。
COM ON/OFF/STOP	通信回線からの割込みの許可、禁止、停止を制御する。	ON COM GOSUB文で定義された、通信回線からの割込みの許可、禁止、停止を制御するときに使用します。 ON COM GOSUB命令実行時には、COM OFF状態になっています。
PC ON/OFF/STOP	PCからの割込みの許可、禁止、停止を制御する。	ON PC GOSUB文で定義された、PCからの割込みの許可、禁止、停止を制御するときに使用します。 ON PC GOSUB命令実行時には、PC OFF状態になっています。
KEY ON/OFF/STOP	キー入力による割込みの許可、禁止、停止を制御する。	ON KEY GOTO, ON KEY GOSUB文で定義されたキー入力による割込みの許可、禁止、停止を制御するときに使用します。 ON KEY GOTO, ON KEY GOSUB命令実行時には、KEY OFF状態になっています。

3. BASIC命令の改善

詳細は、「7-3 BASIC命令の説明」をご参照ください。

項 目	機 能	使用例
PC READ ("@"), PC WRITE ("@") 命令のS型フォーマット	フォーマットに従い、データを配列変数に格納、または配列変数から転送する。	配列変数を用いて、1命令で255CHまでのデータ転送を行うときに使用します。 PCとの転送データ数が多いときに有効です。
OPEN命令の X ON/X OFF指定	X ON/X OFFによる通信制御の指定を行う。	受信用バッファの使用度による通信制御を行うときに使用します。 通信するデータ量が多いときに有効です。
* OPEN命令のRTS指定	RTS信号の定常時のON/OFF指定をする。	周辺装置から常にデータを受付けるときに使用します。通信割込み、キー割込みをいつでも受けることができます。
USR命令のWDTリフレッシュ制御	機械語プログラム実行時のウォッチドッグタイマ・リフレッシュの指定を行う。	機械語プログラム実行時の暴走防止を行うときに使用します。

4. BASIC運転動作の改善

詳細は、「7-3 BASIC命令の説明」をご参照ください。

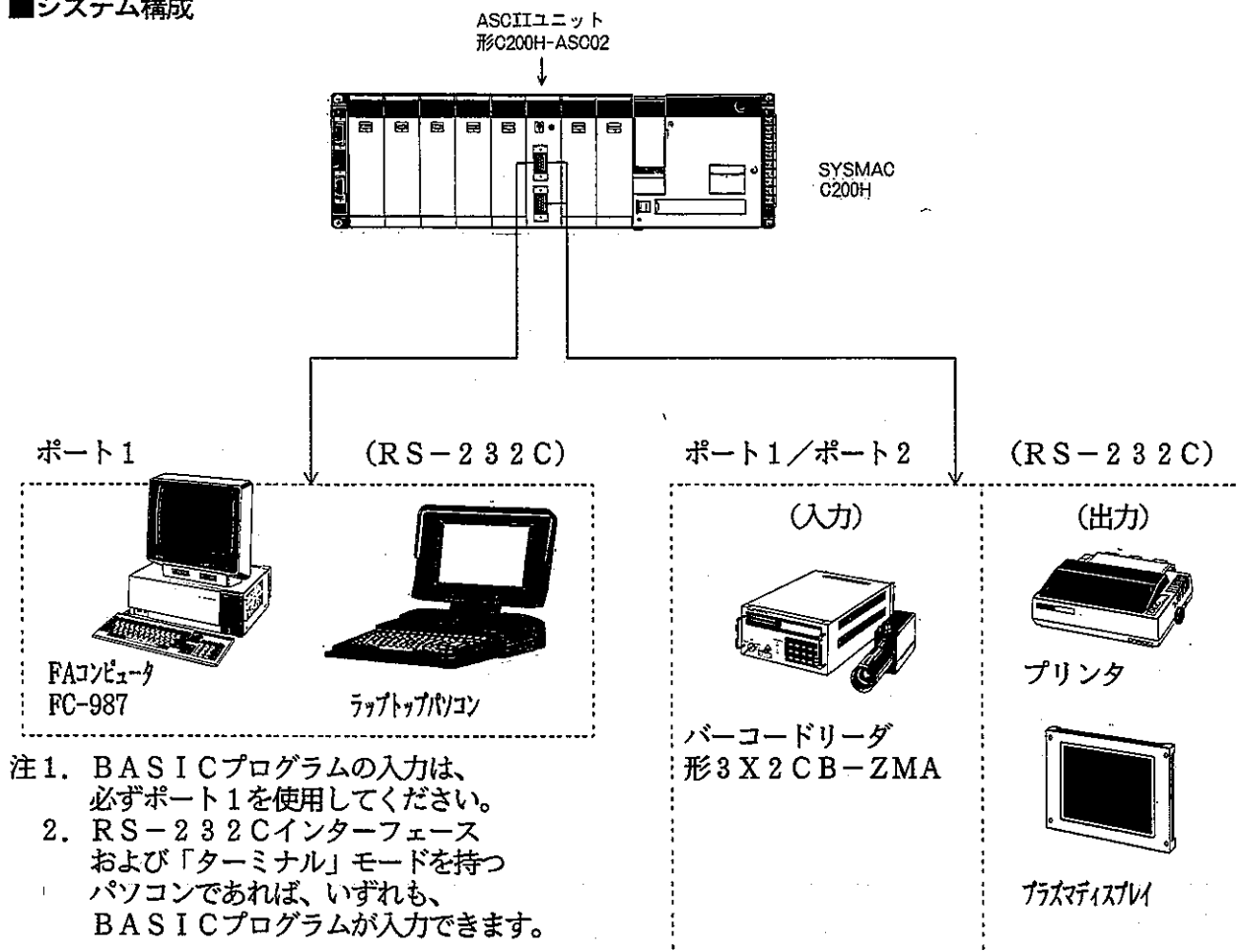
項 目	ASC01	ASC02
* ON PC GOSUB ON KEY GOTO ON KEY GOSUB命令 の割込制御状態	本命令実行時から、割込許可 (PC ON、KEY ON) 状態となります。	本命令実行時には、割込禁止 (PC OFF、KEY OFF) 状態となります。 PC ON/OFF/STOP、KEY ON/OFF/STOP命令により状態を変更できます。
* ON COM GOSUB ON PC GOSUB ON KEY GOSUB命令 の多重割込み	各割込みによるサブルーチン・プログラム実行中は、他の割込みによるサブルーチン分岐が禁止 (OFF) されています。	各割込みによるサブルーチン・プログラム実行中は、他の割込みによるサブルーチン分岐が停止 (STOP) されています。 多重割込みなどは、COM ON/OFF/STOP、PC ON/OFF/STOP、KEY ON/OFF/STOP命令により選択できます。
ON PC GOSUB命令 割込時の ASCIIビジーフラグ (PC ON/STOP時)	WRITEフラグの立上りでASCIIビジーフラグがONし、サブルーチン分岐するときにASCIIビジーフラグがOFFします。	WRITEフラグの立上りでASCIIビジーフラグが1スキャンONします。 PCからの割込みがASCIIユニットで受け付けられたことが確認できます。

《参考》

ASC01で作成したBASICプログラムをASC02にて動作させるときには、*印の項目について、BASICプログラムの変更が必要です。

1-2 システム構成

■システム構成



《参考》

●ASCIIユニットの装着可能数

- ASCIIユニットは、CPU装置、I/O増設装置、リモートI/O子局装置に、他の高機能I/Oユニット（位置制御ユニットなど）を含めて、合計10ユニットまで装着できます。
- リモートI/O子局装置への装着は、1つの装置に対し、最大4ユニットまでとしてください。
- CPUユニットの隣2スロットには装着すると、上位リンクユニットやプロコンなどの周辺ツールが取付けられなくなります。

第2章 仕様と各部の機能

2-1 仕様

■一般仕様

SYSMAC Cシリーズに準じます。

■性能仕様

項目	仕様
通信方式	半二重
同期方式	調歩同期
伝送速度	ポート1 300/600/1200/2400/4800/9600 BPS ポート2 300/600/1200/2400/4800/9600/19,200 BPS (内部スイッチで設定)
伝送方式	ポイント・ツー・ポイント
伝送距離	最大15m
インターフェース	RS-232C準拠2ポート (Dサブ 9P コネクタ使用)
メモリ容量	BASICプログラムエリア } 24Kバイト (RAM) BASICデータエリア } (プログラムはバッテリーによりバックアップ) BASICプログラム保管エリア 24Kバイト (EEP-ROM) (次ページ参照) プログラムメモリ領域を3分割して3種類のプログラムを格納することができます。
転送容量	最大255CH ただし、20CH/1スキャン
時計機能	年、月、日、曜日、時、分、秒 (うるう年対応) (バッテリーバックアップ) 精度: 月差±30秒 (25℃にて)
診断機能	CPUウォッチドッグタイマ、バッテリー電圧LOW検出
バッテリー寿命	25℃で5年間 (これ以上高い温度で使用する場合は、寿命が短くなります。)
内部消費電流	DC5V 200mA以下
外形寸法 (mm)	35 (W) × 130 (H) × 100.5 (D)
質量	400g以下

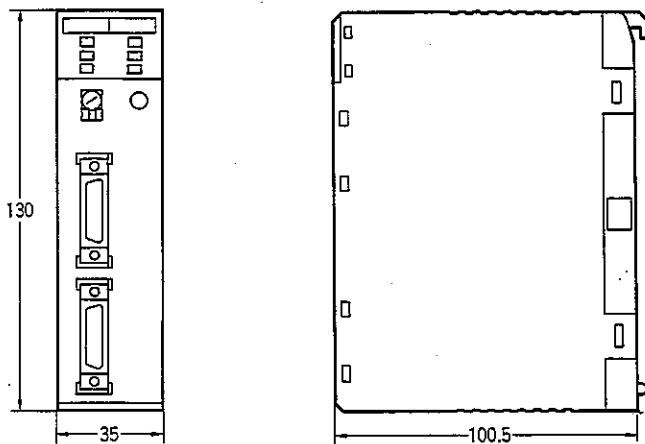
《お願い》

電源投入時の初期処理中は、ポートに不正な出力が出ることがあります。接続機器側で電源投入時の不正出力を無視する (受信データのクリアなど) ようにしてください。

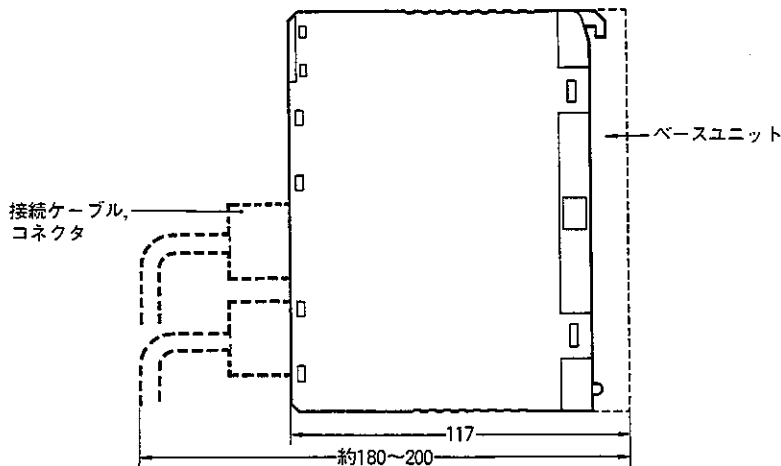
■EEP-ROMについて

- ・EEP-ROMは、RAMに書込まれたプログラムの保管エリアとして使用します。
- ・RAMからEEP-ROMへプログラムを転送するには「SAVE」命令を使用します。
- ・EEP-ROMからRAMへプログラムを転送するには、「LOAD」命令を使用します。
- ・裏面左側スイッチのNo.2をONに設定すると、電源投入時およびリスタート時に自動的にEEP-ROMからRAMへプログラムを転送します。
- ・書き込み（SAVE）、読出し（LOAD）は、全プログラムを一括して行います。
- ・EEP-ROMへの書き込み（SAVE）回数には寿命があります。プログラムの書き込み（SAVE）は5,000回以下を目安としてください。

■外形寸法



■ユニット装着時の寸法



- ・本ユニットをご使用になる場合は実装する制御盤の奥行きを十分に考慮してください。
- ・付属のコネクタを組立てて接続ケーブルを完成させてください。

2-2 各部の名称と機能

■ユニット前面部

表示部

RUN : 運転中	BASIC: RUN中
1 { T/R:伝送中 ERR:異常	2 { T/R:伝送中 ERR:異常

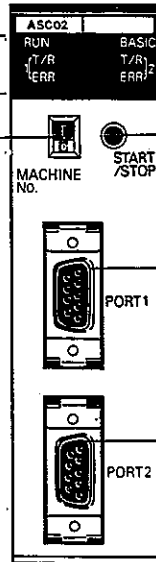
設定スイッチ

MACHINE No. : 号機No設定

- ・⊖ドライバで設定してください。途中で止めたままにすると誤設定となります。また溝がつぶれないようご注意ください。
- ・設定はPC本体の電源をOFFにして行ってください。

操作スイッチ

START/STOP : スタート/ストップスイッチ



RS-232Cインターフェース
コネクタ ポート1
周辺機器接続用コネクタです。(Dサブ9ピンコネクタ)
(BASICプログラムの入力はこのポートから行ってください。)

RS-232Cインターフェース
コネクタ ポート2
周辺機器接続用コネクタです。(Dサブ9ピンコネクタ)
(BASICプログラムの入力はこのポートからはできません。)
付属のコネクタを組立てて接続ケーブルを製作してください。

■表示の機能 (○点灯、◐点滅、●消灯)

表示の名称	表示	機能
RUN (運転中)	○ (緑)	正常動作中のとき点灯します。異常時消灯します。
T/R (伝送中) (1,2はポート1,ポート2を示します。)	◐ (緑)	データを送信中または受信中のとき点滅します。
ERR (異常) (1,2はポート1,ポート2を示します。)	○ (赤)	伝送異常 (パリティエラーなど) のとき、または送信可条件待ちのとき点灯します。
	◐ (赤) (10分)	バッテリー電圧が低下したとき、またはバッテリーが未挿入のとき点滅します。
BASIC (RUN中)	○ (緑)	BASICプログラムがRUN中のとき点灯します。
	◐ (緑)	BASICプログラムが停止中のとき、またはBASICプログラムRUN中の入力待ちのとき点滅します。
	● (緑)	モニタモードのとき消灯します。

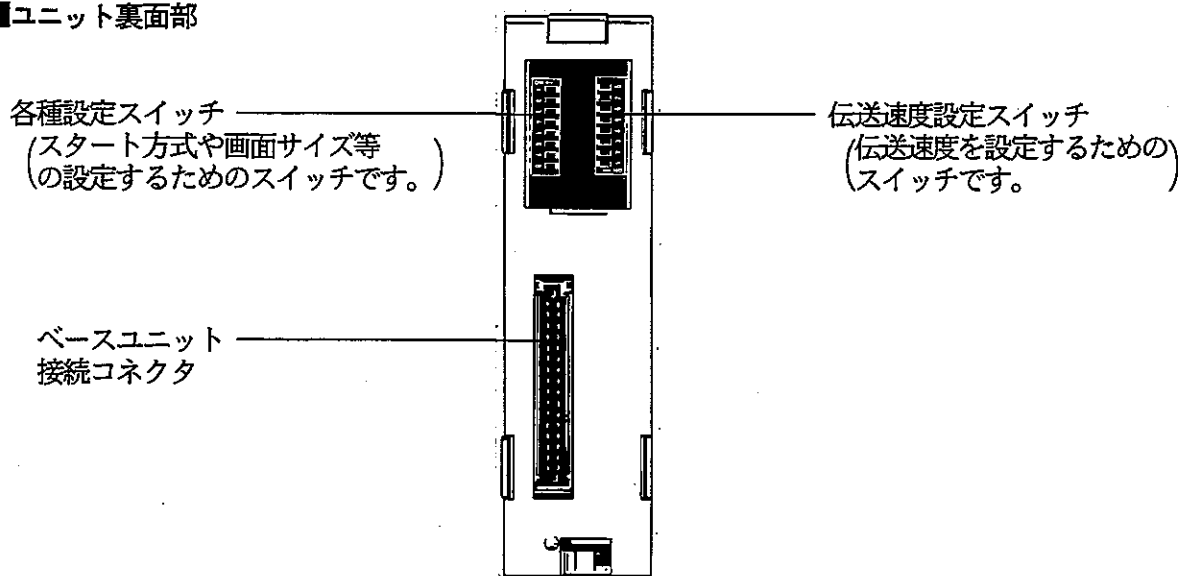
■設定スイッチの機能

スイッチの名称	機能
MACHINE No. (号機No設定)	0～9までの号機Noを設定します。他の高機能I/Oユニットと同じ番号を設定すると、「I/Oユニットオーバー」となり動作しませんのでご注意ください。

■操作スイッチの機能

スイッチの名称	機能
START/STOP (スタートストップスイッチ)	BASICプログラムをスタート/ストップさせます。RUN中に押すとBASICプログラムは停止し、停止中に押すとBASICプログラムはRUNします。

■ユニット裏面部



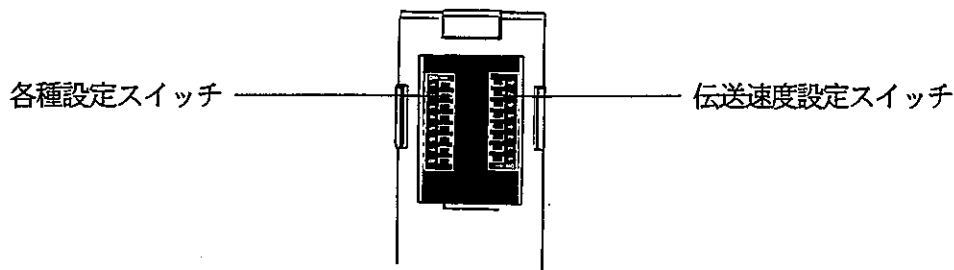
●左側スイッチの機能 (各種設定スイッチ)

スイッチNo.	機 能	
1	スタート方式設定	BASICプログラムの起動方式 (自動/手動) を設定します。
2	EEP-ROM→RAM 自動転送設定	電源投入時およびリスタート時にBASICプログラムを自動的にEEP-ROM→RAM転送するか否かを設定します。
3	プログラムNo設定	BASICプログラムを使用するプログラムNoを設定します。 プログラムNoは、PGEN命令 (7・40ページ参照) で変更することができます。
4		
5	未使用	必ずOFFにしてください。
6	画面サイズ設定	BASICプログラムを入力するパソコンの画面サイズに合わせて設定します。
7		
8		

●右側スイッチの機能 (伝送速度設定スイッチ)

スイッチNo.	機 能	
1	ポート1 伝送速度設定	ポート1に接続する周辺機器の伝送速度に合わせて設定します。
2		
3		
4	未使用	必ずOFFにしてください。
5	ポート2 伝送速度設定	ポート2に接続する周辺機器の伝送速度に合わせて設定します。
6		
7		
8	未使用	必ずOFFにしてください。

2-3 スイッチの設定方法



■左側スイッチの設定（各種設定スイッチ）

↑ OFF : 0
↓ ON : 1



スタート方式設定

スイッチNo.	1	機 能
設定内容	0	手動スタート 電源投入時BASICプログラムは実行しません。START/STOPスイッチまたはパソコン（ポート1）からの起動命令によりスタートします。
	1	自動スタート 電源投入により自動的にBASICプログラムを実行します。

(0 : OFF / 1 : ON)

BEP-ROM → RAM 自動転送設定

スイッチNo.	2	機 能
設定内容	0	RAMのみを使用するときは“0”に設定します。
	1	電源投入時およびリスタート時にBASICプログラムを自動的にBEP-ROMからRAMへ転送させるときは“1”に設定します。

(0 : OFF / 1 : ON)

プログラムNo.設定

スイッチNo.	3	4	使用プログラムNo.
設定内容	0	0	No. 1
	1	0	
	0	1	No. 2
	1	1	No. 3

(0 : OFF / 1 : ON)

画面サイズ設定

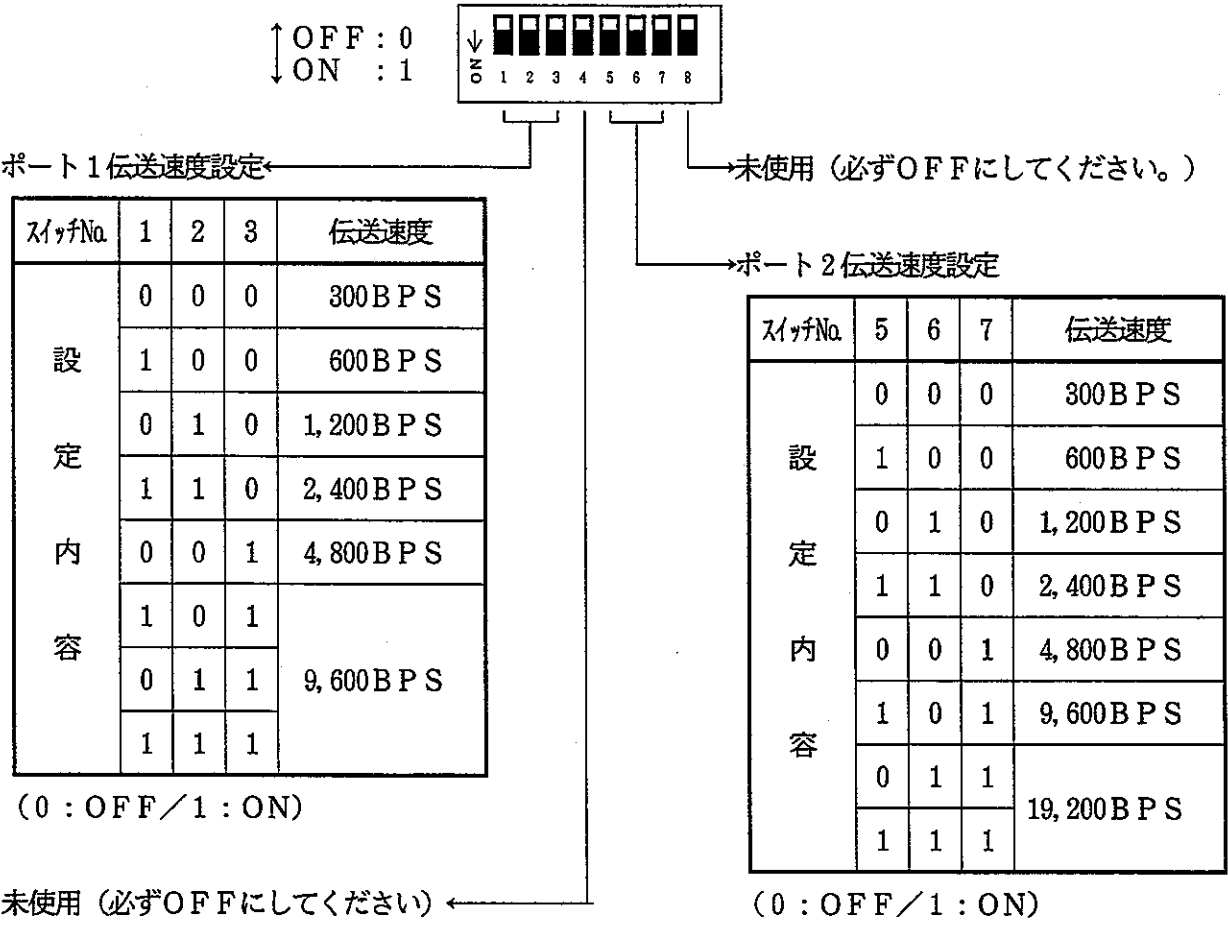
スイッチNo.	6	7	8	画面サイズ
設定内容	0	0	0	40列× 7行
	1	0	0	40列× 8行
	0	1	0	40列× 15行
	1	1	0	40列× 16行
	0	0	1	80列× 15行
	1	0	1	80列× 16行
	0	1	1	80列× 24行
	1	1	1	80列× 25行

(0 : OFF / 1 : ON)

→未使用（必ずOFFにしてください。）

電源投入後、またはリセット解除時、どのプログラムNo.で実行するかを選択します。その後は、PGEN命令（7・16ページ参照）でプログラムNo.を変更することができます。

■右側スイッチの設定 (伝送速度設定スイッチ)



2-4 インターフェース

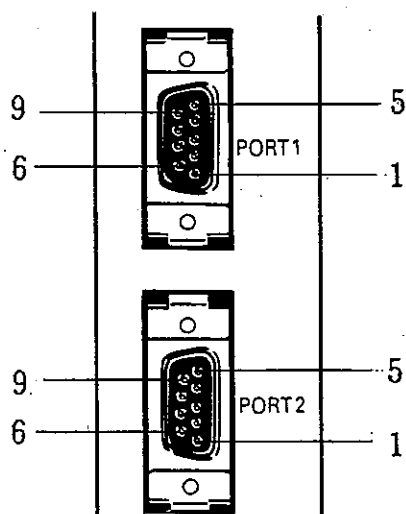
ASCIIユニットと周辺機器とは、RS-232Cインターフェースを通して接続されます。周辺機器との接続には接続ケーブルが別途必要です。

ご使用者においてASCIIユニット付属の接続コネクタを使用して配線、組立てを行ってください。ケーブルの配線はピン番号、信号名称を確認のうえ行ってください。

■コネクタのピン番号と信号名称

RS-232Cインターフェース

- ・電気的特性：EIA RS-232Cに準拠
- ・信号線はポート1、ポート2で共通ですが、BASICプログラムの入力ポート1を使用してください。
- ・Dサブ9ピンコネクタを使用しております。



ピン番号	信号記号	信号名称	信号方向
1	F G	保安用接地	—
2	S D	送信データ	出力
3	R D	受信データ	入力
4	R T S	送信要求信号	出力
5	C T S	送信可能信号	入力
6	—	使用していません。	—
7	D S R	送受信可能信号	入力
8	D T R	送受信可能信号	出力
9	S G	信号用接地	—

注 信号方向はASCIIユニット側から見た方向です。

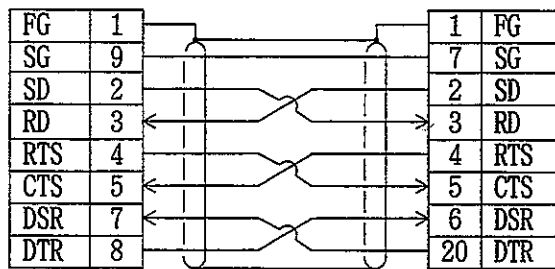
- ・適合コネクタ
プラグ：形XM2A-0901
フード：形XM2S-0911（オムロン製）または同等品
（プラグ、フードは各2個、本ユニットに付属しています。）
- ・推奨ケーブル：
AWG28×5P IFVV-SB（藤倉電線製）
CO-MA-VV-SB 5P×28AWG（日立電線製）
- ・線路長：15m

■接続方法

●パーソナルコンピュータとの接続例

ASCIIエニツ側

パーソナルコンピュータ側

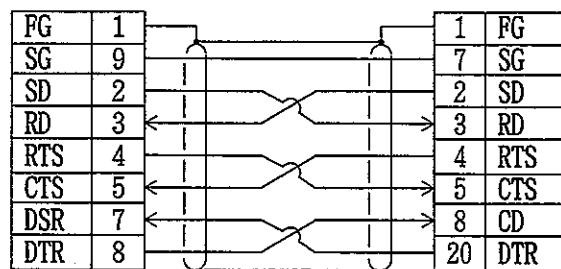


数字はピン番号を示します。

●バーコードリーダー 形3X2CB-ZMA
(オムロン製) との接続例

ASCIIエニツ側

バーコードリーダー側

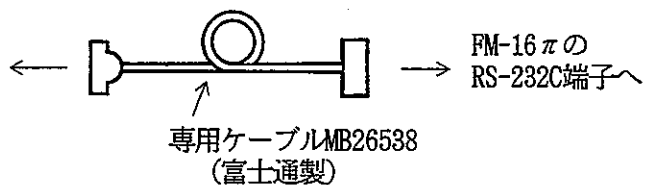
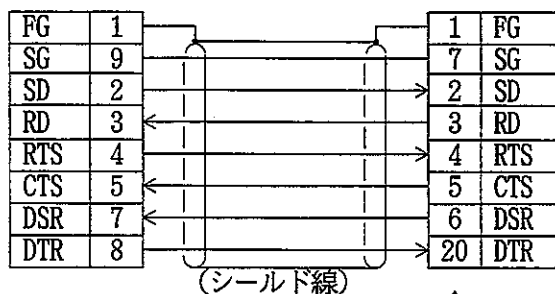


●ハンドヘルドコンピュータFM16π (富士通製) との接続例

ASCIIエニツ側

中継コネクタ

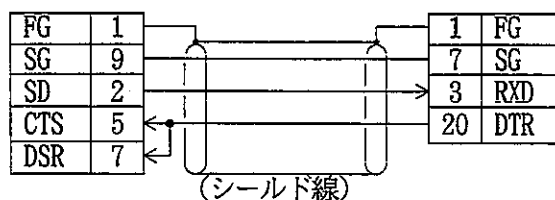
FM-16π側

9P↔25P変換ケーブル (別売)
形C200H-CN223
 ソケット : XM2D-2501 }
 フード : XM2S-2511 } (オムロン製) または同等品
 固定具 : XM2Z-0001 } 2個または同等品

●プリンタとの接続例

ASCIIエニツ側

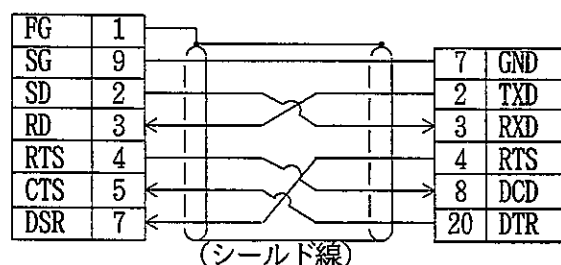
プリンタ側



●プラズマディスプレイとの接続例

ASCIIエニツ側

プラズマディスプレイ側



《参考》

上図はパーソナルコンピュータ、プリンタ、プラズマディスプレイ等の例ですが、周辺機器によっては接続が異なりますので、周辺機器の説明書にしたがって接続を行ってください。

■インターフェース信号のタイミング

電源投入またはリスタート直後、各ポートをOPENにするまでは、ポート1は周辺装置名TERM、ポート2はLPRTの設定状態となっています。

各ポートからの入出力時、RTS、CTS、DTR、DSRの各信号は以下のように扱われます。

(1) ASCII ユニットから周辺機器への送信時

- ・DTRは初期時、ポート1はON、ポート2はOFF状態となっています。
OPEN命令で周辺装置名により以下ようになります。

周辺装置	TERM	SCRN	KEYB	COMU	LPRT
DTRの状態	ON	OFF	ON	ON	OFF

ON : HIGHレベル
OFF : LOWレベル

- ・RTSはOPEN命令で有効信号線が×（通常OFF）の場合OFFとなり、有効信号線が○（常時ON）の場合、OPEN直後からCLOSEされるまでON状態を維持します。
- ・PRINT命令実行時はRTSをONした後、CTSとDSRがともにONしていることを確認してデータを送信します。ONしていない場合、ERROR LEDを点灯し、ONするまで待ちます。データ送信中にCTSがOFFすると出力を中断します。

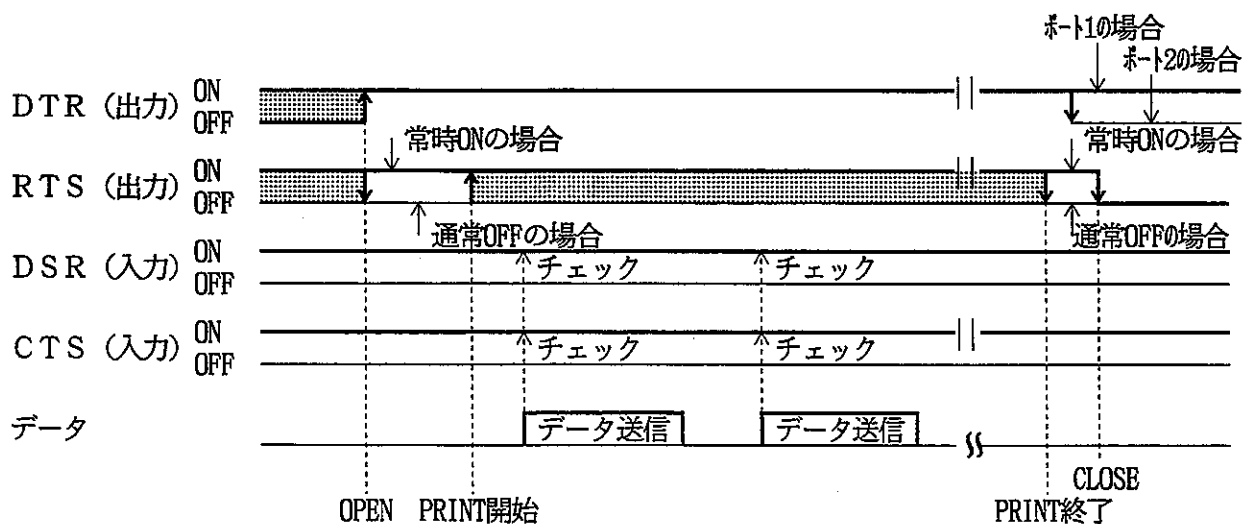
注 DSRまたはCTSが無効指定になっている場合、無効指定になっている信号線は無視します。
ただし、ポート2でCTSを無効にする場合は、CTSの信号をONにするか、RTS信号に接続するか、いずれかにしてください。

- ・CLOSE時、ポート1はTERM、ポート2はLPRTの設定状態へ戻ります。

《OPEN時の周辺装置名がTERM、COMUの場合》

OPEN命令実行時にRTSをOFF、DTRをONとします。

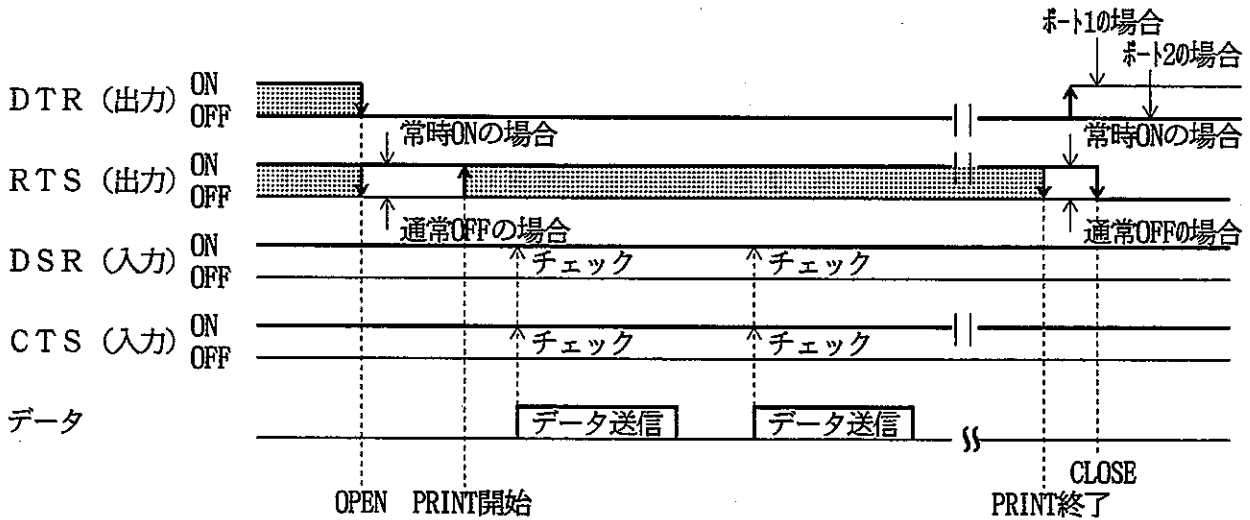
PRINT命令実行時にRTSをONとします。CTSとDSRがともにONならば、データを送信します。



《OPEN時の周辺装置名がSCRN, LPRTの場合》

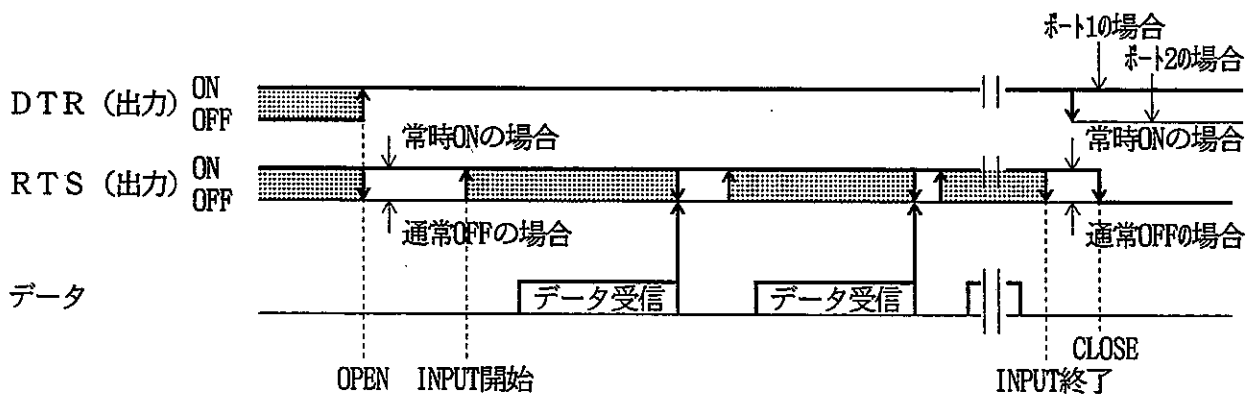
OPEN命令実行時にDTR、RTSをとともOFFとします。

PRINT命令実行時にRTSをONとします。DSR、CTSがともにONならば、データを送信します。



(2) 周辺機器からASCIIユニットへの受信時

- ・DTRはOPEN命令で周辺装置名をKEYBまたはCOMUに設定したときONします。
- ・RTSはOPEN命令で有効信号線が×（通常OFF）の場合OFFとなり、有効信号線が○（常時ON）の場合OPENの直後からCLOSEされるまでON状態を維持します。
- ・RTSが常時ONの場合、INPUT命令の実行に関係なく受信データはバッファに格納されます。
- ・INPUT、INPUT#、INPUT\$命令では、RTSをONし、データがきていれば入力します。（CTS、DSRはチェックしません）
- ・CLOSE時、ポート1はTERM、ポート2はLPRTの設定状態へ戻ります。



- ・周辺機器からの送信は、RTSがONのとき可能になります。

■OPENした周辺装置名による出力の相違

OPEN命令で指定した周辺装置によるPRINT命令などによる出力の違いを示します。

(リセット直後、ポート1はTERM、ポート2はLPRTに自動的に指定されています)

ポート2でSCRNを指定した場合、出力はCOMUと同様になります。

(○: 出力する ◎: 他のコードを付加して出力する ×: 出力しない)

コード 16進 10進 略称			TERM 注(1)		SCRN 注(2)		LPRT 注(3)		COMU	
			ポート1	ポート2	ポート1	ポート2	ポート1	ポート2	ポート1	ポート2
00	0	NUL	×	使用不可	×	○	○	○	○	○
01	1	SH SOM	×		×	○	○	○	○	○
02	2	SX EOA	×		×	○	○	○	○	○
03	3	EX EOM	×		×	○	○	○	○	○
04	4	ET EOT	×		×	○	○	○	○	○
05	5	EQ WRU	×		×	○	○	○	○	○
06	6	AK RU	×		×	○	○	○	○	○
07	7	BL BEL	×		×	○	○	○	○	○
08	8	BS FEO	◎注(4)		◎注(4)	○	○	○	○	○
09	9	HT TAB	×		×	○	○	○	○	○
0A	10	LF LF	×		×	○	○	○	○	○
0B	11	HM VT	○		○	○	○	○	○	○
0C	12	CL FF	○		○	○	○	○	○	○
0D	13	CR CR	◎注(5)		◎注(5)	○	○	○	○	○
0E	14	SO SO	×		×	○	○	○	○	○
0F	15	SI SI	×		×	○	○	○	○	○
10	16	DE DCO	×	使用不可	×	○	○	○	○	○
11	17	D1 XON	×		×	○	○	○	○	○
12	18	D2 TAP	×		×	○	○	○	○	○
13	19	D3 XOF	×		×	○	○	○	○	○
14	20	D4 TAP	×		×	○	○	○	○	○
15	21	NK ERR	×		×	○	○	○	○	○
16	22	SN SYN	×		×	×	○	○	○	○
17	23	EB LEM	×		×	×	○	○	○	○
18	24	CN CAN	×		×	○	○	○	○	○
19	25	EM S1	×		×	○	○	○	○	○
1A	26	SB EOF	×		×	○	○	○	○	○
1B	27	EC ESC	×		×	○	○	○	○	○
1C	28	→ S4	○		○	○	○	○	○	○
1D	29	← S5	○		○	○	○	○	○	○
1E	30	↑ S6	○		○	○	○	○	○	○
1F	31	↓ S7	○		○	○	○	○	○	○

《参考》

- (1) TERMはポート1専用で、ポート2では指定できません。
- (2) SCRNはポート2では&H16 (カーソルON)、&H17 (カーソルOFF) 以外すべてのコードを出力します。
- (3) LPRTは&H0A (LF)、&H0B (HM)、&H0C (CL)、&H0D (CR) のコードがくるか、それ以外のコードが80キャラクタたまると&H0A (LF) を付加して出力します。それまではバッファにためていきます。
CLOSE実行時、このバッファにデータが残っている場合はそれを出力してからポートを閉じます。
- (4) 現在の表示位置に対応したカーソル移動コードを送出して1文字左に移動します。
- (5) &H0C (CR) に&H0A (LF) を付加して出力します。
- (6) COMUは、すべてのコードがバッファに入るごとに出力されます。
- (7) ポート1をTERM以外でOPENした場合、プログラムを停止するときには必ずCLOSEを実行してください。プログラム停止後、ターミナルからキーが受け付けられない場合は、「CTRL」+「X」キーを入力してください。

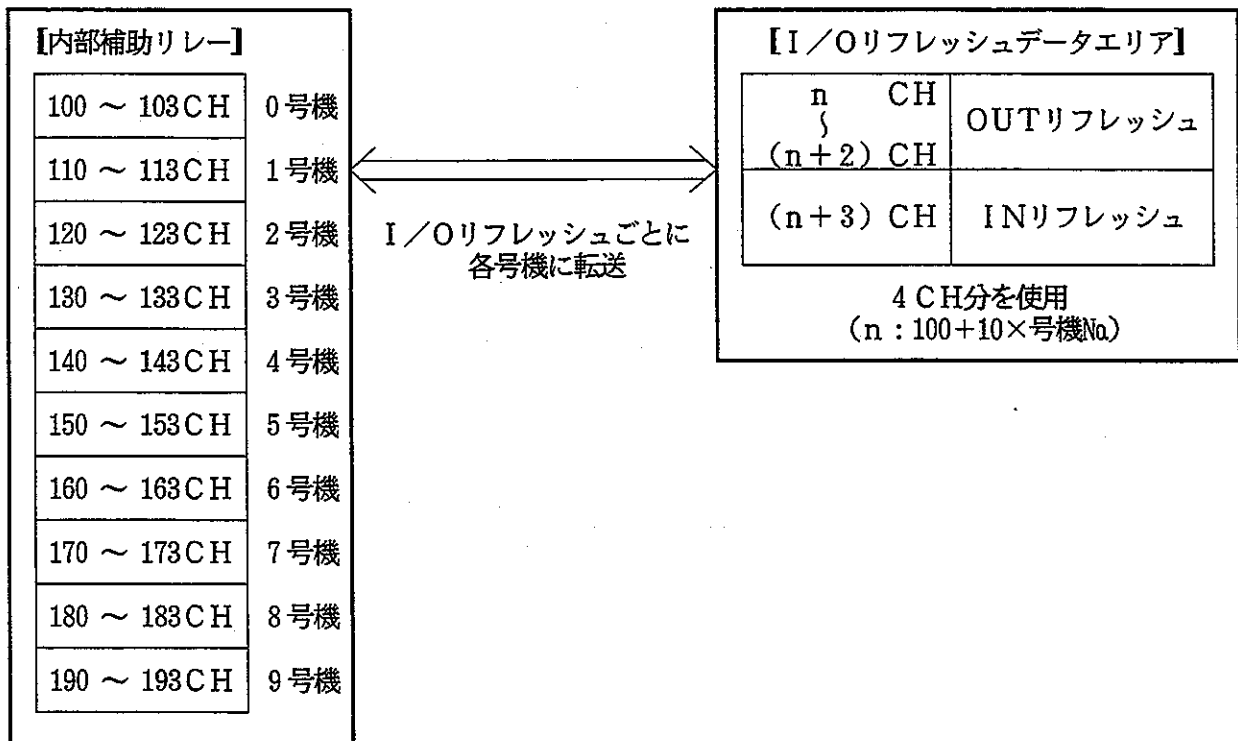
第3章 リレー番号の割付け方法

3-1 チャネルの設定

- ・形C200H-ASC02では、高機能I/Oユニット用内部補助リレーエリアの100～199CHのうち各号機ごとに4CH分ずつをI/Oリフレッシュデータエリアとして占有します。
- ・各高機能I/Oユニットに割付けられている内部補助リレーエリア10CHのうち、形C200H-ASC02で占有する4CH分以外の未使用エリアは、内部補助リレーとして使用できません。

SYSMAC
C200H、C200HS
C200HX/HG/HE

形C200H-ASC02



《参考》

号機Noの設定は、他の高機能I/Oユニットを含めて同一Noを設定すると、「I/Oユニットオーバー」となり動作しません。

3-2 リレー番号の割付け

■内部補助リレーの割付け

入出力 の区別	チャネル 番号	ビット	リレー名称	機 能																													
出力 (n 100 + 10 × 号機 No.)	n	0 0	——	使用していません。																													
		0 1	W R I T E (PC→ASCII)	“1” のときPC READ命令が実行されると転送CH数、転送開始CHに従い、PC→ASCIIへデータが転送されます。 “0” のときPC READ命令は待ち状態になります。 ON立上り(OFF→ON)時にON PC GOSUB命令で使う割込番号を有効とします。																													
		0 2	R E A D (ASCII→PC)	“1” のときPC WRITE命令が実行されると転送CH数、転送開始CHに従い、ASCII→PCへデータが転送されます。 “0” のときPC WRITE命令は待ち状態になります。																													
		0 3	リスタート	OFF立下り(ON→OFF)時にインisialからスタートします。 このフラグがONのときASCIIユニットはリセット状態となります。必ず微分出力としてください。(次ページ注参照)																													
		04-07	割込番号	BASICプログラムでON PC GOSUB命令を使う場合の割込番号となります。 この4ビットを16進数に変換したときの01～15が割込番号となり、00は無視されます。																													
		08-15	出力データ	PC側よりASCII側へ出力されるデータで、PC側のMOV命令などで書込み、BASICプログラムのPC GET命令でASCII側へ読出されます。 注) この出力データは書込みデータとは別に、8ビットデータをASCII側へ転送することにより、ASCII側のプログラム分岐などに使用することができます。																													
	n+1	00-11	転送チャネル数	BASICプログラムでPC READ, PC WRITE命令を使う場合の転送CH数となります。最大255としてください。																													
		12-15	——	使用していません。																													
	n+2	00-12	転送開始 チャネルNo.	BASICプログラムでPC READ, PC WRITE命令を使う場合の転送開始CHとなります。																													
		13-15	リレー種別	BASIC プログラムでPC READ, PC WRITE命令を使う場合の転送データ種別を決めます。																													
				ビットNo.			データの種別	15	14	13	0	0	0	データメモリ (DM)	0	0	1	入出力・内部補助リレー	0	1	0	保持リレー (HR)	0	1	1	補助記憶リレー (AR)	1	0	0	リンクリレー (LR)	1	0	1
	ビットNo.			データの種別																													
	15				14	13																											
	0			0	0	データメモリ (DM)																											
0	0			1	入出力・内部補助リレー																												
0	1			0	保持リレー (HR)																												
0	1	1	補助記憶リレー (AR)																														
1	0	0	リンクリレー (LR)																														
1	0	1	タイマ/カウンタ (TIM/CNT)																														

入出力 の区別	リレー 番号	ビット	リレー名称	機 能
入力 ($n \parallel 100 + 10 \times \text{号機No.}$)	n+3	0 0	ASCIIビジー	ASCIIユニットがPC本体とのデータ転送処理中にONします。
		01-03	——	使用していません。
		0 4	ポート1異常	ポート1が伝送異常（パリティエラーなど）のときONします。
		0 5	ポート2異常	ポート2が伝送異常（パリティエラーなど）のときONします。
		0 6	バッテリー異常	バッテリー電圧が低下したとき、またはバッテリーが未挿入のときONします。
		0 7	BASIC RUN	BASICプログラムがRUN状態のときONします。
		08-15	入力データ	ASCII側よりPC側へ出力されるデータでBASICプログラムのPC PUT命令で書込み、PC側のMOV命令などで読出されます。 注) この入力データは読出しデータとは別に、8ビットデータをPC側へ転送することにより、PC側のプログラム制御に使用することができます。

● 「リスタート」フラグについて

ASCIIユニット前面の「RUN」LEDが消灯時、またはPC本体で高機能I/Oユニット異常発生時には、この「リスタート」フラグは無効となります。

このようなときには、PC本体内の「高機能I/Oユニットリスタートフラグ」AR01CHnビット（nは号機No.）をOFF→ON→OFFとしてリスタートさせてください。

第4章 動作説明

4-1 ASCIIユニットを動作させるにあたって

ASCIIユニットを動作させるにあたって、次の2つのプログラムが必要です。

1. PC側ユーザプログラム
2. ASCII側BASICプログラム

ASCII側BASICプログラムはパーソナルコンピュータ等の入力装置からポート1を通して入力します。

■BASICプログラムの入力にあたって

BASICプログラムの入力はパーソナルコンピュータ等から行います。
下記の機能を持つパーソナルコンピュータ等の装置を準備してください。

RS-232Cを通じてターミナルモードで動作可能なもので、
2・5ページの画面サイズが設定可能なパーソナルコンピュータまたはラップトップパソコンを準備してください。

装置例：FC-987（オムロン製）

PC-9801LV21（NEC製）

形FIT10-CPU01+形FIT10-MF331（ターミナルパック）（オムロン製）

*パーソナルコンピュータ等の設定は

データ8ビット、ストップビット2ビット、パリティチェックなし

全二重、エコーなし、XON/XOFFバッファビジー制御なし、自動LF（ラインフィード）なし

としてください。

また伝送速度はASCIIユニット裏面右側スイッチのポート1伝送速度の設定に合わせて設定してください。

*製品によっては、キーコードが異なったり、必要なキーが一部無かったりするものがありますのでご注意ください。

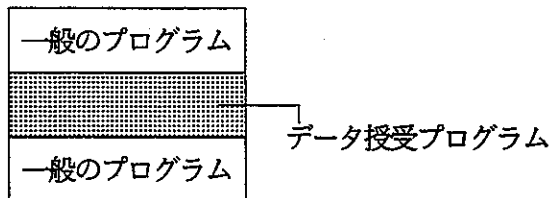
パーソナルコンピュータ等の立上げ方法については、「5-2 BASICプログラムの作成手順」の項に記載していますので参照ください。

4-2 プログラムについて

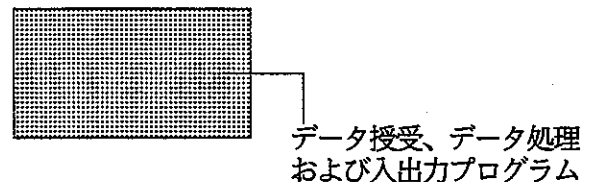
ASCIIユニットを動作させるには、PC側、ASCIIユニット側のそれぞれにプログラムを作成する必要があります。PC側にはユーザプログラムをASCIIユニット側にはBASICプログラムを作成します。

ただし、ASCII側BASICプログラムにてPC READ “@…”、PC WRITE “@…” 命令を使用した場合にはPC側ユーザプログラムは必要ありません。

・ユーザプログラム (PC側)



・BASICプログラム (ASCII側)



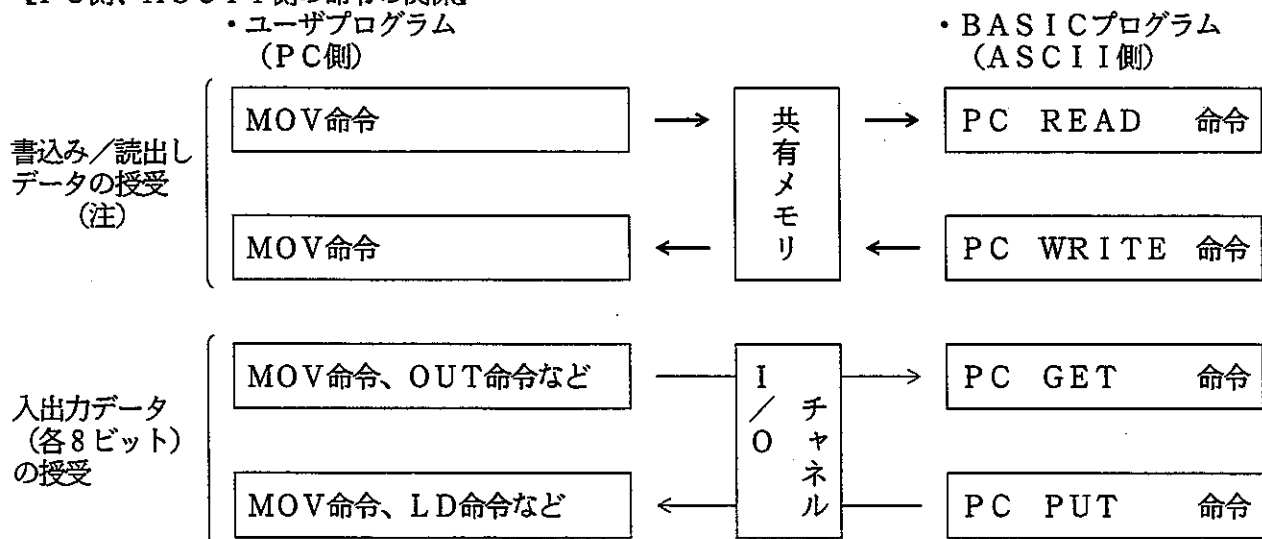
【PC側プログラム】

ユーザプログラムにASCIIユニットとデータを授受するためのプログラムが必要です。MOV命令等を使用してプログラミングします。

【ASCII側プログラム】

BASIC言語でプログラミングします。PCとデータを授受するためには専用のPC READ、PC WRITE、PC GET、PC PUT命令を使用します。データの演算、外部機器への入出力は一般のBASIC命令で行います。

【PC側、ASCII側の命令の関係】



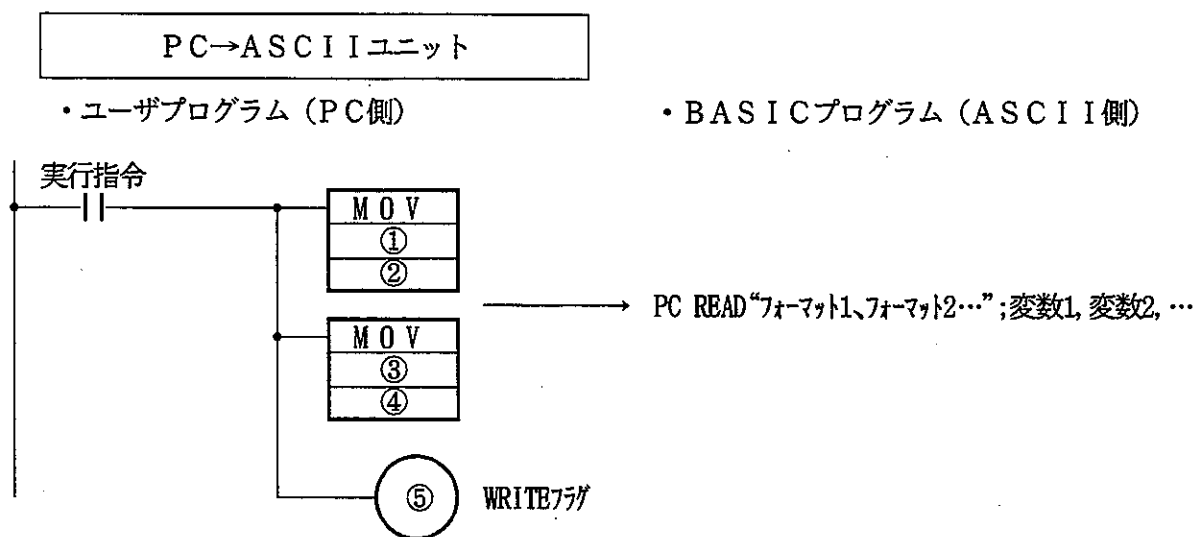
注 PC READ “@…”、PC WRITE “@…” 命令を使用した場合には、PC側のユーザプログラムは必要ありません。

PCの動作状態とASCIIユニットの関係

- ・PCとのデータの授受を行わない場合またはPC READ “@…”、PC WRITE “@…” 命令を用いてデータ授受を行う場合は、PCがプログラムモードでもASCII側のプログラムは動作可能です。(ASCIIユニット単体で動作できます。)
- ・PC側がリセット状態の場合は、ASCIIユニットもリセット状態となります。

4-3 PCとのデータの授受について

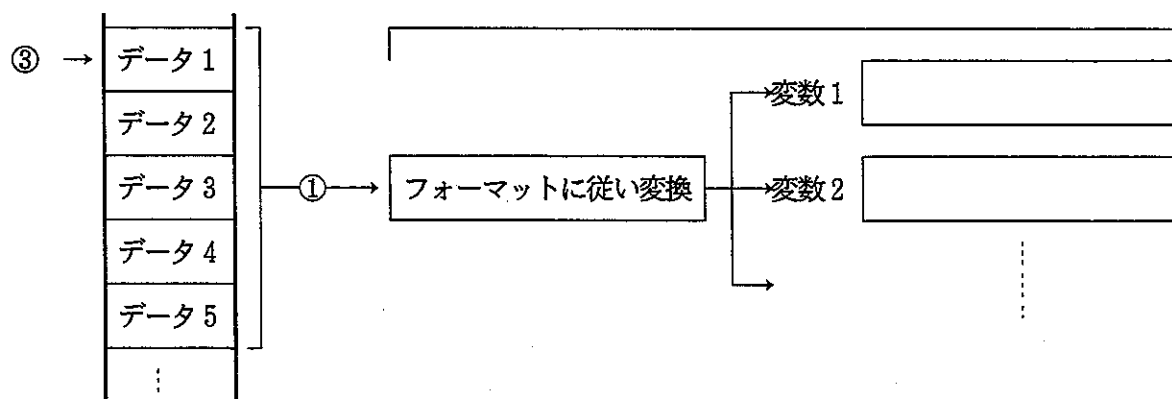
データの流れを中心にPC側、ASCIIユニット側のプログラムの関連を説明します。



- ①: 転送チャンネル数
- ②: (n+1) CH
- ③: 転送元先頭チャンネル番号
- ④: (n+2) CH
- ⑤: WRITEフラグ (nCHの01ビット)
(n=100+10×号機No)

・ユーザプログラム (PC側)

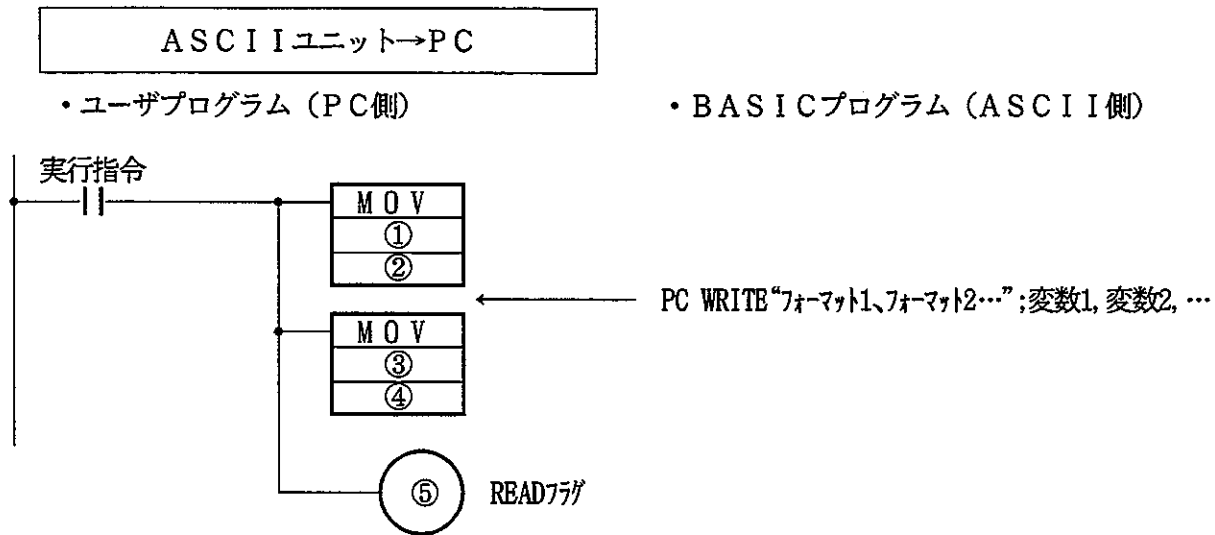
・BASICプログラム (ASCII側)



- ・PCからデータエリアが設定された後、PC READ命令を実行すると、データを読み込みフォーマットに従って変換された後、変数にセットされます。
- ・PC READ命令が実行されるのは、WRITEフラグ (nCHの01ビット) がONしているときです。
- ・WRITEフラグがONしているときにPC READ命令が実行されないと、PC→ASCIIユニットへデータは転送されません。また、WRITEフラグがOFFしているときにPC READ命令を実行すると、ASCIIユニット側では、PC READ命令で待ち状態になります。
- ・ASCII側へ転送できるデータは最大255CH分です。ただし一括して転送できる最大チャンネル数は20CHです。

《参考》

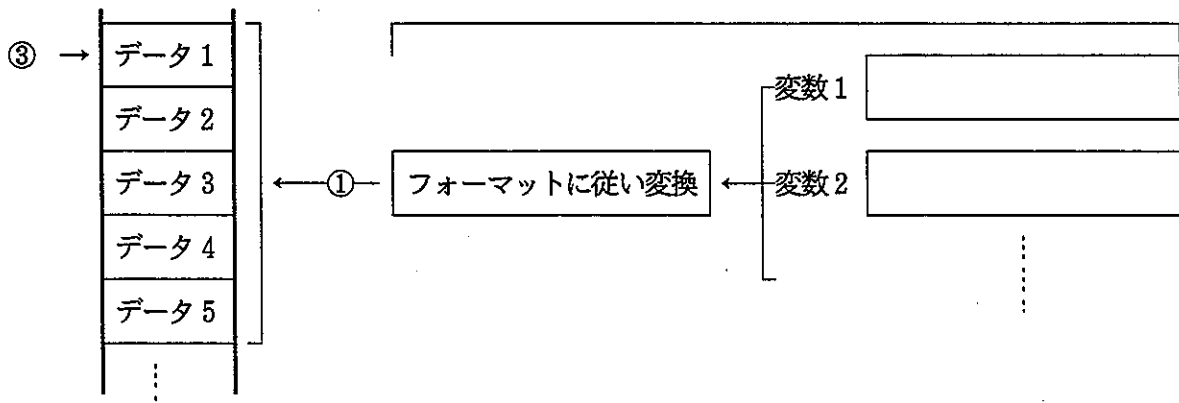
- ・転送チャンネル数はユーザプログラムにより 255CHまで指定可能ですが、データの転送は1スキャンに最大20CHとなっています。従って、データの同時性を要求される場合は (1スキャン内のデータが必要な場合) には20CH以下の指定としてください。
- ・20CHを超えますと、データの転送に数スキャンを要し、同時性が保たれません。
- ・データ転送時間は1CH当り約0.2ms要します。



- ①: 転送チャンネル数
- ②: (n+1) CH
- ③: 転送先先頭チャンネル番号
- ④: (n+2) CH
- ⑤: READフラグ (nCHの02ビット)
(n=100+10×号機No)

・ユーザプログラム (PC側)

・BASICプログラム (ASCII側)



- ・PCからデータエリアが設定された後、PC WRITE命令を実行すると、変数のデータがフォーマットに従って変換され、PC側へ転送されます。
- ・PC WRITE命令が実行されるのは、READフラグ (nCHの02ビット) がONしているときです。
- ・READフラグがONしているときにPC WRITE命令が実行されないと、ASCIIユニット→PCへデータは転送されません。また、READフラグがOFFしているときにPC WRITE命令を実行すると、ASCIIユニット側ではPC WRITE命令で待ち状態になります。
- ・PC側へ転送できるデータは最大255CHです。ただし、一括して転送できる最大チャンネル数は20CHです。

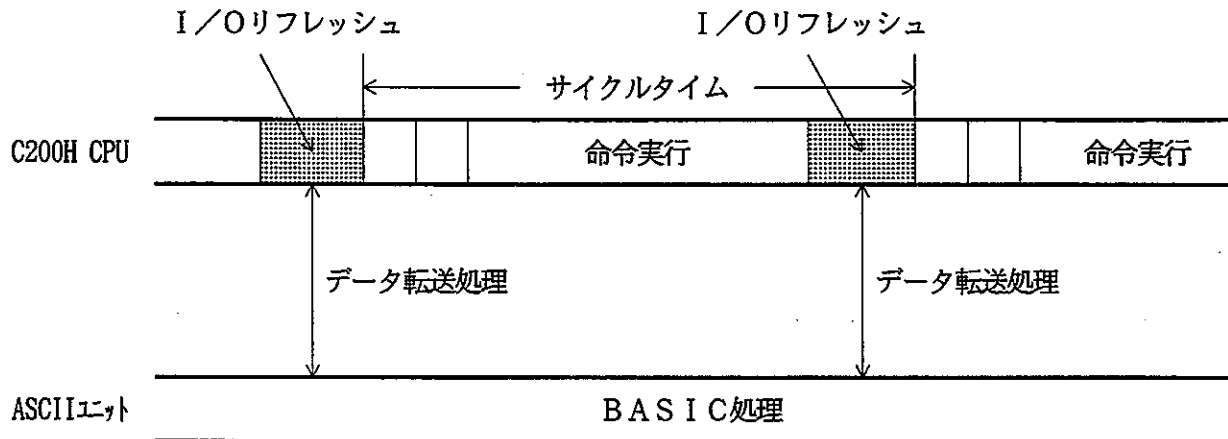
《参考》

- ・転送チャンネル数はユーザプログラムにより 255CHまで指定可能ですが、データの転送は1スキャンに最大20CHとなっています。従って、データの同時性を要求される場合は (1スキャン内のデータが必要な場合) には20CH以下の指定としてください。
- ・20CHを超えますと、データの転送に数スキャンを要し、同時性が保たれません。
- ・データ転送時間は1CH当り約0.2ms要します。

4-4 PC命令実行とリフレッシュタイミングの関係

■各命令とリフレッシュタイミングの関係

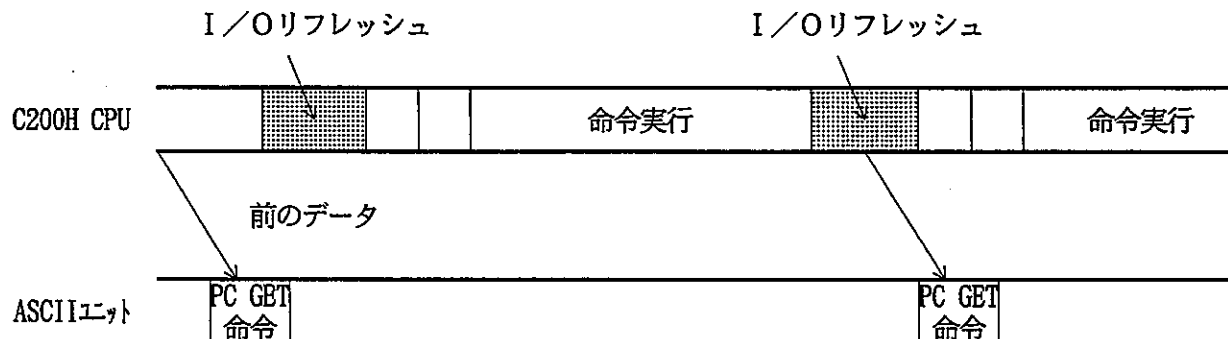
PC本体のI/Oリフレッシュ時にASCIIユニットとPC間のデータ転送が実行されます。



■BASIC命令とPCサイクルタイムの関係

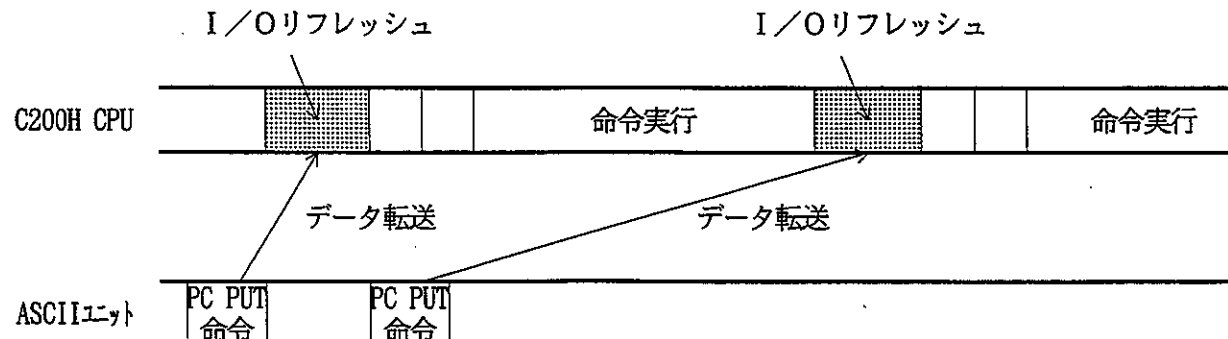
●PC GET命令

PC GET命令では、ASCIIユニットはPC GET命令の実行する以前のPC本体のI/Oリフレッシュで得たデータを取込みます。



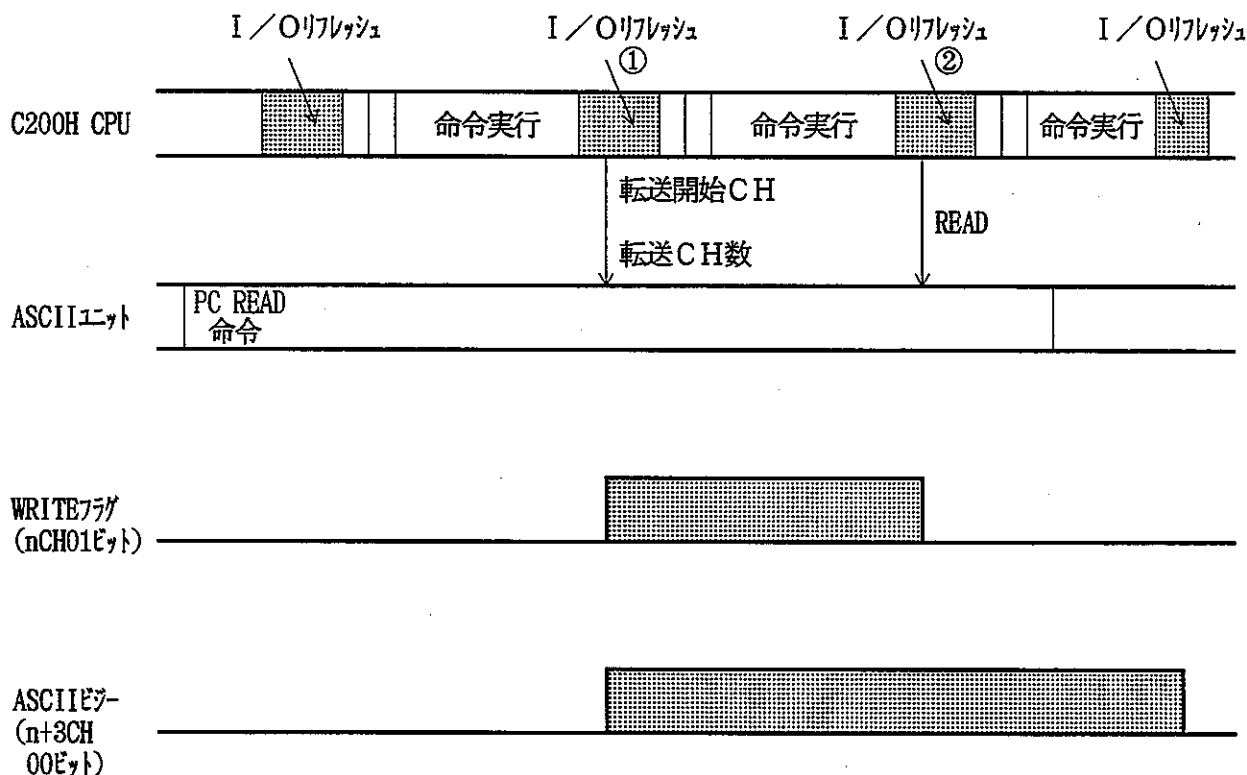
●PC PUT命令

PC PUT命令では、ASCIIユニットはPC PUT命令を実行した後の最初のPC本体のI/Oリフレッシュでデータを転送します。



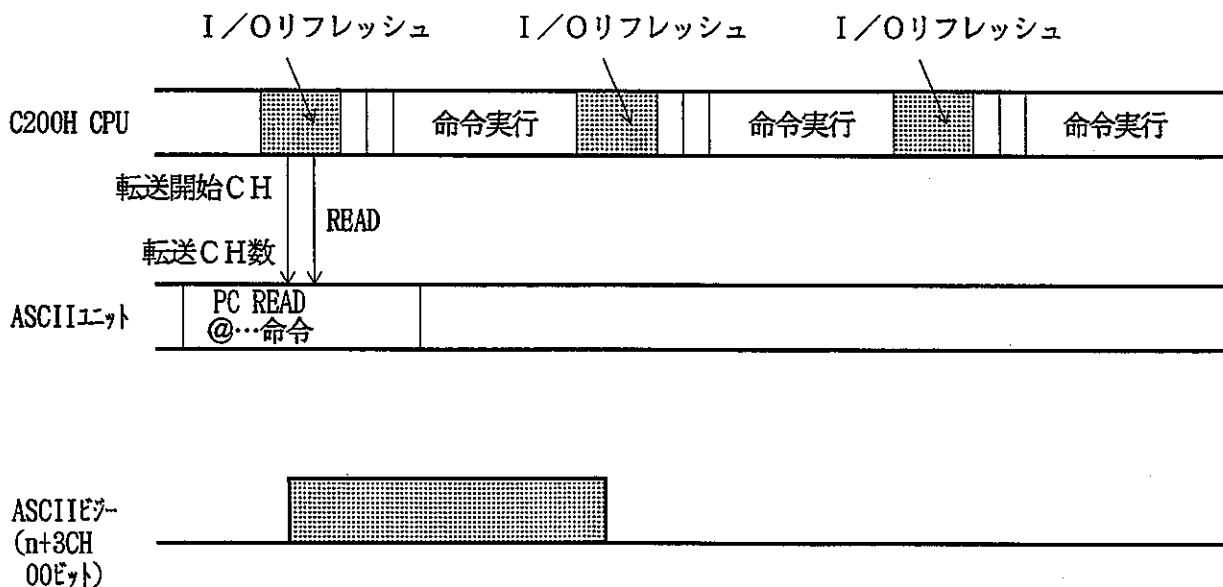
●PC READ命令

PC READ命令は、PC本体のWRITEフラグがONしたとき、そのI/Oリフレッシュ①にて転送開始CH、転送CH数を得て、次のI/Oリフレッシュ②でデータを読み出し（READ）します。



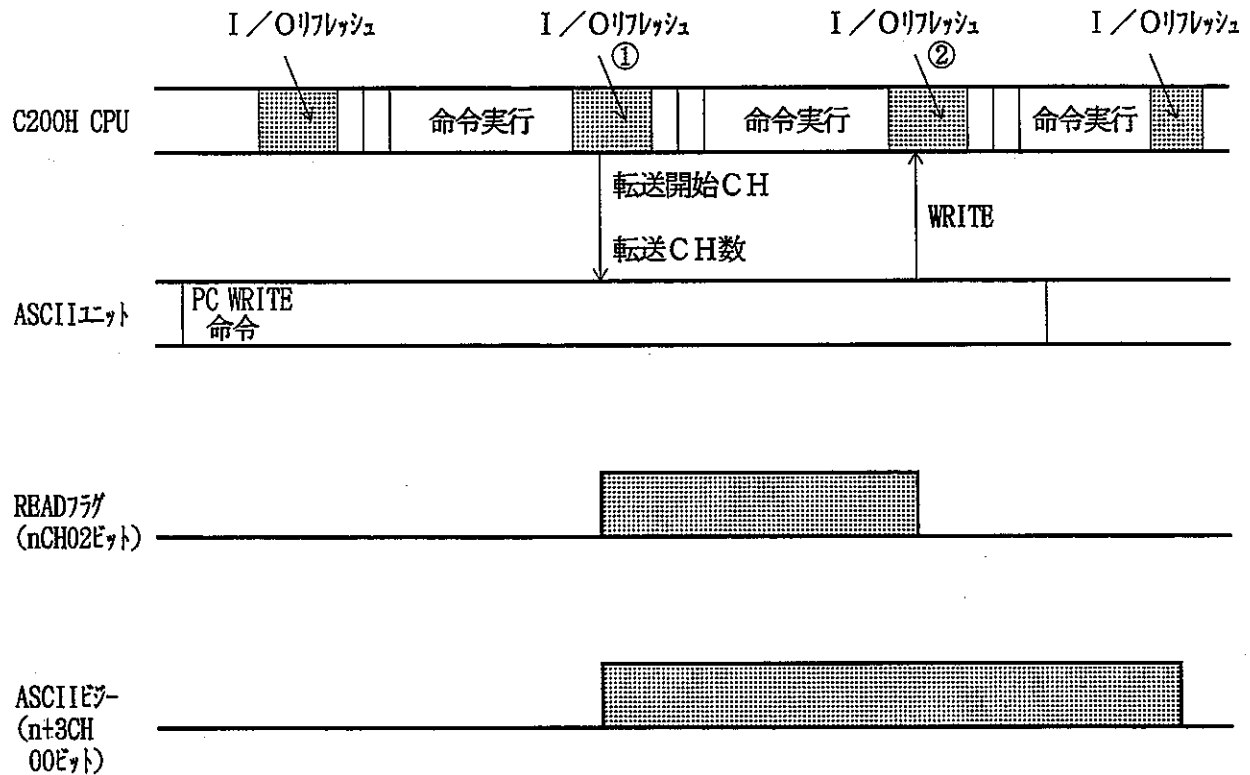
●PC READ @...命令

PC READ @...命令はPC本体のWRITEフラグに関係なく、PC READ @...命令実行時より後の最初のI/Oリフレッシュでデータを読み出し（READ）します。



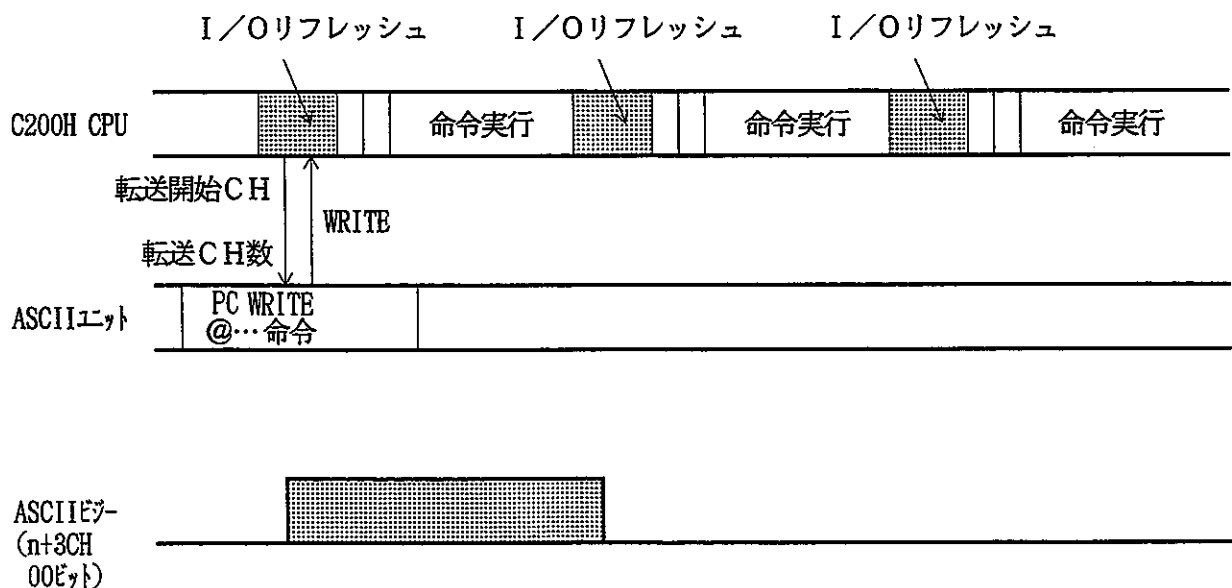
●PC WRITE命令

PC WRITE命令は、PC本体のREADフラグがONしたとき、そのI/Oリフレッシュ①で転送開始CH、転送CH数を得て、次のI/Oリフレッシュ②にてPC本体へデータを転送します。



●PC WRITE @...命令

PC WRITE @...命令はPC本体のREADフラグに関係なく、PC WRITE @...命令実行時より後の最初のI/OリフレッシュでPC本体へデータを転送します。

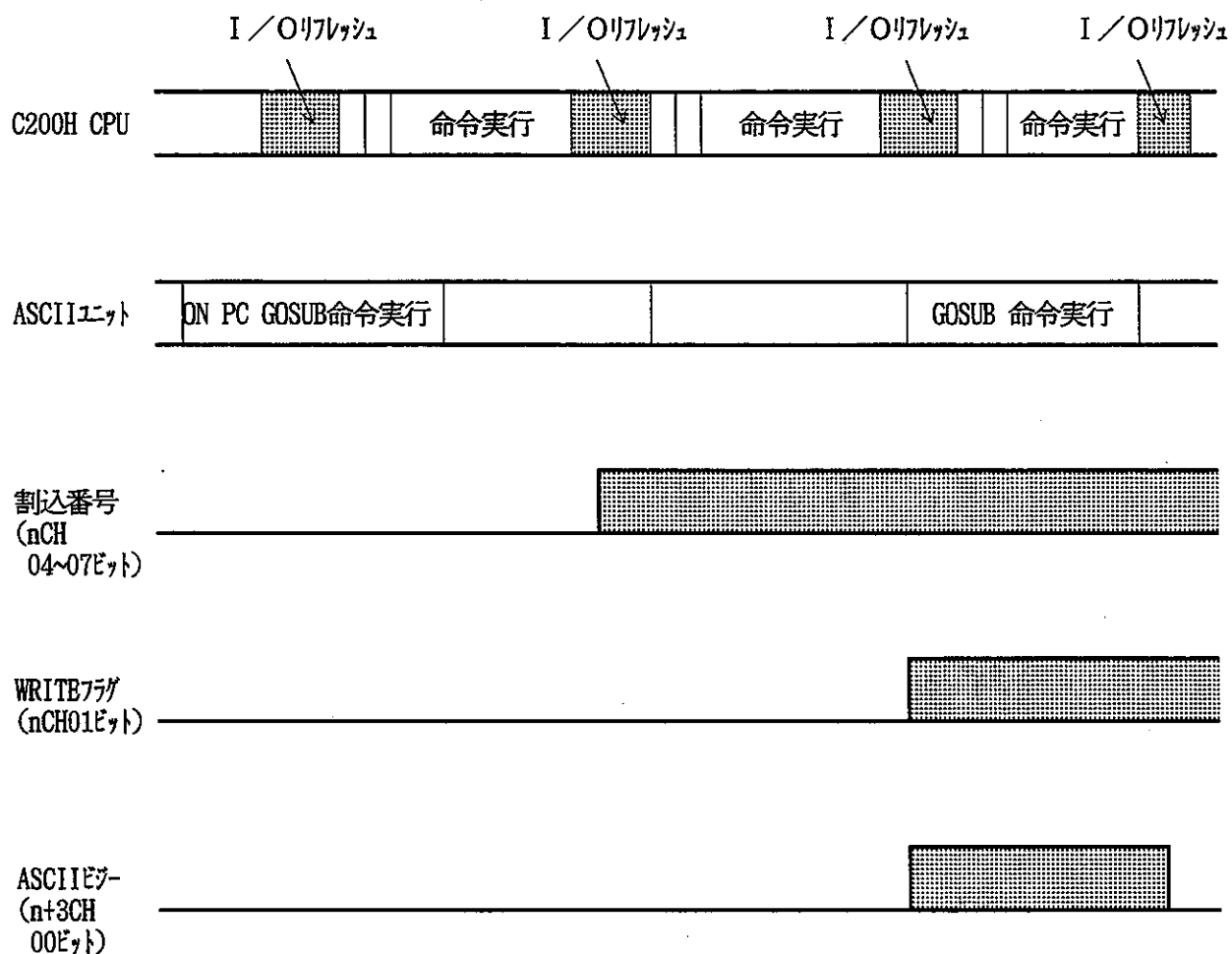


●ON PC GOSUB命令

<PC ON状態のとき>

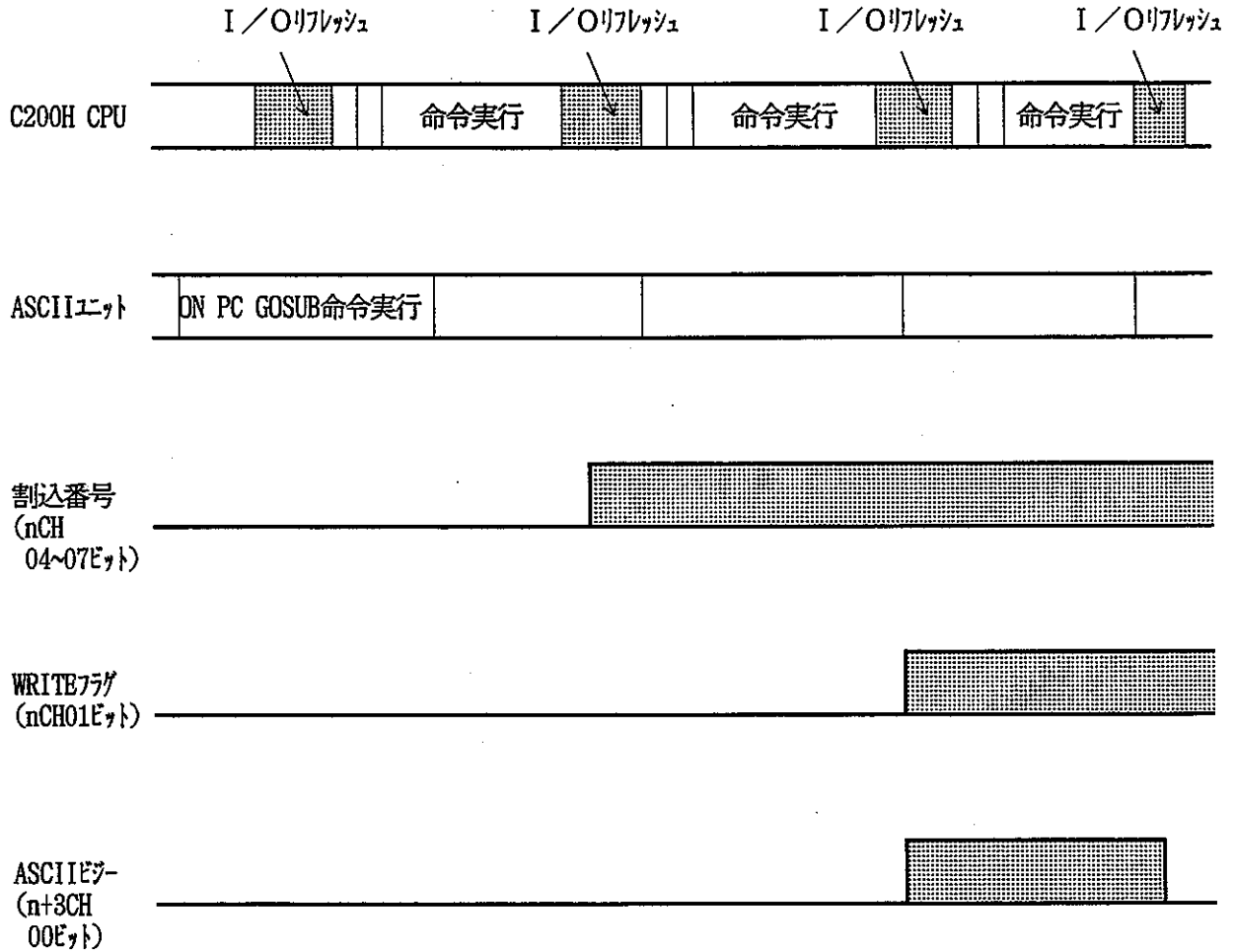
ON PC GOSUB命令を一度実行した後、PC本体の該当割込番号が書込まれ、WRITEフラグが立上ったときに、ASCIIビジーが1スキャンONしGOSUB命令を実行します。

ON PC GOSUB命令は、WRITEフラグの立上り時のみ実行します。

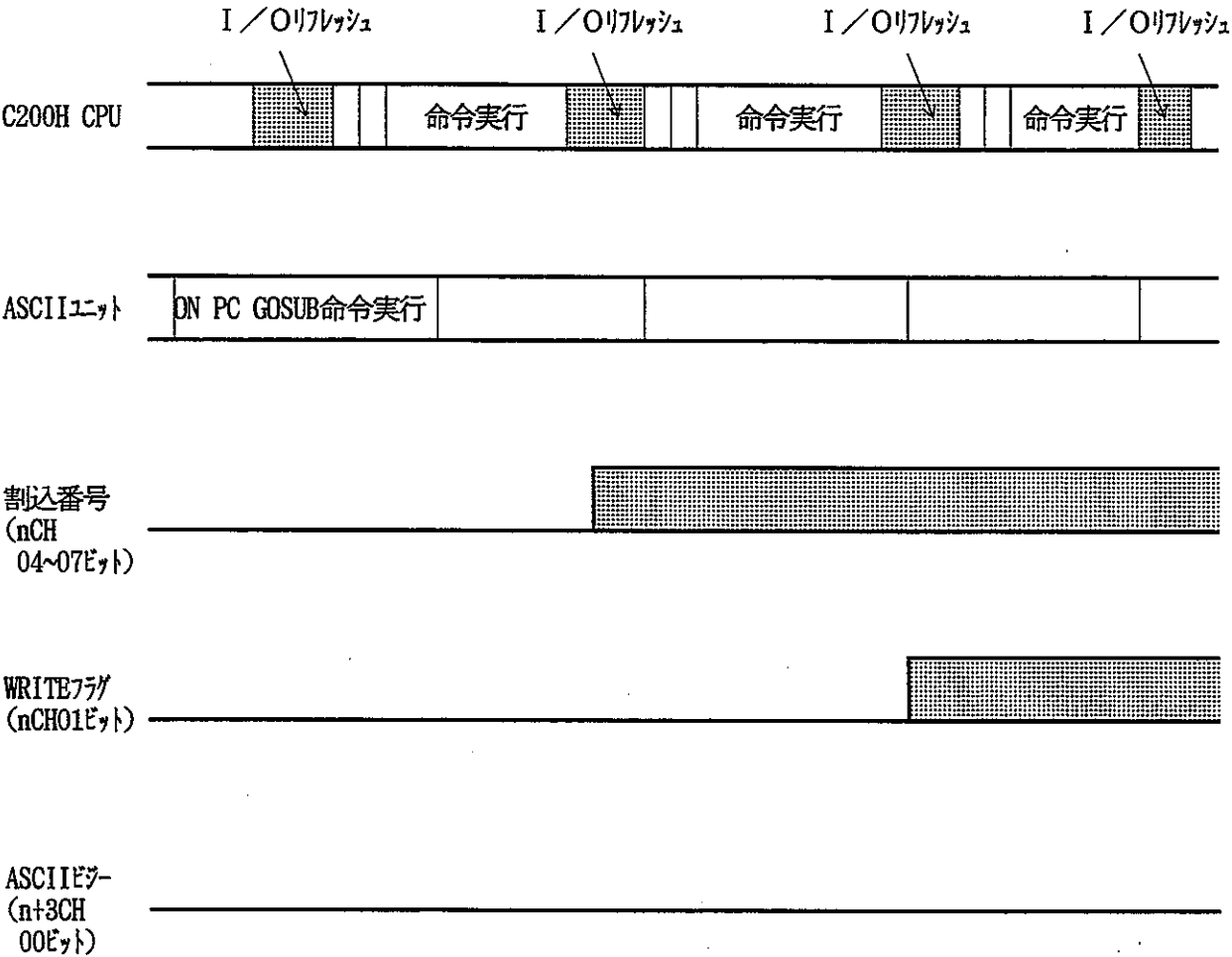


<PC STOP状態のとき>

ON PC GOSUB命令を一度実行した後、PC本体の該当割込番号が書込まれ、WRITEフラグが立上ったときに、ASCIIビジーが1スキャンONしますが、GOSUB命令は実行しません。
PC ON命令が実行されたときGOSUB命令を実行します。



<PC OFF状態のとき>
ON PC GOSUB命令を一度実行した後、PC本体の該当割込番号が書込まれ、WRITEフラグが立上っても、GOSUB命令は実行しません。ASCIIビジーもONしません。



第5章 プログラミング手順

5-1 プログラムの作成について

■PC側のユーザプログラムの作成

- 書き込み/読出しデータは、MOV命令で内部補助リレーエリアに転送チャンネル数、リレー種別、転送開始CHNoをセットし、転送します。
- 入出力データはMOV命令またはOUT命令などで転送します。
- 使用方法は「第6章プログラム例」を参照ください。

■ASCIIユニット側のBASICプログラムの作成

- ASCIIユニット側のBASICプログラムを作成するには、「ターミナル」モードで動作可能なパーソナルコンピュータまたはラップトップパソコン（注）をASCIIユニットのポート1に接続し、キーボードを操作して、1命令ずつASCIIユニット内のRAMに書き込みます。
この時、パーソナルコンピュータなどは、ターミナルとして扱われますので、そのコンピュータ固有のBASICプログラムとは無関係となります。
- または、パーソナルコンピュータ単独であらかじめASCIIユニットのBASICプログラムの文法に従ってプログラムを作成し、パーソナルコンピュータのSAVE“COM”命令でASCIIユニットに転送することもできます。
- RAMに書込んだプログラムは電源OFF時もバッテリーにより保持されます。
- また、RAMの内容をEEP-ROMに転送して保存することもできます。
RAMからEEP-ROMへプログラムを転送するには「SAVE」命令を使用し、EEP-ROMからRAMへプログラムを転送するには「LOAD」命令を使用します。
また、裏面左側スイッチのNo.2をONに設定すると、電源投入時およびリスタート時に自動的にEEP-ROMからRAMへプログラムを転送します。

（注）FC-987（オムロン製）、PC-9801LV21（NEC製）など

Windows付属のターミナルを使用したプログラム作成については巻末の付録を参照してください。

《参考》

- ターミナルから入力されたプログラム、または、ラップトップパソコンなどから転送されたプログラムは、ASCIIユニットのRAMに格納されます。
- EEP-ROMに書き込むには、SAVE命令（7-3 BASIC命令の説明参照）を実行してください。
SAVE命令を実行しなかった場合は、EEP-ROM→RAM 自動転送設定時にプログラムは消滅します。
- ASCIIユニットの故障の際を考慮し、作成されたプログラムはパーソナルコンピュータ側で保存されることをお勧めします。

5-2 BASICプログラムの作成手順

■準備

①入力機器の伝送条件の設定

ポート1へ接続するパソコンの伝送条件を次のように設定します。

項 目	設 定 値
伝送速度	ASCIIユニット側のポート1 伝送速度設定値に合わせて設定してください。(2・6ページ参照)
語長	8ビット
パリティ	な し
ストップビット	2ビット
その他	全二重、エコーなし、XON/XOFFバッファビジー制御なし、自動LF(ラインフィード)なし

②ASCIIユニット側の設定

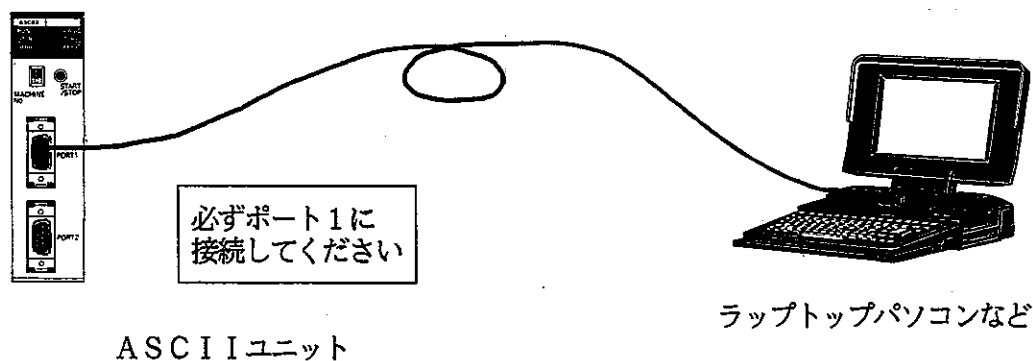
「2-2各部の名称と機能」、「2-3スイッチの設定方法」に従い、表面、裏面スイッチを各々設定してください。

③ASCIIユニットの装着

ASCIIユニットをPC本体へ装着してください。

④ケーブルの接続

パソコン側のPS-232Cコネクタの配線を十分ご確認の上、「2-4インターフェース」で説明したピン配列に合致した接続ケーブルを、ASCIIユニットのポート1に接続してください。



■パーソナルコンピュータの立上げ

FC-987、PC-9801LV21、FM-16πの立上げ例を次に示します。
 操作の詳細な説明、表示については各々のオペレーションマニュアル等をあらかじめ参照ください。
 パーソナルコンピュータは、ターミナルモードで使用します。

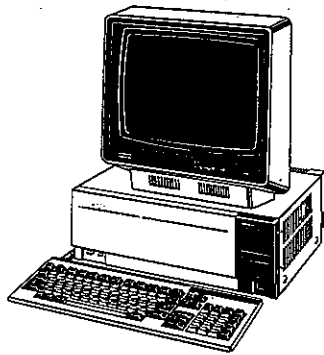
●FC-987、PC-9801LV21の例

・画面は80列×24行です。

①FC-987の電源スイッチをONまたはリセットスイッチを押す

⇒BASICの初期画面が表示される

●BASICモードが立上る



FC-987

②機械語モードにする

●MON「」とキーイン

⇒モニタモードに入り
h)が表示される



PC-9801LV21

③メモリスイッチSW1、SW2を設定する

●h) SSW1「」
 XX-CC「」
 h) SSW2「」
 XX-28「」

とキーインする
 (詳細は次ページ参照)

④BASICモードへ戻る

●「CTRL」+「B」
 キーを押す

⇒OKが表示される

BASICモードに復帰させるには

●「SHIFT」+
 「STOP」をキーインする

⇒OKを表示しBASIC
 モードへ戻る

BASICプログラム
 の作成および起動

⑤ターミナルモードを実行する

●TERM "COM:
 N83NSNCL",
 F「」

《参考》

- ・キー入力例えば「CTRL」+「B」表示は「CTRL」キーを押しながら、「B」キーを押すことを示します。
- ・PC-9801LV21を最初に使用される場合PC-9801LV21本体左側面のディフスイッチSW2の5「メモリ初期化の指定」をONにしてから、ユーティリティプログラム「Switch n88」をロードし、モードを「使う」に設定してシステムを再起動(電源OFF→ONまたはリセットスイッチON)してください。

5-2 BASICプログラムの作成手順

●FC-987、PC-9801LV21メモリスイッチの設定一覧

論理スイッチ名	メモリ番地	データ (ビット位置)								機 能	
		7	6	5	4	3	2	1	0		
SW1	A3FE2								0	Xパラメータ	Xパラメータ無効
									1		Xパラメータ有効
								0	通信方式	全二重	
								1		半二重	
						1	0		データビット長	7ビット長	
						1	1			8ビット長	
					0				パリティチェック	なし	
					1					あり	
				0					パリティ指定	奇数パリティ	
				1						偶数パリティ	
	システム 既定値 (48) ₁₆	0	1						ストップビット 長	1ビット	
		1	0							1.5ビット	
		1	1							2ビット	
		1	1	0	0	1	1	0		0	
		C				C				→この値を入力してください。	

どちらでも可

どちらでも可

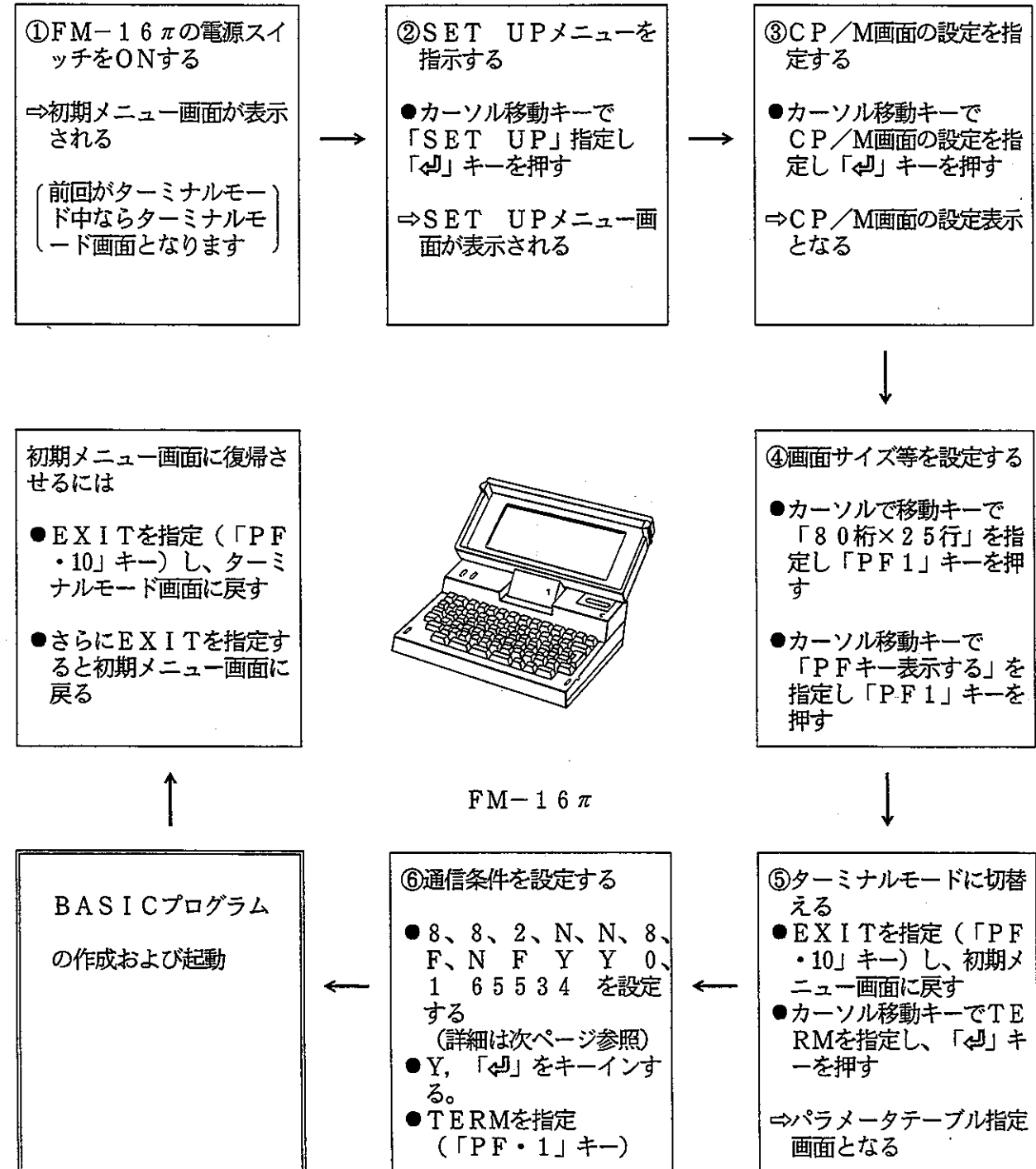
論理スイッチ名	メモリ番地	データ (ビット位置)								機 能	
		7	6	5	4	3	2	1	0		
SW2	A3FE6					0	0	0	0	ボーレート	無 効
						0	0	0	1		75 ボー
						0	0	1	0		150 ボー
						0	0	1	1		300 ボー
						0	1	0	0		600 ボー
						0	1	0	1		1200ボー
						0	1	1	0		2400ボー
						0	1	1	1		4800ボー
						1	0	0	0		9600ボー
					0						日本語 シフト コード
				1					KI = (1A70) ₁₆ KO = (1A71) ₁₆		
				0					C _R /C _R ・ L _F コード 受信時動作	C _R (0D0A) ₁₆ 受信時: 復帰+改行	
				1						(C _R ・L _F (0D0A) ₁₆ 受信時: 復帰+改行 C _R (0D) ₁₆ 受信時:復帰)	
			0						RETURNキー 押下時送信コード	C _R (0D) ₁₆ コード	
			1							C _R ・L _F (0D0A) ₁₆ コード	
	システム 既定値 (05) ₁₆	0								Sパラメータ	無 効
		1									有 効
		0	0	1	0	1	0	0	0		
		2				8					→この値を入力してください。

9600ボーの場合

9600ボーの場合

●FM-16πの例

・画面は80列×25行です



●FM-16 π 通信条件の設定一覧

項 目	設 定 値	設 定
ボーレート (BPS)	1)75 2)150 3)300 4)600 5)1200 6)2400 7)4800 8)9600	8
データビット長 (bit)	7または8	8
ストップビット長 (bit)	1または2	2
パリティ形式	E (偶数パリティ)、O (奇数パリティ)、N (パリティなし)	N
通信ビジー制御	Y (行う)、N (行わない)	N
J I Sコード系 (bit)	7または8	8
全/半二重通信	F (全二重) または H (半二重)	F
オートLF	Y (行う)、N (行わない)	N
オーダ動作系	F (F-BASICオーダ)、A (ADM-3Aオーダ)	F
ダウンロードEOF	Y (行う)、N (行わない)	Y
アップロードEOF	Y (行う)、N (行わない)	Y
CRディレー時間	0~255	0
ブレーク信号時間 (×100ms)	1~30	1
受信バッファサイズ (byte)	256~65534	65534

※ ただし、非漢字モード時は項目が英語で表示されます。

上記の14項目すべてを入力後、「Y」、「」で設定完了します。

■BASICプログラムの作成

初期画面

- ①パーソナルコンピュータ側の立上げおよびPC電源の投入
 パーソナルコンピュータ側の立上げおよびPCの電源を投入してください。
- ②ASCIIユニットの起動
 「CTRL」+「X」キーをキーインしてください。
 初期画面が表示されプログラミングが可能となります。

```
C200H-ASC02 (CF-BASIC) .....
COPYRIGHT OMRON .....
READY
```

- ③ASCIIユニットのRAMのクリア
 BASICプログラムRAMの内容を、NEWコマンドでクリアしてください。

NEW↵

(注) 「↵」は「RETURN」キーを示します。
 部分は1文字ずつキーインすることを示します。

- ④行番号の自動表示
 プログラムの各行に自動的に番号をつけるためにAUTOコマンドを入力してください。
 (10から10ずつ増加する行番号となります)

AUTO↵

- ⑤BASICプログラムの入力
 ・キーボードから1文字ずつ入力してプログラムを作成してください。

例) 10 B=1 2 3 4↵
 20 PRINT B↵
 30 END↵

- ・「↵」までを1行(論理的な)としてRAMに格納します。
- (注) ASCIIユニットは「↵」がキーインされた時点で1行と認識します。
- ・一度作成したプログラムを修正する場合は、EDITコマンドで1行ごとに編集してください。
- ・EEP-ROMに書込むためには、SAVE命令を実行してください。
- ・プログラムの編集機能には以下のものがあります。

項 目	操 作 す る キ ー
文字の入力	7・1ページ文字セット参照
カーソルの移動	上:「↑」、下:「↓」、左:「←」、右「→」
カーソルホーム	「CTRL」+「K」
1行の格納	「↵」または「CTRL」+「M」
1文字の削除	「BS」または「CTRL」+「H」
挿入モード指定/解除	「INS」または「CTRL」+「R」
画面一括消去	「HOME」または「CTRL」+「L」

カーソル位置と実行結果は以下のようになります。

例)

1 2 3 4 5	↑	→Aに変更	1 2 A 4 5	カーソルの文字が変更されます。
		→2を削除	1 3 4 5	カーソルの左側の文字が削除されます。
		→Aを挿入	1 2 A 3 4 5	カーソルの左側に挿入されます。

カーソル

(注) カーソルは現在編集中の行の範囲でのみ移動できます。前後の行へは移動できません。

5-3 BASICプログラムの起動／停止方法

プログラミングの完了したBASICプログラムあるいは停止しているBASICプログラムは、次の3種のいずれかで起動（RUN）させることができます。

これに先立って実行するBASICプログラムのプログラムNoを「プログラムNo選択スイッチ」で設定しておいてください。

また、実行中のBASICプログラムも同様3種の方法のいずれかで停止させることができます。

①パソコンからの指令

ポート1に接続したパソコンからRUNコマンドを入力し、起動させます。

停止させるときは「CTRL」＋「X」キーを入力します。

- ・起動 ----- RUN
- ・停止 ----- 「CTRL」＋「X」

プログラム実行中、エラーが発生するとパソコンの画面に表示されます。

(注) 1 プログラム上でポート1をターミナル以外でOPENさせた状態では、「CTRL」＋「X」キーは受付られません。

2 ポート2からの起動／停止はできません。

3 プログラム内に入出力命令がない場合、または入出力命令があまり使用されていない場合には、「CTRL」＋「X」がきかない場合があります。

このときは、OPEN命令で有効信号線設定をRTS ON指定にしてください。

②START／STOPスイッチによる方法

ASCIIユニットのSTART／STOPスイッチを押すと起動します。

再度押すと停止します。（交互にRUN↔STOPとなります）

③ディップスイッチによる自動起動方法

ASCIIユニット裏面左側ディップスイッチNo.1を「ON」側に設定しておくでPCの電源投入時、またはASCIIユニットのリスタートフラグによるリセット時、自動的に起動します。

プログラム確定後の動作はこの方法で行ってください。

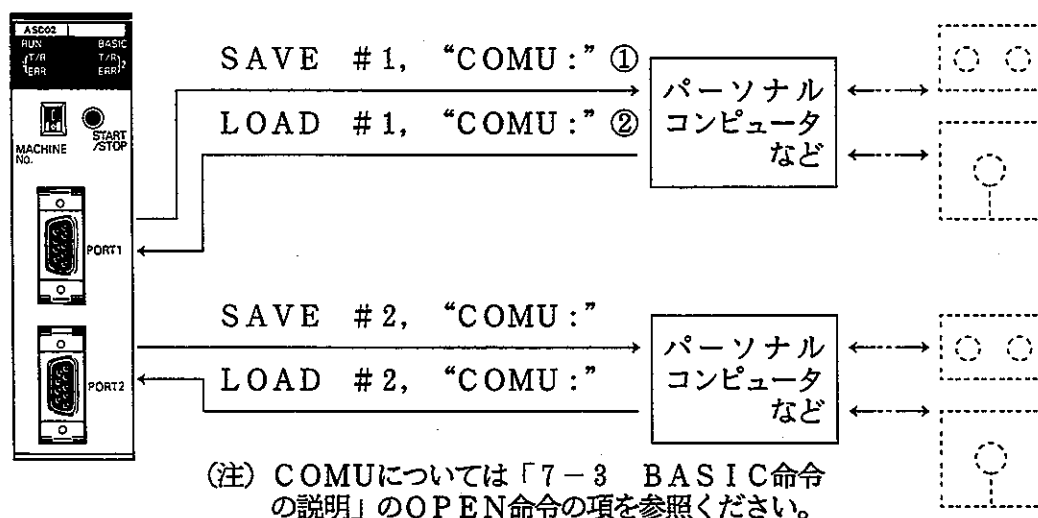
停止させるときはSTART／STOPスイッチを押してください。

《参考》

「CTRL」＋「X」は、「CTRL」キーを押しながら「X」キーを押すことを示します。

5-4 BASICプログラムの転送

通常は、ASCIIユニットに接続したパーソナルコンピュータやラップトップパソコンなどをターミナルとして、直接ASCIIユニット内のBASICプログラムメモリにプログラムを作成しますが、パーソナルコンピュータやラップトップパソコン単独でBASICプログラムを作成し、後でそのプログラムをASCIIユニットに転送したり、逆にASCIIユニット内のプログラムをパーソナルコンピュータやラップトップパソコンに転送することができます。



① パーソナルコンピュータなどへの転送

パーソナルコンピュータなどのRS-232Cポートへデータを送信できる機能

(DOWN LOADまたはLOAD命令)を使って、ASCIIユニットのBASICプログラムをパーソナルコンピュータ側で保存、再生することができます。

転送したBASICプログラムはパーソナルコンピュータ側で1つのファイルとして取扱い、外部記憶機器(カセットテープ、フロッピーディスク)により保存できます。

② パーソナルコンピュータなどからの転送

パーソナルコンピュータなどのRS-232Cポートからデータを受信できる機能

(UP LOADまたはSAVE命令)を使って、パーソナルコンピュータなどに格納しているBASICプログラムをASCIIユニットへ転送することができます。

《参考》

- ・転送時、パーソナルコンピュータ側またはASCIIユニット側の受信バッファの処理速度および容量によりオーバーフローして、データの読取りミスをすることがあります。
このときは、伝送速度を遅く設定するかXON指定にしてください。
- ・「PNAME」でプログラムに名前が設定されていると、プログラムの転送ができません。
- ・「PNAME “ ”」でプログラム名を解除した後、プログラムの転送を実行してください。

■プログラム転送例(1)

FC-987、PC-9801LV21とのプログラム転送例を示します。本文では代表としてFC-987と記して説明します。

(1)FC-987→ASCIIユニット

伝送速度 4,800BPS、データ長8ビット、パリティ無し、2ストップビット、CTS有効、DSR無効の設定とします。

- ① FC-987をBASICモードにします。 (5・3ページの①～④参照)

FC-987でASCIIユニットへ転送するためのBASICプログラムを作成します。

- ② FC-987をターミナルモードにして(5・3ページの⑤参照)「CTRL」+「X」をキーインします。

- ③ LOAD #1, "COMU: (43)" とキーインします。(ASCIIユニットのコマンド)

ASCIIユニットがLOADコマンドを受信すると「BASIC」LEDの点滅が速くなります。

- ④ASCIIユニットの「START/STOP」スイッチを1回押します。

「BASIC」LEDが点灯することを確認してください。この時点でASCIIユニットはLOADコマンドを実行し、ポート1に転送されるデータの受信待ちとなります。

もし、「BASIC」LEDが点灯せず、点滅となった場合は「CTRL」+「X」キーイン後、③の操作から再度やり直してください。

- ⑤ FC-987をBASICモードにします。 (「SHIFT」+「STOP」をキーインする。)

①で作成したBASICプログラムがあることを確認してください。

- ⑥ SAVE "COM:N83" とキーインします。(FC-987のコマンド)

このコマンドが実行されるとFC-987側で作成したBASICプログラムがASCIIユニット側に転送され、「T/R (ポート1)」LEDが転送データに応じて点滅します。

FC-987が「OK」の表示を行うまで待ちます。

- ⑦ASCIIユニットの「START/STOP」スイッチを1回押します。

「BASIC」LEDは通常の点滅となります。

- ⑧ FC-987をターミナルモードにして(5・3ページの⑤参照)「CTRL」+「X」をキーインします。

「READY」と表示されることを確認してください。

- ⑨ LIST とキーインして、ASCIIユニット内にFC-987からのプログラムが転送された

ことを確認してください。

(注) COMUについては、「7-3 BASIC命令の説明」のOPEN命令の項を参照ください。

《参考》

- ・「CTRL」+「X」、「SHIFT」+「STOP」は、「CTRL」キーを押しながら「X」キーを押す、および「SHIFT」キーを押しながら「STOP」キーを押すことを示します。
- ・EEP-ROMに書込むためには、SAVE命令を実行してください。

(2) ASCIIユニット→FC-987の転送

伝送速度 4,800BPS、データ長8ビット、パリティ無し、2ストップビット、CTS有効、DSR無効の設定とします。

- ① FC-987をターミナルモードにして(5-3ページの⑤参照)「CTRL」+「X」をキーインします。
- ② SAVE #1, "COMU: (43)" とキーインします。(ASCIIユニットのコマンド)
ASCIIユニットがSAVEコマンドを受信すると「BASIC」LEDの点滅が速くなります。
- ③ FC-987をBASICモードにします。(「SHIFT」+「STOP」をキーインする。)
- ④ LOAD "COM: N83" とキーインします。(FC-987のコマンド)

⑤ ASCIIユニットの「START/STOP」スイッチを1回押します。

「BASIC」LEDが点灯することを確認してください。この時点でASCIIユニット内のBASICプログラムがFC-987側に転送され、「T/R (ポート1)」LEDが転送データに応じて点滅します。

すべてのプログラムの転送が終了すると、「BASIC」LEDが通常で点滅し、「T/R (ポート1)」LEDが消灯しますので確認してください。

もし、「START/STOP」スイッチを押しても「BASIC」LEDが点灯せず、いきなり通常の点滅となった場合は、①の操作から再度やり直してください。

⑥ FC-987の「STOP」キーを押します。

「DIRECT STATEMENT IN FILE」と表示されることを確認してください。

⑦ LIST とキーインして、FC-987内にASCIIユニットからのプログラムが転送されていることを確認してください。

この後、必要に応じてFC-987のSAVEコマンドでプログラムの保存を行うなどしてください。

《お願い》

ASCIIユニットからFC-987にプログラムを転送するとき、転送プログラムが長い場合は、FC-987側の受信処理が間に合わず、プログラムの一部が抜けることがあります。このような場合は、伝送速度を遅くするか、XON指定をして実行してください。

■プログラム転送例(2)

FM-16πとのプログラム転送例を示します。

(1) FM-16π→ASCIIユニット

伝送速度 4,800BPS、データ長8ビット、パリティ無し、2ストップビット、CTS有効、DSR無効の設定とします。

- ① FM-16πをBASICモードにします。 (5ページの①～⑥を実行後、初期メニュー (F・10) キー)

に戻って「BASIC」を選択する。)

FM-16πでASCIIユニットへ転送するためのBASICプログラムを作成します。

- ② SAVE "A:ASCII.TXT", A とキーインします。(FM-16πのコマンド)

①で作成したBASICプログラムを「ASCII.TXT」というファイル名でASCII形式によりFM-16πのメモリディスクへセーブします。

- ③ FM-16πをターミナルモードにして (初期メニュー (F・10) キー)に戻って「TERM」を選択する)

「CTRL」+「X」 をキーインします。

- ④ LOAD #1, "COMU: (43)" とキーインします。(ASCIIユニットのコマンド)

ASCIIユニットがLOADコマンドを受信すると「BASIC」LEDの点滅が速くなりますので、確認してください。

- ⑤ ASCIIユニットの「START/STOP」スイッチを1回押します。

「BASIC」LEDが点灯することを確認してください。この時点でASCIIユニットはLOADコマンドを実行し、ポート1に転送されるデータの受信待ちとなります。

もし、「BASIC」LEDが点灯せず、点滅となった場合は「CTRL」+「X」キーイン後、④の操作から再度やり直してください。

- ⑥ FM-16πの「PF6」 を押し、 A:ASCII.TXT をキーインします。(FM-16πのコマンド)

このコマンドが実行されると、FM-16πのメモリディスクへ②でセーブしたBASICプログラムがASCIIユニットに転送され、「T/R (ポート1)」LEDが転送データに応じて点滅します。

FM-16πが「Upload End」の表示を行うまで待ちます。

- ⑦ ASCIIユニットの「START/STOP」スイッチを1回押します。

「BASIC」LEDは通常の点滅となります。

- ⑧ 「CTRL」+「X」 をキーイン後、 LIST とキーインし、ASCIIユニット内に

FM-16πからのプログラムが転送されたことを確認してください。

(注) COMUについては、「7-3 BASIC命令の説明」のOPEN命令の項を参照ください。

《参考》

- ・「CTRL」+「X」は、「CTRL」キーを押しながら「X」キーを押すことを示します。
- ・EEP-ROMに書込むためには、SAVE命令を実行してください。

(2) ASCIIユニット→FM-16π

伝送速度 4,800BPS、データ長8ビット、パリティ無し、2ストップビット、CTS有効、DSR無効の設定とします。

- ① FM-16πをキーボードにして(5.5ページの①～⑥を実行する)「CTRL」+「X」をキーインします。

- ② SAVE #1, "COMU:(43)" とキーインします。(ASCIIユニットのコマンド)

ASCIIユニットがSAVEコマンドを受信すると「BASIC」LEDの点滅が速くなります。

- ③ FM-16πの「PF7」を押し、A: ASCII. TXT とキーインします。(FM-16πのコマンド)

- ④ ASCIIユニットの「START/STOP」スイッチを1回押します。

「BASIC」LEDが点灯することを確認してください。この時点でASCIIユニット内のBASICプログラムがFM-16πのメモリディスクに「ASCII. TXT」というファイル名で転送され、「T/R (ポート1)」LEDが転送データに応じて点滅します。

すべてのプログラムの転送が終了すると、「BASIC」LEDが通常で点滅し、「T/R (ポート1)」LEDが消灯しますので確認してください。

もし、「START/STOP」スイッチを押しても「BASIC」LEDが点灯せず、いきなり通常の点滅となった場合は、①の操作から再度やり直してください。

- ⑤ FM-16πでは、転送されたプログラムを保存するかどうかを聞いてきます。

Y とキーイン後、「CTRL」+「X」をキーインします。

「READY」と表示されることを確認してください。

- ⑥ FM-16πをBASICモードにします。(初期に「F・10」に戻って「BASIC」を選択する。)

- ⑦ LOAD "A: ASCII. TXT" とキーインします。(FM-16πのコマンド)

FM-16πが「Ready」の表示を行うまで待ちます。

- ⑧ LIST とキーインして、FM-16π内にASCIIユニットからのプログラムが転送されていることを確認してください。

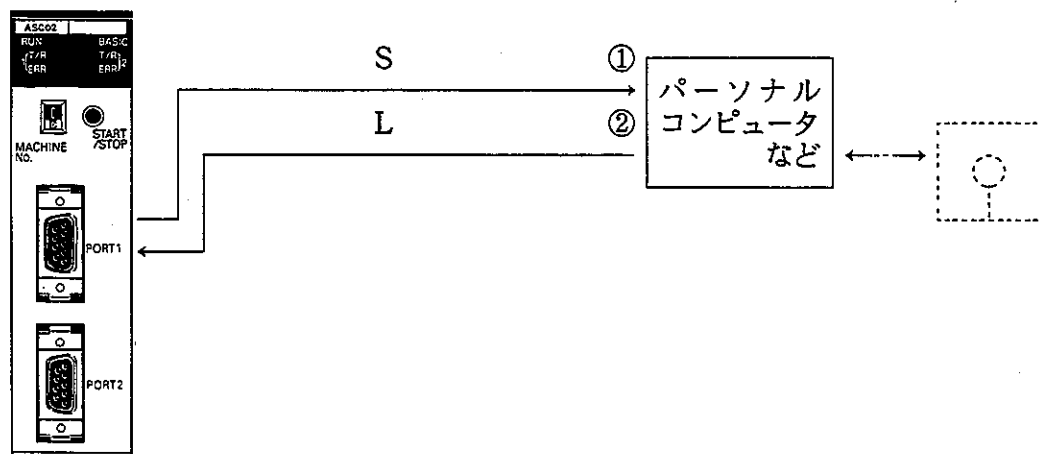
この後、必要に応じてFM-16πのSAVEコマンドでプログラムの保存を行うなどしてください。

《お願い》

ASCIIユニットからFM-16πにプログラムを転送するとき、転送プログラムが長い場合は、FM-16π側の受信処理が間に合わず、プログラムの一部が抜けることがあります。このような場合は、伝送速度を遅くするか、XON指定をして実行してください。

5-5 機械語プログラムの転送

機械語モニタのL、Sコマンドを使用することにより、パーソナルコンピュータやラップトップパソコンなどに機械語プログラムを保存しておくことができます。



① パーソナルコンピュータなどへの転送

(パーソナルコンピュータなどのRS-232Cポートへデータを送信できる機能)

パーソナルコンピュータ側でデータ受信用のプログラム(5・17ページ参照)を使用し、ASCIIユニットの機械語プログラムをデータとして保存することができます。

転送した機械語プログラムは、パーソナルコンピュータ側で1つのファイルとして取扱い外部記憶装置(フロッピーディスク)により保存することができます。

② パーソナルコンピュータなどからの転送

(パーソナルコンピュータなどのRS-232Cポートからデータを受信できる機能)

①において保存しておいた機械語プログラムをASCIIユニットへ転送することができます。

(注) L、Sコマンドは、Sフォーマットで入出力するため、データはSフォーマットのイメージのまま保存されます。また、PROMライタ等のSフォーマットでの送受信による転送機能を使うことにより、PROMライタのRAMエリアに直接データを書込み、読出しすることができます。

Windows 付属のターミナルを使用した機械語プログラムの転送方法については巻末の付録を参照してください。

■プログラム転送例

PC-9801LV21とのプログラム転送例を示します。

転送のためのPC-9801LV21側のプログラムは添付プログラム(5・17ページ参照)を使用します。

(1) ASCIIユニット→PC-9801LV21の転送

伝送速度 4,800BPS、データ長8ビット、パリティ無し、2ストップビット、CTS有効、DSR無効の設定とします。

- ① PC-9801LV21をBASICモードにします。 (5・3ページの①～④参照)

PC-9801LV21でASCIIユニットと機械語プログラムを転送するためのBASICプログラム(添付プログラム)を作成します。

- ② PC-9801LV21をターミナルモードにして(5・3ページの⑤参照)「CTRL」+「X」をキーインします。

- ③ MON↵ S2000.2FFF↵ とキーインします。(ASCIIユニットのコマンド)

Sコマンドに機械語プログラムエリアをつけて実行します。

ここでは、&H2000～&H2FFFが機械語プログラムとします。

- ④ PC-9801LV21をBASICモードにします。 (「SHIFT」+「STOP」をキーインする。)

①で作成したプログラムがあることを確認してください。

- ⑤データフロッピーディスクをドライブ1にセットし RUN↵ とキーインします。

- ⑥「FILE NAME?」と聞いてきますので、ファイルネーム DATA↵ とキーインします。

初めて転送するときのファイルネームは何でもかまいませんが、同一プログラムを次ページ(2)でASCIIユニットに再転送するときは(2)と同じファイルネームを使用してください。

- ⑦「LOAD OR SAVE L/S?」と聞いてきますので S をキーインします。

セーブコマンドを実行しますので、Sとキーインします。リターンキーはキーインする必要はありません。

- ⑧ASCIIユニットの「START/STOP」スイッチを1回押します。

スイッチが押されると転送を開始します。

転送を開始するとASCIIユニットの「T/R (ポート1)」LEDが転送データに応じて点滅します。

すべてのプログラムの転送が終了すると「T/R (ポート1)」LEDが消灯しますので確認してください。

- ⑨ PC-9801LV21の「STOP」キー をキーインします。

「STOP」キーが押されるとPC-9801LV21が転送データをデータフロッピーディスクに書込みます。「OK」の表示がでるまで待ちます。

- ⑩ PC-9801LV21をターミナルモードにして(5・3ページの⑤参照)「CTRL」+「X」をキーインします。

「*」が表示されることを確認してください。

《参考》

- ・FC-987も同様にプログラム転送ができます。

(2) PC-9801LV21→ASCIIユニット

伝送速度 4,800BPS、データ長8ビット、パリティ無し、2ストップビット、CTS有効、DSR無効の設定とします。

- ① PC-9801LV21をBASICモードにします。 (5・3ページの①～④参照)

PC-9801LV21でASCIIユニットと機械語プログラムを転送するためのBASICプログラム (5・17ページ参照) を作成します。

- ② PC-9801LV21をキーボードにして (5・3ページの⑤参照) 「CTRL」+「X」 をキーインします。

- ③ MON↵ L↵ とキーインします。(ASCIIユニットのコマンド)

このとき、機械語エリアが確保されていることを確認してください。

- ④ PC-9801LV21をBASICモードにします。 (「SHIFT」+「STOP」をキーイン。)

①で作成したBASICプログラムがあることを確認してください。

- ⑤データフロッピーディスクをドライブ1にセットし RUN↵ とキーインします。

データが入ったフロッピーディスクがドライブに入っていることを確認してください。

- ⑥「FILE NAME?」と聞いてきますので、(1)で作成したデータのファイルネーム DATA↵ とキーインします。

- ⑦「LOAD OR SAVE L/S?」と聞いてきますので L をキーインします。

ロードコマンドを実行しますのでLとキーインします。リターンキーはキーインする必要はありません。

- ⑧ASCIIユニットの 「START/STOP」スイッチを1回押します。

- ⑨ PC-9801LV21のキーを1つ押します。

キーが押されると転送を開始します。

転送を開始すると、ASCIIユニットの「T/R (ポート1)」LEDが転送データに応じて点滅します。PC-9801LV21が「OK」の表示を行うまで待ちます。

すべてのプログラムの転送が終了すると「T/R (ポート1)」LEDが消灯しますので確認してください。

転送開始までフロッピーディスクからデータを読み出しますので、転送量に応じて時間がかかります。

- ⑩ PC-9801LV21をキーボードにして (5・3ページの⑤参照) 「CTRL」+「X」 をキーインします。

「*」が表示されることを確認してください。

- ⑪ASCIIユニット内にPC-9801LV21に保存していた機械語プログラムが転送されたことを確認してください。

■ 機械語プログラムを転送するためのBASIC (N88 BASIC) プログラム例
(PC-9801LV21側)

```

100 '*****
110 ' * C200H-ASCII エット カイゴ LOAD, SAVE PROGRAM *
120 '*****
130 DIM MDATA$(1000)
140 FIELD#1,250 AS RAFBUF$,2 AS RECOAD$,2 AS NAGASA$
150 INPUT "FILENAME ? ", FILENAME$
160 OPEN FILENAME$ AS #1
170 OPEN "COM:N83NN" AS #2
180 PRINT "LOAD OR SAVE L/S ?"
190 *WAIT0
200 MKEY$=INKEY$
210 IF MKEY$="" THEN *WAIT0
220 IF MKEY$="L" THEN *MLOAD
230 IF MKEY$="S" THEN *MSAVE
240 GOTO *WAIT0
250 '*****
260 ' * LOAD ルーチン *
270 '*****
280 *MLOAD
290 PRINT "LOAD ルーチン START パソコン -> ASCII"
300 PRINT "START/STOP SW ヲ オンテカラ キー ヲ オンテクダサイ"
310 *WAIT3
320 IF INKEY$="" THEN *WAIT3
330 GET #1,1
340 NAGASA$=CVI(NAGASA$)
350 RECOAD$=CVI(RECOAD$)
360 I=1
370 IF RECOAD=1 THEN *ENDRCHK
380 FOR I=1 TO RECOAD-1
390   GET #1,I
400   MDATA$(I)=RAFBUF$
410 NEXT I
420 *ENDRCHK
430 GET #1,I
440 MDATA$(I)=MID$(RAFBUF$,1,NAGASA$)
450 '*****
460 ' * TENSOU MDATA$(I)->FD *
470 '*****
480 *TENSOU
490 FOR I=1 TO RECOAD
500   PRINT #2,MDATA$(I);
510 NEXT I
520 CLOSE #1,#2
530 END

```

次ページに続く

前ページより

```
540 '*****
550 ' * SAVE ルーチン *
560 '*****
570 #MSAVE
580 PRINT "SAVE ルーチン START ASCII -> パソコン"
590 ON STOP GOSUB #MSTOP
600 STOP ON
610 I=1
620 PRINT "START/STOP SW ヲ オシテカダサイ"
630 PRINT "ツウシン ガ シュウリョウ シタラ STOP キー ヲ オシテカダサイ"
640 #WAIT1
650 IF EOF(2) THEN #WAIT1
660 MCOM$=INPUT$(1, #2)
670 IF MCOM$ < > "S" THEN #WAIT1
680 #WAIT2
690 IF LOC(2) < 249 THEN #WAIT2
700 MDATA$=INPUT$(249, #2)
710 MDATA$(1)=MCOM$+MDATA$
720 I=I+1
730 #LOOP
740 IF LOC(2) < 250 THEN #LOOP
750 MDATA$=INPUT$(250, #2)
760 MDATA$(1)=MDATA$
770 I=I+1
780 GOTO #LOOP
790 #*****
800 ' * STOP ルーチン *
810 '*****
820 #MSTOP
830 NAGASA=LOC(2)
840 IF I=1 THEN #FIRST
850 MENDDATA$=INPUT$(NAGASA, #2)
860 #MSTOP1
870 NULLDATA$=STRING$(250-NAGASA, CHR$(0))
880 MDATA$(1)=MENDDATA$+NULLDATA$
890 FOR J=1 TO I
900 LSET RAFBUF$=MDATA$(J):LSET RECOAD$=MKI$(1):LSET NAGASA$=MKI$(NAGASA)
910 PUT #1, J
920 NEXT J
930 STOP OFF
940 CLOSE #1, #2
950 PRINT "NAGASA", NAGASA
960 PRINT "RECOAD", I
970 END
980 #FIRST
990 MENDDATA$=INPUT$(NAGASA, #2)
1000 NAGASA=NAGASA+1
1010 MENDDATA$="S"+MENDDATA$
1020 GOTO #MSTOP1
1030
```

第6章 プログラム例

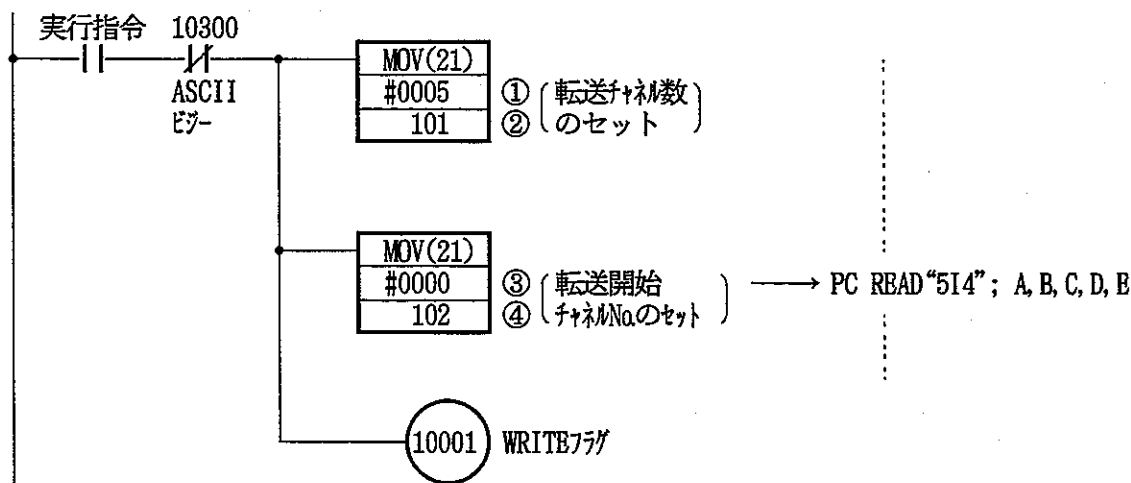
6-1 プログラム基本例

- ・PCとASCIIユニットのデータ授受に関する部分について、いくつかのプログラム例を示します。
- ・なお、以下の例において号機No、PC側データ格納エリアは次の条件とします。
号機No 0号機
PC側データ格納エリア データメモリエリア

1 PCのデータをASCIIユニットへ転送する場合 (PC READ命令)

- (1) PC本体で読出エリアを指定する場合
・ユーザプログラム (PC側)

・BASICプログラム (ASCII側)



- ①転送チャネル数 ② (n+1) CH ③転送元先頭チャネル番号 (DM0000を意味します。) ④ (n+2) CH
(n=100+10×号機No)

説明

- ・実行指令フラグがONすると、MOV命令によって転送チャネル数5チャネル、転送開始チャネルDM0000をセットし、WRITEフラグ (nCHの01ビット) をONさせます。
- ・ASCII側でPC READ命令を実行させると、DM0000から5チャネル分のデータを順に10進数4桁に変換して変数A～Eにセットします。(PC READ命令はPC側でWRITEフラグがONされるまで命令実行しません。)
- ・PC READ命令実行中はASCIIビジーフラグ (n+3CHの00ビット) がONし、処理が終了すると、ビジーフラグがOFFします。

(注) PC READ命令の詳細については7・30ページをご覧ください。

《参考》

“5 I 4” は5チャネル分の10進4桁データを示します。
詳細については7・33ページをご覧ください。

(2) ASCIIユニットで読出エリアを指定する場合

・ユーザプログラム (PC側)

・BASICプログラム (ASCII側)

この例ではPC側ユーザプログラムは必要ありません

PC READ "@D, 0, 5, 5I4"; A, B, C, D, E

説明

ASCII側でPC READ "@..." 命令を実行させると、DM0000から5チャンネル分のデータを10進数4桁に変換して変数A~Eにセットします。

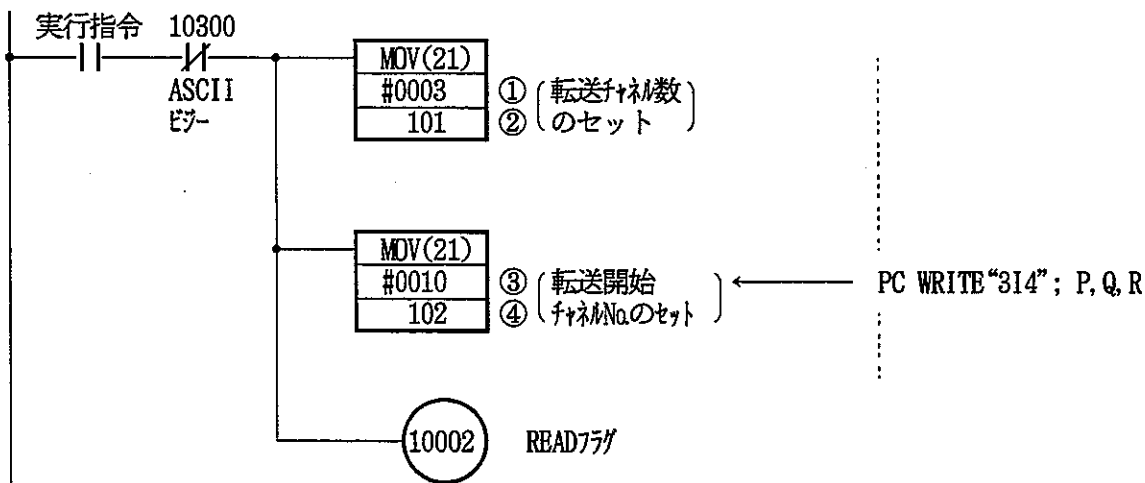
PC READ "@..." 命令実行中は、ビジーフラグ((n+3) CHの00ビット) がONします。

2 ASCIIユニットのデータをPCへ転送する場合 (PC WRITE命令)

(1) PC本体で書込エリアを指定する場合

・ユーザプログラム (PC側)

・BASICプログラム (ASCII側)



- ① 転送チャンネル数 ② (n+1) CH ③ 転送先先頭チャンネル番号 ④ (n+2) CH
(n=100+10×号機No.) (DM0010を意味します。)

説明

- ・実行指令フラグがONすると、MOV命令によって転送チャンネル数3チャンネル、転送開始チャンネルDM0010をセットし、READフラグ (nCHの02ビット) をONさせます。
- ・ASCII側でPC WRITE命令を実行させると、変数P, Q, Rの値を10進4桁に変換してDM0010~DM0012に格納します。
- ・PC WRITE命令実行中はASCIIビジーフラグ (n+3 CHの00ビット) がONし処理が終了すると、ビジーフラグがOFFします。
- ・PC WRITE命令はPC側でREADフラグがONされるまで命令実行しません。

(注) PC WRITE命令の詳細については7・30ページをご覧ください。

《参考》

"3I4" は3チャンネル分の10進4桁データを示します。詳細については7・33ページをご覧ください。

(2) ASCII ユニットで書込エリアを指定する場合

・ユーザプログラム (PC側)

・BASICプログラム (ASCII側)

この例ではPC側ユーザプログラム
は必要ありません

PC WRITE "@D,10,3,3I4";P,Q,R

説明

ASCII側でPC WRITE "@..." 命令が実行されると、変数P, Q, Rの値を10進4桁
に変換してDM0010~DM0012に格納します。

PC WRITE "@..." 命令実行中は、ビジーフラグ (n+3CHの00ビット) がONしま
す。

3 一定時間毎にプリンタにデータを印字する場合 (LPRINT命令)

・ユーザプログラム (PC側)

・BASICプログラム (ASCII側)

この例ではPC側ユーザプログラム
は無関係です

```

100 TH$=MID$(TIME$, 1, 2)
110 IF TH$=TH0$ GOTO 200
120 TH0$=TH$
130 LPRINT TIME$, A

200

```

説明

- ・1時間毎 (××時00分) に時刻とデータを印字する例です。
- ・内部の時計をスキャンしてチェックし時間部分が変化したとき。
(例えば9:××から10:00になったとき) にPRINTを実行します。

・時計は24時間制になっています。

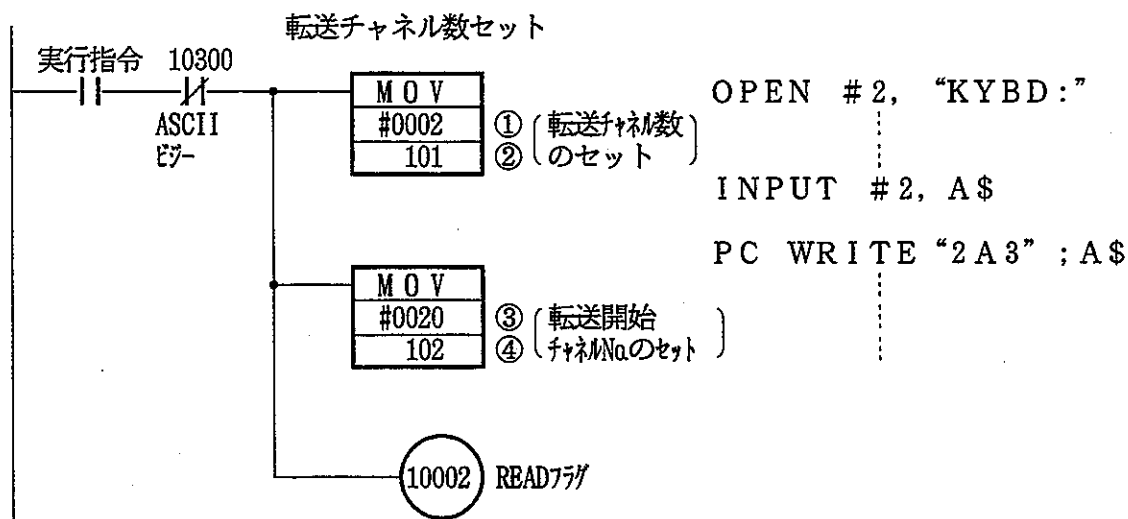
プログラムの実行に先立ち、時刻の設定を行ってください。(7・58ページ参照)

4 キーボードから入力したデータをPCへ転送する場合 (INPUT命令)

(1) PC本体で書込エリアを指定する場合

・ユーザプログラム (PC側)

・BASICプログラム (ASCII側)



①転送チャンネル数

② (n+1) CH

③転送元先頭チャンネル番号

④ (n+2) CH

(DM0020を意味します。)

(n=100+10×号機No)

説明

- ・ポート2に接続したキーボード(ターミナル)からデータを入力し、そのデータをPCへ書込みます。(データは2 CH, 4文字(2文字/CH×2 CH)に変換)
- ・実行指令フラグがONすると、データがDM0020、0021へ格納されます。
- ・ASCII側ではポート2をキーボードとしてOPENし、文字変数A\$に、CRが押されるまでのキーボードからの入力を格納します。
これをPC WRITE命令によって、キーボードからの入力のうちASCIIコードで最初から4文字分をDM0020以降に格納します。(PC WRITE命令の実行は、READフラグ(nCHの02ビット)がONするまで行われません。)
- ・実行指令がONすると、MOV命令によって転送チャンネル数2チャンネル、転送開始チャンネルDM0020をセットし、READフラグをONさせます。

《参考》

“2A3”は2チャンネル分の上位桁、下位桁のASCIIコードデータを示します。詳細については7・35ページを参照ください。

(2) ASCIIユニットで書込エリアを指定する場合

・ユーザプログラム (PC側)

この例ではPC側ユーザプログラム
はありません

・BASICプログラム (ASCII側)

```
OPEN #2, "KYBD:"
```

```
...
```

```
INPUT #2, A$
```

```
PC WRITE "@D, 20, 2, 2A3"; A$
```

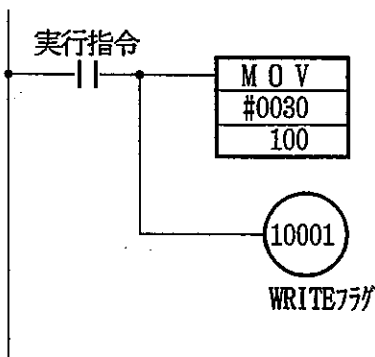
説明

- ・PC WRITE "@..." 命令が実行されると、文字変数A\$の最初から4文字分がASCIIコードに変換されて、DM0020~0021に格納します。
- ・PC WRITE "@..." 命令実行中はビジーフラグ (n+3CHの00ビット) がONします。

5 PCからASCIIユニットのプログラム実行を制御する場合 (ON PC命令)

・ユーザプログラム (PC側)

・BASICプログラム (ASCII側)



```
50  ON PC 3  GOSUB 200
60  ON PC 4  GOSUB 300
70  PC  ON
```

```
200  A=1234:RETURN
```

```
300  A=2345:RETURN
```

説明

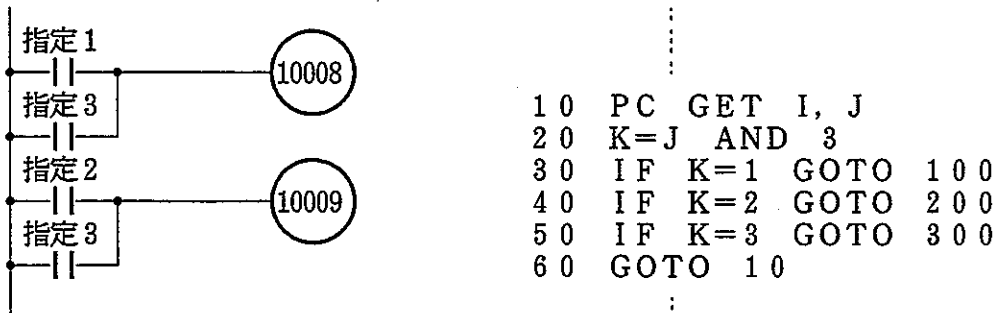
- ・PC側のプログラムによりASCII側のプログラムを割込みにより制御するものです。
- ・ASCIIユニットがON PC GOSUB命令を実行し、PC ON命令で割込みが許可された後、実行指令がONし、PC本体側で割込番号が書込まれます。WRITEフラグ (nCHの01ビット) が立上ると、ASCIIユニット側が割込番号に応じた処理を行います。上記の例では割込番号が3ですから、サブルーチン200行目が実行されます。
- ・ON PC GOSUB命令はWRITEフラグの立上り時のみ実行します。

(注) ON PC GOSUB命令については7・28ページを参照ください。

6 PCからのASCIIユニットのプログラムを分岐させる場合 (PC GET命令)

n CHの08～15ビットのデータをASCII側へ転送することにより、ASCII側のプログラム制御に使用することができます。

- ・ ユーザプログラム (PC側)
- ・ BASICプログラム (ASCII側)



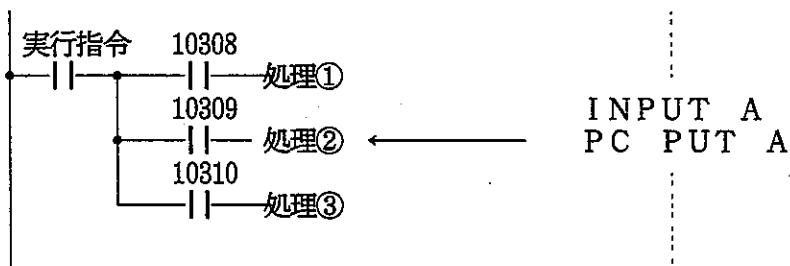
説明

- ・ 出力データ (10008～10015) をセットし、それをASCII側で監視しておくことにより出力データの値に応じてASCII側のプログラムを分岐させます。
- ・ 出力データのセットはMOV命令で行ってもかまいません。
- ・ 上の例では指定1がONの場合、10008がONとなります。
ASCII側では、PC GET文で10008～10015の状態を変数Jに読取っています。10008がONするとJ=1となり、100行目に分岐します。
K=J AND 3は、10010～10015がONしてもこれを無視するためです。
- ・ 指定2、3も上記と同じ処理となります。

7 ASCIIユニットからのPCのプログラムを制御する場合 (PC PUT命令)

n+3 CHの08～15ビットの各データをPC側へ転送することにより、PC側のプログラム制御に使用することができます。

- ・ ユーザプログラム (PC側)
- ・ BASICプログラム (ASCII側)



説明

- ・ ASCIIユニットに入力されたデータの値をPCへ転送し、その値に応じてPC側で実行する処理を選択します。
- ・ ASCII側では、INPUT文でデータを入力し、PC PUT文でそのデータをPCへ転送します。
- ・ 1を入力するとPCの接点10308がONし、処理①は実行可となります。(他の処理も同様です。)

6-2 プログラム実施例

ここでは、プログラムの実際の使用例についてご説明します。

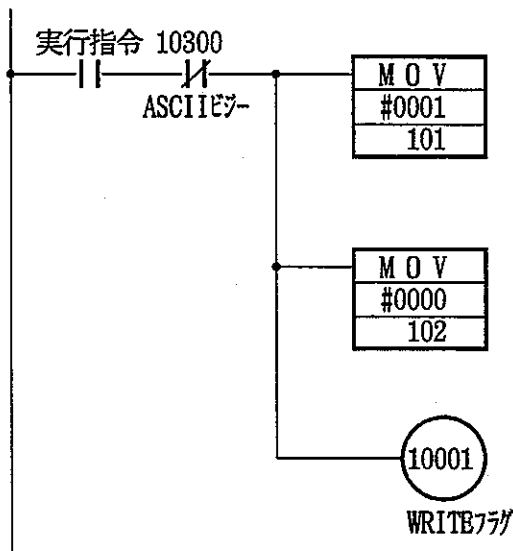
例No.	項 目	ページ
例1.	PC側の1種類のデータを印字する例 (PC READ命令の使用例)	6・8
例2.	ターミナルからのデータをPCに取込む例 (PC WRITE命令の使用例)	6・9
例3.	PC側の複数種類のデータを切替え印字する例 (PC GET命令の使用例)	6・10
例4.	複数種類のデータを異なる起動入力で印字する例 (ON PC命令の使用例)	6・11
例5.	PC側データ・時刻の印字例 (PRINT命令の使用例)	6・12
例6.	表示器への表示例 (PRINT命令の使用例)	6・13
例7.	バーコードリーダによる入力例 (PC WRITE命令の使用例)	6・13
例8.	ASCIIユニット側が主導権をもって、PC→ASCIIのデータ転送を行う例	6・14
例9.	ASCIIユニット側が主導権をもって、ASCII→PCのデータ転送を行う例	6・15
例10.	PC側が主導権をもって、PC→ASCIIのデータ転送を行う例	6・16
例11.	PC側が主導権をもって、ASCII→PCのデータ転送を行う例	6・18
例12.	ASCIIユニット側でデータ処理を行う例	6・20
例13.	PC側でデータ処理を行う例	6・21
例14.	パソコンなどの入力機器からの入力例 (ON COM命令の使用例)	6・22
例15.	起動スイッチによりデータを転送する例 (WAIT命令の使用例)	6・22
例16.	割込みで処理を分ける例	6・23
例17.	機械語プログラムの使用例(1) 文字の分類	6・24
例18.	機械語プログラムの使用例(2) 複数の変数使用	6・27
例19.	機械語プログラムの使用例(3) FCSの計算	6・29
例20.	PC側が主導権をもって、PC→ASCIIのデータ転送を行う例 (リモート子局上にASCIIユニットがある場合)	6・31

例1. PC側の1種類のデータを印字する例(PC READ命令の使用例)

- ・特定時刻毎に生産量(DM0)を印字する。
- ・プリンタ:ポート2に接続、ボーレート:4800BPS

●PC READ命令を使用した場合

- ・ユーザプログラム(PC側)



・BASICプログラム(ASCII側)

```

10 OPEN #2, "LPRT:(47)"
20 A$="00:00":B$=""
30 C$=MID$(TIME$, 4, 5)
40 IF C$<>A$ GOTO 30
50 D$=LEFT$(TIME$, 2)
60 IF D$=B$ GOTO 30
70 B$=D$
80 PC READ "I 4";X
90 PRINT #2, "DM=";X
100 GOTO 30

```

●PC READ "@..." 命令を使用した場合

- ・ユーザプログラム(PC側)

この例ではPC側ユーザプログラムは必要ありません

・BASICプログラム(ASCII側)

```

10 OPEN #2, "LPRT:(47)"
20 A$="00:00":B$=""
30 C$=MID$(TIME$, 4, 5)
40 IF C$<>A$ GOTO 30
50 D$=LEFT$(TIME$, 2)
60 IF D$=B$ GOTO 30
70 B$=D$
80 PC READ "@D,0,1,I 4";X
90 PRINT #2, "DM=";X
100 GOTO 30

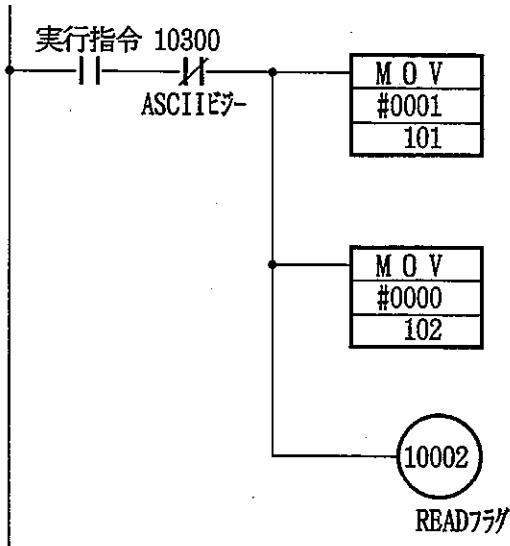
```

例2. ターミナルからのデータをPCに取込む例 (PC WRITE命令の使用例)

- ・キー入力により品種コード (DM0000) を切替える。
- ・キー入力データ: 10進4桁

●PC WRITE命令を使用した場合

- ・ユーザプログラム (PC側)



- ・BASICプログラム (ASCII側)

```

10 INPUT I
20 PC WRITE "I4";I
30 GOTO 10

```

●PC WRITE "@..." 命令を使用した場合

- ・ユーザプログラム (PC側)

この例ではPC側ユーザプログラム
は必要ありません

- ・BASICプログラム (ASCII側)

```

10 INPUT I
20 PC WRITE "@D,0,1,I4";I
30 GOTO 10

```

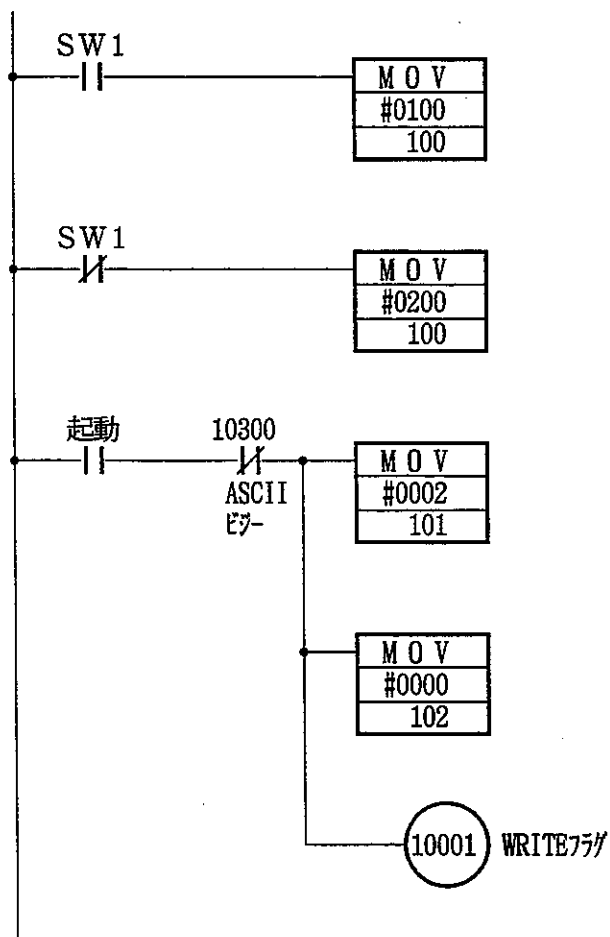
例3. PC側の複数種類のデータを切替え印字する例 (PC GET命令の使用例)

条件・2種類の生産数 (DM0000、DM0001) エリアがPCにあり種類の切替えと、起動により生産数を印字する。

・プリンタ：ポート2に接続、ボーレート：4800BPS

・切替え：SW1 但しSW1は起動前にON/OFFさせているものとする。

・ユーザプログラム (PC側)



・BASICプログラム (ASCII側)

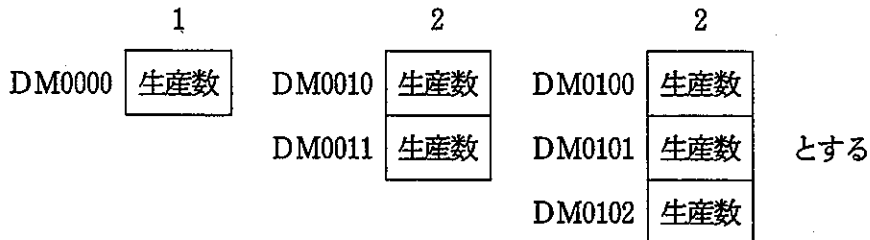
```

10 OPEN #2, "LPRT:(47)"
20 PC READ "2I4";X,Y
30 PC GET I, J
40 IF J=1 THEN GOTO 100
50 IF J=2 THEN GOTO 200
60 GOTO 30
)
100 PRINT #2, "DATA1=";X
)
200 PRINT #2, "DATA2=";Y
)

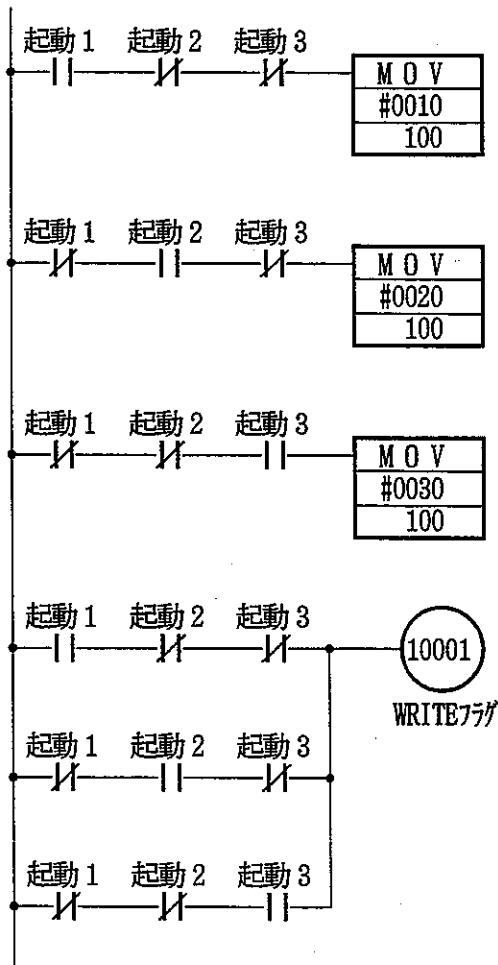
```

例4. 複数種類のデータを異なる起動入力で印字する例 (ON PC命令の使用例)
 条件・3種類の生産数 (DM0000、DM0010、DM0100) があり
 それぞれ異なった起動入力でそれぞれ印字する。
 ・プリンタ: ポート2に接続、ボーレート: 4800BPS

各生産数は



・ユーザプログラム (PC側)



・BASICプログラム (ASCII側)

```

10 OPEN #2, "LPRT:(47)"
20 ON PC 1 GOSUB 100
30 ON PC 2 GOSUB 200
40 ON PC 3 GOSUB 300
50 PC ON
60 GOTO 60

100 PC READ "@D,0,1,I4";X1
110 PRINT #2, "DM0=";X1
120 RETURN

200 PC READ "@D,10,2,2I4";X1, X2
210 PRINT #2, "DM10=";X1
220 PRINT #2, "DM11=";X2
230 RETURN

300 PC READ "@D,100,3,3I4";X1,X2,X3
310 PRINT #2, "DM100=";X1
320 PRINT #2, "DM101=";X2
330 PRINT #2, "DM102=";X3
340 RETURN
  
```

(注) ON PC GOSUB命令はそれぞれ一度実行すれば有効となります。(7・28ページ参照)

例5. PC側データ・時刻の印字例 (PRINT命令の使用例)

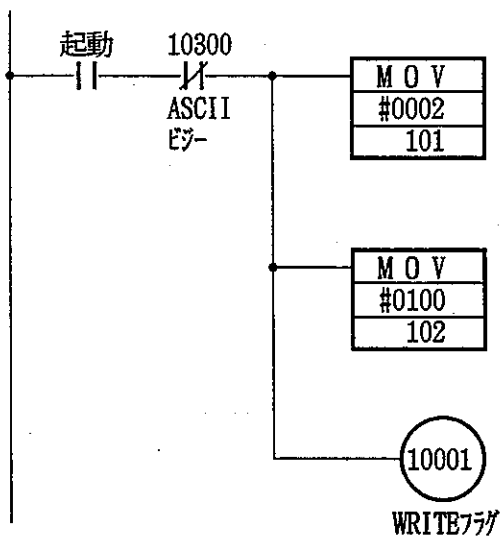
条件・ポート2にプリンタを接続する。

・PCからデータが転送された時、そのデータと時刻を印字する。

・プリンタ: VP-800 (エプソン製)

DM0100	データ1
DM0101	データ2

・ユーザプログラム (PC側)



・BASICプログラム (ASCII側)

```

10 OPEN #2, "LPRT:(47)"
20 PC READ "2I4"; D1, D2
30 PRINT #2, "DATA1=";
  D1, "DATA2="; D2,
  "TIME="; TIME$
40 GOTO 20

```

〈プリント結果例〉

DATA1= 5678	DATA2= 9876	TIME=13:45:03
DATA1= 3249	DATA2= 12	TIME=14:02:51

例6. 表示器への表示例 (PRINT命令の使用例)

条件・ポート2の表示装置に表示する。

- ・表示データはリレー1000のON/OFF状態とする。
- ・表示装置：プラズマディスプレイ

・ユーザプログラム (PC側)

この例ではPC側ユーザプログラム
は必要ありません

・BASICプログラム (ASCII側)

```

10 OPEN #2, "SCRN: (40)"
20 PC READ "@R, 10, 1, B0"; R
30 IF R=0 THEN RS$= "OFF"
   ELSE RS$= "ON"
40 PRINT #2, "RELAY="; RS$

```

例7. バーコードリーダによる入力例 (PC WRITE命令の使用例)

条件・ポート2にバーコードリーダを接続する。

- ・バーコードリーダからの出力データフォーマットは、以下とする。

STX	データ1	データ2	データ3	データ4	データ5	データ6	データ7	データ8	データ9	データ10	ETX
-----	------	------	------	------	------	------	------	------	------	-------	-----

- ・読取ったデータはそのままPCへ転送する。
- ・バーコードリーダ：3X2CB-ZMA (オムロン製)

・ユーザプログラム (PC側)

この例ではPC側ユーザプログラム
は必要ありません

・BASICプログラム (ASCII側)

```

10 OPEN #2, "COMU: (22)"
20 A$=INPUT$ (1, #2)
30 IF A$=CHR$ (2) GOTO 50
40 GOTO 20
50 B$=INPUT$ (11, #2)
60 IF CHR$ (3) =
   RIGHT$ (B$, 1)
   THEN B$=MID$ (B$, 1, 10)
   ELSE GOTO 20
70 PC WRITE "@D, 0, 5, 5A3"; B$
80 GOTO 20

```

注 COMUについては「7-3 BASIC命令
の説明」のOPEN命令の項を参照ください。

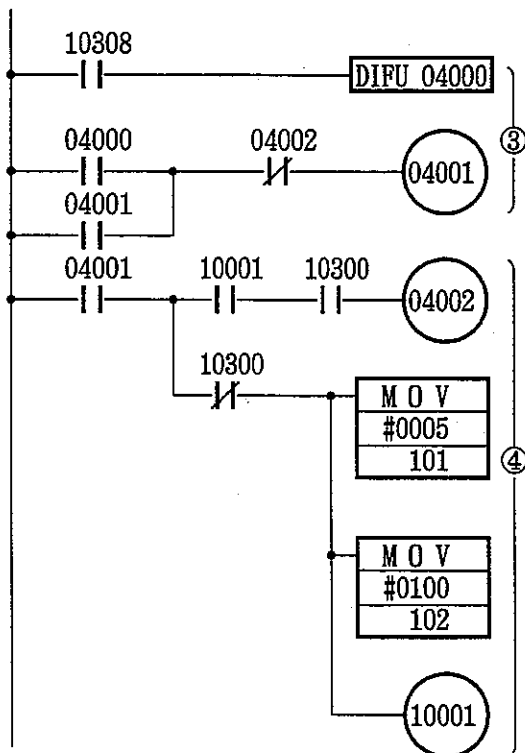
DM0000	データ1	データ2
DM0001	データ3	データ4
DM0002	データ5	データ6
DM0003	データ7	データ8
DM0004	データ9	データ10

例 8. ASCII ユニット側が主導権をもって、PC→ASCII のデータ転送を行う例

(1) PC READ 命令を使用した場合

・ ユーザプログラム (PC 側)

・ BASIC プログラム (ASCII 側)



```

① 1 0 0 PC PUT 1
②⑤ 1 1 0 PC READ "5 I 4" ; A1, A2, A3, A4, A5
⑦ 1 2 0 PC PUT 0
⑧ 1 3 0 PRINT A1, A2, A3, A4, A5
  
```

【動作説明】

- ① ASCII : PC PUT 1 で接点 10308 を ON します。
- ② ASCII : PC READ 命令を実行し、データがくるのを待ちます。
- ③ PC : 接点 10308 の立上り微分をとり、この自己保持 (04001) を ON します。
- ④ PC : 接点 04001 の ON により、ASCII に転送開始 CHNo、転送 CH 数をセットし、WRITE フラグ (10001) ON にて ASCII ユニットにデータを送ります。
- ⑤ ASCII : PC からデータが送られると、BUSY フラグ (10300) を ON します。
- ⑥ PC : ASCII がデータ転送を開始すると BUSY フラグが ON されているので、PC は WRITE フラグを OFF します。また、自己保持回路 (04001) を OFF します。
- ⑦ ASCII : データをすべて転送し終わると、PC PUT 0 で接点 10308 を OFF し次のデータ転送に備えます。
- ⑧ ASCII : 読込んだデータを表示します。

(2) PC READ "@..." 命令を使用した場合

・ ユーザプログラム (PC 側)

・ BASIC プログラム (ASCII 側)

この例では PC 側ユーザプログラム
は必要ありません

```

1 0 0 PC READ "@D, 100, 5, 5I4" ; A1, A2, A3, A4, A5
1 1 0 PRINT A1, A2, A3, A4, A5
  
```

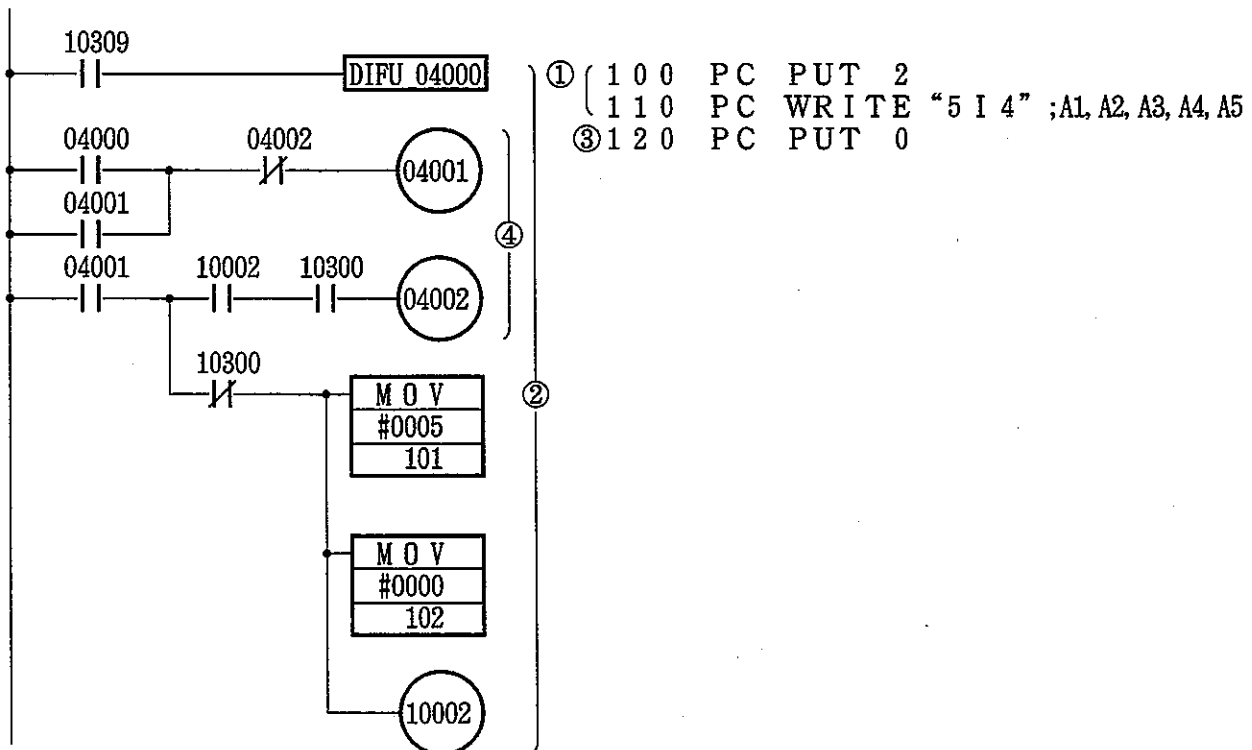
【動作説明】

- ① ASCII : @形式の PC READ 命令で、PC 側ユーザプログラムに関係なくデータを読み込みます。
- ② ASCII : ①で読込んだデータを表示します。

例9. ASCIIユニット側が主導権をもって、ASCII→PCのデータ転送を行う例

(1) PC WRITE命令を使用した場合
 ・ ユーザプログラム (PC側)

・ BASICプログラム (ASCII側)



【動作説明】

- ① ASCII : PC PUT 2によって、接点10309をONします。
 PC WRITE命令を実行し、PCから起動されるのを待ちます。
- ② PC : 接点10309がONすると、PC WRITEを起動します。
- ③ ASCII : PC WRITE命令が終了するとPC PUT 0によって接点10309をOFFします。
- ④ PC : PC WRITE命令が終了すると (ASCIIビジー (10300) がOFFすると) 自己保持回路04001をOFFします。

(注) このプログラムを繰返し実行する場合、PC PUT 0で接点10309をOFFしてから PC PUT 2でONするまでの時間が、PCのスキャンタイムより短いと10309のON/OFFがPCで検知できませんので、時間待ちをしてください。

(2) PC WRITE "@..." 命令を使用した場合
 ・ ユーザプログラム (PC側)

・ BASICプログラム (ASCII側)

この例ではPC側ユーザプログラムは必要ありません

```
1 0 0 PC WRITE "@D, 0, 5, 5I4" ; A1, A2, A3, A4, A5
1 1 0 END
```

【動作説明】

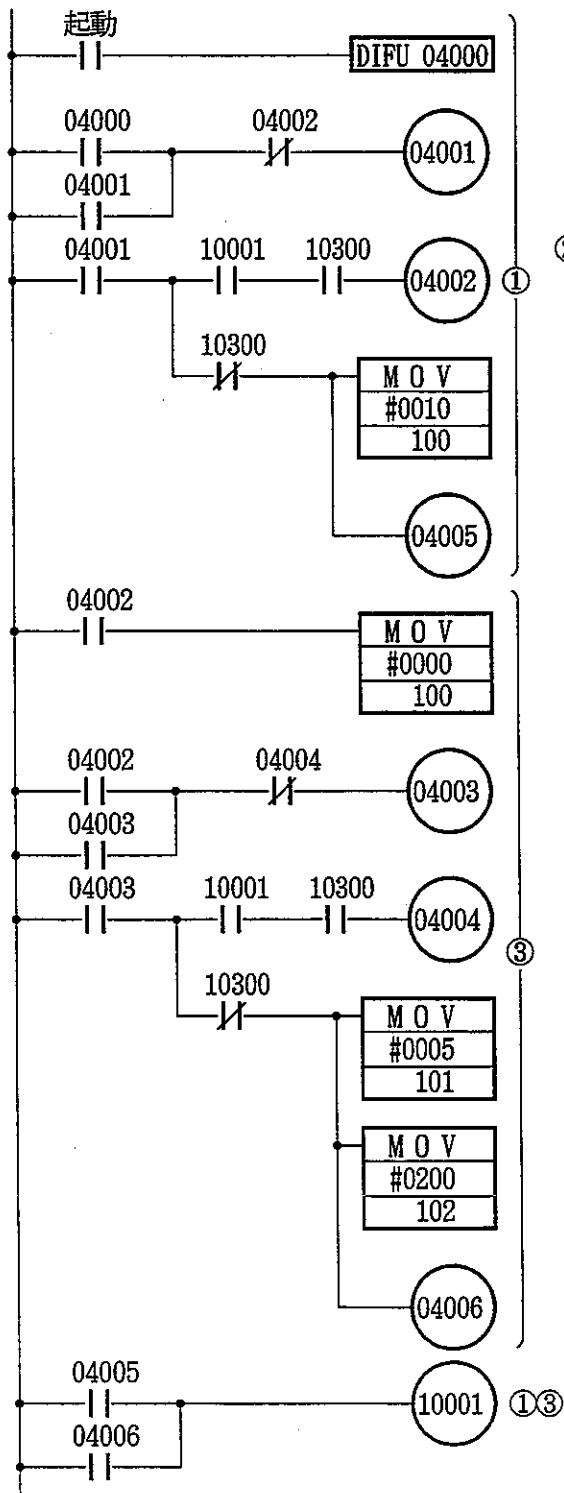
- ① ASCII : @形式のPC WRITE命令で、PC側ユーザプログラムに関係なくデータを書込みます。

例10. PC側が主導権をもって、PC→ASCIIのデータ転送を行う例

(1) PC READ命令を使用した場合

・ユーザプログラム (PC側)

・BASICプログラム (ASCII側)



```

10 ON PC 1 GOSUB 100
20 PC 1 ON
30 (通常処理)
  )
90 GOTO 30
②④ 100 PC READ "5 I 4" ;A1, A2, A3, A4, A5
    110 PRINT A1, A2, A3, A4, A5
    120 RETURN
  )

```

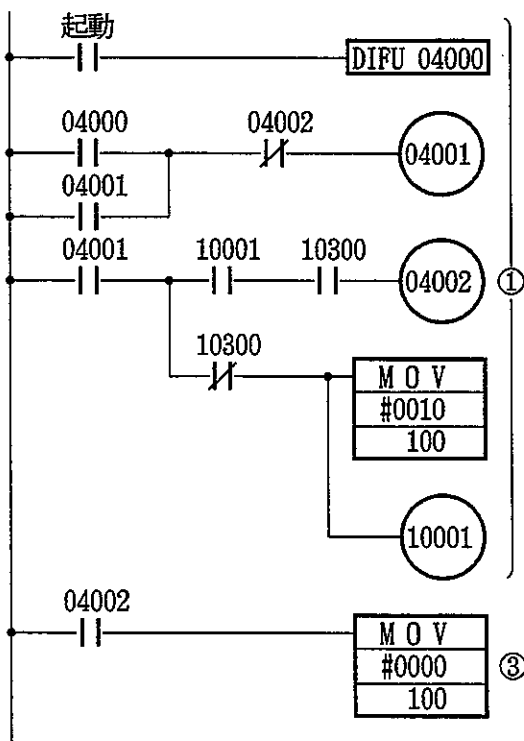
【動作説明】

- ①PC : 起動条件ONの立上り微分をとり、この自己保持 (04001) をONし、ON PC 1 GOSUBを実行させるべく割込番号をセットし、WRITEフラグ (10001) をONします。
- ②ASCII : ON PCの割込みがかかると割込処理ルーチン (100~120行) に分岐し、PC READ命令がPC側から起動されるのを待ちます。
- ③PC : ON PCの割込みが開始されると (ASCIIビジー (10300) ON) 割込番号を0にして、再度割込みが入るのを防ぎます。
また、PC READ命令のパラメータをセットし、WRITEフラグ (10001) をONして、PC READ命令を起動します。
- ④ASCII : PCから起動がかかると、PC READ命令を実行し、READしたデータを表示します。
その後、割込処理ルーチンからメインルーチンに戻り、割込みが入るのを再び待ちます。
- ⑤PC : PC READ命令が終了すると (ASCIIビジー (10300) ON→OFF) 初期状態に戻ります。

(2) PC READ “@...” 命令を使用した場合

・ユーザプログラム (PC側)

・BASICプログラム (ASCII側)



```

10 ON PC 1 GOSUB 100
20 PC 1 ON
30 (通常処理)
   }
90 GOTO 30
②100 PC READ"@D,200,5,5I4";A1,A2,A3,A4,A5
110 PRINT A1,A2,A3,A4,A5
120 RETURN

```

【動作説明】

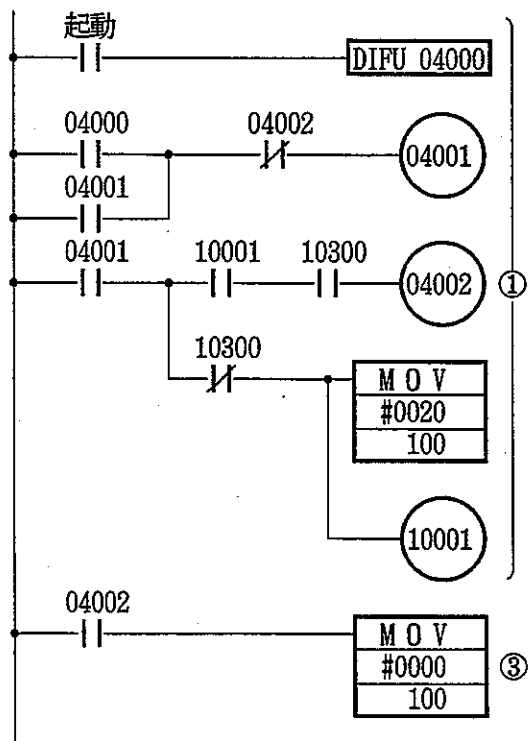
- ①PC : 起動条件ONの立上り微分をとり、この自己保持 (04001) をONし、ON PC 1 GOSUBを実行させるべく割込番号をセットし、WRITEフラグ (10001) をONします。
- ②ASCII : ON PCの割込みがかかると、割込処理ルーチン (100~120行) に分岐し、@形式 PC READ命令によってPCのデータを読み込みます。
- ③PC : ASCIIへのデータ転送が終了すると (ASCIIビジー (10300) がON→OFF) 割込番号を0にして、割込みが発生しないようにします。

例11. PC側が主導権をもって、ASCII→PCのデータ転送を行う例

(1) ON PC命令とPC WRITE "@..." 命令を使用した場合

・ユーザプログラム (PC側)

・BASICプログラム (ASCII側)



```

10 ON PC 2 GOSUB 100
20 PC 2 ON
30 (通常処理)
  }
90 GOTO 30
②100 PC WRITE "@, 400, 5, 514"; A1, A2, A3, A4, A5
110 PRINT A1, A2, A3, A4, A5
120 RETURN

```

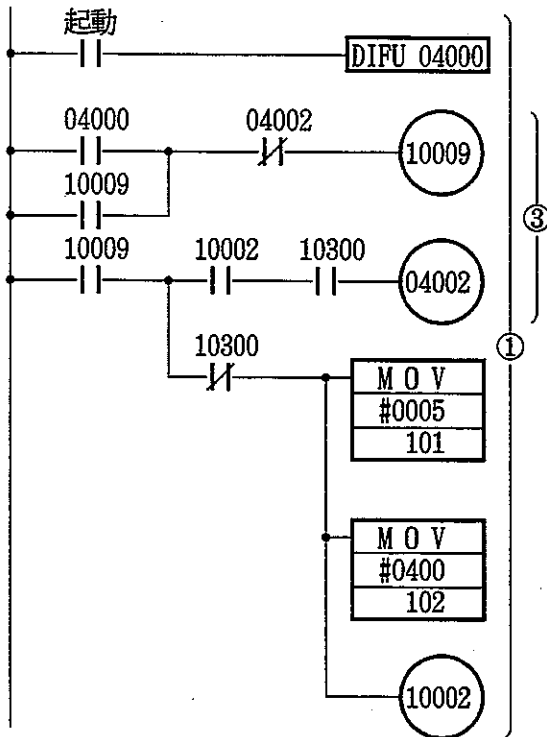
【動作説明】

- ①PC : 起動条件ONの立上り微分をとり、この自己保持 (04001) をONし、ON PC 1 GOSUBを実行させるべく割込番号をセットし、WRITEフラグ (10001) をONします。
- ②ASCII : ON PCの割込みがかかると、割込処理ルーチン (100~120行) に分岐し、@形式 PC WRITE命令によって、PCへデータを書込みます。
- ③PC : ASCIIからのデータ転送が終了すると (ASCIIビジー (10300) がON→OFF) 割込番号を0にして、割込みが発生しないようにします。

(2) PC GET命令とPC WRITE命令を使用した場合

・ ユーザプログラム (PC側)

・ BASICプログラム (ASCII側)



```

100 PC GET I, J
110 K=J AND 2
②120 IF K<>2 THEN 100
130 PC WRITE "5 I 4" ;A1, A2, A3, A4, A5
140 END

```

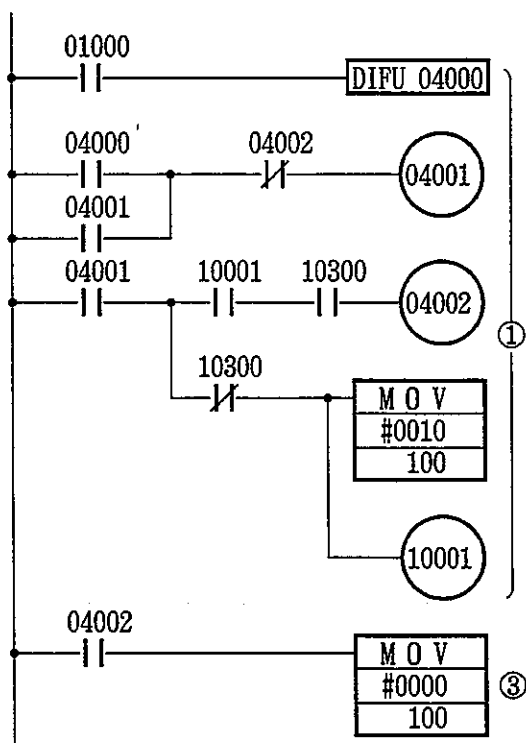
【動作説明】

- ①PC : 起動条件ONの立上り微分をとり、この自己保持(10009)をONしASCIIを起動します。PC WRITEに必要なパラメータをセットし、WRITEフラグをONして、PC WRITE命令の実行を要求します。
- ②ASCII : PC GETによって、PCから起動指令がくると、PC WRITE命令を実行します。
- ③PC : PC WRITE命令が終了すると (ASCIIビジー (10300) ON→OFF) 自己保持接点をOFFします。

例12. ASCIIユニット側でデータ処理を行う例

・C200Hの入力「01000」がONするごとに、C200HのDM0100より10CH分をASCIIユニットへ転送し、ASCIIユニット内で演算した後、その結果を15CH分、C200HのDM0200～0214へ転送する例。

・ユーザプログラム (PC側)



・BASICプログラム (ASCII側)

```

100 ON PC 1 GOSUB 1000
110 PC 1 ON
    )
④190 GOTO 190

{
1000 PC READ"@D,100,10,10H4";A1 ---- A10
1010 (演算処理)
②  : B1~B15へ結果を代入する
    }
1100 PC WRITE"@D,200,15,15H4";B1 --B15
1110 RETURN

```

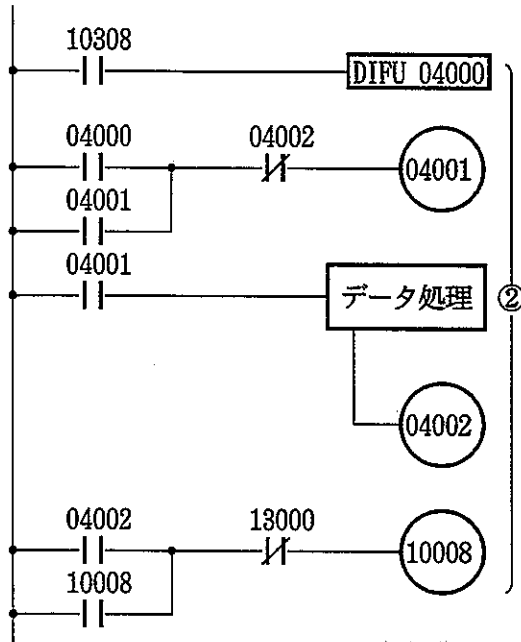
【動作説明】

- ①PC : 接点01000をONするとこの立上り微分をとり、この自己保持 (04001) をONし、ON PC 1 GOSUBを実行させるべく割込番号をセットし、WRITEフラグ (10001) をONします。
- ②ASCII : ON PCの割込みがかかると割込処理ルーチン (1000～1110行) に分岐し、@形式のPC READ命令によって、PCのデータを変数A1～A10に読みます。A1～A10のデータを使用しASCIIで演算し、その結果をB1～B15に格納します。この結果を@形式のPC WRITE命令によって、PCへ転送します。
- ③PC : ON PC命令の実行が終了すると (ASCIIビジー (10300) ON→OFF) 割込番号を0にして、割込みが発生ないようにします。
- ④ASCII : @形式PC WRITE命令の実行を終了するとメインルーチンに戻り、次のON PCの割込みがかかるのを待ちます。

例13. PC側でデータ処理を行う例

- ・ASCIIユニットに3CH分のデータが入力されるごとに、PC側にそのデータを転送し、PC側でそのデータを処理し、ASCII側で4CH分データとして送り返す例。

・ユーザプログラム (PC側)



・BASICプログラム (ASCII側)

```

100 PC PUT 0
① 110 INPUT A1, A2, A3
120 PC WRITE "@D, 100, 3, 3I4"; A1, A2, A3
130 PC PUT 1
140 PC GET I, J
150 K=J AND 1
③ 160 IF K<>1 THEN 140
170 PC READ "@D, 200, 4, 4I4"; B1, B2, B3, B4
180 PRINT B1, B2, B3, B4
190 GOTO 100

```

【動作説明】

- ①ASCII: PC PUT 0によって、PCを初期状態にします。
INPUT命令によって数値を入力し、@形式PC WRITE命令でデータをPCに書き込み、データ転送が終了すると、PC PUT 1にてPCに起動をかけます。
- ②PC: ASCIIがデータ転送を終了し、ASCIIから起動がかかると、ASCIIからのデータによって、データ処理を行います。
データ処理が終了すると接点04002がONし、これが自己保持接点10008をONし、ASCIIにデータをREADするように要求します。
- ③ASCII: PC GETによって、PCからデータのREAD要求を持ちます。
READ要求がくると、@形式PC READ命令によって、PCからデータをREADしこれを表示します。

例14. パソコンなどの入力機器からの入力例 (ON COM命令の使用例)

条件・ポート1、ポート2にRS-232C機器を接続する。

ポート1またはポート2に通信が入ってきたら、これを文字変数に格納する。

・ユーザプログラム (PC側)

この例ではPC側ユーザプログラムは必要ありません

・BASICプログラム (ASCII側)

```

10  CLEAR 1000
100  OPEN #1, "COMU:"
110  OPEN #2, "COMU:"
120  ON COM1 GOSUB 1000
130  ON COM2 GOSUB 2000
140  COM1 ON:COM2 ON
150  GOTO 150

1000  A=LOC(1)
1010  IF A<>0 THEN
      A$=A$+INPUT$(A, #1)
1020  RETURN

2000  B=LOC(2)
2010  IF B<>0 THEN
      B$=B$+INPUT$(B, #2)
2020  RETURN

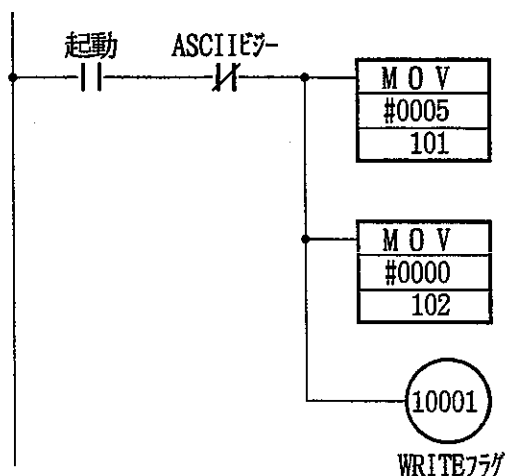
```

例15. 起動スイッチによりデータを転送する例 (WAIT命令の使用例)

条件・ポート1にターミナルを接続する。

- ・STARTと表示がでてから、PC側の起動スイッチが押されると、PC側のデータをASCIIユニットに送り、これを表示する。
- PC側の起動スイッチが10分以内に入らなかった場合、ERROR表示を行い、再度転送を行うかどうか入力する。

・ユーザプログラム (PC側)



・BASICプログラム (ASCII側)

```

100  PRINT "START"
110  WAIT "10:00.0", 1000
120  PC READ "514"; A, B, C, D, E
130  PRINT A, B, C, D, E
140  END

1000  PRINT "ERROR RETRY? Y/N"
1010  F$=INKEY$
1020  IF F$="Y" THEN 100
1030  IF F$="N" THEN END
      ELSE 1010

```

例16. 割込みで処理を分ける例

条件・ポート1にターミナルを接続する。

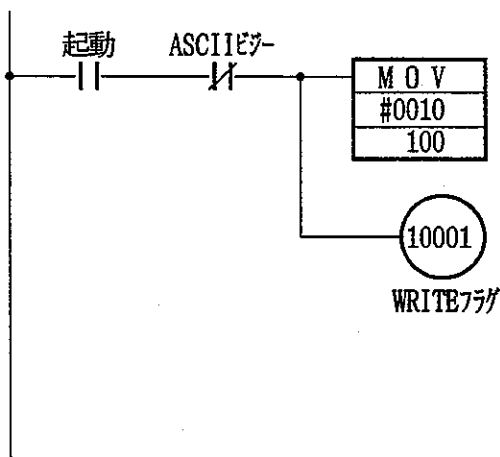
・ポート2にRS-232C機器を接続する。

・ポート1からのキー入力により、次のように処理を分ける。

- ①ポート1からキー1が入った後、ポート2からの通信割込みが入ったときは、そのデータを文字変数に格納し、PCからの割込みが入ったときはPCにデータを転送する。
 ②キー2が入った後、通信割込みが入ると、そのデータを直接PCに転送する。

・ユーザプログラム (PC側)

・BASICプログラム (ASCII側)



```

10 OPEN #1, "TERM:(42)"
20 OPEN #2, "COMU:(42)"
30 ON KEY 1 GOTO 100
40 ON KEY 2 GOTO 200
50 ON PC GOSUB 300
60 ON COM2 GOSUB 400
70 KEY ON: COM2 STOP
80 GOTO 80
① 100 ' KEY 1 ショリ
   110 COM2 ON: PC ON
   120 GOTO 120
   200 ' KEY 2 ショリ
② 210 COM2 ON
   220 IF A=1 THEN GOSUB 300
   230 GOTO 220
   300 ' PC ワリコミ ショリ
   310 B$=MID$(STR$(LEN(A$)),2)
   320 PC WRITE "@D,0,"+B$+" "+B$+"A3";A$
① 330 A=0
② 340 RETURN
   400 ' COM ワリコミ ショリ
   410 IF EOF(2) THEN RETURN
   420 A$=INPUT$(LOC(2), #2)
   430 A=1
   440 RETURN

```

【動作説明】

- ①ASCII: キー1を入力するとキー1割込処理ルーチン(100-120)に分岐し、ポート2からの通信割込みとPCからの割込みを許可します。
 通信割込みが入ると、通信割込処理ルーチン(400-440)に分岐し、通信データを文字変数に格納します。
 PCからの割込みが入ると、PC割込処理ルーチン(300-340)に分岐し、文字変数に格納された通信データをPCに転送します。
- ②ASCII: キー2を入力するとキー2割込処理ルーチン(200-230)に分岐し、ポート2からの通信割込みを許可します。
 通信割込みが入ると、通信割込処理ルーチン(400-440)に分岐し、通信データを文字変数に格納したあと、PC割込処理ルーチン(300-340)に分岐し、文字変数に格納された通信データをPCに転送します。
 このときPCからの起動は必要ありません。

例17. 機械語プログラムの使用例(1) 文字の分類

・キー入力した文字を数字列と文字列に分け、それを1つにするプログラムです。

・BASICプログラム

```
100 DEF USR0=&H2000
110 INPUT A$
120 A$=USR0(A$)
130 PRINT A$
140 END
```

【プログラミング方法】

MSET &H3000

MON

「CTRL」+「A」

\$2000から順にプログラムをキー入力します。

X

「CTRL」+「B」

機械語プログラムエリアを確保します。

機械語モニタに入ります。

ミニアセンブラモードに入ります。

プログラムを入力したらBASICモードまで戻ります。

以下のエリアをプログラムエリア、ワークエリア、バッファエリアとして使用します。

・プログラムエリア

\$2000～\$24FF	プログラムエリア
\$2500～\$2507	ワークエリア
\$2600～\$27FF	バッファエリア

・ワークエリア

\$2500～\$2501	バッファ1 (数字格納用) ポインタ格納用
\$2502～\$2503	バッファ2 (文字格納用) ポインタ格納用
\$2504～\$2505	転送元アドレス格納用
\$2506～\$2507	転送先アドレス格納用

・バッファエリア

\$2600～\$26FF	数字列格納エリア
\$2700～\$27FF	文字列格納エリア

【機械語プログラムの動作説明】

文字変数に入っている数字と文字をそれぞれ数字列用バッファ、文字列用バッファに分けて格納します。
次に、元の文字変数に数字列、文字列の順に格納します。

・機械語プログラム

```

$2000 PSHA
      PSHB
      PSHX
      LDD # $2600
      STD $2500
      LDD # $2700
      STD $2502
      LDAB 0, X
      LDX 1, X
      STX $2504
$2016 LDX $2504
      LDAA 0, X
      INX
      STX $2504
      CMPA # $30
      BLT $2032
      CMPA # $39
      BHI $2032
      LDX $2500
      STAA 0, X
      INX
      STX $2500
      BRA $203B
$2032 LDX $2502
      STAA 0, X
      INX
      STX $2502
$203B DECB
      BNE $2016
      LDD $2500
      LDX # $2600
      STX $2504
      SUBD # $2600
      PULX
      PSHX
      PSHB
      LDX 1, X
      STX $2506
      JSR $2100
      LDX # $2700
      STX $2504
      PULB
      PULX
      PSHX
      LDX 1, X
      ABX
      STX $2506
      LDD $2502
      SUBD # $2700
      JSR $2100
      PULX
      PULB
      PULA
      RTS

```

レジスタ退避

バッファ1先頭アドレスをポインタ1にセット

バッファ2先頭アドレスをポインタ2にセット

文字数 GET

文字変数先頭アドレス GET

DOUNTIL (文字数回)

文字 GET

文字変数アドレスポインタ+1

IF (\$30以上)

THEN

IF (\$39以下 数字のとき)

THEN

バッファ1に数字を格納する

ENDIF

文字をバッファ2に格納する

カウンタ更新

ENDDO

バッファ1から文字変数への転送

バッファ2から文字変数への転送

\$2100	LDX	\$2504	: データ転送サブルーチン
	LDAA	0, X	
	INX		
	STX	\$2504	
	LDX	\$2506	
	STAA	0, X	
	INX		
	STX	\$2506	
	DECB		
	BNE	\$2100	
	RTS		

例18. 機械語プログラムの使用例(2) 複数の変数使用

- ・機械語プログラム内で複数の変数を使用する例で、2つの文字列の対応する1文字間で大きい方を抽出します。
- ・比較元の2つの文字列と結果格納用の合計3つの文字変数を使用します。

・BASICプログラム

```

100 ' *****
110 ' ***** 2ツノ モジ ヘンスウ/ タイオスル/スウチ/ ***** キカイゴプログラム $2170--$21AF
120 ' ***** オキイホウ シラベル プログラム ***** ワークエリア $2000--$2005
130 ' *****
140 CDX$="13426285903581693417" ' ヒカクモト モジ ヘンスウ CDX$
150 CDY$="57201674337291551930" ' = CDY$
160 ANS$="00000000000000000000" ' ケツカ カクノウ モジ ヘンスウ ANS$
170 DEF USRO=&H2170 ' キカイゴ カンスウ カクノウアドレス テイギ
180 CX%=VARPTR(CDX$)+1 ' CDX$/ カクノウアドレス サンシュツ
190 POKE &H2000, CX% ¥ 256 ' = ジョウイ --> ワークエリア
200 POKE &H2001, CX% MOD 256 ' = カイ --> ワークエリア
210 CY%=VARPTR(CDY$)+1 ' CDY$/ カクノウアドレス サンシュツ
220 POKE &H2002, CY% ¥ 256 ' = ジョウイ --> ワークエリア
230 POKE &H2003, CY% MOD 256 ' = カイ --> ワークエリア
240 ANS$=USRO(ANS$) ' キカイゴ カンスウ ジョウコウ
250 PRINT ANS$
260 END

```

動作説明

機械語関数プログラム内で使用する変数のアドレスを、あらかじめVARPTRで求めておき、ワークエリアに格納しておきます。本例はこの方法により、機械語関数プログラムで3つの文字変数を使用しています。

ご留意

- ・変数のアドレスは整数変数で計算します。
 - ・文字変数で文字変数名="XXXX"の形式のものは、文字変数領域には格納されず、ソースプログラム内のデータそのものが使用されます。
- 従って、240行の実行後は160行の" "の中の値が変化します。

・機械語プログラム

2170	E8	00		LDAB	\$00, X	' ANS\$の長さ→Bレジスタ
2172	EE	01		LDX	\$01, X	
2174	FF	20	04	STX	\$2004	' ANS\$のアドレス記憶
2177	FF	20	00	LDX	\$2000	
217A	EE	00		LDX	\$00, X	
217C	FF	20	00	STX	\$2000	' SDX\$のアドレス記憶
217F	FE	20	02	LDX	\$2002	
2182	EE	00		LDX	\$00, X	
2184	FF	20	02	STX	\$2002	' SDY\$のアドレス記憶
2187	FE	20	00	LDX	\$2000	
218A	A6	00		LDAA	\$00, X	' SDX\$のデータリード
218C	08			INX		
218D	FF	20	00	STX	\$2000	
2190	FE	20	02	LDX	\$2002	
2193	A1	00		CMPA	\$00, X	' SDY\$のデータと比較
2195	24	02		BCC	\$2199	' SDX\$のデータ<SDY\$のデータ?
2197	A6	00		LDAA	\$00, X	' SDY\$のデータリード
2199	08			INX		
219A	FF	20	02	STX	\$2002	
219D	FE	20	04	LDX	\$2004	
21A0	A7	00		STAA	\$00, X	' 大きい方をANS\$へライト
21A2	08			INX		
21A3	FF	20	04	STX	\$2004	
21A6	5A			DECB		
21A7	26	DE		BNE	\$2187	' 終了?
21A9	39			RTS		

<ワークエリア>

\$2000		文字変数SDX\$の格納アドレス記憶エリア
\$2001		
\$2002		文字変数SDY\$の格納アドレス記憶エリア
\$2003		
\$2004		文字変数ANS\$の格納アドレス記憶エリア
\$2005		

例19. 機械語プログラムの使用例(3) FCSの計算

- ・上位リンクで使用するFCSの計算を機械語で実行するプログラムです。
- ・計算する文字列をDA\$、計算結果を格納する文字列をFCS\$とします。

・BASICプログラム (100行～230行)

```

100 ' *****
110 ' ***** FCS ケイサン ( ケイゴ カンスウ ショウ ) ***** ケイゴ プログラム $2100--$213F
120 ' ***** DA$/ FCSヲ ケイサンシ ANS$トスル ***** ワークエリア $2000--$2001
130 ' *****
140 DA$="@10RR00310123" ' FCSヲ ケイサンスル DATAヲ
150 FCS$="43" ' FCSカクノウ モジツ("XX")ノナイヨウハ RUNジ ダイニュウサレ ヘンカスル
160 DEF USRO=&H2100 ' ケイゴ カンスウ カクノウアドレス テイギ
170 B%=VARPTR(FCS$)+1 ' FCS$/ カクノウアドレス サンジュツ
180 POKE &H2000, B% ¥ 256 ' = ショウイ --> ワークエリア
190 POKE &H2001, B% MOD 256 ' = カイ --> ワークエリア
200 DA$=USRO(DA$) ' ケイゴ カンスウ ショウコウ
210 ANS$=DA$+FCS$ ' FCSヲ フカンシ モジツ サクセイ
220 PRINT ANS$
230 END
240 '
500 ' *****
510 ' ***** FCS ケイサン ( BASICノイレ / ミ ) *****
520 ' ***** DA$/ FCSヲ ケイサンシ ANS$トスル *****
530 ' *****
540 DA$="@10RR00310123"
550 L=LEN(DA$)
560 Q=0
570 FOR N=1 TO L
580 Q=ASC(MID$(DA$, N, 1)) XOR Q
590 NEXT
600 FCS$=HEX$(Q)
610 IF LEN(FCS$)=1 THEN FCS$="0"+FCS$
620 ANS$=DA$+FCS$
630 PRINT ANS$
640 END

```

- ・比較のため500行～640行にBASIC命令で計算するプログラムがあります。
- ・実行時間を比較すると

機械語関数使用	(140行～220行)	29ms
BASIC命令のみ	(540行～630行)	160ms

 となります。

・機械語プログラム

2100	E6	00	LDAB	\$00, X	' DA\$の文字列長→Bレジスタ
2102	EE	01	LDX	\$01, X	' DA\$の格納アドレス→Xレジスタ
2104	4F		CLRA		
2105	A8	00	EORA	\$00, X	' EORを計算
2107	08		INX		
2108	5A		DECB		
2109	26	FA	BNE	\$2105	' 文字列分繰り返し
210B	16		TAB		
210C	C4	0F	ANDB	#\$0F	' FCS値のASCII変換
210E	C1	0A	CMPB	#\$0A	
2110	25	02	BCS	\$2114	' FCSの下位桁≥10
2112	CB	09	ADDB	#\$07	' THEN A~Fに変換
2114	CB	30	ADDB	#\$30	
2116	44		LSRA		
2117	44		LSRA		
2118	44		LSRA		
2119	44		LSRA		
211A	81	0A	CMPA	#\$0A	
211C	25	02	BCS	\$2120	' FCSの上位桁≥10
211E	8B	09	ADDA	#\$07	' THEN A~Fに変換
2120	8B	30	ADDA	#\$30	
2122	FE	20 00	LDX	\$2000	
2125	EE	00	LDX	\$00, X	
2127	ED	00	STD	\$00, X	' FCS\$用エリアに格納
2129	39		RTS		

〈ワークエリア〉

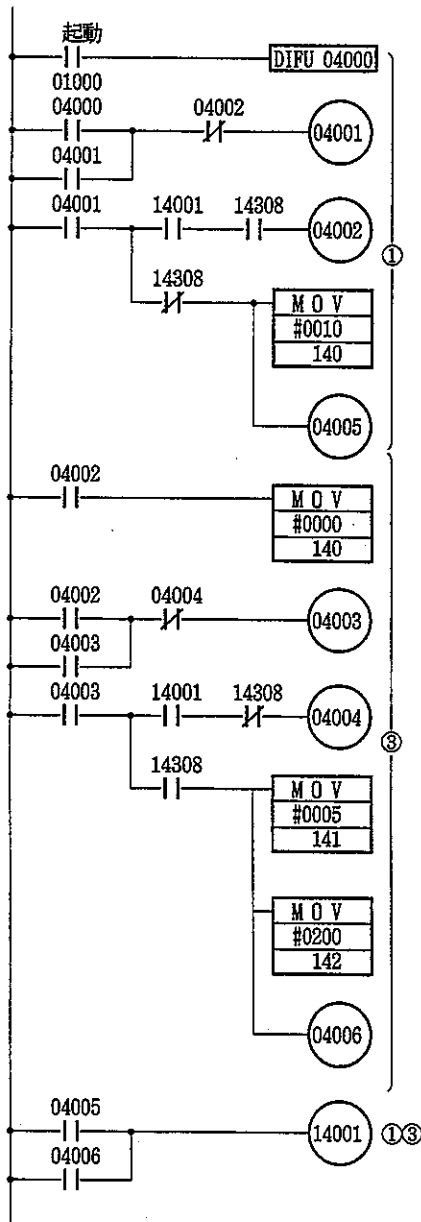
\$2000		文字変数FCS\$の格納アドレス記憶エリア
\$2001		

ご注意

- ・FCS\$のアドレスはあらかじめ\$2000、\$2001に格納してから機械語プログラムを呼び出します。

例20. PC側が主導権をもって、PC→ASCIIのデータ転送を行う例
(リモート子局上にASCIIユニットがある場合)

(1) PC READ命令を使用した場合
・ユーザプログラム (PC側)



BASICプログラム

```

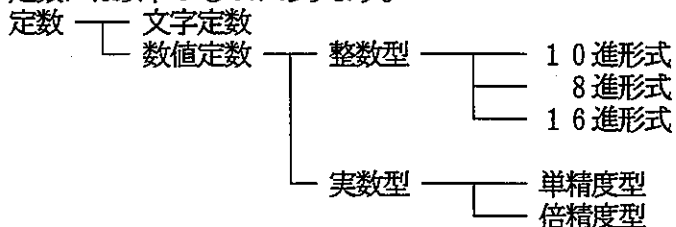
10 PC PUT 0
20 ON PC 1 GOSUB 100
30 PC 1 ON
40 GOTO 40
100 PC PUT 1
110 FOR I=1 TO 50:NEXT I
120 PC READ "514";A1, A2, A3, A4, A5
130 PRINT A1, A2, A3, A4, A5
140 PC PUT 0
150 RETURN

```

*注意: 110 PC PUTが確実に伝わりWRITE
フラグがOFFするまでの時間待ち。
PCスキャン、リモートスキャンの長さにより変
える必要があります。

■定数

定数には以下のものがあります。



●文字定数

文字セットの並びをダブルクォーテーション (") で囲って指定します。文字数の長さは255以下で、特に0の場合は空文字列と言います。

例) "CF-BASIC"

●数値定数

・整数型

10進形式 -32768～+32767の範囲内の10進数(0～9)並びで小数点をつけることはできません。10進数整数では数の後に%を付けて明示することもできます。

例) 1234
-1234
12%

8進形式 &Oまたは&で始まる8進数(0～7)の並びで&O～&177777の範囲です。

例) &O127
&7777

16進形式 &Hで始まる16進数(0～9、A～F)の並びで&H0～&HFFFFの範囲です。

例) &H5E
&HBF4

・実数型

単精度型 -1.70141E+38～1.70141E+38の範囲内の値で、7桁までの精度で格納され、7桁目を四捨五入して6桁で出力されます。次のいずれかで表現します。

- (a)有効桁数が7桁以下の定数 1234.5
(b)Eを用いた指数形式の定数 -1.2E+3
(c)最後に!が付いた定数 2.34!

倍精度型 -1.701411834604692D+38～1.701411834604692D+38の範囲内の値で、16桁の精度で格納され、16桁以下で出力されます。次のいずれかで表現します。

- (a)有効桁数が8桁以上の定数 1.23456789
(b)Dを用いた指数形式の定数 -1.2D-3
(c)最後に#が付いた定数 2.34#

変数

変数には、数値変数と文字変数がありますが、変数に値を与えるまえに数値変数を用いると、その値は0になっています。

また、文字変数の場合には空文字列（何の文字も持たない状態）になっています。

変数名

変数は英字で始まる255文字以内の英数字並びで表現されますが、最初の16文字を有効とします。

変数名の先頭がFN、またはBASICコマンドの予約語であってはけません。

※先頭に予約語が含まれていた変数の場合、シンタックスエラーが発生します。

例えば、「TOTAL」「ABSOL」の場合でも、先頭の「TO」「ABS」が予約語であるためシンタックスエラーが発生し、変数として使用することはできません。

型宣言文字

変数名の後に付けて、その変数の型を定義します。

変数名が同じでも、型が違えば別の変数として扱われます。

型宣言の別方法として、DEFINT、DEFSNG、DEFDBL、DEFSTR文があります。

% 整数型	1変数につき2バイトのエリアを使用する。
! 単精度実数型	1変数につき4バイトのエリアを使用する。
# 倍精度実数型	1変数につき8バイトのエリアを使用する。
\$ 文字型	最大255文字のデータを格納できます。

配列変数

配列は、同じ型のデータを持つ複数個のデータの集まりで、添字付きの同一名で配列の各要素を参照できます。

配列の次元数、および添字の最大値はメモリ容量に依存します。

型変換

数値データを使って演算する場合には、演算実行時に必要に応じて型変換を自動的に行います。

変換規則は以下に従います。

(1)代入文において、右辺の数値データの型と、左辺の数値変数の型が異なる場合には、左辺の型に変換されて代入されます。

ただし、文字型を数値型へまたは、数値型を文字型へ代入することはできません。

例) $A\% = 12.3 - A\%$ には12が代入されます。

(2)代入文において倍精度型データを単精度変数に代入する場合は、四捨五入が行われます。

例) $A = 12.3456789\# \rightarrow A$ には12.34568 (7桁に丸められる) が代入され表示すると12.3457 (6桁に四捨五入) となります。

(3)単精度型と倍精度型のデータ間で演算する場合は単精度型の数値を倍精度型に変換してから演算しますので、結果は倍精度型となります。

例) $10\# / 3 \rightarrow$ 結果は倍精度型

(4)論理演算の場合には、数値はすべて整数型に変換してから演算します。

-32768~32767の範囲内の整数に変換できない場合はエラーとなります。

例) $A = \text{NOT } 12.34 \rightarrow A$ には-13が代入されます。

(5)実数を整数に変換する場合は、小数点以下を四捨五入します。

例) (1)と同じ

■式

定数、変数、関数を演算子で結んだものや、単に数値、変数、文字だけのものも式と呼びます。

式には次の4種類があります。

・算術式 ・関係式 ・論理式 ・文字式

この内、最初の3種はその評価結果が数値であることから、まとめて数値式と呼び文字式と区別します。

●算術式

定数、変数、関数などの数値データを下表の演算子で結んだものです。

算術演算子	記述例	説明
+	A+B	加算
-	A-B, -A	減算または負号
*	A*B	乗算
/	A/B	実数除算
÷	A÷B	整数除算
MOD	A MOD B	整数除算の乗余
^	A^B	べき乗

÷または、MOD演算子でAまたはBが実数の場合は小数部分四捨五入して整数にしてから演算します。

●関係式

関係式は2つの値を比較し、その結果が真なら-1、(&HFFFF) 偽なら0となります。

関係演算子	記述例	説明
=	A=B	等しい
<>, ><	A<>B	等しくない
<	A<B	小さい
>	A>B	大きい
<=, =<	A<=B	小さいか等しい
>=, =>	A>=B	大きい等しい

●文字式

文字定数、文字変数を演算子+で連結したもので、2つの文字データの加算ではなく、2つの文字列を連結して1つの文字列を得る場合に使います。

(例) A\$="CF"
B\$="BASIC"
PRINT A\$+
"-" + B\$
CF-BASICと出力される←

●論理式

定数、変数、関数などの数値データを下表の論理演算子で結んだもので、ビット操作や論理演算をしたり、複数の関係式の結果を調べたりするときに使います。

8進、16進形式でのビットごとの論理演算を実行します。

論理演算子	論 理	
NOT (否定)	A	NOT A
	1 0	0 1
AND (論理積)	A B	A AND B
	1 1	1
	1 0	0
	0 1	0
	0 0	0
OR (論理和)	A B	A OR B
	1 1	1
	1 0	1
	0 1	1
	0 0	0
XOR (排他的論理和)	A B	A XOR B
	1 1	0
	1 0	1
	0 1	1
	0 0	0
IMP (包含)	A B	A IMP B
	1 1	1
	1 0	0
	0 1	1
	0 0	1
EQV (同値)	A B	A EQV B
	1 1	1
	1 0	0
	0 1	0
	0 0	1

論理式の計算例は次ページをご覧ください。

■演算子の優先順位

優先順位

- | | |
|-------------------|---------|
| 1. $\wedge$ (べき乗) | 8. NOT |
| 2. - (負号) | 9. AND |
| 3. *, / | 10. OR |
| 4. $\div$ | 11. XOR |
| 5. MOD | 12. EQV |
| 6. +, - | 13. IMP |
| 7. 関係演算子 | |

*カッコ付きの式および関数は、演算子の優先順位に関係なく先に処理されます。

■論理式の計算例

●NOT (否定)

A% = 1 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
 NOT 1 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 = -2
 NOT A% = -2

●AND (論理積)

A% = 5 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
 B% = 6 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
 A% AND B% = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 = 4

●OR (論理和)

A% = 4 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
 B% = 3 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
 A% OR B% = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 = 7

●XOR (排他的論理和)

A% = -4 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
 B% = 5 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
 A% XOR B% = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 = -7

●EQV (同値)

A% = -4 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
 B% = 5 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
 A% EQV B% = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 = 6

●IMP (包含)

A% = -4 = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0
 B% = 5 = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
 A% IMP B% = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 = 7

7-2 BASIC命令一覧

この一覧表は、全命令をアルファベット順に並べてその概要を示したものです。各命令の詳細については、「7-3 BASIC命令の説明」を参照してください。

表中のステートメントおよび関数欄のアルファベットは次のものを示します。

G：一般ステートメント A：算術演算関数 S：特殊関数
D：デバイスコントロールステートメント C：文字列関数

記述欄の〔 〕は省略可能項目を示します。

命 令	記 述	機 能	コマンド	ステートメント	関数	ページ
ABS	ABS (数値式)	引数で与えた数値の絶対値を得る			A	7・52
ACOS	ACOS (数値式)	引数で与えられた数値の逆余弦（アークコサイン）を得る			A	7・52
ASC	ASC (文字式)	引数で与えた文字列の先頭文字の数値を得る			C	7・55
ASIN	ASIN (数値式)	引数で与えられた数値の逆正弦（アークサイン）を得る			A	7・52
ATN	ATN (数値式)	引数で与えられた数値の逆正接（アークタンジェント）を得る			A	7・52
AUTO	AUTO [行番号] [増分値]	行の先頭に番号を自動的に発生させる	○			7・12
CDBL	CDBL (数値式)	数値を倍精度実数値に変換する			A	7・53
CHR\$	CHR\$ (数値式)	引数で与えた数値に対応する1文字を得る			C	7・55
CINT	CINT (数値式)	数値を四捨五入して整数値にする			A	7・53
CLEAR	CLEAR	数値変数、文字変数の初期化を行う		G		7・19
CLOSE	CLOSE [#ポート番号]	ポートを閉じる		D		7・48
CLS	CLS [#ポート番号]	画面をクリアする		D		7・48
COM ON/OFF/STOP	COM [ポート番号] xx	通信回線からの割込みの許可、禁止、停止を制御する		G		7・19
CONT	CONT	実行停止しているプログラムの実行再開をする	○			7・12
COS	COS (数値式)	引数で与えられた数値の余弦（COS）を得る			A	7・53

命 令	記 述	機 能	マ ボ	ステ ート	関 数	ペー ジ
CSNG	CSNG (数値式)	数値を単精度実数に変換する			A	7・53
DATA	DATA定数 [, 定数]	READ文に与える数値および文字 定数を定義する		G		7・20
DATE\$	DATE\$	日付をセットする、日付を与える			S	7・58
DAY	DAY	曜日をNaでセットする、または与える			S	7・58
DEF FN	DEF FN名前 [(引数 [, 引 数...])] =式	ユーザ作成の関数を定義し、それに 名前を付ける		G		7・20
DEF INT/SNG /DBL/STR	DEF xxx文字の範囲 [, 文字の範囲] ...	変数の型を整数、単精度、倍精度、 文字に宣言する		G		7・20
DEF USR	DEF USR [番号] =開始番地	機械語で作られたユーザ関数の実行 開始番地を定義する		G		7・21
DEL	DEL [行番号1] [- [行 番号2]]	プログラム行の一部を削除する	○			7・12
DIM	DIM変数名 (添字最大値...) [, 変数名 (...)]	配列変数の設定と、領域の割当てを 行う		G		7・21
EDIT	EDIT [行番号]	プログラムの1行編集する	○			7・13
END	END	プログラムの実行を終了し、全ポ ートを閉じる		G		7・21
EOF	EOF (ポート番号)	ポート番号で指定されたポートのバ ッファが空であるかどうかを調べる			S	7・58
ERL/ERR	ERL/ERR	エラーの発生した行番号およびエラ ーコードを得る			S	7・58
ERROR	ERRORエラー番号	エラー発生のシミュレート、エラー 番号を定義する		G		7・21
FIX	FIX (数値式)	引数で与えた数値の整数部分を得る			A	7・53
FOR ...TO... STEP~NEXT	FOR...TO... [STEP] ~NEXT...	FORからNEXTまでを指定回数 繰返す		G		7・22
FRE	FRE (式)	メモリの未使用領域の大きさを与える			S	7・59
GOSUB ~RETURN	GOSUB 行番号 ~RETURN	サブルーチンを呼出し実行し、 RETURNで戻る		G		7・22

命 令	記 述	機 能	マ ボ	ア ト メント	関 数	ページ
GOTO	GOTO行番号	指定の行番号へ分岐する		G		7・22
HEX\$	HEX\$ (数値式)	引数で与えた10進数値を16進数の文字列に変換する			C	7・55
IF...THEN... ELSE IF...GOTO ELSE	IF 論理式 { THEN { 文 行番号 } } { ELSE { 文 行番号 } } GOTO行番号	論理式の結果により、実行文または分岐先を選択する		G		7・22
INKEY\$	INKEY\$ [#ポート番号]	押されているキーのコードを得る			S	7・59
INPUT	INPUT [#ポート番号,] ["プロンプト文",] 変数名	キー入力を指定した変数に読込む		G		7・23
INPUT\$	INPUT\$ (文字数 [, # ポート番号])	キーボードより指定された文字数の文字列を読込む			S	7・59
INSTR	INSTR ([検索開始位置,] 文字式1, 文字式2)	文字列データから文字式2を探しその位置を得る			C	7・55
INT	INT (数値式)	小数点以下を切捨てた整数値を得る			A	7・53
KEY ON/OFF /STOP	KEY [キー番号] $\chi\chi$	キー入力による割込みの許可、禁止、停止を制御する		G		7・23
LEFT\$	LEFT\$ (文字式, 文字数)	文字列の左側から文字数分の文字列を得る			C	7・55
LEN	LEN (文字式)	文字列の文字数を得る			C	7・56
LET	[LET] 変数名=式	式の結果を変数に代入する		G		7・24
LINE INPUT	LINE INPUT [#ポート番号,] ["プロンプト文" ;] 文字変数	キーボードから1行分文字変数に読込む		G		7・24
LIST/LLIST	LIST/LLIST [行番号1] [-行番号2]	プログラムを表示またはプリンタに出力する	○			7・13
LOAD	LOAD	EEP-ROMあるいはポートからプログラムをロードする	○			7・14
LOC	LOC (ポート番号)	ポートにたまっているデータの数を与える			S	7・59
LOG	LOG (数値式)	自然対数を与える			A	7・54
MID\$	MID\$ (文字式, 文字位置 [, 文字数])	指定の文字位置から文字数分取出す		G	C	7・24 7・56

命 令	記 述	機 能	マ ジ ク	ス テ ー ト メント	関 数	ページ
MON	MON	モニタモードに入る	○			7・14
MSET	MSET (番地)	メモリの下限の番地を設定する	○			7・15
NEW	NEW	現在利用のプログラム、変数をすべてクリアする	○			7・16
OCT\$	OCT\$ (数値式)	10進数値を8進数の文字列に変換する			C	7・56
ON COM GOSUB	ON COM [ポート番号] GOSUB 行番号	通信回線からの割込みのサブルーチン分岐先を定義する		G		7・25
ON ERROR GOTO	ON ERROR GOTO 行番号	エラー発生時指定行番号へ分岐する		G		7・25
ON式 GOSUB GOTO	ON式GOSUB 行番号1 GOTO [, 行番号2...]	式の値によって指定された行番号へ分岐する		G		7・26
ON KEY GOTO ON KEY GOSUB	ON KEY [キ-番号] {GOSUB GOTO } 行番号	キー入力による割込みの (サブルーチン) 分岐先を定義する		G		7・26 7・27
ON PC GOSUB	ON PC [割込番号] GOSUB 行番号	PCからの割込みのサブルーチン分岐先を定義する		G		7・30
OPEN	省略	ポートを開く		D		7・49
PC GET	PC GET 変数名1 [, 変 数名2]	PCの出力データの値を変数に読み込む		G		7・30
PC ON/OFF/ STOP	PC [割込番号] X X	PCからの割込みの許可、禁止、停止を制御する		G		7・31
PC PUT	PC PUT数値式	数値式の値をPCの入力データエリアに書き込む		G		7・31
PC READ ("@")	PC READ "フォーマット";変数...	PCからの書き込みデータをフォーマット変換し読み込む		G		7・32
PC WRITE ("@")	PC WRITE "フォーマット";数値式 ...	数値式の値をフォーマット変換しPCへ転送する		G		7・32
PEEK	PEEK (番地)	メモリ上の指定された番地の内容を读出す			S	7・60
PGEN	PGEN数値式	利用するプログラムエリアに切替える	○			7・16

命 令	記 述	機 能	マ ボ	ス ト メント	関 数	ページ
PINF	PINF [{ALL}] 式	プログラムエリアの使用状況を画面に出力する	○			7・16
PNAME	PNAME “プログラム名”	プログラムに名前をつける、または消す	○			7・16
POKE	POKE 番地 数値	メモリ上の指定番地へデータを書込む		G		7・44 4
PRINT/ LPRINT	PRINT/LPRINT [#ポート 番号,] [式…]	式の値を画面またはプリンタに出力する		G		7・44
PRINT USING LPRINT USING	省略	指定フォーマットで文字列、数値を画面またプリンタに出力する		G		7・45
RANDOM	RANDOM (式)	式で与えた乱数の元値により乱数の系列変更する		G		7・45
READ	READ 変数名 [, 変 数名…]	DATA文で定義のデータを変数に読み込む		G		7・45
REM	REM [コメント文]	プログラム中にコメント文を入れる		G		7・46
RENUM	RENUM [新行番号] [,旧行番号] [,増分値]	プログラムの行番号を付直す	○			7・17
RESTORE	RESTORE [行番号]	READ文で読始めるDATA文の行番号を指定する		G		7・46
RESUME	RESUME [{NEXT} 行番号]	エラー処理終了後、再開行を指定する		G		7・46
RIGHT\$	RIGHT\$ (文字式 文 字数)	文字列の右側から文字数分の文字列を得る			C	7・56
RND	RND [(数値式)]	0と1の間の乱数を得る			A	7・54
RUN	RUN [行番号]	プログラムの実行を始める	○			7・17
SAVE	SAVE	プログラムをEEP-ROMまたはポートにセーブする	○			7・17
SGN	SGN (数値式)	引数の符号を得る			A	7・54
SIN	SIN (数値式)	引数で与えられる正弦 (SIN) を得る			A	7・54
SPACE\$	SPACE\$ (文字数)	文字数分の空白文字列を得る			C	7・56
STOP	STOP	プログラムを実行停止する		G		7・46

命 令	記 述	機 能	マ ボ	ス テ ー ト	関 数	ペ ー ジ
STR\$	STR\$ (数値式)	数値を文字列に変換する			C	7・57
STRING\$	STRING\$ (文字数 {キャラコ ード, 文字式})	文字数文の指定文字列を得る			C	7・57
TAB	TAB (桁位置)	指定桁位置まで空白を出力する			C	7・57
TAN	TAN (数値式)	引数で与えられた数値の正接 (TAN) を得る			A	7・54
TIME\$	TIME\$	時刻をセットする、または与える			S	7・60
TRON/TROFF	TRON/TROFF	プログラムのトレースを指示、また は解除する	○			7・18
USR	USR番号 (引数)	DEF USR文で定義した機械語 関数ルーチンを呼出す			S	7・60
VAL	VAL (文字式)	文字列が表す数値に変換する			C	7・57
VERIFY	VERIFY	プログラムをEEP-ROMと照合 する	○			7・18
VARPTR	VARPTR (変数名)	変数の格納されているメモリ番地を 与える			S	7・62
WAIT	WAIT “待ち時間” [, 行番号]	次の命令語の実行終了までの時間を 設定する		G		7・47

7-3 BASIC命令の説明

命令の構成の項で分類した<コマンド>、<ステートメント>（一般／デバイスコントロール）、<関数>（算術演算／文字列／特殊）の順に各命令の詳細を説明します。
なお各分類中の命令はアルファベット順に記載してあります。

コマンド

■AUTO [行番号] [, 増分値]

【機能】 行の先頭に番号を自動的に発生させる。

【記述例】
AUTO
AUTO 100
AUTO 200, 10

【説明】

- ・行番号を最初の番号とし、増分の値ずつ増加してゆく行番号を自動的に発生し、AUTOモードになります。
- ・行番号は、0～63999、増分の値は1～63999で省略時はともに10と解釈されます。
- ・「CTRL」+「X」キーでAUTOモードは解除されます。
- ・プログラムとしてすでに存在する番号を指定した場合は、行番号の直後に*が表示されます。このとき、文字を入力して「CR」キーを押すと新しい内容に置換されます。また「CR」キーのみ押したときは内容はもとのままです。
- ・入出力ポートとしてポート1が選択されます。

■CONT

【機能】 実行停止しているプログラムの実行再開をする。

【記述例】 CONT

【説明】

- ・プログラムの実行中の「CTRL」+「X」キー、またはプログラム中のSTOP文の実行でプログラム実行が停止します。このとき、CONTコマンド入力によって停止プログラムの続行ができます。
- ・入出力装置への入出力中の「CTRL」+「X」キーは、実行打ち切りとなりCONTコマンドで実行再開はできません。
- ・実行停止後にプログラム内容を変更した場合にも、実行再開はできません。

■DEL [行番号1] [- [行番号2]]

【機能】 プログラム行の一部を削除する。

【記述例】
DEL 100
DEL 100-
DEL -150
DEL 100-150

【説明】

- ・行番号1だけの指定時は、その行だけを削除します。
- ・行番号1-の指定時、その行以降を削除します。
- ・-行番号2の指定時は、先頭よりその行までを削除します。
- ・行番号1-行番号2の指定時は、その範囲行を削除します。

■EDIT [行番号]

【機能】 プログラムを1行編集する。

【記述例】 EDIT 100
EDIT

【説明】

- ・指定行番号のプログラムの修正および編集を行います。
- ・指定行番号のプログラムの範囲内で、文字挿入・削除およびカーソルの移動ができます。
- ・表示されているプログラムの行番号を変更して「CR」キーをキーインすると、その行番号のプログラムが新たに作成されます。
- ・EDIT. の場合は最も新しく表示した行、または最も新しく入力した行のプログラムが表示されます。
- ・入出力ポートとしてポート1が選択されます。

■LIST [行番号1] [- [行番号2]] LLIST

【機能】 プログラムの全部または一部を、表示またはプリンタに出力する。

【記述例】 LIST
LLIST
LLIST 100-200
LIST -100
LIST.

【説明】

- ・行番号1だけの指定時は、その行だけを表示またはプリンタに出力します。
- ・行番号1-の指定時は、その行以降を表示またはプリンタに出力します。
- ・-行番号2の指定時は、先頭よりその行までを表示またはプリンタに出力します。
- ・行番号1-行番号2の指定時は、その範囲行を表示またはプリンタに出力します。
- ・出力の中断は「CTRL」+「B」、「CTRL」+「X」キーにて可能です。
「CTRL」+「B」キーの場合は次の「CTRL」+「B」キーのキーインで継続します。
- ・LIST. の場合は、最も新しく表示した行、または最も新しく入力した行番号を表示します。
- ・LISTコマンドではポート1を、LLISTコマンドではポート2を自動的に選択します。
- ・LLISTコマンドではOPEN文に関係なく、周辺装置名LPRTの条件で出力されます。

《参考》

「CTRL」+「B」および「CTRL」+「X」は、「CTRL」キーを押しながら、「B」または「X」キーを押すことを示します。

■LOAD

【機能】 全プログラムエリアにEEP-ROMからロードする。

【記述例】 LOAD

【説明】 ・BASICプログラムエリアおよびMSETコマンドにより確保された機械語プログラムエリアの内容がEEP-ROMからロードされます。

■LOAD #ポート番号, “COMU: [(通信条件・有効信号線)]”

【機能】 RS-232C機器から送られるASCIIコードのプログラムを現在利用しているプログラムエリアにロードする。

【記述例】 LOAD #1, “COMU: (43)”

【説明】 ・本コマンドが起動されると、「BASIC」LEDの点滅が速くなるますので、指定入力ポートに接続したRS-232C機器の送信準備操作を行ってください。
送信準備ができたなら、START/STOPスイッチを押して送信を開始してください。
・LOAD実行中にSTART/STOPのスイッチ入力は処理中断になります。
・通信条件および、有効信号線およびCOMUは次ページ“OPEN”コマンドを参照してください。
・LOAD命令実行直後にプログラムエリアはクリアされます。

■MON

【機能】 モニタモードに入ります。

【記述例】 MON

【説明】 ・BASICモードからモニタモードに制御を移します。
・モニタモードについての詳細は8・2ページを参照してください。
・「CTRL」+「B」キーでBASICモードにもどります。

《参考》

「CTRL」+「B」は、「CTRL」キーを押しながら「B」キーを押すことを示します。

■MSET [メモリの下限番地]

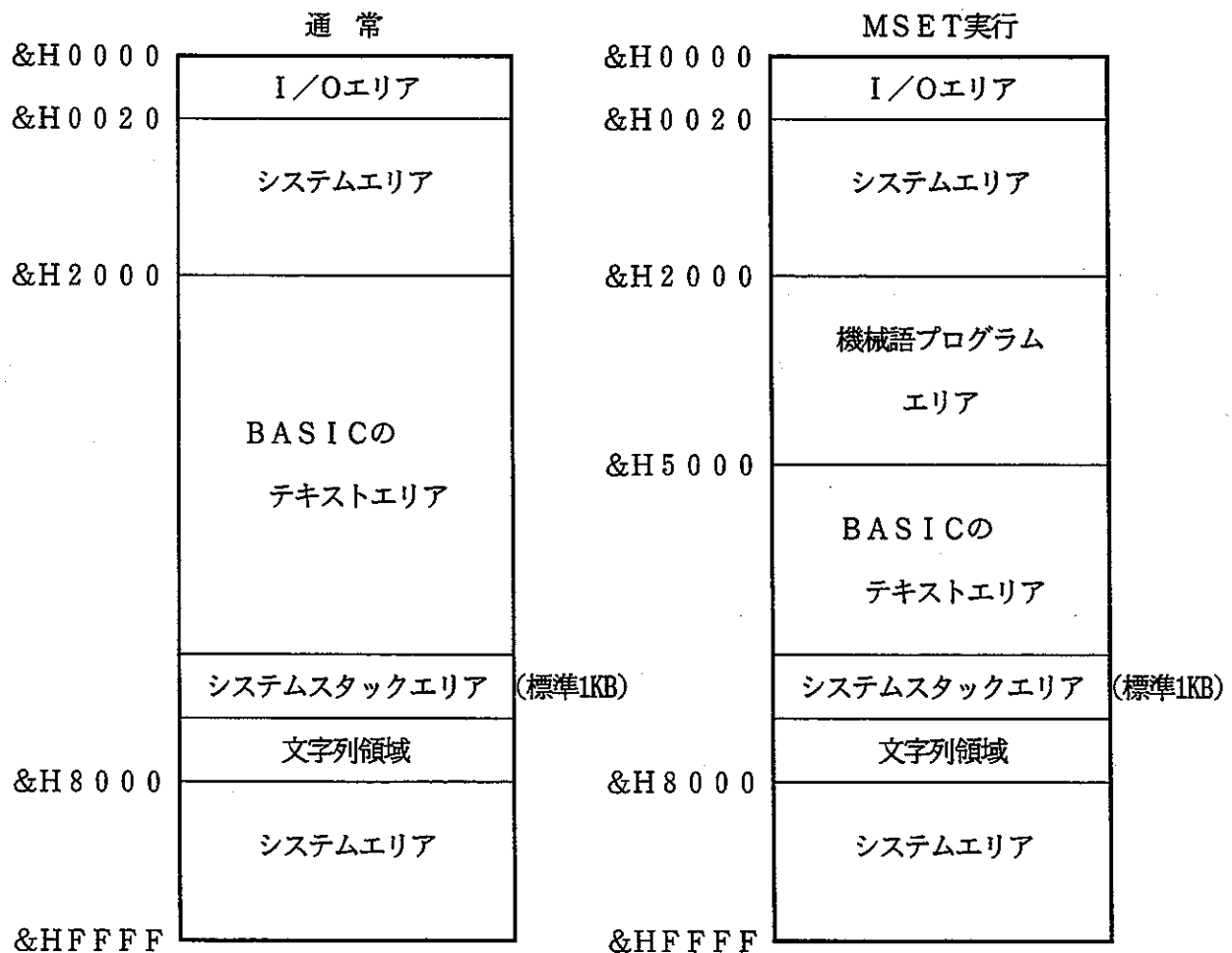
【機能】 メモリの下限を設定する。

【記述例】 MSET &H5000

【説明】

- ・機械語で書かれたプログラムはBASICプログラムテキストエリアの前に配置されるためBASICで使用するメモリの下限を設定し、機械語プログラムエリアを確保する必要があります。
- ・メモリ下限番地を省略した場合は、&H2000に設定されます。
- ・メモリ下限番地は、&H2000から&H7FFFまでの間で設定してください。ただし、&H7FFFより前には、文字列領域、システムスタック等がありますので、これより上に設定してください。
- ・MSETを実行すると、電源をOFF、ONしても状態が保持されますので、通常の状態に戻すときはMSET&H2000を実行してください。

例) MSET &H5000



■NEW

【機能】 現在利用しているエリアのプログラムを消去し、変数をすべてクリアする。

【記述例】 NEW

【説明】

- ・この時点で開いているすべてのポートは閉じられます。
- ・PNAME文によってプログラムに名前が付いている場合は消去できませんので、PNAME “ ” にて名前を消してからNEWコマンドを実行してください。

■PGEN 数値式

【機能】 利用するプログラムエリアに切替える。

【記述例】 PGEN 2

【説明】

- ・数値式は1～3が使用できます。
- ・切替えられたプログラムエリアの使用情報（PINF命令を参照してください）を出力します。

■PINF { {ALL} } 式

【機能】 プログラムエリアの使用状況を画面に出力する。

【記述例】 PINF ALL
PINF 2

【説明】

- ・式の値で示されるプログラムエリアについて、PNAME文で付けられたプログラム名と使用容量を表示します。式の値は1、2または3としてください。
- ・ALLまたは式が省略されたときは、現在使用しているプログラムエリアについて上記情報を出力します。
- ・ALLと指定した場合は、すべてのプログラムエリアについてのプログラム名、使用容量とプログラムの未使用エリア容量を出力します。
- ・出力ポートとして、ポート1が選択されます。

■PNAME “プログラム名”

【機能】 PGENしているエリアのプログラムに名前をつける、または名前を消す。

【記述例】 PNAME “TESTPROG”
PNAME “ ”

【説明】

- ・プログラム名は8文字以内です。
- ・PNAME文で名前をつけられたプログラムエリアは、NEWコマンドから保護されます。NEWコマンドを実行する場合はPNAME “ ” を実行してプログラム名を消しておく必要があります。

■RENUM [新行番号] [, [旧行番号] [, 増分値]]

【機能】 プログラムの行番号を付直す。

【記述例】 RENUM 100, 10, 50

【説明】

- ・新行番号は、新しく付ける最初の番号で省略時は10です。
- ・旧行番号は、付替えようとするプログラムの行番号で、省略時はプログラムの先頭の行番号となります。
- ・増分値は、付直すときの増分量で省略時は10です。
- ・GOTO、GOSUB、THENなどで参照している行番号も対応した新しい行番号に変更されます。
- ・64000以上の行番号は付けられません。

■RUN [行番号]

【機能】 プログラムの実行を始める。

【記述例】 RUN 50

【説明】

- ・行番号を指定したときは、その行番号より実行を始めます。
- ・指定のないときは、プログラムの先頭の行番号より実行を始めます。
- ・RUNコマンドでは、プログラム実行前にすべての変数をクリアし、開いているポートがあれば閉じます。
- ・プログラム実行は「CTRL」+「X」キー、START/STOPスイッチ、END文、STOP文により終了します。

■SAVE

【機能】 EEPROMへ全プログラムをセーブする。

【記述例】 SAVE

【説明】

- ・BASICプログラムエリアおよびMSETコマンドにより確保された機械語プログラムエリアの内容がEEPROMへセーブされます。
- ・EEPROMのため、完了まで若干時間がかかります。

■SAVE #ポート番号, "COMU: [(通信条件・有効信号線)]"

【機能】 現在利用しているエリアのプログラムをRS-232C機器へASCIIコードでセーブします。

【記述例】 SAVE #1, "COMU: (43)"

【説明】

- ・本コマンドが起動されると「BASIC」LEDの点滅が速くなりますので、指定出力ポートに接続したRS-232C機器の受信準備操作を行ってください。
- ・受信準備ができたなら、START/STOPのスイッチを押してください。
- ・セーブ実行中にSTART/STOPのスイッチ入力処理は中断になります。
- ・通信条件および有効信号線およびCOMUは「OPEN」コマンドを参照してください。

《参考》

「CTRL」+「X」は、「CTRL」キーを押しながら「X」キーを押すことを示します。

■TRON/TROFF

【機能】 プログラムのトレースを指示したり、解除する。

【記述例】 TRON
TROFF

【説明】

- ・TRONを実行すると、その後実行するプログラムはすべて文番号を〔10〕〔20〕〔30〕のように表示しながら、進行します。
- ・TROFFを実行すると、トレースを解除します。
- ・NEW、電源OFF、RESETでもトレースを解除します。
- ・出力ポートとしてポート1が選択されます。

ただし、ポート1にターミナル以外が接続されているときは、トレースの表示はされません。

■VERIFY

【機能】 全プログラムエリアの内容とEEP-ROMの内容を照合する。

【記述例】 VERIFY

【説明】

- ・BASICプログラムエリアおよびMSETコマンドにより確保された機械語プログラムエリアの内容について照合します。
- ・一致している場合“READY”を表示します。
- ・一致していない場合“VERIFY ERROR”を表示します。

ステートメント

<一般ステートメント>

■CLEAR [文字領域の大きさ]

【機能】 数値変換、文字変数の初期化および文字領域の大きさを設定する。

【記述例】 CLEAR 1000

【説明】

- ・数値変数を0に、文字変数を空文字列に初期化し、DEF文で定義された情報は無効にします。また、開いているすべてのポートを閉じます。
- ・ON ERROR GOTO文より前に実行させてください。
- ・文字領域の大きさは、BASICが文字列処理のために用いるメモリ領域の大きさにバイト数で設定します。
- ・電源投入またはリスタート時は200バイトに設定されます。

■COM [ポート番号] ON/OFF/STOP

【機能】 ON COM GOSUB文で定義された通信回線からの割込みの許可、禁止、停止を制御する。

【記述例】 COM1 ON

【説明】

- ・COM ON
指定ポートの通信回線からの割込みを許可します。
この命令実行後は、指定ポートに通信が入るごとに割込みを発生し、ON COM GOSUB文で定義されている処理ルーチンに分岐します。
- ・COM OFF
指定ポートの通信回線からの割込みを禁止します。
この命令実行後は、指定ポートに通信が入っても処理ルーチンへの分岐は起こりません。
- ・COM STOP
指定ポートの通信回線からの割込みを停止します。
この命令実行後は、指定ポートに通信が入ってもそのことを覚えておくだけで、処理ルーチンへの分岐は起こりません。しかし、その後COM ONによって割込みが許可されると、先ほどの通信による処理ルーチンに分岐します。
- ・ポート番号を省略すると、ポート1が選択されます。
- ・プログラム終了時には、COM OFFの状態にしてください。
- ・割込処理ルーチンではすべての割込みについて、割込許可、停止状態のものは割込停止状態に、割込禁止状態のものは割込禁止状態になります。
- ・ON COM GOSUB文を実行後に有効となります。
- ・使用例

```

10 OPEN #2, "COMU:"
20 ON COM2 GOSUB 100
30 COM2 ON
40 GOTO 40
100 IF LOC(2) <> 0 THEN A$=INPUT$(LOC(2), #
    2)
110 RETURN

```

■DATA 定数 [, 定数] ...

【機能】 READ文に与える数値および文字数を定義する。

【記述例】 DATA CF, 10, 2.5, "A. : B"

【説明】

- DATA文はデータ定義文で、プログラム中のどこにでも、また複数個を書くことができます。
- 定数には、数値定数と文字定数が許され混在可能ですが、READ文で指定される変数の型の順序と一致している必要があります。
- 文字定数中にカンマ、コロン、意味のある空白を含める場合には、引用符 (") で囲みます。
- DATA文が複数個ある場合は、それらを一連のデータとみなして順番にREAD文が読んでいきますが、RESTORE文を実行した場合は、最初のDATA文より再び読み込むことになります。
- DATA文の後には、コメントはつけられません。

■DEF FN名前 [(引数 [, 引数])] =式

【機能】 ユーザ作成の関数を定義し、それに名前を付ける。

【記述例】 DEF FNA (X, Y, Z) = X * 3 + Y * 2 + Z

【説明】

- 名前は数値変数名または文字変数名で、FN+名前が関数名となります。
- 引数が変数名の並びで、複数の場合はコンマで区切ります。
- 定義した引数の型と、実際に引用して使用する場合の引数の型は一致していなければいけません。
- DEF FN文は、それが引用される前に書かれていなければいけません。
- 引数として書かれた変数名と同じ変数名がプログラム中の他の行で使われていてもかまいません。それらは異なる変数として扱われます。
- ダイレクト実行はできません。
- ユーザ関数の二重定義はできません。

■DEF {INT, SNG, DBL, STR} 文字の範囲 [, 文字の範囲] ...

【機能】 変数の型を整数、単精度、倍精度、文字に宣言する。

【記述例】 DEFSTR A, B-D

【説明】

- 文字の範囲で指定される文字で始まる変数名の型指定をします。
 DEFINTは整数型
 DEFSNG単精度実数型
 DEFDBL倍精度実数型
 DEFSTR文字型
 } に型宣言されます。
- 上記型宣言よりも型宣言文字の方が優先されます。
- 型宣言がされていない変数は、単精度変数として扱われます。

DEF USR [番号] =開始番地

【機能】 機械語で作られたユーザ関数の実行開始番地を定義する。

【記述例】 DEF USR3=&H2000

【説明】

- ・USR関数が呼出す機械語ルーチンの実行開始番地を設定します。
- ・番号は0～9までの値で、最大10個のUSR関数を設定することができます。
- ・番号が省略された場合は0と解釈されます。
- ・使用例


```
100 DEF USR1=&H2100
110 POKE &H2100, &H39
120 A=USR1(A)
130 PRINT A
```
- ・参照 USR、MON、機械語プログラムの作り方、モニタモード

DIM 変数名 (添字最大値 [添字最大値] ...) [, 変数名 (...)]

【機能】 配列変数の次元数と添字の最大値を設定し、メモリ上にその配列変数の領域を割当てる。

【記述例】 DIM A(10, 20), B\$(30)

【説明】

- ・添字として指定できる最小値は0ですが、配列の宣言をせずに配列変数を使った場合の、添字の最大値は10となります。
- ・配列変数の宣言をすると、数値型は0に、文字型は空文字列にそれぞれ初期化されます。
- ・この宣言は電源OFF時保持されません。

END

【機能】 プログラムの実行を終了し、全てのポートを閉じてコマンド待ち状態になる。

【記述例】 END

【説明】

- ・プログラムの最後のEND文は省略できます。この場合、ポートは閉じません。

ERROR エラー番号

【機能】 エラー発生をシュミレートしたり、エラー番号の定義をする。

【記述例】 ERROR 100

【説明】

- ・エラー番号は、1～255の範囲内の数値が使えます。
- ・ON ERROR GOTO文のないプログラムにおいて、ERROR文が実行されると、そのエラー番号に対応するエラーメッセージを出力して実行停止します。
- ・未定義のエラー番号の場合には“UNDEFIND ERROR”メッセージを出力します。
- ・ERR変数にはエラー番号、ERL変数にはERROR文のあった行番号が入っています。
- ・未定義のエラー番号をユーザー独自のエラー番号として使えます。このエラー番号は、ON ERROR GOTO文によるエラー処理プログラムにて使用できます。

■FOR 変数名=初期値 TO 終値 [STEP 増分値] ~NEXT 変数名

【機能】 FORとNEXTで囲まれた一連の文を、変数が初期値から終値に達するまで繰返し実行する。

【記述例】

```
FOR I=0 TO 10 STEP 2
{
NEXT I
```

【説明】

- ・変数はFORループカウンタとして使われ、まず初期値に設定します。
- ・NEXTまでの一連の文が実行された後に、FORループカウンタがSTEPで指定された増分値だけ増加します。
- ・FORループカウンタが終値に達していなければ、FOR文の次の一連の文を再度実行します。一方、終値に達していれば、NEXTの次の文に実行が移ります。
- ・STEPの項を省略したとき、増分値は1とみなします。
- ・増分値>0で初期値>終値または、増分値<0で初期値<終値の場合には、NEXT文の一連の文は1回も実行されずにNEXT文の次へ実行が移ります。
- ・FORループのネスティングができますが、NEXT文が同じ所で終る場合には、NEXT文に変数名をカンマで区切って複数個を書くことができます。

■GOSUB 行番号~RETURN

【機能】 行番号で指定したサブルーチンと呼出して実行し、RETURN文の実行でGOSUB文の次の文に戻る。

【記述例】

```
10 GOSUB 100
{
100 PRINT
{
150 RETURN
```

【説明】

- ・行番号は呼出すサブルーチンプログラムの入口の行番号を指定する必要があります。
- ・サブルーチンプログラムは、RETURN文で終っている必要があります。
- ・指定された行番号が非実行文のときは、その次の文より実行します。

■GOTO 行番号

【機能】 無条件に指定された行番号に分岐する。

【記述例】 GOTO 100

【説明】 ・指定された行番号が非実行文のときは、その次の文より実行します。

■IF 論理式 { THEN { 文 } } { GOTO 行番号 } { ELSE { 文 } }

【機能】 論理式の結果により、実行文または分岐先を選択する。

【記述例】 IF A>=0 THEN A=10 ELSE 100

【説明】

- ・論理式の値が真（値が0以外）ならば、THENまたはGOTO文以降が実行されます。
- ・論理式の値が偽（値が0）ならば、THENまたはGOTO文は無視して、ELSE文以降を実行します。ELSE文が省略されている場合は、次の文を実行します。
- ・IF文のネスティングも可能ですが、1行に書ける範囲内に限られます。

■ INPUT [#ポート番号,] { “プロンプト文” { ; } } 変数名 [, 変数名…]

【機能】 キー入力を指定した変数に読込む。

【記述例】 INPUT “DATA” ; A\$
INPUT #2, “DATA”, A\$, B\$

【説明】

- ・プロンプト文の次が;の場合は、プロンプト文に続いて?が表示され、コンマ(,)の場合はプロンプト文の後には何も表示されません。
また、プロンプト文が省略されたときは?のみが表示されます。
- ・OPEN文でTERM、COMU以外の周辺装置を選択した場合は、プロンプト文や、?は表示されません。
- ・OPEN文でCOMUを選択した場合などで、?を出力したくないときはLINE INPUT命令を使用してください。
- ・ポート入力データは「CR」キー入力で変数に取込まれます。「CR」キー入力までは、BASIC LEDが点滅して入力待ち状態であることを示します。
- ・複数の変数名を書いた場合は、データをコンマ(,)で区切って入力してください。
このとき、変数名の型と入力データの型、および個数が一致している必要があります。
- ・INPUT文のダイレクト実行はできません。
- ・ポート番号の項を省略したときは、ポート1を選択されたものとみなします。

■ KEY [キー番号] ON/OFF/STOP

【機能】 ON KEY GOTO文、ON KEY GOSUB文で定義されたキー入力による割込みの許可、禁止、停止を制御する。

【記述例】 ・KEY ON

【説明】

- ・KEY ON
キー入力による割込みを許可します。
この命令実行後は、キー入力による割込みが入るごとに割込みを発生し、ON KEY GOTO文、ON KEY GOSUB文で定義されている処理ルーチンに分岐します。
- ・KEY OFF
キー入力による割込みを禁止します。
この命令実行後は、キー入力による割込みが入っても処理ルーチンへの分岐は起こりません。
- ・KEY STOP
KEY 割込みを停止します。
この命令実行後は、キー入力による割込みが入っても、そのことを覚えておくだけで、処理ルーチンへの分岐は起こりません。しかし、その後KEY ONによって割込みが許可されると、先ほどのキー入力による割込みによる処理ルーチンに分岐します。
- ・キ番号を省略したときは、キ番号の1～8のすべてについて命令を実行したのと同じに扱ふ。
- ・プログラム終了時には、KEY OFFの状態にしてください。
- ・割込処理ルーチンではすべての割込みについて、割込許可、停止状態のものは割込停止状態に、割込禁止状態のものは割込禁止状態になります。
- ・使用例

```

10 OPEN #1, "TERM: (4 2)"
20 ON KEY 1 GOSUB 100
30 ON KEY 2 GOSUB 200
40 A=0
50 KEY ON
60 GOTO 60
100 PC READ "@D, 0, 1, I 4" ; A
110 RETURN
200 PC WRITE "@D, 0, 1, I 4" ; A
210 RETURN

```

■ [LET] 変数名=式

【機能】 右辺の式の結果を、変数に代入する。

【記述例】 LET A=1.2

【説明】

- LETは省略できます。
- 数値変数=数値式、文字変数=文字式以外は許されません。
- 数値式の場合は、左辺の数値変数の型に合わせて代入します。
- コンマ、コロン、空白をデータとして入力する場合は、文字列全体を“ ”で囲んでください。

■ LINE INPUT [#ポート番号,] ["プロンプト文";] 文字変数

【機能】 キーボードから、1行全体の文字列(255字以内)を文字変数に読込む。

【記述例】 LINE INPUT #2, "DATE"; A\$

【説明】

- ポートより入力されたデータをすべて一連の文字データとして、文字変数に代入します。(カンマ、コロンも文字データとして扱われます。)
- データ入力を促す?は表示されません。
- OPEN文でTERM、COMU以外の周辺装置を選択した場合は、プロンプト文は表示されません。
- ポート入力データは「CR」キー入力で変数に取込まれます。「CR」キー入力までは、BASIC LEDが点滅して入力待ち状態であることを示します。
- ポート番号の項を省略したときは、ポート1を選択されたものとみなします。

■ MID\$ (文字変数名, 文字位置 [, 文字数]) =文字式

【機能】 文字変数の指定した文字位置から文字数分の文字列を文字式の最初から指定した文字数分の文字列で置換える。

【記述例】 MID\$(A\$, 2, 4) = "ABCDEFGH"

【説明】

- 文字数を省略した場合は、文字式のすべての文字と置替わります。
- 文字変数名の文字数を超える文字位置は指定できません。
- A\$ = "123456789" だった場合、A\$の内容は"1ABCD6789"になります。
- 参照 MID\$ (関数) 7・54ページ

■ON COM [ポート番号] GOSUB 行番号

【機能】 ポート番号で指定された通信回線からの割込みによるサブルーチン分岐先を定義する。

【記述例】 ON COM1 GOSUB 1000

【説明】

- ・ポートをOPENした後に実行させないと有効になりません。
- ・ON COM GOSUB文によるサブルーチンプログラムを実行中（ON COM GOSUB文に対応するRETURNを実行するまで）は、ON KEY文、ON PC文、ON COM文によるサブルーチン分岐は許可状態であっても停止されています。
- ・サブルーチンプログラムはRETURN文で終わっている必要があります。
- ・指定された行番号が非実行文のときは、その次の実行文より実行します。
- ・行番号に0を指定した場合は、COM OFF命令を実行したのと同じになります。
- ・ポート番号を省略すると、ポート1が選択されます。
- ・機械語プログラム実行中は、ON COM命令は実行しません。
- ・COM ON命令によって有効になります。
- ・COM OFF命令によって無効になります。
- ・COM STOP命令によって通信回線からの割込みを覚えておきます。
- ・ON COM GOSUB命令実行時には、COM OFF（割込禁止）状態になっています。

・使用例

```

10 OPEN #1, "COMU: (40)"
20 ON COM GOSUB 100
30 COM ON
40 PC READ "@D, 0, 2, 2 I 4" ; A, B
50 PRINT A, B
60 GOTO 30
100 IF LOC(1) = 0 THEN 120
110 PRINT INPUT$(LOC(1), #1)
120 RETURN

```

ポート1の通信回線からの割込みが入るとその内容を表示します。

・参照 EOF, LOC

《お願い》

・割込みルーチン処理中に新たに通信回線からの割込みが入った場合、RETURN文を実行後再度割込みルーチンに分岐します。そのため割込みルーチンに分岐したときはタイミングによってはバッファに何も入っていないことがありますので、割込みルーチン先頭でLOC、EOF命令にてバッファにデータが入っているかを（上記使用例では100行で行っています）確認してください。

・割込みルーチン処理中に受信バッファがオーバーフローするようなことがないように、割込みルーチン処理内容は通信速度と処理時間の関係を検討のうえご使用ください。

■ON ERROR GOTO 行番号

【機能】 エラー発生時に、エラー処理ルーチンへ行くようにする。

【記述例】 ON ERROR GOTO 100

【説明】

- ・この文を実行しておく、エラーが発生したときにエラーメッセージを出力せずに指定された行番号のエラー処理ルーチンへ実行が移ります。
- ・エラーが発生したとき、エラー番号とエラー発生行番号が変数ERRとERLに入っています。
- ・エラー処理ルーチン内でエラーが発生した場合には、エラーメッセージが出力され実行は停止します。
- ・ON ERROR GOTO 0を実行すると、この機能は無効になります。
- ・CLEARコマンドより後で実行するようにしてください。
- ・参照 RESUME, ERR, ERL

■ON 式 GOSUB 行番号1 [, 行番号2...] GOTO

【機能】 式の値によって、指定された行番号の1つに分岐する。

【記述例】 `ON A GOSUB 100, 200`
`ON B GOTO 500, 600`

【説明】 ・式の値が1のときは行番号100へ、式の値が2のときは行番号200へ、というように式の値によって分岐または、サブルーチン呼出しをします。
・式の値は0～255が許されますが、式の値が0、または行番号の個数よりも大きいときは、GOSUB、GOTO文は無視して次の文を実行します。

■ON KEY [キー番号] GOSUB 行番号

【機能】 キー番号で指定されたキー入力による割込みのサブルーチン分岐先を定義する。

【記述例】 `ON KEY 1 GOSUB 1000`

【説明】 ・キー番号は1～8が有効です。キー番号を省略した場合は、1～8のキーすべてについて、分岐するサブルーチンを同じにしたことになります。
・サブルーチン実行後は、キーが押されたときの処理に戻ります。
・ON KEY GOSUB文によるサブルーチンプログラムを実行中（ON KEY GOSUB文に対応するRETURNを実行するまで）は、ON PC文、ON COM文、ON KEY文によるサブルーチン分岐は許可状態であっても停止されています。
・サブルーチンプログラムはRETURN文で終わっている必要があります。
・指定された行番号が非実行文のときは、その次の文より実行します。
・行番号に0を指定した場合は、KEY OFF命令を実行したのと同じになります。
・ON KEY GOTO文で指定したキー番号は指定しないでください。
・本命令語はターミナル接続時のみ有効になります。
・機械語プログラム実行中は、ON KEY GOSUB文は実行しません。
・KEY ON命令によって有効になります。
・KEY OFF命令によって無効になります。
・KEY STOP命令によってKEY割込みを覚えておきます。
・ON KEY GOSUB命令実行時には、KEY OFF（割込禁止）状態になっています。

・使用例

```

10 OPEN #1, "TERM: (42)"
20 ON KEY 1 GOSUB 1000
30 ON KEY 2 GOSUB 2000
40 ON KEY 3 GOSUB 3000
50 KEY ON
    }
1000 PRINT A
    }
1100 RETURN
2000 PRINT B
    }
2100 RETURN
3000 PRINT C
    }
3100 RETURN

```

1、2、3のキーインによりそれぞれ画面に、変数A、B、Cの値を表示します。

ON KEY [キー番号] GOTO 行番号

【機能】 キー番号で指定されたキー入力による割込みの分岐先を定義する。

【記述例】 ON KEY 1 GOTO 1000

【説明】

- ・キー番号は1～8が有効です。キー番号を省略した場合は1～8のキーすべてについて、分岐先を同じにしたことになります。
- ・行番号に0を指定した場合は、KEY OFF命令を実行したのと同じになります。
- ・ON KEY GOSUB文で指定したキー番号は指定しないでください。
- ・本命令語はターミナル接続時のみ有効になります。
- ・KEY ON命令によって有効になります。
- ・KEY OFF命令によって無効になります。
- ・KEY STOP命令によってKEY割込みを覚えておきます。
- ・ON KEY GOTO命令実行時には、KEY OFF (割込禁止) 状態になっています。
- ・使用例

```

10 OPEN #1, "TERM: (42)"
20 ON KEY 1 GOTO 1000
30 ON KEY 2 GOTO 2000
40 ON KEY 3 GOTO 3000
50 KEY ON
    {
1000 PRINT "A"
    {
2000 PRINT "B"
    {
3000 PRINT "C"
    {
1、2、3のキーインによりそれぞれ画面に“A”、“B”、“C”を表示します。

```

■ON PC [割込番号] GOSUB 行番号

【機能】 割込番号で指定されたPCからの割込みのサブルーチン分岐先を定義する。

【記述例】 ON PC 3 GOSUB 1000

【説明】

- ・割込番号は1～15が許されます。
- ・この命令によって割込番号が定義されていない割込要因は無視されます。
- ・割込番号は、ユニット割付先頭チャンネル (nCH) のビット04～07 (1～F (16進データ)) で示されます。
- ・ON PC文によるサブルーチン・プログラムを実行中 (ON PC文に対応するRETURNを実行するまで) は、ON KEY文、ON COM文、ON PC文によるサブルーチン分岐は許可状態であっても停止されています。
- ・サブルーチン・プログラムはRETURN文で終わっている必要があります。
- ・指定された行番号が非実行文のときは、その次の実行文より実行します。
- ・行番号に0を指定すると、PC OFF命令を実行したのと同じになります。
- ・割込番号を省略したときは、割込番号を1～15のすべてについて、分岐先を同じにしたことになります。
- ・機械語プログラム実行中は、ON PC命令は実行しません。
- ・PC ON命令によって有効になります。
- ・PC OFF命令によって無効になります。
- ・PC STOP命令によってPC割込みを覚えておきます。
- ・ON PC GOSUB命令実行時には、PC OFF (割込禁止) 状態になっています。

・使用例

```

10 ON PC 1 GOSUB 100
20 ON PC 2 GOSUB 200
30 PC ON
    {
100 PC READ "H4, I2" ; I, J
110 PRINT I, J
120 RETURN
    {
200 INPUT A
210 PC WRITE "I4" ; A
220 RETURN
    {

```

割込要因番号が1のとき、100行目へ分岐し、PCから2チャンネルのデータを読み込みCRTに表示する。

割込要因番号が2のとき、200行目へ分岐し、KEYにより入力されたデータをPCへ1チャンネル書込む。

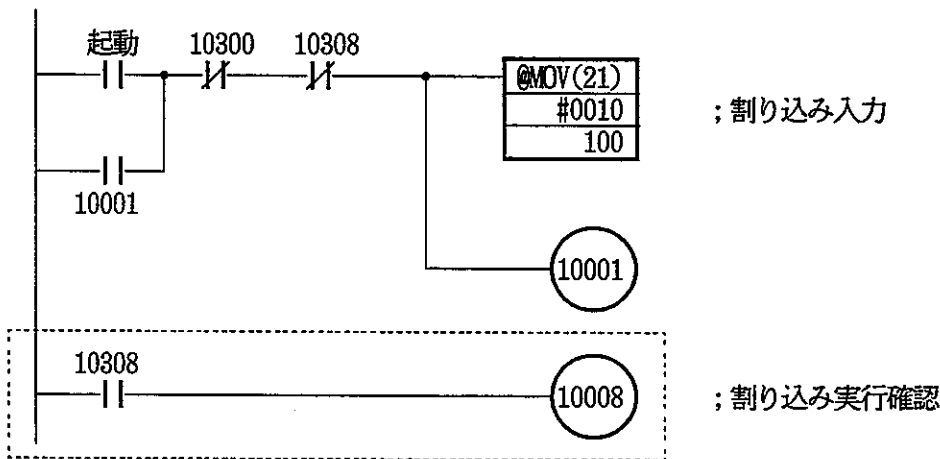
●参考

ASCIIビジーフラグON中はPC本体からの割り込みが禁止されているため、ON PC GOSUB命令は実行されません。そのため、PC READなどの実行時には割り込みがかかりません。

ASCIIビジーフラグがONするような命令(PC READ命令など)とON PC GOSUB命令を使用する場合には、ASCIIビジーフラグON中の割り込み発行は避けてください。また、WRITEフラグONによる割り込み起動を確認することをお勧めします。

(号機No=0時)

・ラダープログラム(PC本体側)



・BASICプログラム(ASC02側)

```
10  ON PC 1 GOSUB 100
20  PC ON
```

```
30  PC GET A, B
40  IF B=1 THEN PC PUT 0
```

; B=接点 10008が1なら
; 10308OFFにする

```
50  GO TO 30
60  END
```

```
100  PC PUT 1
```

; 10308ONにする

```
110  割り込み処理
120  RETURN
```

●プログラムの説明

- ・PC本体側のプログラムによりASCIIビジーフラグONしているタイミングを避けて、ASCII側の割り込みが実行されます。
- ・ASC02はWRITEフラグの立ち上がりにより、ON PC GOSUB命令を実行すると、接点10308をONし、割り込みプログラムが実行されたことをPC本体側に通知します。
- ・PC本体側は、接点10308により割り込みの実行を認識すると、接点10008をONして、ASC02に通知します。
- ・ASC02は接点10008のONにより接点10308をOFFします。

■PC GET 変数名1 [, 変数名2]

【機能】 PCの出力データエリアの値を変数に読み込む。

【記述例】 PC GET I
PC GET I, J

【説明】

- ・変数名1には、ユニット割付先頭nCHのビット0～7の値が読み込まれます。
- ・変数名2には出力データ（ユニット割付先頭チャンネルnCHの8～15ビット）の値が書込まれます。
- ・変数名1および変数名2の値は0～255の範囲です。
- ・PCのnCHの0～7、8～15には00～FFの16進データがPCから書込まれます。ASCIIユニットはこれを10進データ（0～255）に変換して読み込みます。

■PC (割込番号) ON/OFF/STOP

【機能】 ON PC GOSUB文で定義されたPCからの割込みの許可、禁止、停止を制御する。

【記述例】 PC ON

【説明】

- ・PC ON
ON PC GOSUB文で定義されたPCからの割込みを許可します。
この命令実行後は、PCからの割込みが入るごとに割込みを発生し、ON PC GOSUB文で定義されている処理ルーチンに分岐します。
- ・PC OFF
ON PC GOSUB文で定義されたPCからの割込みを禁止します。
この命令実行後は、PCからの割込みが入っても処理ルーチンへの分岐は起こりません。
- ・PC STOP
ON PC GOSUB文で定義されたPCからの割込みを停止します。
この命令実行後は、PCからの割込みが入っても、そのことを覚えておくだけで、処理ルーチンへの分岐は起こりません。しかし、その後PC ONによって割込みが許可されると、先ほどのPCからの割込みによる処理ルーチンに分岐します。
- ・割込番号を省略したときは、割込番号の1～15のすべてについて命令を実行したのと同じになります。
- ・プログラム終了時にはPC OFFの状態にしてください。
- ・割込処理ルーチンではすべての割込みについて割込許可、停止状態のものは割込停止状態に、割込禁止状態のものは割込禁止状態になります。

・使用例

```

10 ON PC GOSUB 1000
20 PC ON
30 GOTO 30
1000 PC READ "312" ; A, B, C
1010 PRINT A, B, C
1020 RETURN

```

ON PC GOSUB命令により、PCからの割込みがあったときの分岐先を定義し、PC ONで割込みを許可します。

■PC PUT 数値式

【機能】 数値式の値をPCの入力データ（ユニット割付チャネルn+3CHの8～15ビット）に書込む。

【記述例】 PC PUT I
PC PUT 123

【説明】

- ・取扱える数値式は0～255の整数です。
- ・ASCIIユニット側の0～255の10進データを、0～FFの16進数データに変換してPC側に書込みます。
- ・数値式の値が小数部分を含む場合は、内部でINT関数を実行し、小数点以下を切捨てて整数化をした値を入力データ（ユニット割付けチャネルn+3CHの8～15ビット）に書込みます。
- ・数値式の値が1～255の範囲外のときは、0を入力データ（ユニット割付チャネルn+3CHの8～15ビット）に書込みます。

■PC READ “フォーマット [, フォーマット...]” ;変数名 [, 変数名...]
PC READ “サブコマンド, 転送開始チャネル番号, 転送チャネル数, フォーマット...” ;変数名...

【機能】 PCの書込みデータを指定したフォーマットに変換して、指定した変数に読込む。
PCの指定されたデータを指定したフォーマットに変換して、指定した変数に読込む。

【記述例】 PC READ “2H1, A3, I4, O2” ;X, Y, A\$, I, J
PC READ “@R, 123, 4, 4I4” ;A, B, C, D

【説明】

- ・サブコマンドは“@R”, “@D”のように指定します。サブコマンド付きの命令を@形式と呼びます。
- ・@形式でない場合は、WRITEフラグ（nCHの01ビット）がONしていなければ、ONするまで実行待ちとなります。WRITEフラグがONしていれば（n+1）CH、（n+2）CHで指定されたリレーエリアのデータを読込みます。（n=100+10×号機No）
- ・@形式の場合は、PC側の動作モードおよびプログラムとは無関係にサブコマンドで指定されたリレーエリアのデータを読込みます。
- ・フォーマットおよびサブコマンドは文字変数でも構いません。
（例 A\$=“2H1, A3, I4, O2”
PC READ A\$; X, Y, A\$, I, J）
- ・PCの出力データ長が変数のデータ長より大きい場合、余分な出力データ部分は無視されます。
- ・PCの出力データ長が変数のデータ長より小さい場合、不足が発生した変数への読込みは実行されません。
- ・@形式において、転送チャネル数の値は1～255の範囲です。転送チャネル数が20を超える場合、1スキャンに20チャネルずつ、数スキャンにわたって処理を行います。
- ・@形式において、リレーエリアを越えて指定した場合“FORMAT ERROR”となります。
- ・フォーマットおよびサブコマンドは大文字で記述してください。

■PC WRITE “フォーマット [, フォーマット…] ” ; 数値式 [, 数値式…]
 PC WRITE “サブコマンド, 転送開始チャンネル番号, 転送チャンネル数, フォーマット…”
 ; 数値式 …

【機能】 指定した数値式の値を指定したフォーマットに変換して、PCへ転送する。
 指定した数値式の値を指定フォーマットに変換して、PCの指定した領域に転送する。

【記述例】 PC WRITE “H4, A2, I3, O4” ; 1234, “AB”, K, L
 PC WRITE “@H, 5, 4, H4, A2, I3, O4” ; 1234,
 “AB”, K, L

【説明】

- ・サブコマンドは“@R”, “@D”のように指定します。サブコマンド付きの命令を@形式と呼びます。
- ・@形式でない場合は、READフラグ (nCHの02ビット) がONしていなければ、ONするまで実行待ちとなります。READフラグがONしていれば、(n+1)CH、(n+2)CHで指定されたリレーエリアにデータを書込みます。
- ・@形式の場合は、PC側の動作モードおよびプログラムとは無関係に、サブコマンドで指定されたリレーエリアにデータを書込みます。
- ・フォーマットおよびサブコマンドは文字変数でも構いません。
 (例 A\$ = “H4, A2, I3, O4”
 PC WRITE A\$; 1234, “AB”, K, L)
- ・数値式の値が小数部分を含む場合は、内部でINT関数を実行し、小数点以下切捨てた値をPCに転送します。
- ・単精度実数型および倍精度実数型の数値式は内部で整数型に変換されます。
- ・@形式において、転送チャンネル数の値は1～255の範囲です。
 転送チャンネル数が20を越える場合、1スキャンに20チャンネルずつ、数スキャンにわたって処理を行います。
- ・@形式において、リレーエリアを越えて書込みを行った場合、“FORMAT ERROR”となります。
- ・フォーマットおよびサブコマンドは大文字で記述してください。
 (フォーマットおよびサブコマンドについては、次ページ以降を参照)

■PC READ, PC WRITE @形式 サブコマンド一覧

サブコマンド	チャンネル
@R	入出力リレー内部補助リレーエリア 0000~0255 PC READ 0000~0252 PC WRITE
@H	保持リレーエリア HR0000~HR0099
@A	補助記憶リレーエリア AR0000~AR0027
@L	リンクリレーエリア LR0000~LR0063
@G	TIM、CNT現在値エリア TIM0000~TIM0511
@D	データメモリエリア DM0000~DM1999

例) @R, 0, 123 入出力リレー 0チャンネルから123チャンネル分のデータを示します。
 @A, 10, 5 補助記憶リレー10チャンネルから5チャンネル分のデータを示します。

《参考》

1. サブコマンドは大文字で記述してください。
2. DM1000~1999のエリアをリード/ライトできるのは、C200H本体のメモリがRAMタイプのときだけです。EEP-ROM、P-ROMタイプのときはライトできません。
3. C200HSにおいても、リード/ライトできる領域はDM0000~1999となります。

■PC READ, PC WRITE命令フォーマット一覧表

フォーマット	意味内容	名称
mIn	mチャンネル分の10進n桁データを示す	I型フォーマット
mHn	mチャンネル分の16進n桁データを示す	H型フォーマット
mOn	mチャンネル分の8進n桁データを示す	O型フォーマット
mBn	mチャンネル分のnビット目データを示す	B型フォーマット
mAn	mチャンネル分のnで指定するASCII文字データを示す	A型フォーマット
SmXn	mチャンネル分のXで指定するn桁(nビット目) データを示す	S型フォーマット

《参考》

- ・mを省略したときは、1チャンネル分となります。
- ・一度に扱えるチャンネル数は最大255チャンネルです。
- ・256チャンネル以上を指定するとエラーになります。
- ・A型およびS型フォーマット以外は1チャンネルごとに1つの変数を対応させる必要があります。
- ・A型フォーマットは、1つのフォーマットに対して1つの変数を対応させます。
- ・S型フォーマットは、1チャンネルごとに1つの配列変数を対応させる必要がありますが、記述上は、1つのフォーマットに対して1つの配列変数を対応させます。
- ・A型およびS型フォーマット以外では1つのフォーマットに対して複数の変数または数値式を対応させることができます。

例) "5H2"; A, B, C, D, E

これは"1H2, 1H2, 1H2, 1H2, 1H2"; A, B, C, D, Eと同じことになります。

- ・フォーマット指定はすべて大文字で記述してください。
- ・一度に転送できるチャンネル数の最大は255チャンネルですが、以下の制限事項があるため、A型およびS型フォーマット以外では、実際には、50~60チャンネルが限度となります。
- ①一行に書くことのできる文字数は最大255文字。
- ②A型およびS型フォーマット以外では、1チャンネルごとに1変数を対応させる必要があります。

例) H型フォーマットの場合

1000 PC READ "59H4"; A0\$, A1\$, A2\$..., B0\$, B1\$, ..., E9\$

19文字

最大236文字

(255-19)

1変数に4文字を使用するとして、書込むことができる変数の最大個数は

$$\frac{236}{4} = 59$$

従って、最大59チャンネル分の転送となります。

- ・A型フォーマットでは、ASCIIユニットの1変数にPC側の複数チャンネルを対応させることができるため、一度に最大255チャンネルまで転送できます。(7・35ページ参照)
- ・ASCIIユニットの1文字変数には最大255文字しか代入できません。従ってA型フォーマットで1変数に複数チャンネルのデータに対応させるときは、255文字以下となるよう設定してください。
- ・S型フォーマットでは、ASCIIユニットの1配列変数にPC側の1チャンネルを対応させる必要がありますが、記述上は、1つのフォーマットに対して1配列変数のみを対応させるだけでよいので、一度に最大255チャンネルまで転送できます。(7・35ページ参照)

(1) I型フォーマット (mIn)

各桁を10進数(0~9)として扱います。

桁	ビット															
n	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	———				———				———				$\times 10^0$			
2	———				———				$\times 10^1$				$\times 10^0$			
3	———				$\times 10^2$				$\times 10^1$				$\times 10^0$			
4	$\times 10^3$				$\times 10^2$				$\times 10^1$				$\times 10^0$			

例) 2I3...2チャンネル分の3桁10進データを示す。

(2) H型フォーマット (mHn)

各桁を16進数(0~F)として扱います。

桁	ビット															
n	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	———				———				———				$\times 16^0$			
2	———				———				$\times 16^1$				$\times 16^0$			
3	———				$\times 16^2$				$\times 16^1$				$\times 16^0$			
4	$\times 16^3$				$\times 16^2$				$\times 16^1$				$\times 16^0$			

例) 3H4...3チャンネル分の4桁16進データを示す。

(3) O型フォーマット (mOn)

各桁を8進数(0~7)として扱います。

桁	ビット															
n	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	———				———				———				$\times 8^0$			
2	———				———				$\times 8^1$				$\times 8^0$			
3	———				$\times 8^2$				$\times 8^1$				$\times 8^0$			
4	$\times 8^3$				$\times 8^2$				$\times 8^1$				$\times 8^0$			

例) 4O2...4チャンネル分の2桁8進データを示す。

(4) B型フォーマット (mBn)

各ビットを2進数(0、1)として扱います。

n	ビット															
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	$\times 2^0$
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	$\times 2^1$	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	$\times 2^2$	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	$\times 2^3$	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	$\times 2^4$	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	$\times 2^5$	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	$\times 2^6$	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	$\times 2^7$	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	$\times 2^8$	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	$\times 2^9$	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	$\times 2^{10}$	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	$\times 2^{11}$	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	$\times 2^{12}$	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	$\times 2^{13}$	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	$\times 2^{14}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	$\times 2^{15}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

例) 5B14…5チャンネル分の14ビット目2進データを示す。

(5) A型フォーマット (mAn)

各桁をASCIIコードとして扱います。

n	ビット															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	———								ASCIIコード							
2	ASCIIコード								———							
3	ASCIIコード								ASCIIコード							

例) 6A2...6チャンネル分の上位桁のASCIIコードデータを示す。

- A型フォーマットでは、ASCIIユニットの1変数にPC側の複数チャンネルを対応することができ、一度に最大255チャンネルまで転送できます。

例) PC READ "50A3, 100A2, 30A1, 75A3"; A\$, B\$, C\$, D\$
 A\$ 50A3で示された50チャンネル分のデータ (50チャンネル×2文字=100文字分) が読み込まれます。
 B\$ 100A2で示された100チャンネル分のデータ (100チャンネル×1文字=100文字分) が読み込まれます。
 C\$ 30A1で示された30チャンネル分のデータ (30チャンネル×1文字=30文字分) が読み込まれます。
 D\$ 75A3で示された75チャンネル分のデータ (75チャンネル×2文字=150文字分) が読み込まれます。

(6) S型フォーマット (SmXn)

- 各桁 (各ビット) をXの指定に従い、扱います。
- Xは、I, H, O, Bを指定します。Aは指定できません。

フォーマット	意味内容
SmIn	mチャンネル分の10進n桁データを示す
SmHn	mチャンネル分の16進n桁データを示す
SmOn	mチャンネル分の8進n桁データを示す
SmBn	mチャンネル分のnビット目データを示す

- フォーマットに対応する変数は、1次元の配列変数としてください。配列変数でないときは、エラーとなります。
- 指示した配列変数から順に、フォーマットに従いデータを配列変数に格納する、または配列変数から転送します。
- S型フォーマットでは、ASCIIユニットの1配列変数にPC側の1チャンネルを対応させる必要がありますが、記述上は、1つのフォーマットに対して1配列変数のみを対応させるだけでよい、一度に最大255チャンネルまで転送できます。

例) PC READ "S100I4, S75H2, S80O3"; A(1), B(11), C(51)
 A(1)~A(100): S100I4で示された100チャンネル分の10進4桁データが読み込まれます。
 B(11)~B(85): S75H2で示された75チャンネル分の16進2桁データが読み込まれます。
 C(51)~C(130): S80O3で示された80チャンネル分の8進3桁データが読み込まれます。

■PC READフォーマット変換例

(1)I型フォーマット

PCのCHデータ

1	2	3	4
---	---	---	---

整数型変数 { PC READ "I 1" ; J ----> J = 4
 PC READ "I 2" ; J ----> J = 3 4
 PC READ "I 3" ; J ----> J = 2 3 4
 PC READ "I 4" ; J ----> J = 1 2 3 4

文字列型変数 { PC READ "I 1" ; A\$ ----> A\$ = "4"
 PC READ "I 2" ; A\$ ----> A\$ = "3 4"
 PC READ "I 3" ; A\$ ----> A\$ = "2 3 4"
 PC READ "I 4" ; A\$ ----> A\$ = "1 2 3 4"

(2)H型フォーマット

PCのCHデータ

8	9	A	B
---	---	---	---

整数型変数 { PC READ "H 1" ; J ----> J = &HB = 1 1
 PC READ "H 2" ; J ----> J = &HAB = 1 7 1
 PC READ "H 3" ; J ----> J = &H9AB = 2 4 7 5
 PC READ "H 4" ; J ----> J = &H89AB = -3 0 2 9 3

文字列型変数 { PC READ "H 1" ; A\$ ----> A\$ = "B"
 PC READ "H 2" ; A\$ ----> A\$ = "AB"
 PC READ "H 3" ; A\$ ----> A\$ = "9AB"
 PC READ "H 4" ; A\$ ----> A\$ = "89AB"

(3)O型フォーマット

PCのCHデータ

1	2	3	4
---	---	---	---

整数型変数 { PC READ "O 1" ; J ----> J = &4 = 4
 PC READ "O 2" ; J ----> J = &3 4 = 2 8
 PC READ "O 3" ; J ----> J = &2 3 4 = 1 5 6
 PC READ "O 4" ; J ----> J = &1 2 3 4 = 6 6 8

文字列型変数 { PC READ "O 1" ; A\$ ----> A\$ = "4"
 PC READ "O 2" ; A\$ ----> A\$ = "3 4"
 PC READ "O 3" ; A\$ ----> A\$ = "2 3 4"
 PC READ "O 4" ; A\$ ----> A\$ = "1 2 3 4"

(4) B型フォーマット

PCのCHデータ

C	1	2	2
---	---	---	---

整数型変数 {
 PC READ "B1" ; J ----> J=2
 PC READ "B2" ; J ----> J=0
 PC READ "B5" ; J ----> J=3 2
 PC READ "B14" ; J ----> J=1 6 3 8 4
 PC READ "B15" ; J ----> J=-3 2 7 6 8
 文字列型変数 {
 PC READ "B1" ; A\$ ----> A\$="2"
 PC READ "B2" ; A\$ ----> A\$="0"
 PC READ "B5" ; A\$ ----> A\$="3 2"
 PC READ "B14" ; A\$ ----> A\$="1 6 3 8 4"
 PC READ "B15" ; A\$ ----> A\$="-3 2 7 6 8"

(5) A型フォーマット

整数型変数は型があわないのでエラーとなります。

PCのCHデータ

5	1	5	2
5	3	5	4

文字列型変数 {
 PC READ "2A1" ; A\$ ----> A\$="RT"
 PC READ "2A2" ; A\$ ----> A\$="QS"
 PC READ "2A3" ; A\$ ----> A\$="QRST"

{
 Q: &H5 1
 R: &H5 2
 S: &H5 3
 T: &H5 4

(6) S型フォーマット

PCのCHデータ

0	1	2	3
4	5	6	7
8	9	0	1
2	3	4	5

整数型変数 {
 PC READ "S4I4" ; A (1) ----> A (1) = 1 2 3
 ----> A (2) = 4 5 6 7
 ----> A (3) = 8 9 0 1
 ----> A (4) = 2 3 4 5
 文字列型変数 {
 PC READ "S4A4" ; A\$ (1) ----> A\$ (1) = "0123"
 ----> A\$ (2) = "4567"
 ----> A\$ (3) = "8901"
 ----> A\$ (4) = "2345"

■PC WRITEフォーマット変換例

(1) I型フォーマット

PCのCHデータ

<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>4</td></tr></table>	0	0	0	4	← PC WRITE "I1" ; J	} 整数型変数 ← J = 1 2 3 4
0	0	0	4			
<table border="1"><tr><td>0</td><td>0</td><td>3</td><td>4</td></tr></table>	0	0	3	4	← PC WRITE "I2" ; J	
0	0	3	4			
<table border="1"><tr><td>0</td><td>2</td><td>3</td><td>4</td></tr></table>	0	2	3	4	← PC WRITE "I3" ; J	
0	2	3	4			
<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	1	2	3	4	← PC WRITE "I4" ; J	
1	2	3	4			

<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	0	0	0	1	← PC WRITE "I1" ; A\$	} 文字列型変数 ← A\$ = "1 2 3 4"
0	0	0	1			
<table border="1"><tr><td>0</td><td>0</td><td>1</td><td>2</td></tr></table>	0	0	1	2	← PC WRITE "I2" ; A\$	
0	0	1	2			
<table border="1"><tr><td>0</td><td>1</td><td>2</td><td>3</td></tr></table>	0	1	2	3	← PC WRITE "I3" ; A\$	
0	1	2	3			
<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	1	2	3	4	← PC WRITE "I4" ; A\$	
1	2	3	4			

(2) H型フォーマット

PCのCHデータ

<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>B</td></tr></table>	0	0	0	B	← PC WRITE "H1" ; J	} 整数型変数 ← J = -30293 = &H 8 9 A B
0	0	0	B			
<table border="1"><tr><td>0</td><td>0</td><td>A</td><td>B</td></tr></table>	0	0	A	B	← PC WRITE "H2" ; J	
0	0	A	B			
<table border="1"><tr><td>0</td><td>9</td><td>A</td><td>B</td></tr></table>	0	9	A	B	← PC WRITE "H3" ; J	
0	9	A	B			
<table border="1"><tr><td>8</td><td>9</td><td>A</td><td>B</td></tr></table>	8	9	A	B	← PC WRITE "H4" ; J	
8	9	A	B			

<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>8</td></tr></table>	0	0	0	8	← PC WRITE "H1" ; A\$	} 文字列型変数 ← A\$ = "8 9 A B"
0	0	0	8			
<table border="1"><tr><td>0</td><td>0</td><td>8</td><td>9</td></tr></table>	0	0	8	9	← PC WRITE "H2" ; A\$	
0	0	8	9			
<table border="1"><tr><td>0</td><td>8</td><td>9</td><td>A</td></tr></table>	0	8	9	A	← PC WRITE "H3" ; A\$	
0	8	9	A			
<table border="1"><tr><td>8</td><td>9</td><td>A</td><td>B</td></tr></table>	8	9	A	B	← PC WRITE "H4" ; A\$	
8	9	A	B			

(3)O型フォーマット
PCのCHデータ

<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>4</td></tr></table>	0	0	0	4	← PC WRITE "O1" ; J	} 整数型変数	←----- J=668=&1234
0	0	0	4				
<table border="1"><tr><td>0</td><td>0</td><td>3</td><td>4</td></tr></table>	0	0	3	4	← PC WRITE "O2" ; J		
0	0	3	4				
<table border="1"><tr><td>0</td><td>2</td><td>3</td><td>4</td></tr></table>	0	2	3	4	← PC WRITE "O3" ; J		
0	2	3	4				
<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	1	2	3	4	← PC WRITE "O4" ; J		
1	2	3	4				

<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	0	0	0	1	← PC WRITE "O1" ; A\$	} 文字列型変数	←----- A\$="1234"
0	0	0	1				
<table border="1"><tr><td>0</td><td>0</td><td>1</td><td>2</td></tr></table>	0	0	1	2	← PC WRITE "O2" ; A\$		
0	0	1	2				
<table border="1"><tr><td>0</td><td>1</td><td>2</td><td>3</td></tr></table>	0	1	2	3	← PC WRITE "O3" ; A\$		
0	1	2	3				
<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	1	2	3	4	← PC WRITE "O4" ; A\$		
1	2	3	4				

(4)B型フォーマット
PCのCHデータ

<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	0	0	0	1	← PC WRITE "B0" ; J	} 整数型変数	←----- J=-32749 =&H8013
0	0	0	1				
<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>2</td></tr></table>	0	0	0	2	← PC WRITE "B1" ; J		
0	0	0	2				
<table border="1"><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	0	0	1	0	← PC WRITE "B4" ; J		
0	0	1	0				
<table border="1"><tr><td>8</td><td>0</td><td>0</td><td>0</td></tr></table>	8	0	0	0	← PC WRITE "B15" ; J		
8	0	0	0				

文字列型変数は型があわないのでエラーとなります。

●フォーマットと変数の関係

B_nのとき 変数を2進数で表したときnビットがONしていれば、PCのnビットをONし他のビットをOFFします。変数のnビットがOFFしていれば、PCのすべてのビットをOFFします。

(例) -32749=&H8013=1000 0000 0001 0011

B0, B1, B4, B15 ----- ON
B2, B3, B5~B14 ----- OFF

(5)A型フォーマット

PCのCHデータ

0	0	5	1
0	0	5	2

← PC WRITE "2A1" ; A\$

5	1	0	0
5	2	0	0

← PC WRITE "2A2" ; A\$

5	1	5	2
5	3	5	4

← PC WRITE "2A3" ; A\$

文字
列
型
変
数

←----- A\$ = "QRST"

{ Q: &H51
R: &H52
S: &H53
T: &H54

整数型変数は型があわないのでエラーとなります。

(6)S型フォーマット

PCのCHデータ

9	8	7	6
5	4	3	2
1	0	9	8
7	6	5	4

← PC WRITE "S4I4" ; A (1)

←----- A (1) = 9876

←----- A (2) = 5432

←----- A (3) = 1098

←----- A (4) = 7654

整数
型
変
数
フ
ォ
ー
マ
ット
の
と
き

■ASCII側命令

[単位ms]

命 令	実行時間
PC READ "I 4" ; A	8.5
PC READ "5 I 4" ; A, B, C, D, E	21.1
PC READ "10 I 4" ; A, B, C, D, E, F, G, H, I, J	43.8
PC READ "100 A 3, 100 A 3, 55 A 3" ; A\$, B\$, C\$	67.7
PC WRITE "I 4" ; A	8.7
PC WRITE "5 I 4" ; A, B, C, D, E	22.1
PC WRITE "10 I 4" ; A, B, C, D, E, F, G, H, I, J	39.3
PC WRITE "100 A 3, 100 A 3, 55 A 3" ; A\$, B\$, C\$	57.9
PC READ "@D, 0, 1, I 4" A	5.0
PC READ "@D, 0, 5, 5 I 4" ; A, B, C, D, E	18.6
PC READ "@D, 0, 10, 10 I 4" ; A, B, C, D, E, F, G, H, I, J	40.3
PC READ "@D, 0, 255, 100 A 3, 100 A 3, 55 A 3" ; A\$, B\$, C\$	65.3
PC WRITE "@D, 0, 1, I 4" A	4.4
PC WRITE "@D, 0, 5, 5 I 4" ; A, B, C, D, E	19.0
PC WRITE "@D, 0, 10, 10 I 4" ; A, B, C, D, E, F, G, H, I, J	37.5
PC WRITE "@D, 0, 255, 100 A 3, 100 A 3, 55 A 3" ; A\$, B\$, C\$	54.4

《参考》

実行時間はPCとのハンドシェイクを含まない時間です。

実際に要する時間は、PCのスキャンタイムにより変化します。

- ・転送チャンネル数20チャンネル以下のとき

MAX PCの2スキャンタイム

@形式 MAX PCの1スキャンタイム

- ・転送チャンネル数20チャンネルをこえるとき

MAX $\text{INT}((\text{転送チャンネル数}-1)/20) + 2$ スキャンタイム@形式 MAX $\text{INT}((\text{転送チャンネル数}-1)/20) + 1$ スキャンタイム

■POKE 番地 数値

【機能】 メモリ上の指定番地へデータを書込みます。

【記述例】 POKE &H2000, &H39

【説明】

- ・番地で指定するメモリに1バイト(8ビット)の数値を書込みます。
- ・数値は1バイトの整数で、0~255(&H0~&HFF)の範囲です。
- ・番地は2バイトの整数で、0~65535(&HFFFF)の範囲です。
- ・&H0000~&H2000、&H8000~&HFFFFはシステムエリアですので、書込まないでください。

**■PRINT [#ポート番号,] [式 [,] [式]]]
LPRINT**

【機能】 式の評価結果を画面またはプリンタに出力する。

【記述例】 PRINT #1, A, B\$; "BASIC"

【説明】

- ・式が文字式の場合はその結果の文字列が出力され、数値式の場合はその結果の値が出力されます。
- ・式をコンマで区切ると、9文字ずつのフィールドに分けて出力し、セミコロンまたは空白で区切ると、前の出力結果に続けて出力されます。
- ・式の最後がセミコロンでない場合が出力後改行します。
- ・数値式の場合には、前後に1個の空白がとられ、前の空白の所はマイナス符号用になっています。
- ・式が1つもない場合は、改行となります。
- ・ポート番号の項を省略したときは、PRINTコマンドではポート1を、LPRINTコマンドではポート2を選択されたものとみなします。
- ・LPRINTコマンドでは、OPEN文に関係なく周辺装置名LPRTの条件で出力されます。

■PRINT [#ポート番号,] USING フォーマット文字列;式 [{;} [式]]
LPRINT

【機能】 指定したフォーマットで文字列、数値を画面またはプリンタに出力する。

【記述例】 PRINT #1, USING "####, # ㍻㍻####";A;B

【説明】

- ・文字出力のフォーマット制御文字
 - (1)! 最初の1文字だけを出力
 - (2)&_& &で囲まれた文字数分出力
 - (3)@ 対応する文字列をそのまま出力
- ・数値出力のフォーマット制御文字
 - (1)# 数字の桁数指定で負符号も1桁として数えられる。
 - (2). 任意の箇所に小数点を挿入する。
 - (3)+ 数値の直前または直後に符号(+)を付ける。
 - (4)- 負の数値の直後に符号(-)を付ける
(フォーマット文字列の最後につける。)
 - (5)* 数値の上位空白桁に*を付ける。
 - (6)㍻ 数値の直前空白桁に1個の㍻を付ける。
 - (7)*㍻ (5)(6)の両機能をあわせもつ。
 - (8), 整数部分を右側より3桁ごとにコンマで区切る。
 - (9)^^^ 指数形式の出力指示(E+nn)(#の後に付ける)
 - (10) に続く1文字をそのまま出力する。

(注) 指定した数値の桁数が大きい場合は、数値の前に“%”が出力されます。

- ・ポート番号の項を省略したときは、PRINT USINGコマンドではポート1を、LPRINT USINGコマンドではポート2を選択されたものとみなします。
- ・LPRINT USINGコマンドではOPEN文に関係なく周辺装置名LPRTの条件で出力されます。

■RANDOM [式]

【機能】 式で与えた乱数の元値より乱数の系列を変更する。

【記述例】 RANDOM 5649

【説明】

- ・式は数値式によって-32768~32767の範囲内で乱数の元値を与えます。
- ・式を省略した場合は、乱数の元値要求メッセージがポート1に出力されますので、乱数の元値をキーボードより入力してください。
- ・参照 RND

■READ 変数名 [, 変数名...]

【機能】 DATA文により定義されたデータを変数に読込む。

【記述例】 READ A, B\$

【説明】

- ・READ文の変数の型と並び、DATA文の定数の型と並びは1対1に対応させて読込む必要があります。
- ・参照 DATA、RESTORE

■REM [コメント文]

【機能】 プログラム中にコメント文を入れる。

【記述例】 REM SAMPLE PROGRAM

【説明】 ・' でREM文を代用できます。
・REM文は非実行文です。

■RESTORE [行番号]

【機能】 READ文で読始めるDATA文の行番号を指定する。

【記述例】 RESTORE 1000

【説明】 ・指定した行番号以降のDATA文よりREAD文を読始めます。
・行番号を省略したときは、プログラム中の最初のDATA文を指定したことになります。

■RESUME [{NEXT}] [行番号]

【機能】 エラー処理終了後、プログラム実行の再開行を指示する。

【記述例】 RESUME 100

【説明】 ・行番号を指定したとき、その行番号から実行再開します。
・NEXTを指定したとき、エラー発生文の次の文から実行再開します。
・指定のないときは、エラー発生文より実行再開しますので、エラー処理ルーチンでエラー原因を取除いておく必要があります。

■STOP

【機能】 プログラム中のSTOP文の所で実行停止して、コマンド入力待ちにする。

【記述例】 STOP

【説明】 ・STOP文を実行すると“BREAK IN ××××”メッセージが出力されて、コマンド入力待ちとなります。
・開かれているポートは閉じられません。
・CONTコマンドによって実行の再開ができます。

■WAIT “待ち時間” [, 行番号]

【機能】 次の命令語の実行終了までの時間を設定する。

【記述例】 WAIT “10:30.5”, 100

【説明】

- ・待ち時間は、0.1秒から59分59秒9まで0.1秒ごとの設定が可能です。
- ・設定された時間以内に処理を終了しないときは、指定された行に分岐します。
- ・行番号が省略された場合は、分岐先は監視対象行の次の行となります。
- ・待ち時間は、“分:秒”のように設定してください。
- ・WAIT命令の後に命令語を続けて書いた場合、監視対象命令はWAIT命令の次の命令語となります。
- ・WAIT命令から次の命令語までは、ON COM, ON KEY, ON PC, ON式, ON ERRORによる割込みはかかりません。
- ・監視対象命令は次の命令語です。
INPUT, INPUT\$, LINE INPUT, PC READ, PC WRITE, PRINT, LPRINT, PRINT USING, LPRINT USING
上記命令以外を指定した場合、その命令実行が設定時間内に終了しなかったときは、正常に処理をしないことがありますので使用しないでください。
- ・使用例

```

10 WAIT “10.0”, 1000
20 PC READ “314”; A, B, C
30 PRINT A, B, C
40 END
1000 PRINT “PC ERR”
1010 GOTO 40

```

PC READ命令に10秒以内にPCからの起動がかからなかったら、“PC ERR”を表示します。

<デバイスコントロールステートメント>

■CLOSE [#ポート番号]

【機能】 ポートを閉じる。

【記述例】 CLOSE #1
CLOSE

【説明】

- ・ポート番号に対応するポートを閉じます。以後、指定されたポートは、異なる周辺装置のために使用するポートを開くのに利用できるようになります。
- ・ポート番号の項を省略したときは、その時、開いているポートを全て閉じます。
- ・閉じているポートに対しては、再びポートを開くまで入出力を行うことはできません。
- ・ポートの周辺装置が出力用の場合には、CLOSEコマンドでバッファに残っていたデータの掃出しを行いますので、出力処理を正しく終了するためには、必ずCLOSEコマンドを実行しなければなりません。
- ・ただし、ポートの周辺装置が入力用の場合には、バッファデータの掃出しは行いません。
- ・END、NEWコマンドは、自動的に全てのポートを閉じる処理を行います。
- ・STOPコマンドはポートを閉じる作業は行いませんので注意してください。

■CLS [#ポート番号]

【機能】 画面をクリアする。

【記述例】 CLS

【説明】

- ・表示画面をクリアし、カーソルをホームポジションに移します。
- ・ポート番号の項を省略したときは、ポート1を選択されたものとみなします。

■OPEN #ポート番号, “周辺装置名: [(通信条件・有効信号線)]”

【機能】 ポートを開く。

【記述例】 OPEN #1, “KYBD:”
OPEN #2, “COMU: (14)”

【説明】 ・周辺装置名で指定された入出力装置を、ポート番号で指定されたポートに開きます。
以後、開かれたポートの周辺装置への入出力は、ポート番号を指示することにより行います。
・周辺装置名は次の中から選択してください。

周辺装置	周辺装置名	ASCIIエントからの出力	ASCIIエントへの入力
ターミナル	TERM:	○	○
キーボード	KYBD:	×	○
ディスプレイ	SCRN:	○	×
プリンタ	LPRT:	○	×
RS-232C装置	COMU:	○	○

・通信条件は次の中から選択してください。

通信条件	1文字の長さ	パリティ	ストップビット
0	7ビット	偶数	2ビット
1	7ビット	奇数	2ビット
2	7ビット	偶数	1ビット
3	7ビット	奇数	1ビット
4	8ビット	なし	2ビット
5	8ビット	なし	1ビット
6	8ビット	偶数	1ビット
7	8ビット	奇数	1ビット

・有効信号線は次の中から選択してください。

信号線 No.	CTS	DSR	RTS	XON/ XOFF
0	○	○	○	×
1	○	○	×	
2	○	×	○	
3	○	×	×	
4	×	○	○	
5	×	○	×	
6	×	×	○	
7	×	×	×	○
8	○	○	○	
9	○	○	×	
A	○	×	○	
B	○	×	×	
C	×	○	○	
D	×	○	×	
E	×	×	○	
F	×	×	×	

(注) ポート2でCTSを無効にする場合は、CTSの信号線をONするか、RTS信号に接続するかのいずれかにしてください。
○は有効を表し、×は無効(信号線を無視する)を表します。

・RTS指定

RTS ON (○) 指定にすると、ポートをOPENしてからCLOSEするまで、RTSがHigh状態になります。

RTS OFF (×) 指定にすると、入出力命令(PRINT、INPUT文等)実行時以外RTSがLOW状態になります。

周辺機器から常にデータを受付ける場合は、RTS ON指定にしてください。

ON COM命令による受信割り込み機能を使用する場合はRTS ON指定にしてください。RTS OFF指定にしますと、受信割り込みがかりません。

・XON/XOFF指定

XON (○) 指定にすると、データ受信時受信用バッファのデータが3/4になると、XOFFコードを送信し、送信の一時停止を要求します。

また、受信用バッファのデータが1/4になると、XONコードを送信し、送信の再開を要求します。

データ送信時、XOFFコードを受信すると送信を一時停止し、XONコードを受信すると送信を再開します。

XOFF (×) 指定の場合、上記制御は行いません。

そのため、バッファがFULLになった場合、その後のデータは受取りません。

(注) FC-987、PC-9801LV21は、パソコンからASCIIユニットへのデータ送信時、XON/XOFF制御は行われません。

・通信条件、有効信号を省略したときは、各装置のデフォルト値が選択されます。

周辺装置	通信条件	有効信号線
ターミナル	4	3
キーボード	4	3
ディスプレイ	4	3
RS-232C装置	4	3
プリンタ	4	5

・既に開かれているポートを再度開くことはできません。

この場合、既に開かれているポートをCLOSEコマンドにより閉じてから再度、OPENコマンドにより開いてください。

・OPENコマンドにより開かれていないポート、またはCLOSEコマンドにより閉じられたポートは、それぞれポート1がターミナル、ポート2がプリンタになります。

・OPENコマンドにより開かれていないポートに対し、ポート番号の指定された入出力命令により入出力を行うことはできません。この場合は、ポート番号を省略した入出力命令により入出力を行ってください。

・ポート2にはターミナルを指定することはできません。

■OPENコマンド実行時のRTS, DTRの出力レベル

デバイス	OPEN時		動作時	
	RTS	DTR	RTS	DTR
TERM	LOW	HIGH	HIGH	そのまま
SCRN	LOW	LOW	HIGH	そのまま
KEYB	LOW	HIGH	HIGH	そのまま
COMU	LOW	HIGH	HIGH	そのまま
LPRT	LOW	LOW	HIGH	そのまま

ポート	CLOSE時	
	RTS	DTR
1	LOW	HIGH
2	LOW	LOW

(注) CLOSEすると各ポートは電源投入時と同じくデフォルト状態としてポート1はTERM、ポート2はLPRTの指定状態となります。

■OPENしたデバイスによる出力の相違

SCRN TERM		・&H0C (CLR) 時スクリーンバッファをクリアする。 ・&H0A (LF), &H0D (CR), &H0B (HOME), &H0C (CLR) の時カラム位置を0にする。(左端になる) ・&H08 (BS), &H1C (→), &H1D (←) の時はそれに応じてスクリーンバッファ上でカーソル移動。 ・&H00～&H09および&H0E～&H1Bコードは無視(出力しない)
	CLOSE 実行時	・なにもしない。
LPRT		・&H0A, &H0D, &H0B, &H0Cの時カラム位置を0にする(左端になる) ・80キャラクタを超えると&H0A (LF) を追加して出力する。
	CLOSE 実行時	・バッファ(80キャラ以内)にキャラクタが残っているときはそれと&H0A (LF) を出力する。
COMU		・バッファにキャラクタが入ってくると出力する。
	CLOSE 実行時	・バッファにキャラクタが残っていれば、それを出力する。

・OPENしたデバイス名により、PRINT文で出力されるデータに対する処理が違います。

算術関数

<算術演算関数>

■ABS (数値式)

【機能】 引数で与えた値の絶対値を得る。

【記述例】 $A = \text{ABS}(-1.5)$

【説明】 ・指定した数値式の絶対値が得られます。

■ACOS (数値式)

【機能】 引数で与えられた数値の逆余弦（アークコサイン）を得る。

【記述例】 $A = \text{ACOS}(1)$

【説明】 ・数値は-1から1までの範囲をとります。
・得られる値の単位はラジアンです。
・得られる値の範囲は0から π までです。
・精度は7桁で与えられます。

■ASIN (数値式)

【機能】 引数で与えられた数値の逆正弦（アークサイン）を得る。

【記述例】 $A = \text{ASIN}(1)$

【説明】 ・数値式は、-1から1までの範囲をとります。
・得られる値の単位はラジアンです。
・得られる値の範囲は $-\pi/2$ から $\pi/2$ までです。
・精度は7桁で与えられます。

■ATN (数値式)

【機能】 引数で与えられた数値の逆正接（アークタンジェント）を得る。

【記述例】 $A = \text{ATN}(1)$

【説明】 ・数値式の単位はラジアンです。
・得られる値の範囲は、 $-\pi/2$ から $\pi/2$ までの範囲です。
・精度は7桁で与えられます。

■CDBL (数値式)

【機能】 数値を倍精度実数値に変換する。

【記述例】 CDBL (2 / 3)

■CINT (数値式)

【機能】 数値を小数点以下を四捨五入して整数値に変換する

【記述例】 A = CINT (B #)

■COS (数値式)

【機能】 引数で与えられた数値の余弦 (COS) を得る。

【記述例】 A = COS (0.5)

【説明】

- ・数値式の単位はラジアンです。
- ・精度は7桁で与えられます。

■CSNG (数値式)

【機能】 数値を単精度実数値に変換する。

【記述例】 B = CSNG (C #)

■FIX (数値式)

【機能】 引数で与えた数値の整数部分を得る。

【記述例】 A = FIX (B / 3)

【説明】

- ・数値の小数点以下を切捨てた値となります。
- ・引数が負のときに、INT関数との違いがでてきます。

■INT (数値式)

【機能】 数値の小数点以下を切捨てた整数値を得る。

【記述例】 A = INT (B)

【説明】

- ・引数の値を超えない最大の整数を返します。
- ・引数が負のときに、FIX関数との違いがでてきます。

■LOG (数値式)

【機能】 自然対数を与える。

【記述例】 $A = \text{LOG}(5)$

【説明】

- ・数値式の自然対数を与えます。
- ・精度は7桁で与えられます。

■RND [(数値式)]

【機能】 0と1の間の乱数を得る。

【記述例】 $A = \text{RND}(1)$

【説明】

- ・数値式が正または省略された場合、同じ系列の次の乱数となる。
- ・数値式が0の場合、1つ前に発生した乱数となる。
- ・数値式が負の場合、新しい固有の乱数系列となる。
- ・正または0の場合には、発生する乱数は、RUNまたはCLEAR文が実行されることに同じ系列を繰返しますが、RANDOM文によって別の系列にすることができます。

■SGN (数値式)

【機能】 引数の符号を得る。

【記述例】 $B = \text{SGN}(A)$

【説明】

- ・引数が正の場合は1を、0の場合は0を、負の場合は-1を関数値として返します。

■SIN (数値式)

【機能】 引数で与えられた数値の正弦(SIN)を得る。

【記述例】 $A = \text{SIN}(1)$

【説明】

- ・数値式の単位はラジアンです。
- ・精度は7桁で与えられます。

■TAN (数値式)

【機能】 引数で与えられた数値の正接(TAN)を得る。

【記述例】 $A = \text{TAN}(3)$

【説明】

- ・数値式の単位はラジアンです。
- ・精度は7桁で与えられます。

<文字列関数>

■ASC (文字式)

【機能】 引数で与えた文字列の先頭文字の数値 (キャラクタコード) を得る。

【記述例】 A=ASC (A\$)

【説明】

- ・空文字列は指定できません。
- ・この逆の機能をもったのがCHR\$関数です。

■CHR\$ (数値式)

【機能】 引数で与えた数値 (キャラクタコード) に対応する1文字を得る。

【記述例】 A\$=CHR\$ (&H41)

【説明】

- ・数値式の値は0～255の範囲内です。
- ・実数値を指定した場合は、小数点以下を四捨五入したものを値とします。
- ・この逆の機能をもったものがASC関数です。

■HEX\$ (数値式)

【機能】 引数で与えた10進数値を16進数の文字列に変換する。

【記述例】 A\$=HEX\$ (B)

【説明】

- ・10進数が小数部分を含む場合は、内部でINT関数を実行し、小数点以下を切捨てた後に変換します。
- ・変換結果は“0”～“FFFF”になります。

■INSTR ([検索開始位置,] 文字式1, 文字式2)

【機能】 文字式1の文字列データの検索開始位置より、文字式2のデータを探し、その位置を得る。

【記述例】 A=INSTR (5, B\$, “BASIC”)

【説明】

- ・検索開始位置は、1～255の範囲内ですが省略されたときは、1として解釈し文字式1データ先頭より検索します。
- ・検索結果が得られない場合は、関数値として0を返します。
- ・文字式2が空文字列の場合には、検索開始位置を関数値として返します。

■LEFT\$ (文字式, 文字数)

【機能】 文字列の左側から、指定された文字数の文字列を得る。

【記述例】 A\$=LEFT\$ (B\$, 5)

【説明】

- ・文字数は0～255の範囲内です。
- ・文字数が0の場合は、関数値は空文字列になります。
- ・文字数が文字式で示されるデータの文字数を超える場合、関数値は文字列データ全体となります。

■LEN (文字式)

【機能】 文字列の文字数を得る。

【記述例】 A=LEN (A\$)

【説明】 ・文字式が空文字列の場合、関数値は0となります。

■MID\$ (文字式, 文字位置 [, 文字数])

【機能】 文字列の中より指定された文字位置から文字数分取出す。

【記述例】 B\$=MID\$ (A\$, 2, 5)

【説明】 ・文字位置は1～255の範囲内です。
・文字数は0～255の範囲内です。
・文字数が0または、文字位置が文字列データ数を超えている場合、関数値は空文字列となります。
・文字数が省略された場合、あるいは文字列データの残りを越えている場合の関数値は、残りの文字列全体になります。

■OCT\$ (数値式)

【機能】 引数で与えた10進数値を8進数の文字列に変換する。

【記述例】 A\$=OCT\$ (B)

【説明】 ・10進数値が小数部分を含む場合は、内部でINT関数を実行し、小数点以下を切捨てて整数化をした後に8進数に変換します。
・数値式の値は、-32768～32767の範囲内で指定してください。

■RIGHT\$ (文字式, 文字数)

【機能】 文字列の右側から、指定された文字数を取り出す。

【記述例】 A\$=RIGHT\$ (B\$, 5)

【説明】 ・文字数は0～255の範囲内です。
・文字数が0の場合、関数値は空文字列になります。
・文字数が文字式で示されるデータの文字数を超える場合、関数値は文字列データ全体となります。

■SPACE\$ (個数)

【機能】 指定個数の空白文字列を得る。

【記述例】 A\$="CF"+SPACE\$ (5)+"BASIC"

【説明】 ・個数は0～255の範囲内ですが、0の場合は空文字列になります。

■STR\$ (数値式)

【機能】 数値を文字列に変換する。

【記述例】 B\$ = "A" + STR\$ (123)

【説明】 ・VAL関数はこの逆の機能となります。

■STRING\$ (文字数, $\left\{ \begin{array}{l} \text{キャラクタコード} \\ \text{文字式} \end{array} \right\}$)

【機能】 指定された1文字よりなる文字数分の文字列を得る。

【記述例】 A\$ = STRING\$ (10, "A")

【説明】 ・文字数およびキャラクターコードは0～255の範囲内ですが、文字数が0の場合は空文字列になります。
・文字式の結果が2文字以上の場合は、その先頭文字だけを使います。

■TAB (桁位置)

【機能】 指定桁位置まで空白を出力する。

【記述例】 PRINT "CF" TAB (10) "BASIC"

【説明】 ・桁位置は0～255まで指定できます。
・桁位置が1行の表示文字数を越えた場合は、1行表示文字数で割った余りを使います。
・現表示位置より小さい値を指定した場合は、次行になります。
・TABはPRINT、LPRINT文のみ有効です。

■VAL (文字式)

【機能】 文字列が表す数値に変換する。

【記述例】 A = VAL (A\$)

【説明】 ・文字列の最初の文字が+, -, &, ., 数字でない場合は関数値として0を返します。
・文字列中の空白は読みとばします。
・数字以外の文字が表れた場合には、それ以降の文字は無視されます。

<特殊関数>

■DATE\$

【機能】 日付をセットする、日付を与える。

【記述例】 DATE\$ = "89/05/23"
PRINT DATE\$

【説明】 ・記述例の1番目のようにして、一度セットすれば、いつでも現在の日付を読出すことができます。
・使用例
10 A\$ = DATE\$
A\$にはYY/MM/DD ("89/11/02" など) の形で代入されます。

■DAY

【機能】 曜日をN_dでセットする、または与える。

【記述例】 DAY = 1
W = DAY

【説明】 ・記述例の1番目のようにして、一度セットすれば、0から6の値が繰返されますので、日曜日から土曜日に対応させて使用できます。たとえば、0を日曜、1を月曜というように決めておきます。
記述例のWには単に0から6までのうちの数値が代入されます。

■EOF (ポート番号)

【機能】 ポート番号で指定されたポートのバッファが空であるかどうかを調べます。

【記述例】 IF EOF (2) THEN CLOSE#1 ELSE GOTO 100

【説明】 ・指定したポートのバッファが空であれば真 (-1) を、空でなければ偽 (0) を返します。
・ポート番号で指定したポートは入力モードでオープンされていなければなりません。

■ERL
ERR

【機能】 エラーの発生した行番号およびエラーコードを得る。

【記述例】 A = ERL
B = ERR

【説明】 ・エラーが発生すると、システム変数ERLはエラーを起こした行番号を、ERRはエラーコードを保持しています。
・直接モード実行時のエラーの発生番号は65535です。
・ERLとERRは、エラー処理ルーチン内でプログラムの流れ制御に使用できます。

■FRE (式)

【機能】 メモリの未使用領域の大きさを与える。

【記述例】 PRINT FRE (0)

【説明】 ・式が数式の場合は、プログラム領域の未使用サイズをバイト数で与えます。
・式が文字式の場合は、文字変数領域未使用サイズをバイト数で与えます。

■INKEY\$ [#ポート番号]

【機能】 キーが押されていればその文字を、押されていなければNULLコードを与える。

【記述例】 A\$=INKEY\$

【説明】 ・「CTRL」+「X」キー以外のみ有効です。
・ポート番号を省略したときは、ポート1を選択されたものとみなします。

■INPUT\$ (文字数 [, #ポート番号])

【機能】 キーボードより指定された文字数の文字列を読み込む。

【記述例】 A\$=INPUT\$ (10, #1)

【説明】 ・文字数は0～255の範囲内です。
・INPUT\$は「CTRL」+「X」キー以外のすべての文字を読み込むため、INPUT文やLINE INPUT文では読めないCR、LFコードもそのまま読み込みます。
・文字数のデータがくるまで、BASIC LEDが点滅して入力待ち状態であることを示します。
・ポート番号の項を省略したときは、ポート1を選択されたものとみなします。
・キーのキャラクターコードを表示するプログラム例

```

10 CLS
20 A$=INPUT$ (1)
30 A$=HEX$ (ASC (A$))
40 PRINT A$
50 GOTO 20

```

■LOC (ポート番号)

【機能】 ポートのバッファにたまっているデータ数をバイト数で与えます。

【記述例】 A=LOC (2)

【説明】 ・ポート番号で指定されたポートは、入力モードでオープンされていなければなりません。
・指定されたポートのバッファ内に受信されてたまっているデータの数を、バイト数で与えます。

■PEEK (番地)

【機能】 メモリ上の指定された番地の内容を読み出します。

【記述例】 A=PEEK (&H3000)

【説明】

- ・番地によって指定されたメモリ番地の内容を与えます。
- ・番地は0～65535 (&H0～&HFFFF) の範囲です。
- ・番地が小数点を含む場合には、小数点以下を四捨五入します。
- ・結果として、1バイトの整数が与えられます。
- ・&H0000～&H1FFF、&H8000～&HFFFFはシステムエリアですので、読出さないでください。
- ・メモリの構成については、8-3項のメモリマップを参照してください。

■TIME\$

【機能】 時刻をセットする、時刻を与える。

【記述例】 TIME\$="09:10:00"
PRINT TIME\$

【説明】

- ・記述例の1番目のようにして、一度セットすれば、いつでも現在の時刻を読み出すことができます。
- ・使用例1
10 PRINT TIME\$
時刻の時、分、秒がHH:MM:SSの形で表示されます。
HHは00から23までの24時間制です。MMとSSは00から59までの数です。
- ・使用例2
TIME\$="10:00:00"
ダイレクト実行で、現在の時刻をセットします。

■USR [<番号>] (<引数>) [, W]

【機能】 機械語で作られた関数を呼出す。

【記述例】 J=USR2 (I), W

【説明】

- ・ユーザの作成した機械語の関数ルーチン（ユーザ関数）を呼出します。
- ・ユーザ関数内では割込みを禁止しないでください。割込みを禁止した場合、正常に動作しません。
- ・ユーザ関数は、あらかじめメモリ内に準備されており、DEF・USR文によって、その実行開始番地が設定されている必要があります。
- ・番号は0から9まで設定でき、DEF・USR文により定義された番号と対応しています。
- ・番号が省略された場合は0と解釈します。
- ・引数により、BASICから機械語ルーチンへ値を受渡しをすることができます。
- ・, W指定をつけると、機械語の関数ルーチン内で、ウォッチドッグタイマのリフレッシュを行いませんので、ユーザでウォッチドッグタイマのリフレッシュを行ってください。
- ・, W指定を省略すると、機械語の関数ルーチン内でウォッチドッグタイマのリフレッシュを行います。
- ・ウォッチドッグタイマとは、プログラムの暴走を防ぐためのタイマです。
タイムアップすると、ユニットにリセットがかかり、C200H-CPUには“コウキノウ ユニット イジョウ”が表示されます。
タイムアップする前にタイマのリフレッシュを行うことにより、プログラムを継続してRUNさせることができます。
- ・ウォッチドッグタイマリフレッシュ方法
AIM #DF, 03
OIM #20, 03
必ず上記2ステップを90msec以内ごとに行ってください。
- ・ウォッチドッグタイマタイムアップ時間 Typ 120msec
Min 90msec

アキュムレータAと引数の型と関係

アキュムレータAの値	引数の型
2	整数型
3	文字型
4	単精度実数型
8	倍精度実数型

インデックスレジスタXは、引数の格納されているメモリのアドレスを示すものですが、引数の型により示すアドレスが以下に示すように異なります。

整数型

	←X
上位8ビット	
下位8ビット	

文字型

文字列の長さ	←X
格納されているアドレス上位	
格納されているアドレス下位	

単精度実数

指数部	←X
仮数上位8ビット	(MSBは常に1)
仮数中位8ビット	
仮数下位8ビット	
符 号 (最上位ビット)	

倍精度実数

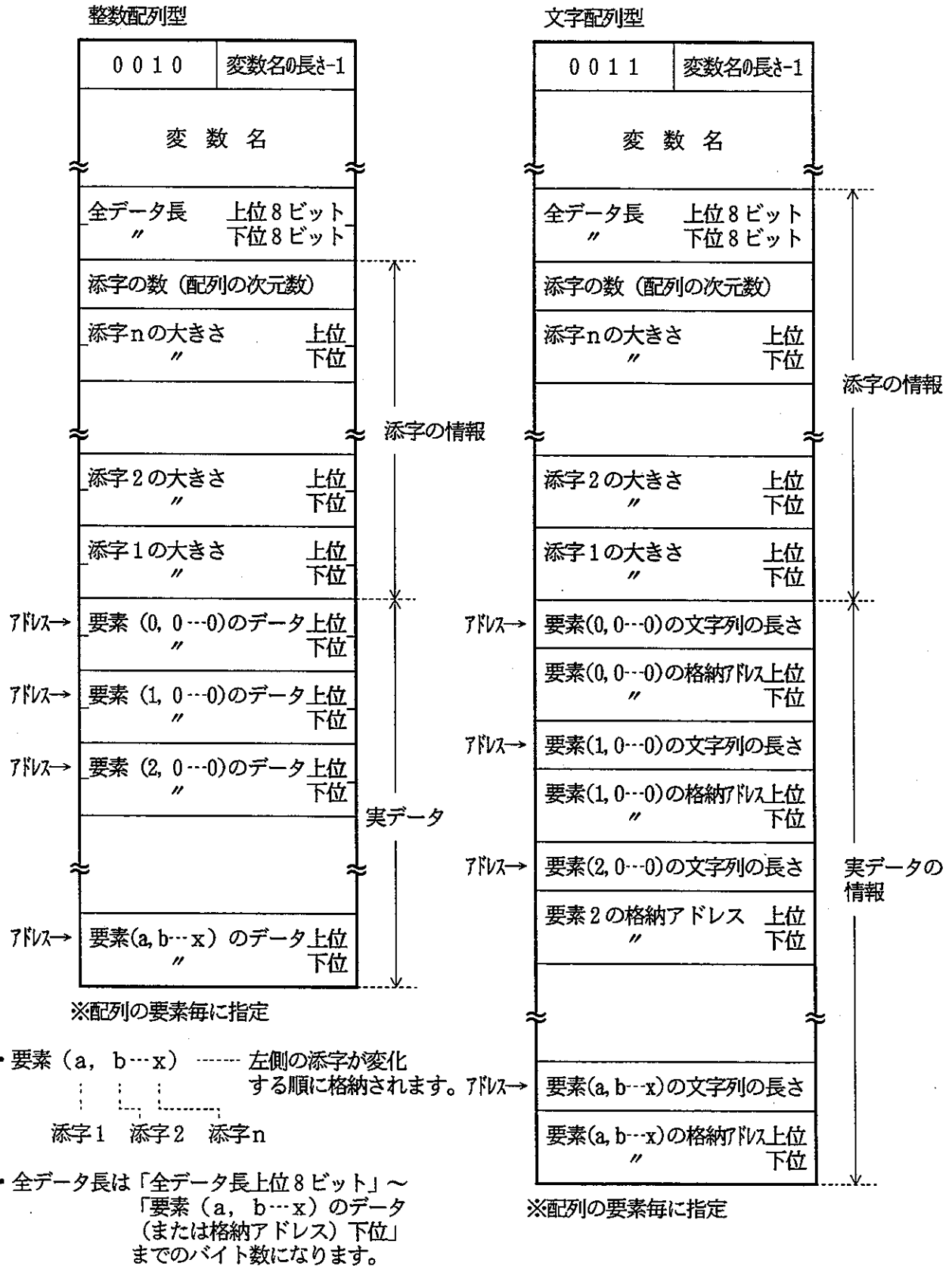
指数部	←X
仮数上位8ビット	(MSBは常に1)
仮数最下位8ビット	
符 号 (最上位ビット)	

【使用例】 BASICプログラム
 100 A%=&H1234
 110 DEF USR0=&H2000
 120 A%=USR(A%)
 130 PRINT A%
 140 END

機械語プログラム
 2000 PSHA
 2001 PSHX
 2002 LDD 2, X
 2004 ADD #10
 2007 STD 2, X
 2009 PULX
 2010 PULA
 2011 RTS

1	2		
3	4		

- ・機械語ルーチンに分岐すると、アキュムレータAに引数の型（整数型）2が入り、インデックスレジスタXに引数の格納されているメモリのアドレスが入ります。引数のデータをアキュムレータDにロードし、&H10を加えます。その結果を引数のデータエリアに書込みます。



第8章 機械語プログラム

8-1 機械語プログラムの作り方

■使用メモリエリア

機械語プログラムを作成する場合、BASICのプログラムおよびデータを壊すことのないように、BASICプログラム領域を使用しないでください。もし重なった場合、BASIC動作の異常時に機械語プログラムが破壊される可能性があります。

- ・メモリ領域の確保は、MSETコマンドによって行います。

■機械語プログラムの作成

機械語プログラムの作成は、以下の2つの方法があります。

(1) モニタ機能を使う。

(2) BASICのPOKE文によって直接メモリに書込む。

(2)は、数ステップの短いプログラムを作成するときに、使用します。また、作成した機械語プログラムは、モニタのS、Lコマンドを使用することにより、機械語プログラムファイルとして、保存したり、読出しすることができます。

SAVE、LOAD命令を使用し、機械語プログラムをEEP-ROMに保存することもできます。

■機械語プログラムの呼出し

(1)USR関数により行います。 USR [<番号>] (<引数>)

BASICには、機械語で書かれたプログラムを呼出す機能としてUSR関数があります。

機械語プログラムを作成・呼出しすることができます。

USR関数により呼出された機械語プログラムには引数として

アキュムレータA : <引数> の型

インデックスレジスタX : <引数> の格納されているアドレス

が渡されます。

ここでいう<引数>とは、USR関数により指定された<引数>を示します。

機械語プログラムからBASICプログラムへ戻るには、機械語のRTS (&H39) を用います。

このとき、以下の注意が必要です。

1) スタックポインタSの値がUSR関数が出された時点と等しくなっていること。

2) 引数が格納されていた位置と同じ書式で、データを格納していること。

このとき、データの値は変わっていてもかまいませんが、データの型は変えないでください。

(2)VARPTR関数により、USR関数で指定した引数以外の変数を機械語プログラムに渡すことができます。

《参 考》

- ・機械語プログラムでは、PCとのインターフェース機能 (BASIC命令のPC READ/ WRITEなど) に当たるものを作成することはできません。

《お願い》

- ・MSETコマンドによる機械語プログラムエリアの確保は、必ず機械語プログラムをメモリ上に格納する以前に行ってください。
- ・MSETコマンドは領域の確保と同時にBASICプログラムの移動を行うため、それ以前に格納されているBASICプログラムを壊すことはありません。
- ・機械語プログラム内では、割込みを禁止しないでください。割込みを禁止した場合、正常に動作しません。
- ・ASCIIユニットの故障の際を考慮し、作成されたプログラムを外部機器にて保存されることをお勧めします。

8-2 モニタモード

BASICモードからモニタモードに入ることによって、機械語モニタができます。

■モニタモードへの入り方/BASICモードへの戻り方

- BASICモードからモニタモードに入るには

READY
mon↵
*

READY表示で
mon↵とキー入力することでモニタモードに入ります。

- モニタモードからBASICモードへ戻るには、「CTRL」キーを押しながら、「B」キーを押すことでBASICモードに戻ります。

■モニタコマンド一覧

No.	コマンド	機 能
1	コマンドなし アドレス	指定アドレスのメモリ内容表示、変更
2	M	メモリ内容の転送
3	C	メモリ内容の比較
4	R	レジスタの表示、変更
5	BP	ブレイクポイントの設定、表示
6	N	ブレイクポイントの解除
7	I	逆アセンブラ
8	S	ポートへの出力
9	L	ポートからのロード
10	V	ポートとのベリファイ
11	G	プログラムの実行
12	T	プログラムのステップ実行
13	ミニアセンブラ	1ラインのアセンブラ
14	コマンドなし +、-	16進数の加減算

■コマンドの説明

1. (Dump)

【機能】 メモリの内容を16進数で表示します。
 ␣のみキーインの場合8バイトまで表示します。

【入力形式】 [〈表示開始アドレス〉]. [〈表示終了アドレス〉]

【例】 (1)* 4 0 0 0␣

4 0 0 0 - 1 0

(2)*␣

* 2 0 3 0 5 0 6 0 7 0 8 0 9 0

*␣

4 0 0 8 - A 0 B 0 C 0 D 0 E 0 F 0 0 0 1 0

(3)*. 4 0 1 A␣

4 0 1 0 - 0 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8

4 0 1 8 - 1 2 3 4 5 6

(4)*. 3 0 0 0␣

4 0 1 B - 7 8

(5)* 3 0 0 0 : 9 8 7 6 5 4 3 2 1␣

* 3 0 0 0 . 3 0 0 7␣

3 0 0 0 - 0 9 0 8 0 7 0 6 0 5 0 4 0 3 2 1

(6)* : 1 1 2 2 3 3 4 4 5 5␣

* 3 0 0 0 . 3 0 0 7␣

3 0 0 0 - 1 1 2 2 3 3 4 4 5 5 0 4 0 3 2 1

【説明】 (1)指定アドレスのメモリ内容を1バイトで表示します。
 (2)直前に表示したアドレスの次から8バイト分のメモリ内容を表示します。
 (3)直前に表示したアドレスの次から指定アドレスまでのメモリ内容を表示します。
 (4)直前に表示したアドレスより指定アドレスが小さい場合。
 直前に表示したアドレスの次のアドレスを表示します。
 (5)指定アドレスから順にデータをセットしていきます。
 (6)指定アドレスがない場合、直前に表示したアドレスから順にデータをセットしていきます。

2. M (Move)

【機能】 転送開始アドレスから転送終了アドレスまでのメモリの内容を転送先アドレス以降に転送します。

【入力形式】 M [〈転送先アドレス〉] < [〈転送開始アドレス〉, 〈転送終了アドレス〉]

【例】 *M 3000<4000, 4007↵

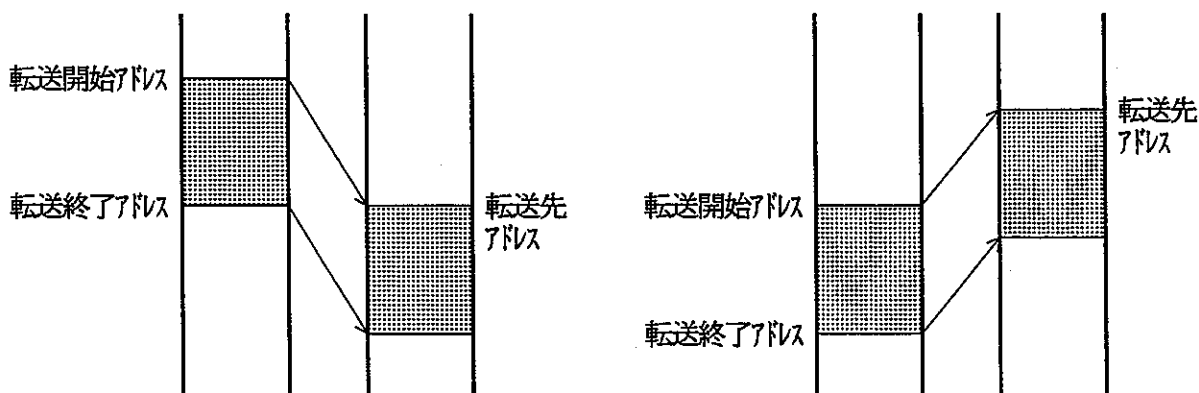
*4000, 4007↵

4000-01 02 03 04 05 06 07 08

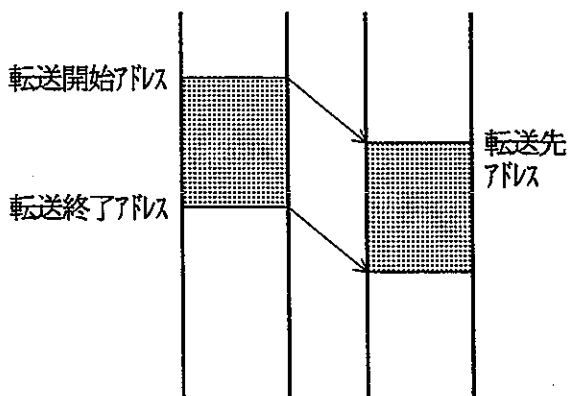
*3000, 3007↵

3000-01 02 03 04 05 06 07 08

【説明】 4000から4007の内容を転送先アドレス3000へ順に転送します。
 このとき、転送前のエリアと転送後のエリアが重ならないようご注意ください。
 転送前後のエリアが重なると（転送開始アドレス<転送先アドレス≤転送終了アドレスのとき）、データが正しく転送されません。



(例1) 上図の例ですと、データが正しく転送されます。



(例2) 上図の例ですと、転送開始アドレス<転送先アドレス≤転送終了アドレスとなり、データが正しく転送されません。

3. C (Compare)

【機能】 比較元開始アドレスから終了アドレスまでのメモリ内容と比較対象アドレスからのメモリ内容とを比較し、内容が異なっている場合は、アドレスとメモリの内容を表示します。

【入力形式】 C [〈比較対象アドレス〉] < [〈比較元開始アドレス〉、〈比較元終了アドレス〉]

【例】 *C 3000 < 4000, 4007↵

4003-FF (03)

*3000, 3007↵

3000-00 01 02 03 04 05 06 07

*4000, 4007↵

4000-00 01 02 FF 04 05 06 07

*

【説明】 3000～3007と4000～4007を比較し、3003と4003の値が違った場合です。この場合、4003のデータFFと3003のデータ03を表示します。

4. R (Register)

【機能】 すべてのレジスタを表示します。
レジスタの値を変更します。

【入力形式】 R
[C] [A] [B] [X] [S] [P] =データ

【例】 (1)*R↵

C-C0 A-00 B-01 X-ABCD S-2EFF P-5000

*

(2)*A=12↵

*X=FF00↵

*R↵

C-C0 A-12 B-01 X-FF00 S-2EFF P-5000

*

【説明】 (1)レジスタを表示します。
(2)指定レジスタを指定データに書替えます。

5. BP (Break Point)

【機能】 アドレスをブレイクポイントとして設定します。

【入力形式】 BP

【例】 (1)*BP 3 0 0 0↵

(2)*BP↵

BP=3 0 0 0 0 0 0 0

(3)*BP 5 0 0 0↵

*BP↵

BP=5 0 0 0 3 0 0 0

【説明】 (1)ブレイクポイントを設定します。
(2)現在設定しているブレイクポイントを表示します。
(3)ブレイクポイントは2ヶ所まで設定できます。

6. N (New)

【機能】 ブレイクポイントをすべて解除します。

【入力形式】 N

【例】 (1)*N↵

*BP↵

BP=0 0 0 0 0 0 0 0

【説明】 直前までに設定されていたブレイクポイントを解除します。

7. I (Inverse Assembler)

【機能】 アドレスより20行の内容を逆アセンブルし表示します。

【入力形式】 I<アドレス>

【例】 (1)*I 3000↵

3000-CE	10 00	LDX	#\$1000	} = 20行
3003-FF	40 00	STX	\$4000	
3006-86	80	LDAA	#\$80	
3030-81	12	CMPA	#\$12	

(2)*I, I↵

3032-26	02	BNE	\$3036	} = 2 × 20 = 40行
3034-A7	00	STAA	\$00, X	
3036-39		RTS		
3080-08		INX		

【説明】 (1)指定アドレスより20行分の内容を逆アセンブルします。
(2)Iを複数個書くと $20 \times (I \text{の個数})$ 行を逆アセンブルします。

8. S (Save)

【機能】 開始アドレスから終了アドレスまでの内容をポート1にSフォーマットで転送します。

【入力形式】 S [<開始アドレス>]. [<終了アドレス>]

【例】 (1)*S3000.300F↵

(2)「START/STOP」スイッチを押す。

S113300086FF971686009717CE0008260001010157
S9030000FC

*

【説明】 (1)本コマンドが起動されると送信待ち状態になりますので、ポート1にRS-232C機器を接続し、受信準備操作を行ってください。
(2)受信準備ができたなら、START/STOPのスイッチを押してください。送信を開始します。

9. L (Load)

【機能】 ポート1からSフォーマットのオブジェクトをロードし、メモリにストアします。

【入力形式】 L <オフセット>

【例】 (1)*L↵

(2)*L 100↵

「START/STOP」スイッチを押す

*3100.310F↵

3100-CE 00 00 08 26 FD 08 26

3108-FD 86 55 97 17 CE 00 00

【説明】 (1)ポート1からのSフォーマットオブジェクトをロードし、メモリにストアします。
(2)コマンドの後にオフセットを付けると、ロードしたアドレスにオフセットを加えたアドレスにオブジェクトをストアします。
本コマンドが起動されると、受信準備状態になりますので、ポート1にRS-232C機器を接続し、送信準備操作を行ってください。
送信準備ができたなら、「START/STOP」スイッチを押してください。受信可能になりますので送信を開始してください。
送信が終了したら、ターミナルにつなぎかえ、「CTRL」+「X」キーをキー入力してください。

10. V (Verify)

【機能】 ポート1からの内容とメモリ内容を比較します。

【入力形式】 V <オフセット>

【例】 (1)*V↵

(2)*V 100↵

「START/STOP」スイッチを押す

3120-12

【説明】 (1)メモリ内容と、ポート1から入ってきたSフォーマットで指定されたアドレスの内容を比較します。
(2)コマンドの後にオフセットを付けた場合は、Sフォーマットで指定されたアドレスにオフセットを加えたアドレスの内容と比較します。
比較を行い一致していないときは、そのアドレスとデータを表示します。
本コマンドが起動されると、受信準備状態になりますので、ポート1にRS-232C機器を接続し、送信準備操作を行ってください。
送信準備ができたなら、「START/STOP」スイッチを押してください。受信可能になりますので送信を開始してください。受信を開始しメモリ内容との比較を行います。
比較を行い一致していないときは1ERR LEDが点灯しますので、接続しているRS-232C機器をターミナルとつなぎかえてください。1ERR LEDが消灯します。
「START/STOP」スイッチを押すと、一致していないアドレスとデータを表示します。
比較を行い、一致しているときは、送信を終了しても「1ERR LED」が点灯しませんので、接続しているRS-232C機器をターミナルとつなぎかえ、「CTRL」+「X」キーをキー入力してください。

11. G (Go)

【機能】 ユーザプログラムの実行を行います。
 アドレスを指定した場合、そのアドレスより実行します。
 アドレス指定のない場合は、現在のプログラムカウンタの示すアドレスより実行します。
 プログラムの実行は、ブレイクポイント、SWIの実行、割り込みの発生により中断され、レジスタの内容を表示します。
 スタックポインタを機械語エリア以外に設定していると正常に動作しませんので、必ず機械語エリアに設定してください。

【入力形式】 G <アドレス>

【例】 *I 3000↵

3000-86 80 LDAA #\$80

3002-B7 40 00 STAA \$4000

3005-20 F9 BRA \$3000

⋮

*BP 3005↵

*G 3000↵

C-C8 A-80 B-FF X-0000 S-2EFF P-3005

*

12. T (Step)

【機能】 プログラムを1ステップのみ実行します。
 アドレスを指定した場合、そのアドレスから1ステップのみ実行します。
 指定なしのときは、現在のプログラムカウンタ (PC) の示すアドレスから1ステップのみ実行します。
 複数のステップを実行させるときは、Tをステップ数分T、Tキーインしてください。
 スタックポインタを機械語エリア以外に設定していると正常に動作しませんので、必ず機械語エリアに設定してください。

【入力形式】 T <アドレス>

【例】 *T 3000↵

3000-86 80 LDAA #\$80

C-C8 A-80 B-00 X-0000 S-2EFF P-3002

*

13. ミニアセンブラ

【機能】 1ラインのアセンブラです。
 1行入力するごとに、ソースプログラムがアセンブルされ、オブジェクトがメモリに格納されます。アセンブルした結果を逆アセンブルし表示します。
 相対アドレスは、アセンブル時に計算します。

【起動方法】 CTRL+A↵

【終了方法】 X↵

【例】 *CTRL+A↵

```
! 3000: LDAA #$80↵
3000-86 80 LDAA #$80
! LDAB #$7F↵
3002-C6 7F LDAB #$7F
! STD $4000↵
3004-FD 40 00 STD $4000
! ASLA↵
3007 48 ASLA
! BNE $3000↵
3008 26 F6 BNE $3000
! X↵
*
```

(注) CTRL+Aは、「CTRL」キーを押しながら「A」を押すことです。

《参考》

- ・メモリエリア&H0000~&H1FFF、&H8000~&HFFFFまでは、ASCIIユニットのシステムで使用しているので、ユーザでは使用しないでください。
 使用すると誤動作する場合があります。

14. 16進数の加減算

【機能】 16進の4桁のデータの加算、減算を行います。

【入力形式】 <16進データ> + <16進データ>

<16進データ> - <16進データ>

【例】 * 1 2 3 4 + 5 6 7 8↵

* 1 2 3 4 + 5 6 7 8 = 6 8 A C

* A B C D + E F 0 1↵

* A B C D + E F 0 1 = 9 A C E

* 1 2 3 4 - 5 6 7 8↵

* 1 2 3 4 - 5 6 7 8 = B B B C

* A B - 1 2↵

* A B - 1 2 = 0 0 9 9

8-3 メモリマップ

■メモリ構成

メモリ構成は以下のようになっています。番地は&H0000~&HFFFF (0~65535) で、バイト単位の構成です。&H2000~&H7FFFの24Kバイト (24576バイト) がプログラムエリアです。このプログラムエリアは、PEEK関数 (7-60ページ参照) によって読み出し可能です。&H0000~&H1FFF、&H8000~&HFFFF (下図  部) はシステムで使用されるため、読み出し禁止です。

7 0		メモリ構成の説明	
		メモリエリア	内 容
&H0000	I/O エリア1	I/Oエリア1 (&H0000~&H001F)	マイクロプロセッサ63B03の内部 ポートのエリアです。
&H0020	システム ワーク	システムワーク (&H0020~&H1FFF)	システムで使用しているRAMエリア です。
&H2000	機械語 プログラム エリア	機械語 プログラム エリア	機械語プログラムが格納されるエリア です。 MSET命令によってエリアの大きさが 変化します。
MSET 命令に→ よって 設定	BASIC テキスト エリア	BASIC テキスト エリア	BASICプログラムが中間言語コード で格納されているエリアです。 MSET命令 (7-15ページ参照) に よってエリアの大きさが変化します。
CLEAR 命令に よって 設定	システム スタック エリア	システム スタック エリア	システムで使用しているスタックエリ アです。
→	文字列領域	文字列領域	文字列が格納されているエリアです。 通常200バイト。この領域の大きさは CLEAR命令 (7-19ページ参 照) によって設定します。
&H8000	共有メモリ エリア	共有メモリエリア (&H8000~&H8FFF)	PCとのインターフェース用のRAM エリアです。 このエリアをアクセスすると高機能ユ ニット異常となることがありますので、 アクセスしないでください。
&H9000	I/O エリア2	I/Oエリア2 (&H9000~&H9FFF)	ACIA、PTM、RTCのポートが 割付けられているエリアです。
&HA000	システム エリア	システムエリア (&HA000~&HFFFF)	システムROMのエリアです。
&HFFFF			

 : プログラムエリア

 : システム使用エリア

■ポートアドレス一覧

●ポート1 HD63B03 SCI使用

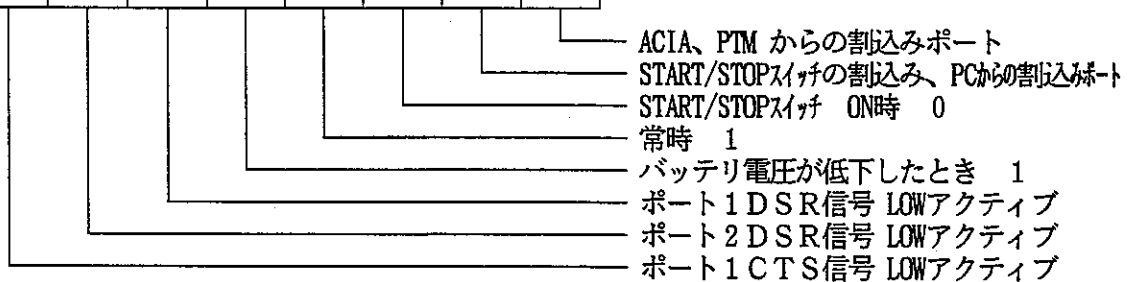
アドレス	R/W	内 容	システム初期設定値
\$0010	R/W	転送レート/モードコントロールレジスタ	\$34
\$0011	R/W	TX/RXコントロールステータスレジスタ	\$00
\$0012	R	レシーブデータレジスタ	——
\$0013	W	トランスミットデータレジスタ	——

●ポート2 HD63B50 ACIA使用

アドレス	R/W	内 容	システム初期設定値
\$9400	R	ステータスレジスタ	——
\$9400	W	コントロールレジスタ	\$11
\$9401	R	レシーブデータレジスタ	——
\$9401	W	トランスミットデータレジスタ	——

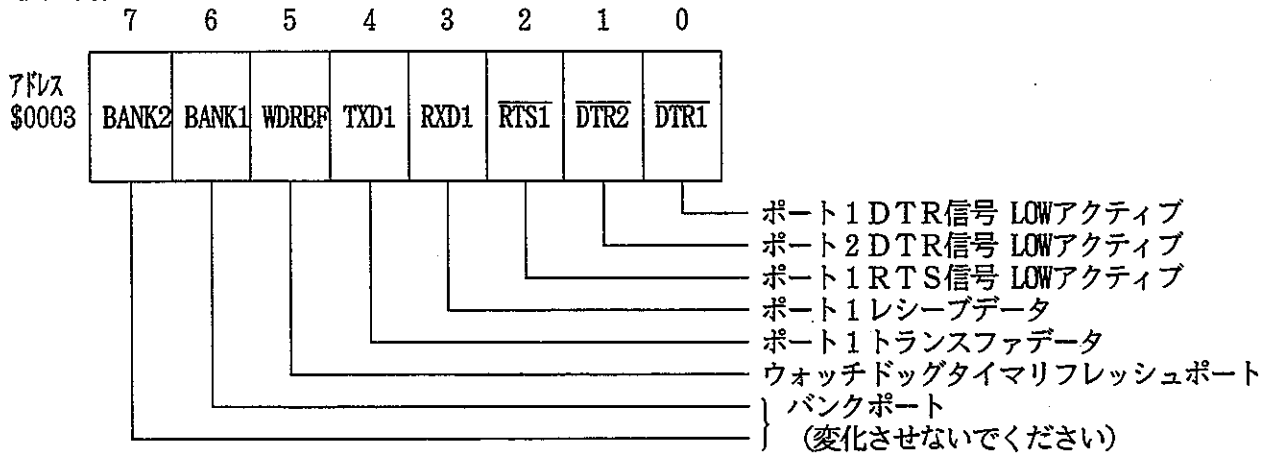
(入 力)

	7	6	5	4	3	2	1	0
アドレス \$0015	CTS1	DSR2	DSR1	BAT LOW	1	START /STOP	TRQ2	TRQ1



8-3 メモリマップ

(出力)



●PTM HD63B40

アドレス	R/W	内 容	システム初期設定値	備 考
\$9800	R	0	——	
	W	コントロールレジスタ #1, #3	\$82	#3にライト
\$9801	R	ステータスレジスタ	——	
	W	コントロールレジスタ #2	\$00	
\$9802	R	タイマ#1カウンタ上位バイト	——	
	W	上位バイト(MSB)バッファレジスタ	——	
\$9803	R	下位バイト(LSB)バッファレジスタ	——	
	W	タイマ#1ラッチ下位バイト	——	
\$9804	R	タイマ#2カウンタ上位バイト	——	
	W	上位バイト(MSB)バッファレジスタ	——	
\$9805	R	下位バイト(LSB)バッファレジスタ	——	
	W	タイマ#2ラッチ下位バイト	——	
\$9806	R	タイマ#3カウンタ上位バイト	——	
	W	上位バイト(MSB)バッファレジスタ		ポートの値によって変化
\$9807	R	下位バイト(LSB)バッファレジスタ	——	
	W	タイマ#3ラッチ下位バイト	——	

●RTC-62461 リアルタイムクロック

アドレス	R/W	内 容	システム初期設定値
\$9000	R/W	1秒桁 0~9	——
\$9001	R/W	10秒桁 0~5	——
\$9002	R/W	1分桁 0~9	——
\$9003	R/W	10分桁 0~5	——
\$9004	R/W	1時桁 0~9	——
\$9005	R/W	PM/AM 10時桁 0~2	——
\$9006	R/W	1日桁 0~9	——
\$9007	R/W	10日桁 0~3	——
\$9008	R/W	1月桁 0~9	——
\$9009	R/W	10月桁 0~1	——
\$900A	R/W	1年桁 0~9	——
\$900B	R/W	10年桁 0~9	——
\$900C	R/W	週 桁 0~6	——
\$900D	R/W	コントロールレジスタ D	D0に0セット
\$900E	R/W	コントロールレジスタ E	——
\$900F	R/W	コントロールレジスタ F	D0、1、3に0セット

MPUへの供給クロック 4.9152MHz
 ACIA、PTMへの供給クロック 1.2288MHz
 ACIAへの TXCLK はPTMから供給
 RXCLK

詳細は各IC説明書をご覧ください。

●送受信関係ワーク

アドレス	内 容	
\$ 0 1 4 5	ポート 1	データ格納ポインタ (受信)
\$ 0 1 4 6		データ抽出ポインタ (受信)
\$ 0 1 4 7		データ格納ポインタ (送信)
\$ 0 1 4 8		受信バッファ 2 5 6 B
\$ 0 2 4 B	ポート 2	データ格納ポインタ (受信)
\$ 0 2 4 C		データ抽出ポインタ (受信)
\$ 0 2 4 D		データ格納ポインタ (送信)
\$ 0 2 4 E		受信バッファ 2 5 6 B
\$ 1 4 4 0	ポート 1	送信バッファ 2 5 6 B
\$ 1 5 4 0	ポート 2	送信バッファ 2 5 6 B

第9章 異常とその処置

9-1 エラーメッセージ一覧

BASIC命令実行中エラーが発生するとパーソナルコンピュータなどの画面に下表に示すメッセージを表示します。

- (注) ポート1がターミナル以外の装置で、開かれている間にBASIC命令の実行が停止すると、ASCIIユニットは各種メッセージの表示待ちとなります。
 上記の場合、ポート1にパーソナルコンピュータを接続して「CTRL」+「X」キーをキーインすると、各種メッセージをパーソナルコンピュータの画面に表示します。
 「CTRL」+「X」は「CTRL」キーを押しながら、「X」キーを押すことを示します。

【表示例】

SYNTAX ERROR IN ××××

- ・番号ありで実行したとき表示されます。
- ・番号なしで実行したとき表示されません。

■エラーメッセージ

エラーメッセージ	エラーコード	エラー内容とその処置
BAD DATA IN PORT ERROR	58	ポートから読出されたデータの形式が間違っています。
BAD I/O MODE ERROR	51	ポートの指定または周辺装置の指定が間違っています。 ポートの指定または周辺装置の指定を確認してください。
BAD PORT DESCRIPTOR ERROR	55	ディスクリプタが間違っています。 ディスクリプタを確認してください。
BAD PORT NUMBER ERROR	50	ポート番号が間違っています。 ポート番号を確認してください。
BAD SUBSCRIPT ERROR	9	規定の範囲外の添え字を使っています。 DIM命令で使用する最大値の添え字を割当ててください。
CAN'T CONTINUE ERROR	17	プログラムの実行が再開できません。 再度、RUN命令等で実行を開始させてください。
DEVICE I/O ERROR	53	周辺装置との通信で、エラーが発生しました。 通信回線を確認してください。
DEVICE UNAVAILABLE ERROR	60	不正なデバイス名を使用しています。 デバイス名を確認してください。
DIVISION BY ZERO ERROR	11	0で除算を行っています。 除算の確認をし、プログラムを修正してください。

エラーメッセージ	エラーコード	エラー内容とその処置
DIRECT STATEMENT IN PORT ERROR	56	プログラムのLOADのとき、行番号のない行を読み込みました。 転送するプログラムを確認してください。
DUPLICATE DEFINITION ERROR	10	配列、ユーザ関数を2重に定義しました。 どちらかの配列、ユーザ関数を別のものに变更してください。
FORMAT ERROR	67	フォーマットまたは、サブコマンドが間違っているか、PC側の転送チャンネル数、転送元先頭チャンネルNoが不正です。 ASCIIユニット、PCのプログラムを確認してください。
FOR WITHOUT NEXT ERROR	23	NEXT文が足りません。 FOR~NEXTの組合せを確認してください。
ILLEGAL DIRECT ERROR	12	ダイレクトモードでは実行できないステートメントを実行しようとしてしました INPUT、LINE INPUTはBASICプログラムの中でしか実行できません。
ILLEGAL FUNCTION CALL ERROR	5	ステートメントや関数の呼び方が間違っています。 ステートメントや関数の呼び方を確認してください。
INPUT PAST END ERROR	54	ポートのデータを読みつくしました。
MISSING OPERAND ERROR	22	必要なパラメータがありません。 プログラムを確認してください。
NEXT WITHOUT FOR ERROR	1	FOR文が足りません。 FOR~NEXTの組合せを確認してください。
NO RESUME ERROR	19	エラー処理ルーチンにRESUME文が足りません。 RESUME文を追加してください。
NO SUPPORT ERROR	64	サポートしていません。
OUT OF DATA ERROR	4	READ文で読むべきデータがありません。 READ文の変数の個数と、DATA文の定数の個数を確認してください。
OUT OF MEMORY ERROR	7	メモリ容量が足りなくなりました。 CLEAR命令、MSET命令でBASICプログラム領域を拡張するか、プログラムの見直しを行ってください。
OUT OF STRING SPACE ERROR	14	文字領域が足りません。 CLEAR命令で文字領域を拡大してください。
OVERFLOW ERROR	6	数値が許される範囲を越えています。 数値の範囲を確認してください。

エラーメッセージ	エラーコード	エラー内容とその処置
PORT ALREADY OPEN ERROR	52	その番号のポートはすでに開かれています。 OPEN文で同じポートを続けて2回開いています。 OPEN文を1つ削除してください。
PORT NOT OPEN ERROR	57	OPENしていないポート番号または入出力の指定を使いました。 OPEN文でポートを開いてください。
PROM ERROR	65	装着されているEEP-ROMが異常、またはEEP-ROMに何も書込まれていません。
PROTECTED PROGRAM ERROR	62	プログラムが保護されています。 プログラムを変更される場合は、PNAME命令で名前を消してください。
RESUME WITHOUT ERROR	20	エラーが発生していないのにRESUME文を実行しました。 プログラムの構成を確認してください。
RETURN WITHOUT GOSUB ERROR	3	GOSUB文の実行以前にRETURN文に出会いました。 プログラムの構成を確認してください。
STRING FORMULA TOO COMPLEX ERROR	16	文字式が複雑すぎます。 文字式を単純にするか、文字式を分けてください。
STRING TOO LONG ERROR	15	1文字変数に255文字を越えて代入しようとしています。 プログラムを修正してください。
SYNTAX ERROR	2	プログラムが文法にそっていません。 プログラムを修正してください。
TYPE MISMATCH ERROR	13	変数の型が合いません。 変数の型を確認してください。
UNDEFINED LINE NUMBER ERROR	8	行番号の指定が間違っています。 分岐先等の行番号を確認してください。
UNDEFINED USER FUNCTION ERROR	18	ユーザ関数が定義されていません。 DEF USR命令で実行開始番地を定義してください。
VERIFY ERROR	66	EEP-ROMの照合時エラーが発生しました。

9-2 異常とその処置

■異常一覧

異常項目	原因	処置
表示が全て点灯していない	PC本体の電源がOFFとなっている。	PC本体の電源をONにしてください。
	ASCIIユニットがしっかりと固定されていない。	ASCIIユニットを正しく装着してください。
	高機能I/Oユニットのいずれかが不良。 (高機能I/O立上り待ち) このときPCは運転を開始しません。	異常のある高機能I/Oユニットを交換してください。 異常ユニットは「I/Oテーブル読出し」操作にて「\$」のみの表示になっています。
	高機能I/Oユニットの号機No設定が重複している。 (I/Oユニットオーバー) このときPCは運転を開始しません。	号機Noが重複しないように設定し直してください。 号機Noは「I/Oテーブル読出し」操作で見ることができます。
	PC本体→ASCIIユニット間のリフレッシュが正常に行われなかった。 (高機能ユニット異常) このときASCIIユニットのみ運転を停止します。	原因解除後、号機Noに応じた補助記憶リレーAR0100～0109でリスタート(OFF→ON→OFF)をかけてください。 リスタートをかけても復帰しない場合は、ユニットを交換してください。
「ERR」 (異常) 表示点灯	接続機器の電源がOFFとなっている。	接続機器の電源をONにしてください。
	接続機器への接続ケーブルが外れている。	接続ケーブルを正しく接続し、ねじ止め固定してください。
	接続ケーブルの断線、コネクタの接触不良	接続ケーブルの修理または、交換をしてください。
	接続機器とASCIIユニットの伝送速度、通信条件があていない。	接続機器とASCIIユニットの伝送速度、通信条件を合わせてください。
「1ERR」 (異常) 表示点滅	バッテリーのコネクタが外れている。	バッテリーのコネクタを正しく接続してください。
	バッテリーの電圧が低下している。	バッテリーを新しいものと交換してください。
初期画面で《PROGRAM MEMORY ERROR》が表示され、「CTRL」+「X」が効かない。	BASICプログラムメモリが破壊されている。	「CTRL」+「I」をキーインしてください。 BASICプログラムメモリがクリアされます。 (EEP-ROMにプログラムがバックアップされていれば、この後「LOAD」命令で復旧することができます。)
正常にプログラムできない。	システムワークが破壊されている。	MSET &H2000↵ MON↵ 2000:00000000↵ :00000000↵ を実行してからリスタートをかけてください。 (PCからのARリレー) 初期画面に《PROGRAM MEMORY ERROR》が表示されます。

(注) 「CTRL」+「X」および「CTRL」+「I」は、「CTRL」キーを押しながら「X」または「I」キーを押すことを示します。

■点検項目

おもな点検項目は、下表のとおりです。

No.	点検項目	点検内容	判定基準	備考
1	周囲環境	周囲温度（盤内温度）は適当か。	0～+55℃	温度計
		周囲湿度（盤内湿度）は適当か。	35～+85%RH結露のないこと	湿度計
		ほこりなどが積もっていないか。	ないこと	目視
2	取付け状態	接続ケーブルのコネクタのビスはゆるんでいないか。	ゆるみのないこと	—ドライバー—
		接続ケーブルは切れかかっているか。	外観異常のないこと	目視

■保守部品

●予備品

故障発生時、すみやかな修復作業を行うための予備のユニットをご用意ください。

●寿命品

・バッテリーセット：形C200H-BAT09

・電池の寿命は25℃で5年間です。

それ以上高い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。

・バッテリーの寿命がなくなると「1ERR」LEDが点滅し、バッテリー異常フラグ（n+3cHの06ビット）がONします。（n=100+10×号機No）

・「1ERR」LEDが点滅後、1週間以内に新しいバッテリーと交換してください。

●バッテリーの交換方法

①電源をOFFにしてください。

最初から電源が入っていないときは、1分以上一旦電源を投入した後、電源をOFFにしてください。

②ASCIIユニットを取外してください。

ベースユニットのロックレバーをドライバなどで下に押しながらユニットを取外してください。

③⊕ドライバを用いてケース矢印部（2箇所）のネジを取外してください。（図1）

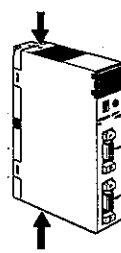
④⊖ドライバを用いてリアケースを取外してください。（図2）

⑤基板を取出してください。

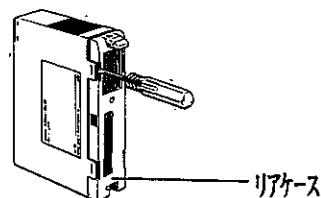
部品やパターンに直接触れないようにするなど、静電気に注意して取扱ってください。

⑥バッテリーはコネクタごと引抜き、新しいバッテリーと交換してください。交換は5分以内に行ってください。（図3）

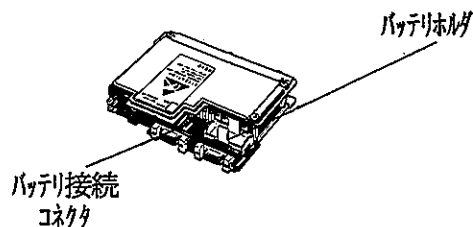
⑦以上の逆の手順で、ユニットを組立てベースユニットに取付けてください。



(図1)



(図2)



バッテリーホルダ
バッテリー接続コネクタ
バッテリーセット
形C200H-BAT09

(図3)

■取扱いの際の注意事項

●ユニット交換は電源を切った後、行ってください。

●ユニット交換した場合、再度ユニットが異常でないか確認してください。

●不良ユニットを修理返却される場合、異常現象をできるだけ詳細に記入し、現品に添付して返送してください。

●ASCIIユニットが故障した場合は、ASCIIユニット内部のRAMまたはEEPROMに書込まれたプログラムを全く読出せないことがあります。作成されたプログラムを外部機器（パーソナルコンピュータ、ラップトップパソコンなど）にて保存されることをお勧めします。

プログラムの転送方法については、「5-4 BASICプログラムの転送」の項を参照ください。

形式一覧表

No.	名 称	仕 様	形 式
①	ASCIIユニット	EEP-ROMタイプ	形C200H-ASC02

●オプション・メンテナンス部品

No.	名 称	仕 様	形 式
②	バッテリーセット	C200H専用 バッテリーバックアップ用	形C200H-BAT09
③	接続ケーブル	9ピン→25ピン変換用 ケーブル長2m C200H ASCIIユニット専用	形C200H-CN223

●周辺装置

No.	名 称	仕 様	形 式
④	ファクトリー コンピュータ FC-987	CPU本体	形3G8F6-CPU01
		キーボード	(専用防塵用カバー別売) 形3G8F6-KB001
		CRT	14インチカラー耐環境性強化 形3G8F4-CD141
⑤	バーコードリーダ	コントローラ	形3X2CB-ZMA
		カメラ	形3X2CA-ZLHV
		標準レンズ	形23Y0111C
		ケーブル	形3X2C9-2
		光 源	形3X2CL-D01A

●ASCIIユニットのプログラムを作成するのに、通常F I T 1 0ターミナルパックまたはN 8 8 - D I S K - B A S I Cが必要でしたが、Windows 9 5に標準添付されているハイパーターミナル等のアクセサリを利用して作成することが可能です。

- ## ■セットアップ

- DOS/V

3	SD
2	RD
7	RTS
8	CTS
6	DSR
4	DTR
5	GND

C200H-ASC02

2	SD
3	RD
4	RTS
5	CTS
7	DSR
8	DTR
9	GND

(a) (b)

(a) D-sub メス 9pin	7-ド : XM2 S-0913
	コネク : XM2 D-0901
(b) D-sub オス 9pin	7-ド : XM2 S-0911
	コネク : XM2 A-0901
(c) D-sub メス 25pin	7-ド : XM2 S-2511
	コネク : XM2 A-2501

- PC-98

2	SD
3	RD
4	RTS
5	CTS
6	DSR
20	DTR
7	GND

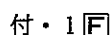
C200H-ASC02

2	SD
3	RD
4	RTS
5	CTS
7	DSR
8	DTR
9	GND

(c)

-

〔電話番号〕 :
接続方法 “Com1へダイレクト”
→ [OK]



〔COM1のプロパティ〕：
 ビット/秒 “9600”
 データビット “8”
 パリティ “なし”
 ストップビット “2”
 フロー制御 “なし”
 →〔OK〕

〔ファイル〕〔プロパティ〕〔設定〕〔ASCII設定〕：
 デレイ（行） “300”
 →〔OK〕

- ・他の設定は標準のままでOKです。
- ・2回目以降は、〔接続の設定〕のところで設定された名前のアイコンから起動することで、再設定は不要になります。
- ・そのパソコンにすでに“モデムの設定”がされている場合は、〔接続の設定〕から設定がはじまります。

ASCIIユニットとのターミナル接続

パソコンからCtrl+Xを入力し、以下のメッセージが出たら接続完了です。

C200H-ASC02 (CF-BASIC) V1.6 1994. 12. 28
(C) Copyright OMRON Corporation 1990
READY

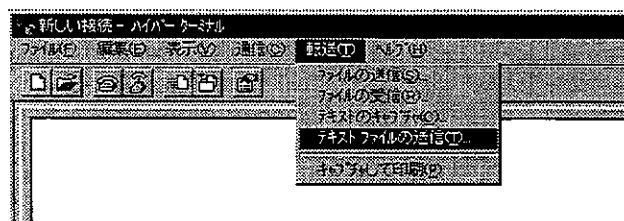
■オペレーション

●プログラムの作成

プログラムは、メモ帳等のテキストエディタを使用して作成し、テキスト形式で保存しておきます。

●プログラムをASCIIユニットに転送する

- (1) ユニット内のプログラムをNEW命令で消去します。
- (2) 〔転送〕〔テキストファイルの送信〕を選択し、保存しておいたプログラムを転送します。

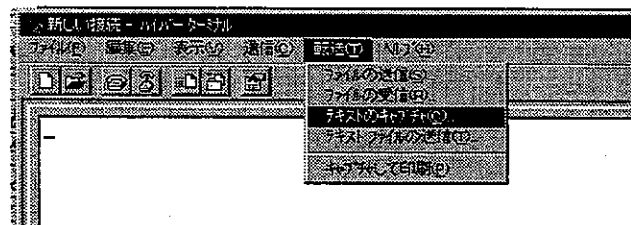


●ASCIIユニット内のプログラムをパソコンへ転送する

- (1) 次のように入力します。

SAVE #1, "COMU: (43)" ↵

- (1) 〔転送〕〔テキストのキャプチャ〕を選び、プログラムをセーブするためのファイル名を指定します。
- (2) ASCIIユニット前面のSTART/STOPボタンを押すと、プログラムの転送が開始します。
- (3) プログラム転送が最後まで終わったら、
 〔転送〕〔テキストのキャプチャ〕
 〔停止〕を選択し、最後に
 Ctrl+Xを入力します。



ターミナルを使用したASCIIユニットの機械語プログラミング方法

Windowsのターミナルを使用してASCIIユニットにて機械語プログラムのプログラミングをする方法について解説します。

ASCIIユニットのセットアップ方法、BASICプログラムのプログラミング方法については、「Windows 95 ハイパーターミナルでASCIIユニットのプログラミングをする方法」をご覧ください。

1. 準備

- (1) マニュアルまたは資料を参照してケーブルの準備、設定を行い、ターミナル接続ができる環境にします。
- (2) MSET命令を使い、ユーザメモリエリア（&H2000～&H7FFF）中に機械語プログラムエリアを確保します。

```
READY
MSET &H3000 ↓      .....機械語エリア確保(&H2000 ～ &H3000)
READY
```

ポイント

- マシン語のプログラミングをするには、HD 6303Xユーザズマニュアル((株) 日立製作所)を別途入手する必要があります。

2. プログラムの作成

ASCIIユニットはミニアセンブラが内蔵されています。ここではミニアセンブラを使用して、プログラムを入力します。

まず、アセンブラを起動します。

```
READY
MON ↓      ..... モニタモードに移る
C200H-ASC02 MONITOR V1.6
*          ..... モニタモードのPrompt
* [Ctrl + A] ..... アセンブラの起動
!          ..... アセンブラのPrompt
```

この後、プログラムを入力していきます。

```
!2000:LDAA #$80
2000- 86 80      LDAA  #$80
! LDAB #$7F
2002- C6 7F      LDAB  #$7F
! STD $4000
2004- FD 40 00   STD   $4000
!X ↓          ..... アセンブラを終了する (Xは大文字)
*
```

Format. ! [アドレス] : [ニモニック] アドレス入力時
 ! [ニモニック] アドレス省略時
 — 空白をいれる

3. マシン語プログラムを吸い上げる

(1) モニタモードで次のように入力します。

*S2000. 2100&H2000～&H2100 までセーブ

Format. S [先頭アドレス]. [終了アドレス]

- (2) テキストファイルの受信を選び、プログラムをセーブするためのファイル名を指定します。
- (3) ASCII ユニット前面の START / STOP ボタンを押すと、プログラムの転送が開始します。
- (4) プログラムの転送が最後まで終わったら、〔転送〕〔テキストのキャプチャ〕〔停止〕を選択し、C t r l - X を入力する。

BASICモードに戻る

モニタモードを抜けてBASICモードに戻るにはC t r l + Bを入力します。

```
*[Ctrl + B]
READY
```

マシン語プログラムをASCIIユニットに転送する。

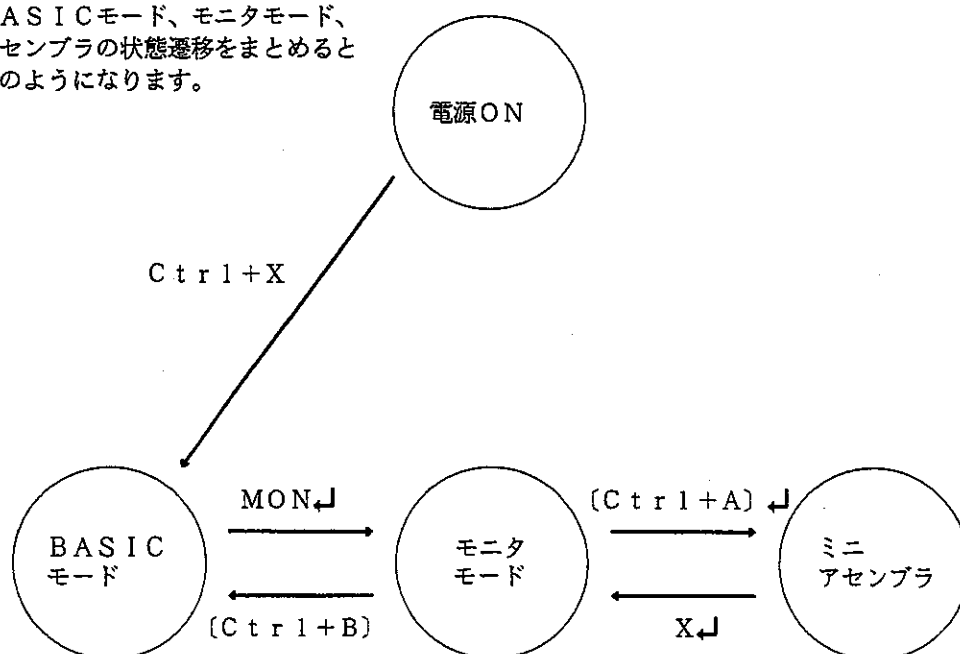
3. で保存したプログラムをASCIIユニットに転送します。

(1) モニタモードで次のように入力します。

*L ↙

- (2) ASCIIユニット前面のSTART/STOPボタンを押します。
- (3) テキストファイルの送信を選択し、保存しておいたプログラムを送信します。

まとめ BASICモード、モニタモード、アセンブラの状態遷移をまとめると次のようになります。



⊗生産中止（予定）機種一覧

アプリケーションの重複する機種や、ニーズに合わなくなり、ここ数年ご注文がない機種につきましては整理・統合を実施させていただいています。今回生産中止対象機種をご紹介します。

■生産中止（予定）機種

商品名称	商品形式
FIT本体	FIT10-CPU01
ターミナルパック	FIT10-MF331
ASCIIユニット	C200H-ASC01

代替機種につきましては別途お問い合わせください。
他社製品につきましては直接各メーカーにお問い合わせください。

保証内容

1.保証期間

納入しました商品の保証期間は、ご指定場所に納入後1年間と致します。

2.保証範囲

上記保証期間中に当社の責により故障を生じた場合は、その商品の故障部分の交換または修理を、その商品のご購入あるいは納入場所において無償で行わせていただきます。ただし、次に該当する場合は、この保証の対象範囲から除外させていただきます。

- (1) 本マニュアルまたは別途取り交わした仕様書等にて確認された以外の不適当な条件・環境・取り扱い並びに使用による場合
- (2) 故障の原因が納入品以外の事由による場合
- (3) 当社以外による改造または修理による場合
- (4) 商品本来の使い方以外の使用による場合
- (5) 当社出荷当時の科学・技術の水準では予見できなかった事由による場合
- (6) その他、天災、災害など当社側の責ではない原因による場合

なお、ここでいう保証は、納入品単体の保証を意味するもので、納入品の故障により誘発される損害はご容赦いただきます。

3.サービスの範囲

納入品の価格には、技術者派遣等のサービス費用は含んでおりません。

ご要望により、別途ご相談させていただきます。

以上の内容は、日本国内での取引および使用を前提としております。日本以外での取引および使用に関しては、別途当社営業担当者までご相談ください。

- 本誌に記載のない条件や環境での使用、および原子力制御・鉄道・航空・車両・燃焼装置・医療機器・娯楽機械・安全機器、その他人命や財産に大きな影響が予測されるなど、特に安全性が要求される用途への使用をご検討の場合は、定格・性能に対し余裕を持った使い方やフェールセーフ等の安全対策へのご配慮をいただくとともに、当社営業担当者までご相談いただき仕様書等による確認をお願いします。
- 本製品の内、外国為替及び外国貿易法に定める輸出許可、承認対象貨物（又は技術）に該当するものを輸出（又は非居住者に提供）する場合は同法に基づく輸出許可、承認（又は役務取引許可）が必要です。

オムロン株式会社 営業統轄事業部

現在販売されていないオプション・アクセサリ・消耗品等が記載されている場合があります。

また記載されている営業拠点の電話番号等は変更されています。

お問い合わせはつぎのフリーコールでお願いいたします。

カスタマサポートセンタ

クイック

オムロン



0120-919-066

■営業時間：8:00～21:00（365日）

携帯電話、PHSなどではご利用になれませんので、その場合は下記におかけください。

電話：055-982-5015（通話料がかかります）

オムロン株式会社 インダストリアルオートメーションビジネスカンパニー