

# パーソナル・コンピュータ (FM-7/8) を用いた画面編集ターミナル

統計数理研究所 仁 木 直 人

(1985年6月 受付)

## 1. はじめに

スクリーンエディタを用いれば、プログラム開発やデータの作成・編集などを行う際、作業の能率が著しく向上する。この事実は、パーソナル・コンピュータやワードプロセッサの普及により、広く認めるところとなっている。

しかるに、大型計算機との会話処理に用いることのできるスクリーンエディタは、特定の(しばしば高価な)ターミナル装置を要す上、機能が豊富な代わりに、操作がスムーズに行えないことが多いようである。しかも、その機能や操作法は固定されており、より都合よく変更しようとしても、自由度はすこぶる狭い。

その解決策のひとつとしては、UNIX システムで使用されているスクリーンエディタ vi のように、ホスト側でターミナルの動作を全て監視・制御し、必要な機能を実現することが考えられる。しかし、この方法はホスト側に大きな負担を掛けることになり、また大型計算機の構造上からも実現は困難である。

そこで、パーソナル・コンピュータである FM-7/8 (Fujitsu Micro 7 または 8) にプログラムを組み込み、ホスト側の変更を全く行わずに、せめてそのパーソナル・コンピュータ並のスクリーンエディット機能を持つ TSS ターミナルを実現しようと考えた。その開発の結果がターミナル・コントローラ “.TERMSxx” である。

ターミナル・コントローラ自体は、FM-7/8 のメイン CPU 部にロードされて単独に動作できる機械語プログラムであり、標準言語である BASIC システムの存在を必ずしも仮定しない。但し、現在のところ、パーソナル・コンピュータの利用法の選択や動作パラメータの変更等、利用者の便利を考えた機能に BASIC プログラムを用いている関係上、BASIC システムと共存する構成を標準としている。

ターミナル・コントローラのソース・プログラムは、K というコンパイラ言語 (COMPAC が開発・販売を行っている。FM シリーズの『活用研究』等を参照) で記述され、約 900 ステップの規模である。コンパイラ言語 K は C と BASIC を混ぜたような言語体系を持ち、機械語を 16 進数で直接プログラム中に挿入できるなど、パーソナル・コンピュータのシステム記述に非常に便利である。

本稿の構成は、次のようになっている。まず、第 2 節で、このターミナル・コントローラのホストとなるコンピュータに想定される動作仕様を明確にする。次に、第 3~5 節では、ターミナル・コントローラの持つ主な機能について述べる。第 6 節は、ターミナル・コントローラを使用する上で便利ないように用意されたふたつの BASIC プログラム “TERM” (パーソナル・コンピュータの使用目的の選択、ターミナル・コントローラの起動等) と “TERMGxx” (動作パ

ラメータの設定・変更)について、その使用法の解説を主な話題としている。また、開発をしていく上で気付いた点などについて述べ、最後の第7節としている。なお、付録Aに画面編集用の特殊キーの使い方の一覧表を掲げ、付録Bに、ターミナル・コントローラの構造に関し、メモリ・マップと動作パラメータ・テーブルの図を中心とした解説がある。また、付録Cで、ターミナル・コントローラの動作の実現方法に関し、簡単な説明を加えた。

ターミナル・コントローラ“TERMSxx”の名前の末尾のxxは、プログラム構造の変化に対応して、いくつかの種類が存在する。ここで紹介する内容は、現在の一般用バージョン2.4a(“TERMS23”)に基づく。

なお、本稿を読む上で、末尾に掲げた文献が参考になると思う。ただし、FM-8に関する文献は、新規購入等が困難かもしれない。FM-8所有者にあたってみるのも一方である。

## 2. ホスト・コンピュータ

このターミナル・コントローラを用いたTSSターミナルのホストとなるコンピュータとしては、一応、日立製作所のHITAC Mシリーズを想定している。しかし、できるだけ、Mシリーズ独特の機能に合わせることを避け、一般性を失わないような考慮を払っているので、他の多くのコンピュータにそのまま、あるいは、簡単な変更で接続できると思われる。

想定されているホスト側の動作は、次の通りである。

a. 無手順で、かつ、リモート・エコーを返さない半二重的運用を行っている。但し、割り込みはターミナルへの送信中も受付ける(回線は全二重)。

b. ターミナルからの受信行の終了を示すCR(復帰)に対して、LF(改行)を返送する。

c. ターミナルへの送信行は、CR, LFまたはLF, CRで終了し、行の途中にこれらの制御文字を含まない。

d. 入力を要求するため、ターミナルに特別な文字列(2種類・各5文字以下)を送出する。Mシリーズでは、システム標準のSI, BEL(ASCII/JISコード体系で0F<sub>16</sub>, 07<sub>16</sub>なる値を持つ。以下同様)と、ユーザが指定したSI, NUL(0F<sub>16</sub>, 00<sub>16</sub>)またはSI, “.”BS(0F<sub>16</sub>, 2E<sub>16</sub>, 08<sub>16</sub>)という制御文字を主とした文字列を用いることが多い。ターミナル・コントローラの移植性という点からは、この項目が最も大きい障害となる可能性があるので注意しておく。

e. 行番号指定で編集のできるエディタが利用できる。パーソナル・コンピュータの標準的なBASICシステムのように、プログラムのリストがそのまま行番号指定の編集コマンドとして使えるようになっていることが望ましい。Mシリーズでは、EDITがそれに当たる。

f. 通信には7ビットASCIIコードを使用し、カナ文字を扱う場合にはSI/SO(0F<sub>16</sub>/0E<sub>16</sub>)による切替え方式を採用している。なお、近々、8ビットASCIIを使用した通信にも対応できるようにターミナル・コントローラを改良する予定である。

さらに、次の様な機能がある場合には、ターミナル・コントローラにその機能を生かすオプションを用意してある。

g. ターミナルからのCRの送信をセッション開始の要求とみなす(自動CR送信オプション)。

h. XON/XOFF制御文字(11<sub>16</sub>/13<sub>16</sub>)による出力制御ができる(PF10キーによるXOFF/XON交互送信機能、割り込み前後でのXOFF/XON送出機能オプション)。

### 3. 画面編集機能

画面編集機能は、FM-7/8 のサブ CPU 部 (画面およびキーボードを制御) が持っている画面編集機能をそっくりそのまま利用している。従って、利用者は、パーソナル・コンピュータをスタンドアローンで用いるときと、全く同じ操作で画面編集が行えることになる。

FM-7/8 の画面編集機能を簡単に述べれば、

1. 画面 (最大 25×80 字) をいくつかの『フィールド』と呼ばれる編集対象単位に区切ることができる。
2. カーソルを移動させて適当な変更位置を決めることができ、文字または制御キーの操作により変更された全フィールドを、RETURN キー (FM-7 では  $\curvearrowright$  キー、以下同様) の入力により順次読み出すことができる。
3. RETURN キーの代わりに、CLEAR キー (FM-7 では CLS キー、以下同様) または CTRL-C (CTRL キーを押しながら C のキーを同時に押す) あるいは CTRL-X を入力することにより、画面編集動作を中止できる。CLEAR キーの場合は同時に画面を消去し、画面全体を未使用フィールドとする。

という 3 点に尽きる。ターミナル・コントローラでは、この機能を大型計算機の画面編集ターミナルとしての動作に適合させるよう、共有 RAM を通じたサブ CPU の動作制御を行っている。例えば、フィールドの分割に関しては、「ホストから送られてきた CR, LF または入力要求文字列はフィールドを分割する境界とし、また、キーボードから画面の未使用部分に入力を行った場合は、画面編集の終了時、入力のあった行とその次の行の間にも境界を置く」などの処理がある。

ターミナル・コントローラにより制御されたパーソナル・コンピュータは、ホスト側から見れば単純な無手順ダム・ターミナルである。画面上で変更された各フィールドは、画面の上の方から順に (変更の順と無関係に)、そのフィールド全体が手で入力されたものとしてホストに送られる。フィールドの後には CR が付加される。複数の変更フィールドがある場合には、始めのフィールドの送出後、ホスト側から入力要求文字列を受けとるごとに、1 フィールド分ずつ順番に送出を行う。

従って、ホスト側の状態 (コマンド待ち、EDIT サブコマンド待ち、ユーザ・プログラムからのデータ入力待ち、など) によらず、また、画面上の任意の表示部分をコマンドあるいはデータとして利用できる。例えば、特定のファイルの一覧表を表示し、その表中のファイル名 (データセット名) をパラメータの一部として使うコマンドを生成するようなことも簡単である。

#### 3.1. 文字セット

画面編集対象として用いることのできる文字セットは、パーソナル・コンピュータの内部コードで

20<sub>16</sub>~7E<sub>16</sub>: JIS (ASCII) 95 文字セット

80<sub>16</sub>~9F<sub>16</sub>: グラフィック文字-1

A1<sub>16</sub>~DF<sub>16</sub>: カナ文字

E0<sub>16</sub>~FE<sub>16</sub>: グラフィック文字-2

に対応する文字で構成される。この内、80<sub>16</sub>~9F<sub>16</sub> (グラフィック文字-1) に対応する文字は、JIS コード 00<sub>16</sub>~1F<sub>16</sub> に対応する制御文字をホストに送るために使用する。例えば、グラフィック文字 “ $\text{J}$ ” (9B<sub>16</sub>) は制御文字 ESC (拡張, 1B<sub>16</sub>) に変換されてホストに送信される。

カナ文字 (A1<sub>16</sub>~DF<sub>16</sub>) は SO (Shift Out) 側文字として、それらが連なる部分の前に SO (0E<sub>16</sub>) を付加し、SI 側の文字が現れる直前に SI (0F<sub>16</sub>) を挿入する。

残りのグラフィック文字 (E0<sub>16</sub>~FE<sub>16</sub>) については、現在のバージョンではカナ文字と同様の取扱いとしているが、次のバージョンからはホストへの送信時に指定文字列に変換される文字 (およびそのための制御文字) としてその一部を使う予定である。文字列指定も画面上でできるようにするので、長い文字列になりがちなファイル指定などに便利になると思われる。

これまで挙げなかった内部コードを持つ文字は、その多くが画面編集のための特殊キーに対応している。画面編集に用いる特殊キーについては、付録の『操作法』で簡単に説明してあるが、詳しいことは、FM-7/8 の『システム仕様』、『システム解説』等のマニュアルを参照して欲しい。

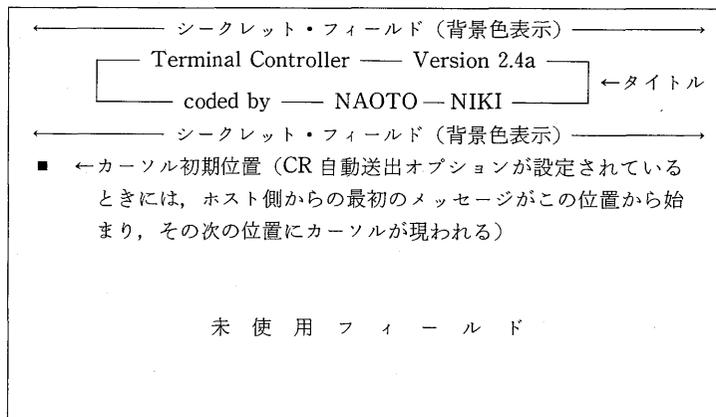
### 3.2. 初期画面とフィールド

ターミナル・コントローラ起動時の画面は、図 3.1 のようになっている。画面の最上行と枠のすぐ下 (すなわちカーソルの初期位置のすぐ上) の行は、シークレット・フィールドとして、背景色と同じ色で文字の表示を行うように設定してある。この 2 行にカーソルを移動すると (最上行への移動は HOME キーのみでよい)、カーソルが消えたように見えるはずである。文字も表示が見えないだけで、画面編集の対象となる一般のフィールドと何の変わりもない。パスワード等の入力に便利であろう。

枠に囲まれたタイトルから 1 行空いた、その次の行の先頭から画面の最後までは未使用フィールドとなっている。むしろ、未使用フィールドは画面の外にまで続いていると考えた方が良くかも知れない。

画面編集モードに入る際、カーソルは常に未使用フィールドの先頭に置かれる。また、未使用フィールドは、キーボードからの入力あるいはホストから受取った文字列により、必要に応じて区切られ、その区切られた部分はひとつの『既存フィールド』を形成する。『既存フィールド』は以後、移動したり (画面全体のスクロールを除く)、拡大・縮小されることがなく、画面編集の基本単位となる。

図 3.1. 初期画面



## 4. ホストとの通信機能

ホスト・コンピュータとの通信には 2 種類あり、ひとつはホストと入出力の同期をとりなが

ら行う通常のデータの交換(半二重的通信)であり、もうひとつは端末割り込み(ブレイク)発生および XON/XOFF による出力制御のために必要な非同期の(全二重)通信である。

本節では、通信に関して、一般ユーザが必要とする知識のみを扱っている。通信に関するプログラム技法等については付録 B に説明がある。

#### 4.1. データ交換

ホストとの通常のデータ交換(コマンド、データ、リストなどの送受)は、画面編集を実現し、また、低速のプリンタの制御を行わねばならないことなどから、ユーザのキー操作や画面上の表示とは、必ずしも同期していない。

ホストから受信したデータは、一旦、受信用のバッファに入り、そこから順番に取り出されて、画面への表示やプリンタへの出力等に使われる。低速のプリンタへの出力を考慮して、大きめ(11KB)のバッファを用意してある。バッファに入りきれなくなると、ベルが鳴り、バッファの内容は捨てられる(XON/XOFFの項を参照)。

画面編集で画面より取り出された各フィールド(文字列)も、区切りとして後に CR が付加され、ホストへの出力用バッファ(2KB)に一時蓄えられる。先に画面に表示しておくべきデータが受信用バッファに溜っていなければ、最初の1フィールド分の文字列がホストに向けて送信される。2番目以降のフィールドは(もしあれば)、ホストから入力要求文字列が送られてきており、かつ、前の送出文字列に対するホストの応答を全て表示(またはプリンタ出力)し終わった、という2条件が満たされた時、再び1フィールド分の文字列が送られる。

ホストからの入力要求に対して、送出すべき文字列が出力用バッファにないときには、画面編集のモードとなり、カーソルが表示され、キーボード入力が有効となる。

#### 4.2. ブレーク

端末割り込み(ブレイク、アテンション)は、ブレイク信号(200 ms 以上続く論理 0 信号)をホストに送出することにより、ホストの動作を中断したり、ホストに要求を与える“きっかけ”を得るためなどに用いられる。Mシリーズでは、入力行の途中取消しや連続入力の終了を知らせる目的などにも使われている。

ブレイク信号は、STOP キーが押され、それが元に戻った時点から 300~500 ms の間、ホストに向けて送出される。なお、後述する XON/XOFF 制御オプションが選択されているときには、ブレイク信号送出後に XON 制御文字をホストに送る。この機能により、XOFF でホストからの出力を停止していた場合、ブレイクを掛けると同時に出力停止状態も解除できる。

ホストからブレイク信号を送るような運用は通常しないので、ブレイク検出は行っていない(ブレイクが送られてきた場合、無意味な文字として表示されるかもしれない)。

#### 4.3. XON/XOFF

全二重通信では、XON/XOFF 制御文字により、相手からの出力の送出/停止を制御する XON/XOFF 手順(手順というほどのこともないが)が良く使われ、低速の入出力機器の動作待ちや表示の一時停止などに用いられている。大型計算機とその TSS 端末との間では、リモートエコーを返す典型的な全二重通信を行うことは稀である。しかし、無手順端末を接続する場合には、非同期のブレイク信号を受付る関係上、回線は全二重、運用は半二重的という使い方が多い。そこでリモートエコーは返さない全二重通信(3/4 二重とでも呼べば適当か)を実現することは、ホスト側の構成を小変更すれば済み、そう難しいことではないと思われる。現に HITAC Mシリーズでは、通信制御装置に全二重通信用バッファを増設することによりこの

ような『3/4二重』通信が可能であり、XON/XOFF制御を組み込むことができる。

この制御機能を簡単に使えるように、ターミナル・コントローラにはXON/XOFF制御オプションが選択できるようになっている。このオプションが選択されていると、ブレイク後にXONが送信される(前述)とともに、PF 10キーを押すと直ちにXOFFまたはXONを交互にホストに送信する機能が付加される。また、入力データ・バッファが満杯に近くなったときにもXOFFを送り、しばらくの間、ホストからのデータ送出を停止させる。

#### 4.4. 起動時のCR(復帰: 0D<sub>16</sub>)自動送信機能

ターミナル・コントローラの起動時、初期画面の表示後、制御文字CRを2秒程度ごとに間欠的に自動送信するオプションを選択できる。自動送信はホスト側からCRまたは入力要求文字列が返送されてきたとき終了し、画面編集処理に入る。また、このオプションが選択されていないときは、直ちに画面編集処理に入る。

### 5. ファンクション・キー(PF 1~10)による補助機能

ターミナル・コントローラには、その使い勝手を良くするため、種々の補助機能が用意されている。それらの機能はファンクション・キーの押し下げによって操作される。

#### 5.1. 常用文字列への置き換え(PF 1~5)

上下2段になっているファンクション・キー群のうち、下段のPF 1~5は画面編集に押されると15文字以下の文字列に、直ちに、置き換えられる。文字列の中には画面編集のために使われる制御文字が入っていても良く、その通りキーボードから入力されたのと同様な取扱いを受ける。

良く使うTSSコマンドなどを定義しておけば、キー操作の回数を減らすことができ、また、操作ミスも少なくなる効果が期待できる。例えば、

<u>R</u>	<u>U</u>	<u>N</u>	<u>CR</u>		(4文字, CR=0D <sub>16</sub> )			
<u>L</u>	<u>SP</u>	<u>*</u>	<u>SP</u>	<u>2</u>	<u>0</u>	<u>CR</u>		(7文字, SP=20 <sub>16</sub> )

などの文字列を組み込むことができる。PFキーに対応させる文字列の定義は、事前に『動作パラメータ設定プログラム』で行う。なお、1キーあたり最大15文字では短すぎると考えるかも知れないが、TSSには必ずマクロ・コマンドまたはコマンド・プロシジャなどと呼ばれる高度なコマンド処理機能があるので、その有効利用を図れば、特殊な用途を除き、まず充分の文字数であろう。

なお、関連する機能として、特定のグラフィック文字を『送信時に』指定文字列に置き換える機能、対応文字列定義の動的変更機能があり、バージョン2.5からサポートする予定である。

#### 5.2. プリンタ出力制御(PF 6)

処理結果、あるいは、プログラムやデータのリストを保存する場合等、ホストとの会話記録をプリンタに残したいときには、PF 6を押す。押した時点から、次に再びPF 6を押すまで、ホストからの受信文字列を画面に表示するとき、および、ホストに文字列を送信するとき、それらの文字列をプリンタにエコーバックする。

プリンタ上の表示は、それゆえ、画面上の表示とは必ずしも一致しない。画面編集で多数のフィールドを一度に修正した場合などには、実際に行われたホストとの会話処理をプリンタ上

で確認できるであろう。

プリンタへのエコーバックを行っている間は、画面の左上に黄色のドットを表示している。行っていないときは青色のドットである。ただし、画面編集集中に PF 6 を押した場合には、画面編集が終了するまでドット色は変化しないから、注意が必要である。

### 5.3. 送信文字列の表示 (PF 7)

ホストに送信する文字列を確認する必要があるとき、または、再度似たようなコマンドを送るため、なるべく次のキー入力位置近くに送信文字列を表示しておきたいときなどには、PF 7 を押す。次に PF 7 を押すまでの間、ホストに実際に送信を行う時点で、送信文字列を画面に表示する。この機能を用いて表示された文字列は、ひとつのフィールドを作り、再び画面編集の対象となる。

### 5.4. 強制的画面編集開始 (PF 9)

ホスト側が明らかに入力待ち状態であるにも拘らず、入力を要求する特別な文字列が、何らかの原因により、受信されなかったとき、PF 9 を押すことにより、直ちに画面編集処理に入る。

ターミナル・コントローラ起動時の CR 自動送信中に押すと、自動送信を終了させ、直ちに画面編集を開始する。

### 5.5. XON/XOFF 送出 (PF10)

この補助機能については、すでに『ホストとの通信機能』のところで述べた。XON/XOFF オプションを選択していない場合、PF 10 はブレーク・キーとして使われる。ただし、STOP キーと異なり、画面編集集中の場合、その終了まではブレーク信号をホストに送らない。

### 5.6. 特殊機能 (PF 8)

その他の特殊な機能を、PF 8 を押すと表示されるメニューから選んで、働かせることができる。画面編集集中に押すと、終了まで待ち、メニューの表示を行う。

メニューの現在の内容は

```

      ♡ Special Functions ♡
1 — Set Tab Stops
2 — Emulate Tektronix 4010 Graphics
3 -X- Send a File to the Host
4 -X- Recieve a File from the Host
5 — Put a String for Printer Control
6 — Restart (Warm Start)
7 — Exit
      * Enter 1-7
  
```

となっている。ここで、1~7 を選択すれば、それぞれの番号に対応した機能が起動される。

[1] 水平タブ位置の変更を行う。現在のタブ位置を“◆”で示す行と修正位置を示す“^”が表示されるので、カーソル移動キー“→”と“←”を用いて位置を指定し、タブ位置設定 (“↓”キー)または解除 (“↑”キー)を行う。設定または解除後、修正位置は右に1だけ動くので、“↑”キーを押し続けることにより、全てのタブ位置を解除することも簡単にできる。変更の終了は“RETURN”キーで知らせる。

[2] Tektronix 4010 シリーズのグラフィック・ディスプレイ用の出力形式に則ったデータを、画像情報として表示するモードとする。このモードは PF 8 を再び押すと解除される。もともと 4010 シリーズはモノクロのディスプレイであるが、カラー表示を可能にするため、ESC 1 (1B<sub>16</sub>, 31<sub>16</sub>)~ESC 8 (1B<sub>16</sub>, 38<sub>16</sub>) により、8色のグラフィック・ペンを使えるよう拡張を行っている。なお、このエミュレーション機能（他の機器のための命令を代替実行する機能）の使用に先立って、画面の論理的ポイント数を 4010 シリーズと同じ 1024×780 とするか(512×195 にスケーリングする)または 640×200 とするか（指定された通り FM-7/8 の画面に対応させる）を、ホスト側の動作に合わせて正しい動作パラメータを選択しておく必要がある。

[3] [4] ホストのファイルとパーソナル・コンピュータのディスクとの間のデータ転送を行う機能であるが、バージョン 3.0 よりサポートする予定である。

[5] パーソナル・コンピュータに接続されたプリンタに特殊制御文字列などを送る。たとえば、改ページのために FF (0C<sub>16</sub>) を送ったり、文字の種類、改行ピッチの変更などの制御を行うために使用できる。プリンタに送出する文字列の定義中は、画面編集に用いるキーを含め、全ての制御文字が普通の文字と同様に取扱われる。定義終了の印として、GRAPH-P (“GRAPH” キーを押しながら “P”) を押すとそこまでに入れた文字列がプリンタに送られる。代わりに GRAPH-C とすると、プリンタに送らずに処理を終える。

[6] ターミナル・コントローラを再スタートする。オプションの指定があれば、CR の自動送出手も行われるので、一度回線を切断した会話処理を再開する場合などに用いると便利である。また、動作開始時の画面にある非表示領域を利用して、他人に見られたくない情報の入力を行いたい場合にも使用できる。

[7] ターミナル・コントローラの動作を終了する。ターミナル・コントローラはサブルーチンとして動作できるような構造を持っている。この番号の指定により、退避しておいたレジスタ類や割り込み処理環境などを回復し、呼出しを行ったプログラムに復帰する。通常は “TERM” という BASIC プログラムに返り、機能選択表示が現れる。

## 6. ユーティリティ・プログラム

### 6.1. 使用目的選択ルーチン (TERM)

ターミナル・コントローラを外部記憶装置（通常フロッピー・ディスク）から内部メモリにロードした後、FM-7/8 の使用目的をユーザに問い合わせる。その応答に従ってそれぞれの目的に合った環境（メモリ使用上限など）を設定し、必要なプログラムを呼出す（または、BASIC システムに制御を返す）BASIC プログラムである。

ターミナル・コントローラが含まれているフロッピー・ディスクからシステム起動を行うと、自動的にこの TERM が RUN するような構成としてある。また、ターミナル・コントローラの終了時、および、次に述べる『動作パラメータ設定ルーチン』の正常終了時にも、TERM に入る。ただし、ターミナル・コントローラの終了時には、新たに外部からターミナル・コントローラ・プログラムをロードすることはしない。

使用目的の選択には、次のような表示に対して、対応する番号のみを入力すればよい。

* BASIC system .....	1
* Terminal Configuration .....	2
* Terminal Controller .....	0
Enter 0, 1 or 2 .....	■

ここで0を指定すればターミナル・コントローラが開始される。1の指定ではメモリの使用上限を6FFF<sub>16</sub>番地、文字列バッファ長を300バイトと設定した後、

\*\* Disk Drive=2, Disk File=1 \*\*

とディスク・ファイルの使用環境を表示し、さらにプログラム自身を消去して BASIC システムに制御を委ねる。指定を2とすれば、次のユーティリティが起動される。

### 6.2. 動作パラメータ設定ルーチン (TERMG23)

このユーティリティ・プログラムは、付録Bで詳しく述べる『動作パラメータ』を簡単に設定するためにある。まず、各パラメータの現在の設定値が図6.1のように表示される。始めの3行はタイトル部分で、これからパラメータ設定を行うターミナル・コントローラの名称、バージョン、作成日付を示している。

図6.1. 動作パラメータ設定画面

```

Configuration starts for
— Screen Edit Terminal Controller in 'TERMS23'
  Version=2.4a, Create Date=23/ 4/85

Mode of Communication Interface=F7E1
  Keyin Prompt String-1=$0F$07
  Keyin Prompt String-2=$0F$00
  Sending C/R's until Reply=ON
  Color for /BG/Rx/Tx/Keyin/= / 0/ 7/ 5/ 4/
  Ringing Bell=OFF
  String set in PF1=PF1$0D
  String set in PF2=PF2$0D
  String set in PF3=RUN$0D
  String set in PF4=
  String set in PF5=

Screen Size of 4010 Emulation=1 (639,199)
  Pen Color for 1, 2, ..., 8=/ 7/ 6/ 5/ 4/ 3/ 2/ 1/ 0/
  Rule for Mixing Colors=PSET
  XON/XOFF (PF10)=ON
  Tab Stops=/ 1/ 7/13/19/25/31/37/43/49/55/61/67/
             /73/80/80/80/80/80/80/80/80/80/80/
    
```

それ以下の部分が個々の動作パラメータと対応している行である。各行とも、“=”の左側がパラメータの種別、右側がパラメータの設定状態を表している。

反転表示によるカーソルが始めの行の“=”の右側の最初の文字に置かれているはずである。このカーソルを『カーソル移動キー(←, →, ↑, ↓)』を使って必要な位置に動かし、画面上でパラメータ設定を変更する。特殊文字(空白=20<sub>16</sub>を含む)は、“\$”に続けて、その ASCII

(JIS)コードを16進数2桁で指定することで表現できる("\$"を表すときには"\$\$"とする)。なお、『EL』キーにより、その行のカーソル位置以降を消去できる。

設定が終了したなら、RETURN キーを押す(一行ごとには不要)。再度、動作パラメータの設定値が確認のため表示され、

\*\* OK? (Y or N)

と聞いてくるので、正しく設定されていれば“Y”または“y”を応答する。この応答により、変更されたターミナル・コントローラはディスク上にしまわれ、続いて『使用目的選択ルーチン』が再起動される。もし“N”または“n”と応答すれば、パラメータの変更を続けることができる。

もし、それまで行ってきたパラメータ変更を無効にしたい場合には、STOP(またはBREAK)キーを押してプログラムの実行を止め、

RUN “TERM”

として、『使用目的選択ルーチン』を起動すればよい。

以下、順に設定方法を説明する。

a. 回線インフェースの動作モード。4文字からなる指定で、始めから、

$\left\{ \begin{array}{l} \text{F} \\ \text{S} \end{array} \right\}$ :  $\times 16$  のクロックを使用  
 $\left\{ \begin{array}{l} \text{E} \\ \text{O} \end{array} \right\}$ : 偶数パリティ (他の指定は現在のバージョンでは無視)  
                   奇数パリティ  
 $\left\{ \begin{array}{l} 1 \\ 2 \end{array} \right\}$ : ストップ・ビットは  $\begin{array}{l} 1 \text{ ビット} \\ 2 \text{ ビット} \end{array}$

を表す。HITAC M シリーズ用は図 6.1 に示すように『F(またはS) 7E1』である。

b. 入力要求文字列-1, 2. それぞれ5文字以下の文字列を指定する。図の例は、SI, BEL および SI, NUL を指定したものである。

c. 起動時の CR (改行) コードの自動送出。この機能を利用する場合は ON, 利用しない場合は OFF を指定する。

d. 画面表示色。背景(BG), 受信文字列(Rx), 送信文字列(Tx), キー入力文字列(Keyin)の別に、対応する位置のカラー・コード(表 6.1)が表示色になる。

表 6.1. カラーコード

色コード	0	1	2	3	4	5	6	7
2進コード	000	001	010	011	100	101	110	111
表示色	黒	青	赤	紫	緑	水色	黄	白

e. ベル音。ホストからの BEL 制御文字に対して、実際にベル音を出すとき ON, 出さないとき OFF を指定する。

f. PF 1~5 に対応する文字列。各キーに対して、それぞれ15文字以下の文字列を設定できる。指定の先頭位置で『EL』キーを押すと、そのキーを無効にする。図の例では、PF 1~3 に設定があり、RETURN キーを押さなくともいいように、それぞれ改行コード(CR=0D<sub>16</sub>)が付加されている。

g. グラフィック・ディスプレイ 4010 のエミュレーション。この部分の始めの行では、画面

座標の上限値を 1: (639, 199), 2: (1023, 779) から番号で選ぶ。次の行はペン番号に対応するカラー・コードを順に指定する。このとき、同色があっても構わない。3 行目では、同一点に複数の色を着ける場合の混合規則を表 6.2 から選択する。

表 6.2. 色の混合規則コード

コード	論理	説明
0	PSET	最後に指定した色をそのままその点の色とする。
2	OR	カラーコードの論理和をその点の色とする (加色混合)。
3	AND	カラーコードの論理積をその点の色とする (減色混合)。
4	XOR	カラーコードの排他的論理和をその点の色とする。

h. XON/XOFF オプションの指定. PF10 を XON/XOFF 制御のために使用する場合は ON, ブレークとして使用する場合は OFF を指定する。

i. 水平タブ位置. 『TAB』 『DUP』 キーの動作を規定する水平タブ位置を設定する。このユーティリティを使用する場合は最大 24 個所までである (ターミナル・コントローラの動作時に変更する場合は、全ての欄をストップ位置にできる)。なお、第 80 欄は必ずストップ位置になる。

## 7. おわりに

このターミナル・コントローラは、もともと半二重通信 (割込みに関してのみ全二重) を前提として創られた。その場合には特別な入力要求がホストから送られてくることが多く、画面編集モードに入るきっかけをうまく握ってきた。しかし、UNIX 系のマシンの普及とともに、XON/XOFF による出力制御および入力要求を待たない先行入力の要求が高まり、大型汎用計算機の TSS も全二重通信をサポートしつつある。

現に、統計数理研究所の HITAC M-280 システムでもすでに全二重通信を実現しており、ターミナル・コントローラの方でもバージョン 2.2 から、XON/XOFF をサポートしてきた。ただ、先行入力に関しては、現在のターミナル・コントローラの基本構成では対応が難しい (もっとも、複数行の修正が可能な画面編集というのも一種の先行入力ではあるが)。欧米人に比べタイプ入力がかかり遅い日本人が、比較的良好な TSS 環境で使っている分には、何の支障もなく済んでしまうかも知れないが、それでも、目の健康を考えてなるべく画面を見ないようにしているユーザの存在を無視できない (小さな音で補助すれば良いか?)。今後この機能の要求が強まれば、抜本的な手直しが必要となろう。

漢字入出力のサポートの要求も強い。しかし、漢字データの取扱いについて統一的な規格がない現状では、どうしてもホストの規格に依存してしまうことを免れない。漢字機能を組込むことは (相当な改造を覚悟すれば) できないことはないが、ホストからのある程度の独立性を確保しようとする設計思想の重要な部分が失われる危険がある。

いろいろ書いてきたが、最近のワークステーションの充実ぶりを見ると、TSS システムの下、大型計算機と直接会話をしながら大勢が利用する、という時代はそろそろ終りに近いように思う。画面編集端末というのものも、ワークステーション時代への布石の一部かも知れない。このターミナル・コントローラもワークステーションの強力なデータ作成・プログラム開発環境には全く太刀打ちできそうもない。本格的なワークステーション時代への『ツナギ』として、多くのユーザに使われれば、本システムの役割は充分果たしたと言わなければならない。

**謝 辞**

査読者の方々および編集委員の方からは有益なコメントをいただいた。ここに記して感謝いたします。

## 付録 A. ターミナル・コントローラ操作法

### A.1. 一般的な起動法

FM-7/8 の裏面にある DIP スイッチが、通常良く使われる ROM/DISK モードまたは DOS モード用のブート・ローダ選択に設定されていることを確認しておく。

ドライブ番号 0 のフロッピー・ディスク・ユニットに、FM-7 用 (New 7 を含む) または FM-8 用に区別された『ターミナル・コントローラ・システム・ディスク』をセットする。電源が入っていない場合には、ディスク・ユニットおよび本体の電源を入れると、使用目的選択画面 (本文 6.1. 参照) が出るので、『0』を選択する。すでに電源が入っている場合には、本体裏面のリセット・スイッチを押せばよい。また、すでに BASIC システムの下にある場合は RUN “TERM” で起動することができる。

ターミナル・コントローラの初期画面 (本文 3.2. 参照) が表示され、(CR 自動送付機能を選択している場合にはホスト側からの応答を待って) 文字反転ブリンクによるカーソルが現われ、ただちに画面編集モードに入る。なお、初期画面の最上行およびカーソル初期位置の直上行にはシークレット・フィールド (背景色で文字を表示するので見えない) を設定してあるので、パスワードの入力等に便利である。

なお、PF 8 で開始する操作によりターミナル・コントローラ使用中での再起動が可能なことを付記しておく。

### A.2. 特殊キーの使用法

#### A.2.1. 画面編集入力を終了するキー

RETURN または 

画面編集により変更された全フィールドをホストに送る。

CTRL + X または CTRL + C

それまでの変更を取消し、画面編集を終了する (通常は、やり直す)。

CLEAR または CLS

画面を消去して、変更を取消し、画面編集を終了する (通常は、やり直す)。

#### A.2.2. カーソル移動に関するキー

, , , 

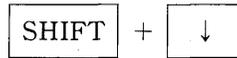
カーソル位置を指定した方向に移動する。移動は同一行または欄で循環する。

SHIFT + , SHIFT + 

カーソル位置を、指定した方向にある、英数字列の先頭に移動する。

SHIFT + 

カーソル位置を行の先頭に移動する。すでに先頭にある場合は上の行に移動する。



カーソル位置を下の行の先頭に移動する。



カーソル位置を画面の先頭（左上）に移動する。

### A.2.3. 編集キー



挿入モードの設定・解除を行う。挿入モード（キー横のLEDで表示）では、入力された文字がカーソル位置にあった文字の前に挿入される。



カーソル位置にある文字を抹消する。



カーソル位置の直前（左側）にある文字を抹消する。



カーソル位置以降の文字列を消去する。

### A.2.4. 文字列に置き換えられるキー



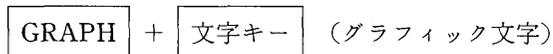
次のタブ・ストップ位置まで、スペース（空白）を埋める。タブセットについてはPF8についての説明および本文5.6および6.2.参照。



次のタブ・ストップ位置まで、1段上の行を複写する。



設定されている文字列に置き換えられる。設定法は本文6.2.参照。



内部文字コード  $80_{16} \sim 9F_{16}$  であるグラフィック文字は、ホストへの送信時に、最上位ビットを0としたとき対応するASCII (JIS) 制御文字（内部コード  $00_{16} \sim 1F_{16}$ ）に変換される。

A.2.5. 制御に使われるキー

STOP
         
 または
         
 BREAK

ホストに端末割り込み（ブレイク）信号を送る。

PF 6

会話記録のプリンタ出力の実行・非実行の切替え。

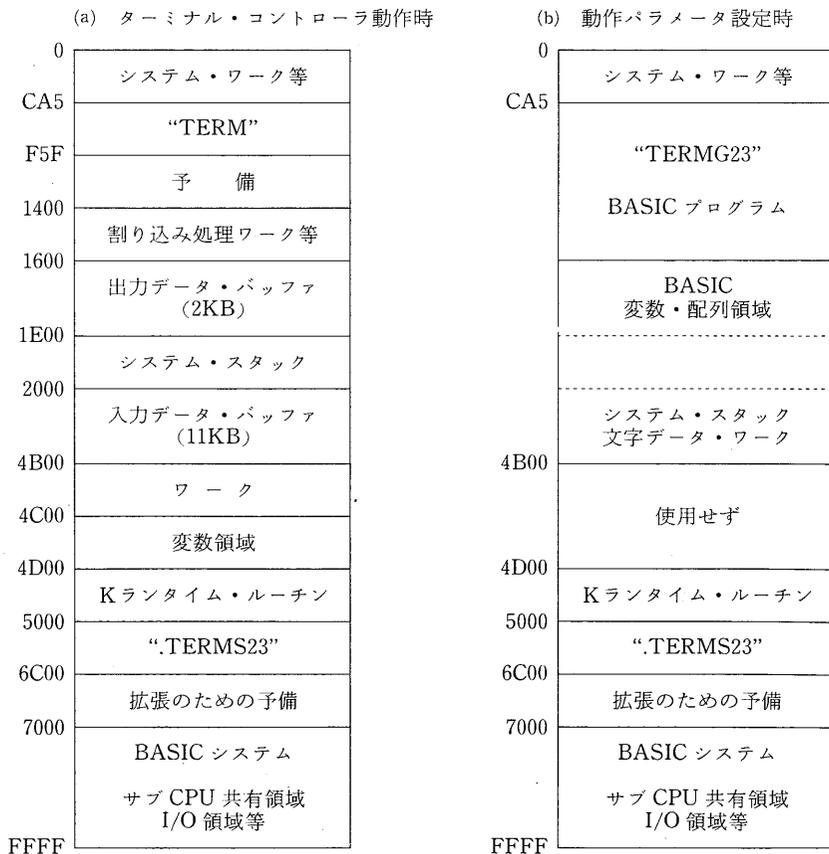
PF 7

ホスト側に送信した文字列の表示・非表示の切替え。

PF 8

メニュー選択による特殊機能の開始、タブセット、4010 エミュレーションなどを行える。本文 5.6. を参照。なお、画面編集中はメニューが出てこない。その際は、A.2.1. のいずれかのキーで編集モードを終了させる必要がある。

図 B.1. メモリ・マップ



PF 9

ホストからの入力要求を待たずに画面編集モードに入る。

PF 10

XON/XOFF 制御文字を交互にホストに送る。XON/XOFF 制御オプションを選択していない場合には、ブレイク信号を送る。

## 付録B. ターミナル・コントローラの構造と動作パラメータ

### B.1. メモリ・マップ

ターミナル・コントローラとその動作に必要なワーク・エリア、および、併用して用いる BASIC で書かれたユーティリティ・プログラムの主メモリ上の配置を、図 B.1 に掲げる。

図 B.1. (a) は、ターミナル・コントローラとして動作を行っているときのメモリ・マップで、コントローラを呼出す“TERM”という簡単な BASIC プログラムと共存する態勢となっている。

ターミナル・コントローラのソース・プログラムは K 言語で書かれているので、K のランタイム・ルーチン群、K の単純変数領域、および、ユーザ・スタックのための領域が、 $4B00_{16}$  ~  $4FFF_{16}$  番地に確保されている。コントローラのオブジェクト・プログラムは、K のオブジェクトの標準的な位置である  $5000_{16}$  番地より置かれ、約 7KB のサイズを持つ。この位置に今後も置く場合には、機能拡張のためのスペースがあと 1KB 程度しかないが、K のオブジェクトはリロケータブル（再配置可能）であるから、別の所に移すことが容易で、拡張に伴う問題は少ない。また、BASIC システムとの共存を止めれば、直ちに 31KB の領域が解放されるので、資源には当分事欠かない。

コントローラのワーク・エリアとして、入出力データ・バッファがそれぞれ 11KB および 2KB (1 画面分 +  $\alpha$ ) 用意してある他、システム・スタック領域、割り込み処理のためのワーク・エリアを十分な余裕をみて確保してある。割り込み処理用ワークは、IRQ 処理用と FIRQ 処理用を分けて、主に各処理時のユーザ・スタック領域として使用している。

メモリの先頭番地付近にある「システム・ワーク」部分は、ほとんど BASIC システムでの使用形式のままになっている。ターミナル・コントローラで使用しているのは一個所のみで、割り込み処理ルーチンへのジャンプ命令を保持する  $1DA_{16}$  ~  $1E2_{16}$  番地を書き換えて、“TERMS23”の中の各処理ルーチンに必要な制御が移るようにしてある。なお、動作終了時(5.6 [7] 参照)には、元の BASIC システム用に戻している。

図 B.1. (b) は、BASIC プログラム“TERMG23”を用いて、ターミナル・コントローラの動作パラメータを設定・変更している場合のメモリ・マップを示す。この場合、“TERMS23”はプログラムというよりメモリ上に置かれたデータと考えた方がよい。

パラメータ設定プログラム“TERMG 23”は、BASIC システムで使用可能なメモリ上限を  $4B00_{16}$  番地までとし、また、改めて“TERMS 23”を外部記憶からメモリにロードすることをしない。従って、それまで使用していたターミナル・コントローラおよび K ランタイム・ルーチンは壊れることなく、パラメータ設定の対象になる。設定の方法については、第 6 節に説明がある。

図 B.2. 動作パラメータ・テーブル

5000	入口：レジスタの保存と設定, パラメータ・テーブルをスキップ
5010	名称：“.TERMS23”
5018	バージョン, 作成日付: 2.4a, 23/04/85
5020	ワークおよびフラグ 20: SI 側/SO 側文字インディケータ 21: 最後に回線入出力を行った文字 22: ホストからの入力要求受取りフラグ 23: 回線出力状態 (可/不可) インディケータ 24~2A: FIRQ フラグ 24: STOP, 25: TIMER, 26~2A: PF6~10
502B	ホスト通信関係パラメータ 2B~2E: 文字表示色 2B: 背景, 2C: 受信文字, 2D: 送信文字, 2E: キー入力文字 2F: 8251A (同期/非同期通信用 IC) に設定する動作モード 30~36: ホストからの入力要求文字列定義-1 (5文字以下) 37~3D: ホストからの入力要求文字列定義-2 (5文字以下) 3E: 動作開始時に CR (0D <sub>16</sub> ) を応答が来るまで送出する/しない 3F: BEL (07 <sub>16</sub> ) 制御文字に対してブザーを鳴らす/鳴らさない 40~4B: TEKTRONIX 4010 シリーズ・エミュレーション 40: 予備 41: X 方向スケールリング・ファクタ 42: Y 方向スケールリング・ファクタ 43: 表示色決定論理 (PSET, OR, AND, XOR) 44~4B: 1~8 番ペンの色 4C: XON/XOFF コントロールを行う/行わない 4D~4F: 予備
5050	ファンクション・キー (PF1~5) に対応する文字列指定
50A0	水平タブ位置
50AA	パラメータ・テーブル予備
50B0	ターミナル・コントローラ本体
6C00	機能拡張のための予備
7000	

## B.2. 動作パラメータ

適正な『パラメータ』を設定することにより、ターミナル・コントローラの動作をホスト側の仕様に合わせ、また、ユーザが希望する使用法に適合させることができる。

動作パラメータ（一部にワークを含む）は、ターミナル・コントローラ本体の始めの部分に、表としてまとめられている。その詳細を図 B.2 に示す。この動作パラメータ・テーブルに含まれるパラメータは、BASIC プログラム“TERMG23”を用いて容易に設定できる（第6節）。ここではターミナル・コントローラを使用または拡張・変更する場合に必要な知識を提供する。

『入口』部は、動作パラメータではないが、ここで説明しておく。この部分には、ターミナル・コントローラを呼出した時点の各レジスタをシステム・スタックに保存し、Kの動作環境に合うように必要なレジスタの設定を行ったのち、パラメータの表を飛越えて、ターミナル・コントローラとしての動作を開始するためのプログラムと、システム・スタック・ポイントの値の保存場所が含まれている。この部分を変更することにより、システム・スタック領域やKの変数・ワーク領域の位置を変えることができるが、ここでは説明しきれない知識を要するので、その方法の説明は省略する。

『名称、バージョン、作成日付』部は、初期画面の一部に表示する他、ターミナル・コントローラと動作パラメータ設定プログラムの適合性を調べるためにも使われる。

『ワークおよびフラグ』部は、ホストとのデータの受け渡しやユーザの操作とともに変化し、ターミナル・コントローラの状態を示す部分である。

『ホスト通信関係パラメータ』部には、背景および各フィールドに与える表示色の指定部分、同期/非同期通信用 LSI である 8251A に設定する動作モード、ホストからの入力要求を示す文字列の定義部分、カラー・グラフィック出力機能に関するパラメータ部分、オプション動作設定部分、が含まれている。

表示色をうまく指定することにより、フィールドの境界を明確にでき、また、見やすい画面とすることができる。表示色は、背景、ホストからの受信文字列フィールド、ホストへの送信文字列フィールド（5.3. 参照）、画面編集時のキー入力文字列フィールドの別に指定できる。色指定は 0~7（8~15 も可能だが使用しない方がよい）のカラーコード（表 6.1）で行い、普通は順に 0（黒）、7（白）、5（水色）、4（緑）が用いられている。なお、モノクロ画面では色番号は輝度（番号の大きい程明るい）に対応している。

通信動作を規定するパラメータとしては、8251A に設定する動作モード指定バイトそのものを格納しておく。ただし、現在のバージョンでは、非同期通信で文字データ長が 7 ビットの場合に限る。例えば、ボーレートをクロック周波数の 1/16 とし、偶数パリティ・ビットを付加し、ストップ・ビットを 1 ビットとしたときには、動作モード指定は  $7A_{16}$  である。

ホストからの入力要求文字列としては、5 文字以下の文字列 2 種を指定することができる。通常は、画面に表示されない制御文字を使用することが多く、HITAC-M シリーズでは一方を SI ( $0F_{16}$ ) BEL ( $07_{16}$ ) の 2 個の制御文字からなる文字列としておく必要がある。もう一方には、SI ( $0F_{16}$ ) NUL ( $00_{16}$ ) あるいは SI ( $0F_{16}$ ) “.” ( $2E_{16}$ ) BS ( $08_{16}$ ) などが良く使われている。

動作開始時に CR ( $0D_{16}$ ) を応答があるまで送出する機能を生かすオプションを選択するときは、パラメータの値として 0 を与える。選択しないときは  $80_{16}$  である。

ホストから送られてきたベル音を鳴らす制御文字 BEL ( $07_{16}$ ) に対して、実際にベル音を出す場合にはパラメータの値として  $80_{16}$  とし、出さない場合には 0 とする。

グラフィック・ディスプレイである TEKTRONIX-4010 シリーズのための描画データに基

づいて、画面上にグラフ表示を行う機能を有している。このとき必要となるパラメータは、X およびY方向へのスケーリング・ファクタ、8本ある仮想ペンの色指定および色を重ねる規則である。

スケーリング・ファクタは指定された座標値の何分の一を画面上の座標値とするかを決定するために用いる。通常は、共に1 (640×200の座標系で使用の場合)、または『x座標に対して2, y座標に対して4』(4010シリーズの1024×780の座標系使用の場合)である。

ペン色は0~7のカラーコード(表6.1)が指定可能で、同色のペンが複数あっても構わない。同一の点を複数回ペンが通る場合の色決定論理は、表6.2に従う。

XON/XOFF コントロールを行う場合の指定は0、行わない場合は80<sub>16</sub>である。この指定により、PF 10の使用目的が変化することに注意が必要である。

常用文字列を呼出すファンクション・キーPF 1~5に対応づけるそれぞれ15文字以下の文字列を定義する部分は、PF 1から順に16バイトずつの領域を割当ててあり、各領域の先頭は文字列の長さデータ(1バイト)である。

水平タブ位置は80ビット(10バイト)のデータで指定される。先頭バイトの最上位ビット(MSB)が行の先頭に、また、10バイト目の最下位ビット(LSB)が行末に対応している。なお、この部分は「水平タブ位置の変更(5.6参照)」を行うと、その設定に従って同時に変更される。

## 付録C. ホストとの通信方式と割り込み処理

ホスト・コンピュータとの通信に関しては、すでに第4節で、ユーザ側から見える動作を中心に説明を加えてある。ここでは、それらの動作がどのような仕組みで実現されているかを話題とし、本稿のような通信プログラムの開発・改良に興味を持つ読者の参考としたい。

### C.1. データ交換

通常のデータ交換(コマンド、データ、リストなど)は、パーソナル・コンピュータ内ではIRQ (Input/Output Request: 入出力要求) 割り込み処理を用いて行われている。

ホストから送信されたデータの各バイトごとに、シリアルI/OポートからIRQがメインCPU(6809)に送られる。割り込み処理ルーチンでは、割り込み原因を調べ、シリアル入力による割り込みであれば、I/Oポートから1バイトのデータを読み出して入力データ・バッファに入れ、バッファ書き込み位置ポインタをひとつ進める。ターミナル・コントローラのメイン・ルーチンでは、バッファの書き込み位置ポインタと読み出し位置ポインタの値が異なることで入力データの存在を知ることができるので、両位置ポインタを参照・更新しながらデータを読み出し、画面への表示(または画面編集のための必要な操作など)等を行う。入力データ・バッファは11KB分の大きさがあり(付録B参照)、循環的に使用している。

IRQ割り込み処理ルーチンは、また、ホストから入力要求文字列が送られて来たことを、フラグを立てることにより、メイン・ルーチンに知らせることも行っている。

ホストへの出力にも同じくIRQ割り込みが利用されている。これはFM-8ではシリアルI/OポートからのIRQを入出力別に扱っていないため(FM-7では別にできる)、割り込みを使う積極的な理由があるわけではない。画面編集で画面(実際にはそのバッファ)より取り出された文字列は、フィールドの区切りとして後にCRを付加し、出力データ・バッファ(2KB)に一時蓄えられる。シリアルI/Oポートを『出力可』とすると、もしそのポート出力バッファが空いていれば、ポートからIRQがメインCPUに送られる。IRQ割り込み処理ルーチンでは、出

力要求割り込みであることが判れば、出力データ・バッファから1バイト取り出しシリアル I/O ポートに送る。以後、ポート出力バッファが空くたびに IRQ が掛かり、出力すべきデータが次々に送りだされることになる。

IRQ 割り込み処理ルーチンは、また、フィールドの区切りである CR を送り出すとシリアル I/O ポートを『出力不可』とし、一旦ホストへの送信を停止する。送信の再開はメイン・ルーチンが行う。入力データ・バッファが空で、かつ、ホスト側から入力要求文字列が送られてきている状態になったとき、まだ出力データ・バッファにデータが残っていればシリアル I/O ポートを『出力可』として次のフィールドの送信を開始し、残っていなければ画面編集処理を起動する。

### C.2. ブレーク

端末割り込み（ブレーク）を掛けるには、ホストの動作とは非同期に、ブレーク信号をホストに送出しなければならない。また、ターミナル・コントローラもキーボード入力可能（画面編集集中）の場合と入力不能の両方の状態が考えられる。そこで、STOP キーを押すと FIRQ (Fast IRQ) 割り込みが発生することを利用することにした。

FIRQ 処理ルーチンでは、STOP キーが押されたことを示すフラグを立てるとともに、ターミナル・コントローラの動作状態を強制的にリセット（サブ CPU が動作中であれば中断させ、入出力データ・バッファをクリアするなど）する。メイン・ルーチンはフラグを検出すると、STOP キーが元の押されていない状態に戻るまで待ち（チャタリングも考慮して）、シリアル I/O ポートを『ブレーク送出・入力可・出力不可』状態に 200 ms 以上（実際は 300 ms 程度）おく。必要な動作が終了しフラグがクリアされるまで、FIRQ 処理ルーチンは全ての FIRQ を無効にする。

なお、後述する XON/XOFF 制御オプションが選択されているときには、ブレーク信号送出後に XON 制御文字をホストに送る。この機能により、XOFF でホストからの出力を停止していた場合、ブレークを掛けると同時に出力停止状態をも解除できる。

### C.3. PF 6~PF 10 による特殊機能

PF 6~PF 10 の操作により特別な動作を指示するときも、ホストとの通信状態や画面編集状態等と無関係に行えるよう、FIRQ 割り込みが使われている。ターミナル・コントローラは動作の開始時にサブ CPU 側の動作指定を行い、PF 6~10 キーの押し下げによりメイン CPU に FIRQ 割り込みが掛かるようにしてある。

メイン CPU 側の FIRQ 割り込みルーチンでは、ホストとの非同期通信を正しく行うため、時間を要す処理を行えない。そこで、PF 6~10 キーに対応するフラグを変化させ、メイン・ルーチンに知らせることのみを行っている。メイン・ルーチンは適当な条件が成り立ったときフラグを見に行き、必要な機能（4.3, 第5節）を実行することになる。

## 参 考 文 献

- ・ K コンパイラ, FM-8/7 のシステム構成, その他  
「FM-8 活用研究」(1982), 工学社.  
「FM-7/8 活用研究」(1983), 工学社.  
「FM-7/11 活用研究」(1984), 工学社.
- ・ FM-8/7 仕様, コミュニケーション・インタフェース  
「FUJITSU MICRO 8 システム解説」(1981), 富士通.

- 「FUJITSU MICRO 7 システム仕様」(1982), 富士通.
- 「FUJITSU MICRO New 7 システム仕様」(1984), 富士通.
- HITAC M シリーズ
  - 「VOS 3 概説」(第 10 版, 8090-3-001) (1982), 日立.
  - 「TSS 解説」(第 4 版, 8090-3-136) (1982), 日立.
- グラフィックス
  - 「4010 型グラフィック・コンピュータ・ターミナル/インストラクション・マニュアル」(1977), *Sony Tektronix*.
  - 「汎用図形出力ルーチン集 GPSL—第 1~第 4 分冊—」(8080-7-096~099) (1981), 日立.

Screen-Edit Terminal Controller  
Using  
Personal Computer (FM-7/8)

Naoto Niki

(The Institute of Statistical Mathematics)

The Screen-edit Terminal Controller is a software tool which lets any Fujitsu Micro 7/8 series personal computer emulate a screen-edit terminal with TEKTRONIX 4010\* graphics.

The Controller promotes a consistent method of operation, because the features of screen-editing are the same to the users of the personal computers. The Controller also provides hardcopy output via printers and accepts 4010 interactive commands with the enhanced 8 color graphics. Communication with a host has two options, without protocol or in XON/XOFF protocol. User definition of function keys and tabstops and some other auxiliary functions are also available.

---

\* A trademark of Tektronix, Inc.