

ターミナル・エミュレータ“TSS45”と分散処理データ通信*

古宇田 亮 一**

KOUDA, Ryoichi (1983) The terminal emulator “TSS45” and the distributed processing data communication. *Bull. Geol. Surv. Japan*, vol. 34 (12), p. 593-626.

Abstract: The Research information processing system of Tsukuba Research Center of Ministry of International Trades and Industries employs a large computer system and two types of data communication networks for common users and remote satellite stations. The network system of data communication operates the on-line TSS (Time Sharing System) and RJE (Remote Job Entry) format. A mini- or micro-computer user must develop a software routine of terminal emulation with a hardware of the data communication, when he needs to connect his computer as an intelligent terminal with a data communication network to a main large computer. The author has established the terminal emulator, “TSS45”, for a desk-top-size micro-computer system, YHP9845T, to link with the asynchronous data communication channel for the large computer FACOM-M200 through the RS-232C serial interface.

The TSS45 emulator employs the interactive TSS and RJE format of FACOM-M200 through the data communication network for common users. The user of YHP9845T can also interact every data with FACOM-M200 with the external mass storage device of YHP9845T. The TSS45 provides semi-segmented sub-programs for the data communication routine and utilities as follows: user-oriented special key operations and data transmission, data receive, all communicating data back-up, data file monitoring, and data file editing using the mass-storage devices. The situation status of RS-232C serial interface device was examined and monitored through several data communication experiments in order to set up the data communication channel through TTY (Teletype) format. Functions of utilities are also tested through the experiments to adjust the “timing” for sending and receiving of the data of data communication format with the FACOM-M200 operating system. A YHP9845T user can handles the terminal emulator “TSS45” as an intelligent terminal system not only for the on-line TSS and RJE network but also for a distributed processing network.

1. はじめに

地質調査所で扱っている地球科学関係のデータは、今のところ各個別で処理されている。地熱データベース¹⁾の開発のように統合化の動きが始まった今、データを如何にして統合・運用するかが大きな問題になりつつある。地球科学のデータは、対象が複雑なため、本来的に分散処理指向を持つ。一方では、地球科学の大テーマに包含統合され得る可能性を持つので、データを結合した時の利益も大きい。この矛盾を解決して、データの統合

を推進しようとする時、融通性のあるデータ通信が必要となる。この問題は、最近重要になってきた分散処理データ通信、つまり計算機同士を通信回線で結合した処理方式の問題でもある。

マイクロエレクトロニクス技術の進歩によって、かつての大型計算機の持つ能力が、現在の小型計算機(ミニ・コンピュータ、又はミニコンと略す。あるいは、マイクロ・コンピュータ、マイコンと略す)においても実現できるようになった。大型計算機の占める大規模・多機能処理の中核機能とその発達をメインフレームと呼ぶことにしよう。かつてのメインフレームは、現在の大型・超大型計算機によって発展的に引継がれている。一方、ミニコンとマイコンの急速な技術的進歩により、低価格ながら高機能な計算業務が可能になる。たとえば、ミニコンの機能をマイコン上で、より発展的に実現した16ビ

* 本論の紹介は、RIPS ニュース, No. 6, p. 2-4 (1983)に掲載された。同ニュース, No. 6, p. 2-40の特集記事は参考になるだろう。

** 鉱床部

1) 花岡尚之・矢野雄策・津宏治・浦井稔・仲澤敏・佐藤功・小川克郎(1982)地熱情報データベース・システムについて。地質ニュース, 第335号(1982-7), p. 33-41。

ット μ CPU (micro Central Processing Unit) の一つであるモトローラ MC 68000 のアドレスバス幅は 24 ビット、アクセス可能な記憶空間が 16 MB (Mega Bytes) もある。これは、1960年代から70年代にかけての大型計算機の代表といわれる IBM/370 シリーズの能力に匹敵する。巨大な部屋を占有する IBM/370 シリーズにくらべて、MC 68000 は小指にも乗せられる小ささであり、周辺装置の小型化も進んでいるため大して場所をとらない。

高性能なミニコンやマイコンを通信回線を介して互いに接続したり、他の計算機、特に大型計算機と接続すると何が起こるだろうか。接続の技術的問題が解決されたあとには、次の利点が残るだろう。

- ① 別々のデータ集合が通信回線を介して結合されるので、有機的結合性をもつ多種データ処理の効率化が促進される。
- ② 計算機処理の上で多大な労力が使われる入力の手間を幾重にも重ねる無駄が省ける。データ移転が容易である。
- ③ 複数の計算機のそれぞれの能力が結合されて利用可能となり、単体で処理をするより乗算的能力拡大が期待できる。
- ④ 複数プロジェクトの情報資源が結合でき、また共用化できるので、各プロジェクトが独立になされた場合よりも大きな成果が期待できる。
- ⑤ 一つあれば済む設備に対する二重投資の無駄が省ける。このため、一つの開発プロジェクトに要する費用が節約でき、しかも既存計算機施設の利用が促進されるので研究開発に対する投資効果の増大が期待できる。

従って、大型ホスト計算機とミニコン・マイコンを通信回線で結合して効率的に利用するなら、ミニコン・マイコンのレベルでも大型ホスト計算機の持つ強大な機能と多彩な周辺装置を共有することができる。大型ホスト計算機もまた、入出力に伴う負荷と無駄から解放されることができよう。それでは、効果的な利用を期待するには、何をすればよいのだろうか。

- ① ハードウェア：低価格かつ必要な機能を搭載した分散処理向きのミニコン(マイコン)を、周辺装置を含めて、一つの開発プロジェクトに適正な規模で設置し、通信回線と接続可能にする。高価な周辺装置などは、通信回線経由で既存設備を低価格で利用する。
- ② ソフトウェア：データ通信を強力に支援できるインタフェースを制御し、通信目的を実現するソフトウェア(データ通信用エミュレータ)を開発する。

ハードウェア的な条件は、現在光ファイバ・ケーブルや電話回線の利用などで困難な点は解消しつつある。静止衛星を利用したマイクロ波通信などが整備されれば更に充実するであろう。問題はソフトウェア的な条件である。特に異機種計算機同士を結合する時には、汎用性のあるインタフェースを通じて、通信手順を制御しなければならない。その一部は、ハードウェア化できる所もある。しかし、実際の通信業務では予期しないこともおこり得るため、柔軟性に富んだソフト部分が残されなければならないことがある。その部分は、通信事情に応じて手直しする必要がある。

たとえば、市販ソフトウェア(ターミナル・エミュレータ)を導入したとき、この手直しができないと、半身不随の通信しかできないおそれがある。普通、エミュレータは、端末機がホスト計算機の端末としてのみ動くところまでしか作られていないことが少なくない。ところが計算機使用者(ユーザー)は、端末装置としてミニコン(マイコン)を通信に利用するだけでは飽き足りない。同時に、ミニコン(マイコン)としての性能も十分に使いたいと考えている。ミニコン(マイコン)で開発したプログラムやデータ資源を有効に活用したい。そのため、エミュレータに必要な修正を加えることがある。もし、市販のエミュレータがソース・コードや内部フォーマット(フォーム、又は様式)をユーザーに公開していなければ、この要求は実現しがたい。市販エミュレータのメーカーが求めに応じてシステムの拡張をはかってくれればよいが、概して応じてくれることは少なく、また、ユーザーの要望の内容を正しく理解して解決してくれるかどうか、はじめから期待しない方がよいことが多い。エミュレータを拡張するには、次の条件が必要である。

- ① ある程度までユーザーの必要性を満足している。
- ② ソース・コードがユーザーに公開されている。
- ③ ソース・コードがユーザーの理解できる言語で書かれている(つまり、機械語や滅多に使わない種類のアセンブラ語ではなく、高級言語に準ずる言語で書かれている)。
- ④ ユーザーの変更しやすい柔軟な構造を持っている。

異なるメーカーの計算機を結合しようとしても、双方から同じレベルの十分な支援が必ずしも期待できないので、ユーザーによる変更は、実際上必須の条件になることが多いだろう。上述の条件を満足できない市販エミュレータを導入した場合には、業務拡張に伴う簡単な変更すら困難となる。せっかくデータ通信を開通しても、逆に自分の手足を縛る結果になりかねないのでは、効果的

な利用どころの話では無くなってしまふ。

工業技術院は、RIPS(研究情報処理システム)と呼ぶ大型計算機のオンライン通信回線網を持っており、中核となる集中処理大型計算機はFACOM-M200である(1983まで)。この回線網は光ファイバを使用している。筆者の属する研究室では、RIPSの端末として、マイコンシステム(YHP 9845T)を用いている。筆者は、この回線網との接続が同時に分散処理データ通信の道も開けるように、そのソフトウェア“TSS 45”を開発した。同様の試みは、工業技術院の中でも行なわれているが、パッチ処理には問題が無いものの、TSS環境での分散処理データ通信のターミナル・エミュレータの成功例はまだ聞いていないという(電子技術総合研究所田中隆氏(RIPS担当)談)。当研究室のマイコンは代表的機種の一つでもあり、工業技術院内でのユーザーも多く、また同系列の機種も含めると、“TSS 45”で試みた例が適用できるケースはかなり多いと思われる。さらに“TSS 45”の主要部は汎用性のある高級言語BASICで書かれているため、他機種でも参考になると考えられる。特に、マイコンを単なるオンライン端末の代わりにしたり、パッチ処理ができるだけのインテリジェント端末のレベルにとどまらざるを得なかったユーザーが、分散処理データ通信のためのマイコンへと機能を高めたい時には、“TSS 45”の経験が参考になるのではないだろうか。

米国で設計された卓上型マイコンの一典型であるYHP 9845T(横河ヒューレット・パッカー社販売)と、日本製大型ホスト計算機の一典型であるFACOM-M200(富士通社製)とを光ファイバによる通信回線で結合した時、多くの困難が発生したものの、前述の諸条件(特にハード面とインタフェース)は基本的に実現できた。たまたま両社が、データ通信と分散処理を推進しようとする積極的姿勢を打ち出していたことが幸いしているだろう。しかし、データ通信のためのエミュレータに関しては、必ずしも目的のものが得られたわけではない。特に、FACOM-M200(以後M200と略す)の作業手順に合わせてYHP 9845T(以後9845Tと略す)のデータ資源を有効に活用しようとした時、既存エミュレータが十分に支援してくれないことが問題となった(機械研究所筒井氏、化学技術研究所杉江氏談)。筆者の研究室では、9845TとM200の双方の既存データ資源を有機的に結合して業務をすすめることを目的にデータ通信用インタフェース(YHP 98046 B)とその制御コードの書かれたROM(読み出し専用記憶素子)を購入し、光ファイバ・ケーブル先端のモデムと結合していた。そのため独自のエミュレータ開発の必要に迫られた。そこで横河ヒュー

レット・パッカー社(以後YHP社と略す)と富士通社(以後F社と略す)の公表されたマニュアル類を用いて、目的に合うエミュレータを開発し、いくつかの通信実験をおこなって改良した(インタフェース購入の4月半ばから完成まで約1カ月間、昭和56年度)。エミュレータを9845Tに実装してのち1年間の使用経験を踏まえて、どのようなデータ通信ソフト・ウェアが有効となりうるのかを、実際に使用しているプログラム・リストを紹介しながら説明し、諸賢の御批判を仰ぐ次第である。プログラム・リストの紹介の前に、まず分散処理データ通信について回顧しよう。なお、9845Tのユーザーは、直接3.から読み始められてもよい。

2. 分散処理とデータ通信

1960年代中頃に登場した第3世代計算機では、すでに実現されていたデータの遠隔計算処理(RJE: Remote Job Entry)のための通信回線を拡張する試みがなされていた。計算機におけるネットワーク(回線網)システムとは、独立した機能を持つ計算機システムが、ハードウェア、ソフトウェア、それにデータを共用できるように通信回線を通じて結合したシステムであると定義できる。計算機のネットワーク・システムは1960年代後半から70年代初めにかけて開発され完成したものとなった。その過程で登場した方法が分散処理と呼ばれる方法である。

ネットワークに先行するオンライン・システムでは、全体を支配する計算機が全ての計算処理を受け持つ。これを集中処理とよぶ。集中処理は支配計算機に負荷がかかりすぎて、何らかの理由から停止するとシステム全体が強い影響を受ける。これをダウンと呼ぶ。支配計算機のCPU(中央処理装置)の負担も大きい。そこでCPUにかかる入出力制御や周辺装置の制御など各種の機能をマイクロ・プロセッサなどを使って分散させる方法がとられるようになった。現在の大型(超大型)計算機はこの機能分散型で構成されている。さらに、処理業務の増大に対応して、同じ処理を複数の計算機に分担させる負荷分散方式もとられるようになった。藤枝(1974)²⁾は計算機ネットワークの利点を次の2つにまとめている；

- ① 共同利用：ハードウェアとしてのCPU、記憶装置入出力装置、その他の周辺装置や、プログラムなどのソフトウェア、さらにデータベースなどのデータ資源の共同利用が実現できる。
- ② 利用効率：共用処理にかかる負荷をバランスよく分散させるだけでなく、共用処理のハードウェア、

2) 藤枝純教(1974)大型情報処理体系(電子計算機基礎講座②)、共立出版、p. 15-19.

ソフトウェア、データ資源の結合によって処理能力と効率を高める。

この分散処理の考え方をおし進めて、より大規模なデータ通信回線網と結合したものが分散処理データ通信の考え方である。データ通信回線さえ確保されれば、異なる機種、異なる能力を持つ独立の計算機が結合され、共同利用とその効率が促進される。集中処理方式や、初期の分散処理では同一のオペレーティング・システム(OS)が必要であったが、現在では異なるOSが流れていても、エミュレータによる制御を通じて分散処理機能が発揮できる。集中処理方式のように一体の支配計算機が全体を監督する必要がないので、一つの計算機が故障しても、他の計算機の処理業務には影響しない。それ故、安全性が高まる。公開したくないデータは、データ通信回線網に乗せなければいけなないので、データ保護がやり易い。この分散処理データ通信、分散処理ネットワーク(回線網)の実用的意味を明らかにするため、一つの事例を比較対照してみよう(これは銀行の例であるが、銀行の各支店を研究室や工場、現場に置き換えてみれば、同様な状況が広く発生している)。

McGlynn(1978)³⁾は、計算機によるデータ通信とデータ処理が大規模に使われる身近な例として、以下の金融機関の事情を考察している。

「金融機関のデータに対する関心は、他産業より著しく厳しい。初期のバッチ処理からデータ通信回線を利用するオンライン処理が普通になった。しかし現在では分散処理への移行が実現し始めている。オンライン処理の段階では、各出先機関の端末が中央の集中処理大型計算機とデータ通信回線網で接続されている。データの処理は大型計算機が一括支配していた。業務の拡大に伴って大型計算機の負担は耐えがたいものになってきた。計算機に故障はつきものだが、中央の大型機がダウンしただけで、全オンライン端末が動かなくなるのでは、業務上の支障も大きくなる。技術革新の必要性が求められた。

金融機関の出先(支店)の作業環境は、分散処理の実現に好都合だった。データ入力や確認、データ処理、顧客へのハードコピー(プリンター)出力、内部記録、監査などは、オンライン処理による集中化よりも、各支店レベルで全て経済的に実行できる。支店はデータ通信回線と“密度の低い接続”をしている。つまり集中処理機が常に監督・支配しているわけではない。顧客がある取引を

したい場合、本人自身の支店ではない支店を利用することもできる。支店同士がデータ通信回線で結合されているからである。各支店は管理運営目的のために支店からの適切な集約データを定期的に収集する上位支店に相互接続されている。この場合は通信回線でも、又は磁気媒体(磁気テープ、磁気ディスクなど)を介してもよい。

注意すべき点がある。集中処理から分散処理に変化するのには、組織環境が変化して分散処理が有利に転じた場合のみという点である。ニューヨーク市のシティ銀行の経験を検討しよう。

各支店は、自身の活動責任を負うものとされる。これにはデータ処理業務の責任も含まれる。支店の管理者は、その活動に使うデータの特性と処理条件に精通している。そのため、目的に最適の分散処理計算機構成を支店レベルで選択できる。実際、各支店ごとに各メーカーの異機種が混在している。

支店での処理は、中央の全社的経営と離れて行なわれる。しかし、金融統制と経営監査や見直しは、通信回線網を使って支店と本店の計算機システム同士で対話させることにより実行する。

分散処理における最重要な点は、銀行データベースを完全に分散することと、集中データベース・集中計算機処理機器の集中度を徐々に低下させる点にある。本店に大型計算機を置いて単一大規模・多重プログラム・多重処理を集中的に行なうシステムと、各支店に分散したミニコンの異機種混合によるシステムとの効率比較が銀行の厳密さで徹底的に検討された。次の4点がポイントとなった。

- ① 大規模集中型データ処理システムを有効に使いこなすには、業務内容について高度な作業知識を必要とする。大規模データ処理そのものも一つの業務であるため、データ処理作業を管理する経験豊富な専門家を必要とする。しかし、そのような専門家は、一般的に、ユーザーの希望条件をかなえる計算機構成を効果的に設計するだけの業務知識を持っていないのが現実だった。つまり、大規模データ処理システムの担当管理者は、計算機のスループット、仮想記憶装置、最適化コンパイラ、及び性能監視などでは、効率的な処理を実行してくれる。だが、銀行の作業担当者にとって値打ちのある特定情報を効果的な形で処理して提供できないのが、経験によれば通例だった。(このようなシティ銀行の評価が、他の企業組織にもあてはまるとは限らない。多くの企業体にとって、単一大規模集中処理の作業規模と集中管理の方が、分散型データ処理より経済効果が大きいと考えられる例はいくちもあるだろう

3) McGlynn, D. R. (1978) Distributed processing and data communications. John Wiley and Sons. (邦訳、分散処理とデータ通信(酒井重恭訳)。共立出版。p. 201-209)。

う。集中処理が問題化するの、は、複数の大型計算機システムを利用せざるを得ぬ大組織体である。この場合には、多数の小型データ処理作業に分散した方が単一大規模集中データ処理よりも管理が効率的になり、エンド・ユーザーへも有効な情報が提供し得る。ただし、昨今のように分散処理向きの小型機が高性能化し、かつ低価格化してくると、それほど大規模でなくとも分散処理を考えた方がよい場合もでてくるだろう。

② 大型(超大型)計算機の導入や、現存機種からより上位機種に変更する時には、その企画と開発に予想以上のリード・タイムが必要だった。大規模集中処理システムでは、変化する顧客の要望に速やかに対応できない。システム設計が最終的に実行可能な形になった時には、なんとこの大規模システムはすでに古くさくなり、顧客の要望に適するものではなくなっていた。つまり、大規模集中処理システムにだけ頼る場合、銀行事業の変化の早さにくらべて、企画・提案・実行に時間がかかり過ぎる。

③ 大規模集中処理システムでは、非常用電源、空調、床あげ、機密保護手段などの環境条件が集中型データ処理作業を一層複雑化し、効率性を削減させている。

④ 大型(超大型)計算機の持つ技術的複雑さも問題点となった。事業の実務上の問題や重要なユーザーの要望は複雑な大型(超大型)計算機の中にそう簡単には翻訳できない。つまり、銀行の管理者は優先順位や区分など事業そのものの必要事項を計算機の専門語で語らないし、計算機のシステム管理者は、計算機の技術的な問題をユーザーの基本的必要事項に必ずしも結びつけられるとは限らなかった。

以上4点の評価から、銀行的決断力の良さでシティ銀行の分散処理ネットワークが発足した。1975年のことである。当初は50台を超えるミニコンの導入から始まり、翌1976年には100台以上になった。各ミニコンは、DEC社、インタデータ社、データ・ゼネラル社、HP社などの異機種による混合構成である。これは、他支店の計算機種を無視し、自支店の作業にもつとも効果的で効率を高める専用計算機システムを配置してよいとする、シティ銀行の分散管理方針によっている。互換性を無くした理由は、汎用データベースの共有を無くし、支店や作業間の情報を自動的にではなく特別要求に応じる形で交換することにしたためである。財務管理と経営監査に関しては、磁気媒体かデータ通信回線により、各支店が結合されている」。

以上要約してきた銀行業務における分散処理の事例から、大規模集中処理システムのみにも頼ることの短所は次の3点にまとめられる。

- ① システム管理者：個々のユーザーのパラパラな用途をきちんと把握し、それぞれに応じた適切な処理手段を提供してくれることは、経験的に期待できない。
- ② ユーザー：大型(超大型)計算機を、自己の必要に合わせて使いこなせるまでに、多くのリード・タイムが必要である。使いこなせるころには、ユーザー本来の専門分野の変化に追いつけなくなる可能性がある。その頃には計算機も進歩して、次の機種が導入されるかもしれない、また新しく勉強し直さなければならない。
- ③ 計算機：技術的内容が複雑なため、導入から平常業務をこなせるまでに多くの時間が必要となる。もし、集中処理計算機が故障すれば、その影響は即時かつ全面的に全支店に波及する。

こうして見ると、多彩な分野があって、変化に富む研究開発業務においても同様の問題が生じてくるだろう。一つのプロジェクトの下にある研究であっても、常時オンラインで結合されて即時密接な連絡をとらねばならない業務は少ない。地震情報の監視のような場合は測定器とデータ処理機がオンラインでつながる必要がある。しかし、オンラインの必要性はそこまでであり、たとえば岩盤力学の実験的研究とオンラインでつながる必要はない。通常この2つの業務は相互に無関係に進行していても構わない。そして必要があった時に結合されればよい。たとえば、実験結果の理論的解析に現場情報を比較検討しようとする場合である。もし、この2つの業務が完全に独立していて相互に何の関連手段も持ち合わせていなければ、データの入力だけでも膨大な時間と労力と予算が必要となってしまう。磁気媒体による結合ができるなら、それを運搬する手間がかかるだけである。データ通信回線網で結合されているなら、即座に必要な情報を呼び出して業務に提供することができる。

分散処理を採用した時には、上述の集中処理の問題点は次のように変わる。

- ① システム管理者：大型(超大型)計算機のスループット、仮想記憶装置、最適化コンパイラ、及び性能監視などや、データ通信回線網の効率的な運用に専念できる。
- ② ユーザー：必要に合った計算機を使用できる。直接の相手がミニコンやマイコンであることが多いので、大型(超大型)計算機使用の習得にかけるほど労

力を費さなくてすむし、管理も容易である。複雑なシステムの運用は、①のシステム管理者の支援が期待できる。

- ③ 計算機：過度の負担から解放されるので、本来の高性能が必要な時に使用できる。
- ④ 回線網：複数のユーザーの業務が通信回線網を通して必要に応じた結合を実現できる。

さて、ニューヨーク・シティ銀行の事例では、汎用データベースを持たない方向だった。各支店が遠隔地にあることもあって、単一大規模集中処理は廃止したが、さらに大型(超大型)計算機そのものの必要性も無くそうとしている。分散処理業務をミニコンのみで実行するのは、一つの行き方であり、シティ銀行の例はその極端な場合の一つである。このような形態の分散処理は、その処理の規模に対する投資の割合が低く押さえられる。

しかし、大型(超大型)計算機の高性能を利用しつつ分散処理化をすすめたい場合もあるだろう。これには、大型(超大型)計算機を分散処理端末の一つにしてしまう場合と、データ通信回線網の運用制御を通じて、回線網で結合される各分散処理計算機の介添役となり各分散処理機で手に負えない業務を手伝うホスト役にする場合とが考えられる。特に各端末がそれほど遠隔地にあるわけでもなく、共通の汎用データベースを扱う時などは、ホスト役としての大型(超大型)計算機がある方が便利なおことがある。これは費用対効果の問題でもある。

工業技術院筑波研究センターの大型計算機M200には、共用回線としての光ファイバ・ケーブル網が存在しているので、分散処理方式への移行が容易である。もちろん集中処理方式の単なる端末としてデータ通信回線を使うこともでき、現在はこちらが主流である。しかし近い将来より高度の分散処理ネットワークに移行するためにホスト計算機M200の回線網制御機能の充実(当然これも分散処理)の他に、M200の位置付けが問題となる。回線網につながる一端末として位置付けられた時は、たかだか高性能計算機も端末の一つにあるということになる。しかし、回線網制御と汎用データベースの運用を扱うホスト計算機に位置付けた場合には、次の役割が期待されてよいだろう。

- ① 中立性：回線網制御にあたって、各分散処理計算機同士の結合をバッファする中立的機能。各端末の独立性が強まるほど、相互の結合を仲介し、円滑にデータ通信を通常したり、異なるネットワーク同士を結合する中立的・代表幹事役的性格をもつ。
- ② 高性能：高性能を必要としつつも高額な装置は共通して使う方が費用対効果が著しく向上する。空き

時間の多い周辺装置もこれに加えてよい。これらは、集中管理した方が効率的であろう。たとえば巨大並列処理装置などをホスト役が管理していれば、各分散処理端末が必要な時に共同利用できる。

- ③ 汎用データベース：大規模な記憶装置を常時動かし続けなければならないような処理を、端末レベルで独占しているのはもったいないので共同利用化が促進される。ホスト役が集中管理していれば、各端末の負担が減少する。

以上のうち本質的に重要と思われる機能は①であり、②と③は大型(超大型)計算機が一端末という場合でも変わらない。回線網の制御や大型(超大型)計算機の性能を引き出す仕事は専門家にまかせる方がよい、というのがシティ銀行の事例で示された4つのポイントの第1点と第4点であろう。

ところで、M200のもつネットワーク制御機構を利用して分散処理端末化を実現する上での問題点は何か。端末側はホスト計算機M200に対してデータ通信レベルでの応答にのみ注意を払えばよく、回線網全体の制御やポーリングを気にする必要がないとすれば、問題は9845 T上で走らせるエミュレータの構成にあるといえよう。

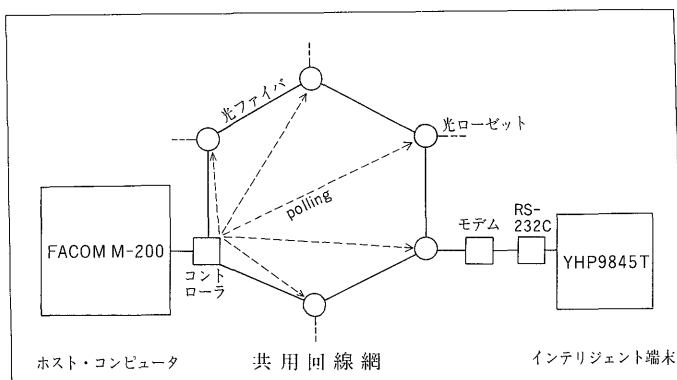
データ通信のための通信回線標準仕様には、IEEE仕様、EIAのRS-232C、同RS-422、同RS-423仕様や、CCITTのV.10、V.11、V.24、V.28、米国国防総省のMIL STD188Bなどがある。9845 Tでは、RS-232C仕様を採用している。これはM200と接続できる。RS-232Cについては付録を参照されたい。

3. “TSS 45”の概要と実験手順

M200は工業技術院筑波情報計算センターが昭和55年末に発足した時導入された。各研究所の各研究室には、光ファイバによる通信回線網に連結された光ローゼットがノードとして置かれ、端末装置はモデムを介して通信回線に結合されている(第1図)。これにより時分割方式の集中処理(TSS)が開始されたのが翌昭和56年4月からであった。

当研究室ではTSS専用の端末ではなく、今まで使用していた9845 Tにデータ通信機能を持たせることで端末の役割も果たせることにした。9845 T上に蓄積したソフトウェアとデータ資源をM200にスムーズに結合したかったからでもあるが、9845 Tをインテリジェント端末として、さらにデジタルイザやXYプロッタなどの周辺装置を作動させる分散処理端末化する目的があった。

M200のTSS専用端末の場合には、スイッチを入れるだけで端末と“ての機能が発揮される(すなわちLOGON



第1図 M200 と 9845T の接続

M200 をホスト 計算機として各研究所の端末が接続されている。回線様式には専用と共用があり、それぞれ内容に若干の相違がある。9845T は共用回線のノードに接続される。この端末がインテリジェント端末の場合は分散処理ができる。現在、各ノード同士のデータ通信や、ノード同士の結合した処理ができるようには、まだ整備されていないため、分散処理ネットワークと呼ぶべきかどうか問題があるかもしれないが、それを指向していると言ってよいだろう。このネットワークはリング型と呼ばれる形式の一つである。

できる。LOGON とは TSS の一つのセッションを開設することであり、LOGOFF はそれを終了するコマンド文である)。ところが9845T は自身がマイコンでM200とは別の OS (オペレーティング・システム) が流れている。独立した OS の流れに M200 との信号のやり取りを割り込ませるためには工夫がいる。まず M200 の回線網制御規約が 9845T の通信仕様と合わなければいけない。通信に使われる文字コードも一致していなければいけない。そのもとに、双方の計算機としての機能を損うことなく、データのやり取りがなされなければいけない。これらの要求を満足するための端末側データ通信ソフトウェアが今回開発したターミナル・エミュレータ“TSS45”である。

はじめに市販エミュレータで通信実験をおこない、順次改良するつもりであったが、市販エミュレータのプログラム・ステップ数は膨大かつ複雑で問題のあることがわかった。ユーザーの必要に合うサブプログラムを付加することも困難だということだった。TSS 環境の中にテープデータを自由に参加させることもむずかしいようだった。

そこで既に購入していた YHP 社のインタフェース使用マニュアル (HP 98046-90010 と HP 98046-90030) を参考にして、新たにエミュレータ“TSS 45”を開発した。開発期間およそ1カ月で実用的なエミュレータ“TSS 45” V1.0 を完成した。V1.0 の作成中、及び完成後に種々の通信実験をおこなって機能向上をはかった。たまたま GEOREF 文献データベースが F 社製のソフト

ウェア (FAIRS-I) 上で実験的に動いていたので、データベースの検索処理を通じて送・受信データのやり取りとその内容を確認した。この間に、データとプログラムの送受信を自動的に実行するサブルーチンや、FAIRS の入力様式を YHP 側の仕様に変更するインタフェース・プログラム、及び外部磁気記憶装置 (カセット磁気テープ：CMT や、フレキシブル磁気ディスク：FD のドライブ装置) を動かして CMT/FD データ及びプログラムをデータ通信に参加させるソフトウェアを整備した。簡単な有限要素法構造解析プログラムを作成して通信実験に使ってみたが、9845T のようなインテリジェント端末 (自ら処理能力を持つ端末) に対する全画面 (フルスクリーン) 編集支援をホスト計算機の M200 が行っていないので、複雑なプログラムの修正と改良は手易くないことがわかった。

TSS 手順におけるインテリジェント端末の有利な点を1つあげよう。TSS の共用回線は、各研究所のサテライト・ステーションの専用回線にくらべて、いく分機能が劣っている。たとえば3分間応答しないでいると、ポーリングの際に入力催促警報が出る。これが3回出されると回線を切断されてしまう。リストを検討したり、CMT/FD (カセット磁気テープ/フレキシブル磁気ディスク) を準備したり、あるいは RJE (Remote Job Entry) を実行して出力待ちのため READY 状態のままにしていづのままに10分間経つと切断されてしまうのである。普通の端末はこれに対処できない。しかし、インテリジェント端末は、それ自身の OS のもとで作動するプログ

ラムを組めるので対策をたてることができる。たとえば、入力催促警報が受信された時、直ちにその時点での作業に影響しない信号(たとえば CR(Carriage Return)コードや STATUS コマンド)を送信することにする。こうするといつ回線が切られるか時計と相談して心配し続ける手間と、回線が切られたあと再接続の手続きをする手間が省ける。そのかわりに利用料金はそれなりに嵩むことになる。

エミュレータ“TSS 45”は、独立した OS が流れ、異なる言語のプログラムを組まねばならない計算機同士を通信回線で接続した時に、片方が端末としての機能を持つるようにしたものである。そのため計算機同士の結合をデータ資源の送受信に限定する仕事の方が向いている。TSS 共用回線では、1200 bps の伝送速度しかないで、データのやり取りはかなり遅くなる。1巻の CMT データを送信するのに TSS 環境で実行すると1時間から2時間もかかってしまう。手作業で入力するのにくらべればずっと早く手間もかからないが、他の転送手段からみると遅い方である。

実際に X 線粉末回折データをいったん 9845T で入力処理しておき、CMT から M200 に転送・収納する実験を1カ月ほど続けてみた。9845T は多重処理ができないので、TSS 端末としてふるまっている間は、他の業務が停滞するという短所が明らかになった。これは、データの入出力、転送、プログラム開発などの複数の仕事を一台で一度に一つしかできないため、分散処理端末とはいうものの、M200 に対しては分散処理ながら自身は集中処理という性格のためである。

エミュレータ“TSS 45”の1年間の使用実績をもとに、さらにユーザーの必要性に合うよう改良したものが、“TSS 45” V2.0 である。V2.0 が現在使われている仕様であり、以下の各節で解説される。V1.0 は個別の用途にそれぞれ特別化させすぎていたし、プログラム構造が複雑に入り組んで、あまり見易くはないであった。V2.0 では、各サブルーチンをできるだけ“独立”させ、一般性を持たせて、ユーザーが改変し易いように作っている。

“TSS45”で使用する言語は、YHP 社製の ANSI 拡張 BASIC である。ANSI (American National Standard Institute) の提案した BASIC の諸機能を持っている。しかし、ELSE 文や WHILE 文が使えないので、十分に構造化できず、セグメントに分けることも十分でなく、いまひとつすっきりしない。次の点は特長的である。

① ラベル分岐：IF 文、GOTO 文、GOSUB 文などにおける分岐指定先にはプログラム・ライン番号を使うのが従来の BASIC 文法だった。これに加えて、ライン番号のあとに14文字以下の英大文字で始まるラベルをつけ、最後にコロンの示してプログラム・ラインを指定することもできるようになった。この利点はユーザーに分かりやすいラベルでプログラム・ラインを指定できる点にあり、作業内容が見通しよく、編集時に大変便利である。ライン番号指定しかできないと、分岐指定をする時には指定先のライン番号が正確に予想できないので、しばしばメモをとって忘れないでおく必要があった。ラベルで分岐先指定をすれば、必要な時にそのラベルで始まるプログラムを付け足せばよい。ただし、長いプログラムでラベルの数が多数にのぼる時には、どのラベルがどのラインにあるか一目で決しがたくなることがある。ELSE 文、WHILE 文など、分岐先指定のいらない構造化プログラミング手法が加えられれば、プログラムを見易いセグメントに分けるのに効果的になる。9845T はまだそこまでできない。

② 文字列配列：変数は英大文字で始まる14文字以内のラベルで指定し、変数の数は容量の許す限り十分多く定義できる。配列変数になった場合には、最大6次元、32767要素までとることができる。このため文字列(配列)は主記憶容量による制限と6次元32767要素の制限以外を考えなくてよい。

③ 行列演算：これは BASIC というより、ユーティリティ ROM であるが、行列への値の代入や行列演算を、あたかも行列を1変数とみなして行なうことができる。逆行列や行列式の値が一行のプログラム・ラインで求められる。配列が文字列の時に威力を持つのが並べ換え命令であり、多数の文字列の集合を、ASCII の順番に整理し直すことができる。一行で実行できる淘汰命令の実行時間が速いので、簡易データベースや表入力の簡易プログラムを作成するのに便利である。

④ 通信機能：送信データは文字列に入れて CWRITE 文で A-1 アセンブリに送る。この時、バッファが存在する。A-1 アセンブリからの受信データも一たんバッファに入れられてから CREAD 文とよって文字列にとり出す。つまり BASIC 文法で通常使われる PRINT 文と INPUT 文と同じような働きに、ユーザーからは見える。

特に①の働きを上手に使うとプログラムをある程度構造化・セグメント化できる。すると、従来よく使われて

第1表 “TSS45” で使われる配列変数

配列変数名	最大要素数	最大文字長	内 容
A\$	1	160	受信データ
B\$	1	160	送信データ, キー入力文字
C	11		外部記憶装置のレコード長, バイト数などの定義
D\$	7	6	外部記憶装置の装置番号
F\$	11	80	送受信ファイル名
K\$	32	80	特殊キーの定義
P\$	1	160	出力データ
Rips\$	11	160	ホスト電算機から受信されるメッセージの代表的文字列
S\$	11	80	Rips\$ に対応して送信するメッセージ
V	801		V\$ のステータス
V\$	801	160	外部記憶装置上のデータ編集用
W\$	1	160	送信データ
Wb\$	3	160	送信データ・メモ(リコール用)
X\$	1	160	外部記憶装置上のデータのモニター
Y\$	1	160	送受信データ1行のバックアップ
Yr\$	1	160	受信データ1行分の文字列

配列宣言が必要な変数とその規模・内容を示す。配列をもたない変数は示していない。

きたフローチャートの類はかえって繁雑で、プログラム作成上は不要の存在になる。そこで、プログラムはフローチャートを作らずに書き下すことにした。プログラム・リストの解説も、ANSI BASIC の知識を前提にして直接読み下して理解してほしい。なお、第2図と第5図一第17図までは一連のプログラムであるが、説明のために分割した。

4. データ通信回線の開設と待機

“TSS 45” (V2.0) で使われる配列変数の規模とその用途を第1表で解説した。初期値の設定と特殊キー(32個まで使用可能)の定義の行程を第2図に示す。初期値は通常のTSS端末として使うのに必要な数値と文字列に与えられる。第2図の55行から78行までは、M200のメッセージに対してユーザーがキーボードから入力しなくても自動的に対応・送信するメッセージ群を定義する。ここでは送信促進警告特有のメッセージ部分(55行)、最初のLOGON(開始)のためのメッセージ(56行)、LOGOFF(終了)後のメッセージ(60行)だけ定義しておいた。61行以下には、ユーザーが必要に応じて、M200から受信するメッセージ(Rips\$(I))とこちらから送信するメッセージ(S\$(I))をそれぞれ一行分、配列番号を合わせて定義できる。この時、メッセージ群の最大数(Mxrips=Iの最大値)を定義する。Mxripsが10を超える場合は第2表の配列宣言で、Rips\$(10)とS\$(10)を必要ならだけ拡張すれ

ばよい。80行から111行までは第3図に示した特殊キーの説明である。

初期設定が済んだら、まず通信回線を開いた状態にしなければいけない。第5図はその手順を示している。第4図はYHP社のマニュアル No.98046-90010による開設手順である。第5図の内容は、基本的には第4図の110行から220行までに従っている。以下第5図に従って説明する。

RS-232C シリアル・インタフェース HP 98046B は、9845Tのシステム・バスに対してセレクト・コード4番と5番により接続されることにしてある。4番は送受信用であり、5番は受信の到来を知らせるコードである。

第5図において、113行は、コード番号4番をすべての接続からいったん切断する作業である。次いで114行でデータ通信用に1500B(バイト)の記憶域を確保する。回線制御様式はこの記憶域を利用する。セレクト・コード4番の状態表示(ステータス)CSTAT(4, n)の要素nは回線制御状態を知らせてくれる。コード4番のステータス要素5は回線の接続状況を示し、CSTAT(4,5)=0の時は回線が繋がっておらず、CSTAT(4,5)の値が2になった時はじめて回線が接続される。

116行は、モデムとは接続を維持するが、通信回線網にはまだデータを送信しない状態にする。これはモデムと接続しながら、データ通信仕様を変えるために必要である。117行でM200のデータ通信仕様に合わせた設定を定

```

20 Default: Hardcopy=16
21 PRINTER IS Hardcopy
22 STANDARD
23 Sc=Select=4
24 Speed=1200
25 Noactivity=0
26 Cr$=CHR$(13)
27 Lf$=CHR$(10)
28 Bs$=CHR$(8)
29 Eof$=C-3&Lf$
30 End$="END"
31 Record=Upload=Backup=Num_cut=Datatype=Heading=Graphics=Edit=0
32 S=I=Write=Device=Step=1
33 !
34 C$="STRING TEXT"
35 C(1)=21000
36 C(2)=10
37 C(10)=72
38 Cursor=95
39 DS(1)=":F15"
40 DS(2)=":F14"
41 DS(3)=":F8"
42 Dev1=1
43 Dev2=2
44 Dev3=3
45 FS(0)=" "
46 FS(1)="TESTR"&DS(Dev1)
47 FS(2)="TESTL"&DS(Dev2)
48 FS(3)="BACKUP"&DS(Dev1)
49 FS(4)="BACKUP"
50 FS(5)="TESTL"
51 Num1=32
52 Num2=254
53 P$=""
54 Rips=0
55 Rips$(0)="KEQ54112A"
56 SS(1)="LOGON TSS 09999/PASSWORD"&Cr$
57 !
58 ! User definition of RIPS message for AUTO-RIPS operation.
59 Mxrips=1 ! Maximum number of operation.
60 Rips$(1)="KEQ54100I" ! Rips$(I):RIPS message.
61 Rips$(2)=" " ! SS(I):HP9845T message.
62 SS(2)=" "
63 Rips$(3)=" "
64 SS(3)=" "
65 Rips$(4)=" "
66 SS(4)=" "
67 Rips$(5)=" "
68 SS(5)=" "
69 Rips$(6)=" "
70 SS(6)=" "
71 Rips$(7)=" "
72 SS(7)=" "
73 Rips$(8)=" "
74 SS(8)=" "
75 Rips$(9)=" "
76 SS(9)=" "
77 Rips$(10)=" "
78 SS(10)=" "
79 !
80 Dkeys: K$(0)="CHR$"
81 K$(1)="RECALL LINE"
82 K$(2)="EDIT LINE MODE"
83 K$(3)="GRAPHICS ON/OFF (NOT USE)"
84 K$(4)="AUTO RIPS ON/OFF"
85 K$(5)="ALLOCATION"
86 K$(6)="LOGON"
87 K$(7)="LOGOFF"
88 K$(8)="BACKSPACE"
89 K$(9)="HARDCOPY ON/OFF"
90 K$(10)="UPLOAD ON/OFF"
91 K$(11)="RECORD ON/OFF"
92 K$(12)="END"
93 K$(13)="EXPLAIN KEY DEFINITION"
94 K$(14)="LIST"
95 K$(15)="BREAK"
96 K$(16)=" "
97 K$(17)=" "
98 K$(18)=" "
99 K$(19)="DUMP GRAPHICS"
100 K$(20)="GCLEAR"
101 K$(21)="REVISE MT"
102 K$(22)="READ ALL DATA"
103 K$(23)="CAT"
104 K$(24)=" "
105 K$(25)=" "
106 K$(26)=" "
107 K$(27)=" "
108 K$(28)=" "
109 K$(29)="BACKUP"
110 K$(30)="CREATE"
111 K$(31)="INITIALIZE"
112 !

```

第2図 暫定値と初期値

配列の初期設定をあらかじめ定義する。この設定のあるものは暫定値でもある。データ通信の基幹規約部分(たとえば Speed=1200 など)は、変更してはならないが、その他は状況に応じて変わりうる。自動応答文や特殊キーの設定もここでこなう。

義する。ここで定義しないものは、あらかじめ内部的に決められた暫定値が採用される。設定内容を第2表に示した。

118行は伝送速度を 1200 bps とし(Speed=1200)、接続はモデムのシグナルをモニターして制御する方式で(HANDSHAKE ON)、RS-232Cの第6,8ピンが400ミリ秒の間作動しない時は接続を止める(LOST CARRIER=400)、受信が無い時の切断する時間を無限にする(NO ACTIVITY=0)。

これで準備完了なので、モデムに対して121行を働きかけ続ける。XONはデータ通信開始の信号をモデムを介してホスト計算機に送る合図であり、ここで始めてデータ通信回線に信号が乗ることになる。ホスト計算機M200から受信可の信号がモデムで変調されて9845Tに到来する。その時121行のCSTAT(4,5)の値が0から2に変わる。そこで123行に移り、ASC II文字をすべて送信することをM200に宣言する。ホスト側が受信可能になったところで、125行でCR(Carriage return)コードを送信する。

126行は送受信時の割り込み先を行ラベル Sc 4 に分岐させ、127行は受信時の割り込み先を行ラベル Receive に分岐させることを設定する。これらの割り込みは最優先される。この割り込み設定はDISABLEコマンドで解除され、ENABLEコマンドで再び作動する。

128行はエラーが生じた時の割り込み先を行ラベル Error に指定してエラー原因を表示させ、エラーの原因を除いてから復帰することを設定する(第9図468-494行参照)。

第5図の129行から133行は待機ループである。通常、送信も受信もない状態でのこのエミュレータの走行は129行から133行までを繰り返して続ける。

データ通信を開設した直後に、M200からLOGON許可の信号を送ってもらうためにBREAK信号を送信する必要がある。それが134行から137行であり、129行と130行でS=1の場合にのみ作動する。いったんLOGON可能状態に入ると、Sは2以上の値をとるので129行と130行はとび越えられる。BREAK信号を送ってからCRコードを送るよう、M200から求められているが、連続して送信したところ、M200が混乱してつながらなかった。M200がうまく反応してくれるよう試行錯誤を続けた結果、BREAKとCRコードの間に1.5秒の間隔を置くと、うまくLOGONできることがわかった。M200の御気嫌が日によってマチマチなので、この間隔はもっと短かくてもよいこともあるが、1.5秒にとっておくと失敗が無く、経験的にはうまくいっている。これで十分である。

(A)

K 16	K 17	K 18	K 19	K 20	K 21	K 22	K 23
			DUMP GRAPHICS	GCLEAR	REVISE MT	READ ALL	CAT
K 0	K 1	K 2	K 3	K 4	K 5	K 6	K 7
CHR\$	RECALL	EDIT	GRAPHICS ON/OFF	AUTORIPS ON/OFF	ALLOCATE	LOGON	LOGOFF
K 24	K 25	K 26	K 27	K 28	K 29	K 30	K 31
					BACKUP MT ON/OFF	CREATE	INITIALIZE
K 8	K 9	K 10	K 11	K 12	K 13	K 14	K 15
BACKSPACE	HARDCOPY ON/OFF	UPLOAD ON/OFF	RECORD ON/OFF	END	EXPLAIN KEY	LISTC	BREAK

```

17 User:      IF B$="!" THEN Break
(B) 18        RETURN
    19        !
    
```

第3図 特殊キーの配置

(A)は特殊キーのキーボード上の配置である。特殊キーは上段8列下段8列で2行並んでいるので、16個のキーしかないようにみえる。シフトキーを押してキーを押すことにより、数字に15を加算した番号のキーの働きをするので、合計32組のキー操作ができる(0-31番まで)。(A)の1行目はシフトキーを押した上段キー、2行目は、シフトキーを押さない上段キー、3行目はシフトキーを押した下段キー、4行目はそのまま(シフトキーを押さない)の下段キー操作を意味する。

(B)はユーザーの定義する普通キーによる特殊キー操作のサブルーチンである。例として!の場合を示した。ユーザは、必要な機能を追加すればよい。

プログラムが走る時には、各プログラム・ラインにプログラム・カウンタ(PC)がおかれて、その行を実行する。インタプリタ型なので各行は独立して実行可能である。プログラム作成時にはこの方が扱い易い。以下の各節ではプログラム走行中の実行ラインを示すためPC(プログラム・カウンタ)を使用する。

5. データ送受信の割り込み

第4図の割り込み方法240-300行と320-360行はきわめて簡単であるが、M200と接続した時には作動しない。RS-232Cは8945Tのシステム・バスに対してセレクト・コード番号4番と5番で接続されている。4番は送受信用であり5番は受信を知らせる。通信実験したところ、5番は受信が終了した時、そのステータスをゼロに戻してしまう。ところが受信された文字コード群は、いったん受信バッファに入れられてからCREADでとり出さなければ、エミュレータの中で扱えない。第4図の方法がうまくいかない理由は、受信が早く終了すると180行の割り込みが働かず、170行も働かずに220行の待機ループに戻ってしまうところにあった。

このための対策をたてるため、入出力内容を表示する

CDUMP コマンドを用いて、システム・バス側の中身をモニターしてみた。しかし、原因がわからなかったのので、次に通信回線制御の状態を示すはずのCSTATUSの各要素をすべて調べてみた。その結果、要素4が入出力キューの残存容量を示しているのので、これを使うことが考えられた。すなわち、入力情報を全部9845Tのエミュレータの文字列A\$に移した時、どの程度まで残存容量が回復するかを調べた。

実験によれば、第2表の設定の時、482以上になれば受信データがすべてA\$に移されたのと同じ意味を持つことがわかった。第5図の行程に続く第6図の146行は、受信による割り込みが働いた時、入出力キュー上に受信データがまだ残っている限りは第5図129行から133行のループに戻らないようにしている。この問題はかくして経験的に解決した。

第6図の139行は、キーボード上で何らかのキー押しがあった時、PC(プログラム・カウンタ)をラベルT(152行)に分岐させてそこから下へと実行させる割り込み命令である。送信時のデータ操作に使われる割り込みであり、受信時の割り込みより優先順位を低くしてある。

キーボードから何か入力された時、たとえば文字Lが

```

10 | *****
20 |
30 | The following program is an example of an interrupt driven LINE MODE
40 | terminal emulator. Defaults are assumed where possible. This is for
50 | a direct (NO MODEM) connection to an H.P. 3000. If a modem is used,
60 | delete "HANDSHAKE OFF" from the CONNECT statement (line 140).
70 | *****
80 | !! YOU MUST PRESS THE CONTINUE KEY TO TRANSMIT THE DATA !!
90 | *****
100 DIM A$(30),B$(160) ! Dimension the string variables
110 DISCONNECT 4 ! Ensure you are disconnected
120 CCOM 1500 ! Specify the CCOM memory allocation
130 CMODEL ASYNC,4 ! Define the CMODEL (ALL DEFAULTS)
140 CONNECT 4;HANDSHAKE OFF ! Link is connected here
150 CCONTROL 4;ACK ON ! Enable ENQ-ACK handshake
160 |
170 ON INT #4,1 GOSUB Transmit ! Interrupt when a PROMPT is detected
180 ON INT #5,2 GOSUB Receive ! Interrupt when an INSEP is received
190 |
200 CWRITE 4;ENDLINE ! Send C/R (OUTSEP) to host (I am ready)
210 |
220 Spin: GOTO Spin ! Wait here for an interrupt
230 |
240 Transmit: !
250 IF NOT CSTAT(4,2) THEN RETURN ! Check for a PROMPT before continuing
260 CREAD 4;A$ ! Get the visual prompt character
270 PRINT A$ ! Print the visual character (e.g.,":")
280 LINPUT "Enter data",B$ ! Enter data, press CONT key to send
290 CWRITE 4;B$,ENDLINE ! Send data and C/R (OUTSEP)
300 RETURN
310 |
320 Receive: !
330 IF NOT CSTAT(4,1) THEN RETURN ! Check for an INSEP (ALEPTN not used)
340 CREAD 4;A$ ! Read the data from the host
350 PRINT A$ ! Print the data from the host
360 GOTO Receive ! Go check for more input data
370 |
380 END

```

第4図 YHP1000/3000 シリーズとの回線開設手順

同じメーカーの機器同士の場合には、簡易なプログラムで開設を実現できる。設定値の内容は第7図と少し異なるが、基本的な手順は第7図に受け継がれている。(図は横河ヒューレット・パッカード株式の許可を得て、同社マニュアル No. 98046-90010 の p.1-3 より転載した)。

```

113 Main: CDISCONNECT Sc
114 Conmem: CCOM 1500
115 ON CSTAT(Sc,5)+2 GOTO Connect,Connect,Main,Start,Main
116 Connect:CDISCONNECT Sc;HOLD
117 CMODEL ASYNC,Sc;CHECK=2
118 CCONNECT Sc;SPEED=Speed,NO ACTIVITY=Noactivity
119 Conloop:IF CSTAT(Sc,5)=2 THEN Conctl
120 DISP "Waiting for connection-Press DTE2 & enter 030 on MODEM"
121 CCONTROL Sc;XON
122 GOTO Conloop
123 Conctl: CCONTROL Sc;READALL ON
124 Start: OFF INT #Sc
125 CWRITE Sc;Cr$
126 ON INT #Sc,1 GOSUB Sc4
127 ON INT #Sc,1,2 GOSUB Receive
128 ON ERROR GOSUB Error
129 Begin: IF S=1 THEN DISP "Begin TSS (Wait until displayed)"
130 IF S=1 THEN Bbreak
131 DISP
132 GOSUB Sc4
133 GOTO Begin
134 Bbreak: GOSUB Break
135 WAIT 1500
136 CWRITE Sc;Cr$
137 GOTO Begin
138 !

```

第5図 データ通信回線の開設

インタフェース YHP98046B内の回路が作動する条件を設定して、信号を M200 に送信し、M200 から開設了解の信号を受信するまでの手順。

入力されたことにした時、PC はどう動くだろうか。まず第6図のラベル T (152行) に分岐する。152行は ROM を使って解読され、CPU の演算可能な形にして実行する。キーボード入力した時の文字コード(ASC II)は KBD\$ の中に L として入るので、送信用文字列 BS に L が代入される。この実行が済んでから PC は次の 153 行におかれる。もし L ではなく、BS に何も入っていな

れば、PC は再びループ129-133行(第5図)に戻る。今は BS="L" のため、PC は次の154行に移る。154行は、第3図(B)に分岐させる実行で、BS の内容がユーザーが定義した特殊実行文字に相当するかどうかを調べる。第3図(B)では、記号!が入っていた時に BREAK信号を送ってM200の作業を中断させ、M200をアイドル状態にする Break ルーチン(第8図(B)406行)に分岐させるオプションのみがついている。さらにオプションを加えたい時には、ユーザーは17行と18行の間にプログラム・ラインを挿入すればよい。当然Lは!ではないから、Break ラベルへの分岐はなされず、また PC が元に戻って来て、次の155行(第6図)に進む。これは、キーボード上の通常の英数字と記号以外のキー(たとえば特殊キー)が押されたかどうかを調べるプログラム・ラインである。

特殊キーを押した時には、KBD\$ の最初の文字列文字の ASC II番号が255のため、PC は次の行にすすんで、どの特殊キーかを判定、各々のルーチンに分岐してから待機ループ129-133行(第5図)に戻る。Lは ASC IIの76番なので、ラベル Tsend(167行)に PC を進めるべきである。167行から179行までは CRコードが入力されたか否かを確認し、CRコードなら CRを含めた一行の文字列として認識されるべきことを送信する。CRコード

第2表 データ通信制御規約の定義

定義名	設定値	内 容
ALERTN	32767	入力データの終了を確認するための、1入力ブロック内最大数。この設定値を超えるごとにターミナルが働くのと同じ状態が入力キューに与えられる。
CHARLENGTH	7	1語の長さが7 bits。
CHECK	2	送・受信とも1語に対して偶数パリティ・チェックを行なう。
FULL DUPLEX		全二重方式をとる。
GAP	0	1語の起動 bit と次の1語の起動 bit の間隔の最小時間を0秒に設定する。
INBUFFER	260	入力バッファの規模を260 bytesにする。
INSEP	CR/LF	入力行の分割指示文字をCRとLFにする。
MEMLIMIT	500	入出力キューの規模を500 bytesにする。
OUTSEP	CR	出力行の分割指示文字をCRにする。
PROMPT	DC1	ホスト側(M-200)の受信可能指示文字をDC1にする。
STOP BITS	1	1語の終了 bit を1個にする。
TBUFFER	130	送信バッファの規模を130 bytesにする。

テレタイプ型(TTY)手順として、第5図の117行で設定する内容。

が入力されなければ、M200側は未完成の行が引き続き送信されてくると期待している。LはCRコードではないから、ラベル Sending(181行)に分岐する。

もし、まだデータ通信を開設しはじめて、LOGONをM200が正しく処理していない時は、進行係数Sが1のままのため、CRT画面上にWait until display READYと表示するのが、181行の実行内容である。S=1なら198行でB\$の中にLOGON情報が代入され、行終端指示子のCRコードを付してM200に送信される。これはB\$に何が入っていてもすべて第2図56行のS\$(1)に置き換えられるので、LOGONの操作以外は何もM200に送信しないことを意味する。

現在ではすでにLOGONされて、Sは2になっているものとして、181行は無視されPCは次の182行から197行までを実行させる。これらはCRT画面上の映りを見易くするためのものである。後述する第7図の説明でもふれるが、モデムを介したデータ通信中は、YHP社のマニュアルに記載されていない現象がしばしば起こる。前述のセレクト・コード4番の第4要素ステータスが482になることが、バッファ内の全受信データを読み出したことを意味することになったのも、マニュアルだけでは解決のつかないことであった。182行から197行も、マニュアル通りに(たとえば第4図の270、350行のように)書くと、その実行行が意味しているはずの作業をしてくれない。

182行から197行の画面表示作業は、プログラム命令としては奇妙な形(CRT特殊機能などへの接続を意味するPRINT USING文中の“#,K”; CHR\$(27)&“&a...

...)”をしているが、カーソル[CHR\$(Cursor):ここでは、Cursor=95(第2図38行)]としてアンダーラインで表示)表示を操作する必要があるものの、試行錯誤によって経験的にこの方がよいことになった。はじめは、Backspaceコードなどを使おうとしたこともあった。

さて、PCが199行に移るとしよう。今はキーボードから入力しているので、199-200行は必要ないが、もし外部記憶装置のデータを転送する場合には有効である。転送モードUpload=1になっている時には、B\$には外部記憶装置上のデータ1行分が入っているはずなので、それをM200に転送する。

199行のCWRITE文で、文字列B\$にさらにCRコードを付加したのは、B\$のまま転送すると、自動的にCRコードが付加されるわけではなかったからである。9845T内では、CRコードがなくても、それとまったく同じ働きをしてくれるが、M200側はCRコードがないと1行が終了したとみなさず、1行に許容限界を超えるデータが次々に入力されたとみなして受け付けてはくれない。

この問題は、CRコードなしの行をまとめて送り、不足部分をキーボードから入力させる方式をとろうとした時、B\$しか送信していないはずなのに、モデムを介してデータを受けとったM200側は、B\$のあとに英数字と記号以外の何らかのASCIIコードを多量に受け取ったと認識してしまう事故が生じた時に更に厄介になった。まったく同一の作業を繰り返しても、10に1つぐらいは、B\$がB\$の内容だけ送られたわけではないことがあったからである。

```

139 Sc4: ON KBD 9 GOTO T
140 IF NOT CSTAT(Sc,3) THEN Transmit
141 Scrcv: DISABLE
142 CREAD Sc;AS
143 GOSUB Disp
144 ENABLE
145 IF CSTAT(Sc,3) THEN Scrcv
146 Transmit: IF CSTAT(Sc,4),482 THEN Scrcv
147 IF Write=1 THEN Sc4
148 PRINT USING "#,K";CHR$(27)&"a0C",CHR$(Cursor)
149 Heading=1
150 GOTO Sc4
151
152 T: BS=KBDS
153 IF NOT LEN(BS) THEN RETURN
154 GOSUB User
155 IF BS/1,1/. CHR$(255) THEN Tsend
156 Keyselect=NUR(BS/2)
157 IF (Keyselect 53) AND (Keyselect.80) THEN TKey2
158 IF Keyselect.15 THEN TKey1
159 IF (Keyselect=18) OR (Keyselect=19) OR (Keyselect=20) OR (Keyselect
=21) THEN Tsend
160 IF Keyselect=22 THEN GOSUB Backspace
161 IF Keyselect=23 THEN BS=""
162 IF Keyselect=17 THEN Pause
163 GOTO Tsend
164 TKey1: ON Keyselect+1 GOSUB Key0,Key1,Key2,Key3,Key4,Key5,Logoff,Back
space,Hardcopy,Key10,Key11,Key12,Explan,List,Break
165 GOTO Tsend
166 TKey2: ON Keyselect-63 GOSUB Key16,Key17,Key18,Key19,Key20,Key21,Key22,Cat,
Key24,Key25,Key26,Key27,Key28,Key29,Key30,Key31
167 Tsend: IF NOT LEN(BS) THEN RETURN
168 IF Write=1 THEN Heading=0
169 IF NUM(BS)7.255 THEN Sending
170 PRINT USING "#,K";CHR$(27)&"a-1C", " "&Eof$
171 DISABLE
172 W$=Cr$
173 IF NOT POS(WB$(1),Cr$) THEN WB$(2)=WB$(1)
174 IF POS(WB$(1),Cr$) THEN WB$(2)=WB$(1)/1,POS(WB$(1),Cr$)-1/
WB$(1)""
175 GOSUB Write
176 ENABLE
177 Write=1
178 RETURN
179
180
181 Sending: IF S=1 THEN BS=""Wait until display "&CHR$(132)&"READY"&CHR$(128)&Eo
f$
IF Heading=1 THEN PRINT USING "#,K";CHR$(27)&"a-1C"
Heading=0
182 IF Upload=1 THEN Upup
183 IF LEN(BS)1 THEN PRINT USING "#,K";CHR$(27)&"a-1C",BS
184 IF LEN(BS)1 THEN Hs
185 IF BS=BS$ THEN Hs
186 IF Write=1 THEN PRINT USING "#,K";BS&CHR$(Cursor)
187 IF Write=1 THEN PRINT USING "#,K";CHR$(27)&"a-1C", " "
188 IF Write=1 THEN PRINT USING "#,K";CHR$(27)&"a-1C",BS&CHR$(Cursor)
189
190
191 Hs: IF Hardcopy=0 THEN PRINTER IS Hardcopy
IF Hardcopy=0 THEN PRINT BS;
GOTO Hw
192
193
194 Upup: PRINT USING
IF Hardcopy=0 THEN PRINTER IS 0
IF Hardcopy=0 THEN PRINT USING "#,K";BS&Cr$
195
196 PRINTER IS 16
197 Hw: IF S=1 THEN BS=S$(1)
IF Upload=1 THEN WRITE Sc;BS&Cr$
IF Upload=1 THEN Rtrn_s
198
199 WS=BS
200 WB$(1)=WB$(1)&BS
201 IF POS(WB$(1),Cr$) THEN WB$(2)=WB$(1)/1,POS(WB$(1),Cr$)-1/
GOSUB Write
202
203 Rtrn_s: ENABLE
Write=Write+1
IF Upload=1 THEN Write=1
IF S.2 THEN S=S+1
204
205 RETURN
206
207
208
209
210

```

第6図 データ送受信の割り込み手順
送信準備と送信手順のソース・リスト。

実は調べてみると、タイミングの問題だった。B \$ & Cr\$ (Cr\$ は CRコード：第2図26行で設定)を送信したつもりでも、続いてすばやく別のBSを送信すると、前に送ったCRコードをM200が取りこぼしてしまうことがあったのである。このような場合には送信間隔をあげる必要がある。

しかし、通常の外部記憶装置上のデータを送信する時は、いったんM200からの送信許可信号を受信してから送信することになっている(後述)ので、取りこぼしは無い(ここに“TSS 45”の工夫点がある)。

転送モード Upload が1以外(通常は0か-1)の時には、キーボード等から入力されたBSを201行から204行(第6図)で送信する。ここでW\$におきかえたのは、しばらく送信文字を保存しておきたいためで、送信のみの

```

211 Receive: IF NOT CSTAT(Sc,1) THEN RETURN
212 R: CREAD Sc;AS
213 GOSUB Disp
214 IF S=1 THEN Sending
215 RETURN
216
217 Rcv_ln: IF LEN(Y$)+LEN(AS) Mxchr-1 THEN Y$=Y$&Cr$
IF LEN(Y$)+LEN(AS) Mxchr THEN Y$=Y$&AS
IF NOT POS(AS,Cr$) THEN RETURN
220 VS(0)=S
221 IF Edit=1 THEN GOSUB Key2end
222 IF Graphics=1 THEN GOSUB Plot
223 Y$=""
224 GOSUB Rcv_op
225 IF Rips=I THEN Rcv_user
226 RETURN
227
228 Rcv_op: IF NOT POS(VS(0),Rips$(Rips)) THEN RETURN
229 BS=Cr$
230 GOSUB Sending
231 PS="DATA ENTRY DELAYED"
232 BEEP
233 GOSUB Print
234 RETURN
235
236 Rcv_user: IF POS(VS(0),Rips$(Rips)) THEN Rcv_up
Rips=Rips+1
IF Rips.=Mxrips THEN Rcv_user
Rips=1
241 Rcv_up: IF Rips=1 THEN Rcv_pas
BS=SS(Rips)&Cr$
242
243 GOSUB Sending
244 PS=SS(Rips)
245 GOSUB Print
246 Rips=1
247 RETURN
248
249 Rcv_pas: PRINT USING "#,K";CHR$(27)&"a-1y0C"
250 PRINT
251 BS=""
252 Y$=""
253 RETURN
254
255 Disp: IF Heading=0 THEN Disp_p
PRINT USING "#,K";CHR$(27)&"a-1C"
IF (AS/1,1/=Cr$) OR (AS/1,1/=Eof$) OR (AS/1,1/=Lf$) THEN PRINT USIN
G "#,K";CHR$(27)&"a-1C", " "
256 Disp_p: PRINT USING "#,K";AS
Heading=0
IF Hardcopy=0 THEN PRINTER IS 0
IF Hardcopy=0 THEN PRINT USING "#,K";AS
PRINTER IS 16
IF POS(AS,Rips$(1)) THEN Rcv_pas
IF NOT LEN(AS) THEN RETURN
265
266 GOSUB Rcv_in
IF Backu=1 THEN GOSUB Bu_rcv
VS(0)=S
IF Record=1 THEN GOSUB Recording
IF Upload=1 THEN GOSUB Uploading
270 RETURN
271

```

第7図 データ受信の割り込み手順
データ受信の割り込みで分岐するソース・リスト。

サブルーチン(ラベル Write: 第9図476行から480行)を独立させるためである。BSに何か入っている時、別のコードを送信する必要がある時に使用される。

202行と203行は、送信データ1行分をWb\$(1)に次々と付け加えて1行にするためであり、CRコードが入った時に、その1行はWb\$(2)に収納されて残される。今入れたデータをまた送信したい時に使われる。Upload=1の時には、自動的にデータ転送がなされているので、このルーチンは省かれる。

205行から209行までは、接続開始モード(S=1)を接続中モード(S=2以上)に変え、次にCRT画面上に表示する時には先頭から書き始めるのか否か(Write=1か2以上か)を決める。Writeは1行中の列を示す。ラベル Sending(181行)に始まり209行至るサブルーチンは、送信と送信の画面処理として以後多用される。かくして文字LがM200に送られた。

第7図は第4図の320行から360行に相当する受信とその表示のためのサブルーチンである。インタフェースのステータス要素1が0の時は受信状態にないはずなの

で、そのまま待機ループに戻ってしまう。0でない時には入力バッファ内のデータを読み出す必要があるので、第7図の212行でバッファ・データをA\$に読み出し、213行からラベルDispへ分岐してCRT画面上に表示したり、必要なサブルーチンに分岐して処理を施してのち、214行に戻り、再び待機ループに戻る。もし、接続開始の場合には、送受信のやり取りの間中、LOGON情報がM200に送信される。

受信が終われば当然M200は入力待ちの状態になっていると9845Tは考えてしまうのだが、実はM200が本当に入力待ちの状態になったのかどうかはっきりさせる手段は、現在のインタフェース手順で解決できにくい。YHP社のミニコンHP1000/3000シリーズの場合は、入力待ちのとき送信促進信号(PROMPT)を送ってくれるため、現在の手順でもわかる。M200のオンライン専用端末では、送信可能の時にCRT画面左下のX-SYSTEMの表示が消える。同期信号をやり取りしている有利さであろう。

非同期式でRS-232Cを使うユーザーには、YHP社以外の機種でもその特典が与えられない。当然、SDLCやHDLCのような高度なネットワーク手順が望まれようが、現在のM200には本格的な分散処理ネットワーク・システムが実現されていないので、次善の対策—非同期式の条件でも経験的方法で克服することにした。

受信したA\$を第7図の255行から258行までで画面表示する。先述したように第4図350行のように1行で済むと思われるかもしれないが、そのようにすると各文字はCRT画面の第1列目にしか表示されず、つまり連続表示指示記号が無視される。その結果、横に並ぶはずの1行文が、縦に並んでしまい、しかも画面から消えてしまう事態がおこった。経験的によりよい方法を見つけ出し、最終的に落ち着いたものが、対症療法的な255-258行である。

通信実験中に、たとえばM200から受信された全文字コードのASCIIをその番号で表示させてみた。この場合回線が切断することがあったので、LOGOFFしてメッセージを受けとる時にモニターしてみた。すると、普通の英数字・記号の他に特殊なASCIIコードも送られてくるのがわかった。そのため、255-258行の方法をとらないで、後述するように第20図261行の形式にすると、第21図のように重ねて出力されたり勝手にLF(ラインフィード)コードが作用してしまう。

第7図260行から269行までは、受信に伴う各種の処理をほどこす分岐先指定である。他のモードについてはあとで述べるが、必ず通過するのがラベルRcv-In(217

行)から始まるサブルーチンである。Rcv-Inから253行まではV1.0の時にはなく、V2.0であらたにつけ加えている。217行から226行までは、受信データをCRコードまでで区切り、V\$(0)に受信データ一行分として代入する。

通信実験をはじめめるまでは、A\$には必ず1行ずつ送られるものと憶測していたが、実際には中途に余計なコードをはさみつつ、CRコードの位置に無関係に文字群が送信されてきた。また、M200のINPUTモードでは、行番号のあとにLFコードが付加されるため、完全な1行と見なされないから、そこに送信データを付加する時にはよいものの、本来のLFコードの働きと別の意味を持つので、取扱いに慎重を期さねばならない。

V\$(0)の中身を検討することによって、M200からの受信データに応じたユーザー定義の仕事や9845Tにおこなわせることができる。たとえば、228行から234行までは、データ入力促進警報メッセージが受信された時、とりあえずCRコードのみを送って、ホストM200が通信回線への接続を断つのを回避させる。236行から247行までは、AUTO-RIPSモードに設定した時、ユーザーが作成した仕事をM200から受信されるメッセージに応じておこなうための識別ルーチンである。この部分は、はじめに第2図59行から78行までの初期設定でユーザーが定義すれば作動する。

ここで、識別ルーチンが実行された時にかかる時間が気になるかもしれない。BASICインタプリタの実行速度は概して遅いので、単なる識別ルーチンにのみあまり時間がかかるのは好ましくない。このルーチンが受信による割り込みルーチンの中にあるので、受信情報が大量にかつ高速に送られてきた時には、途中でV\$(0)の値が変わってしまうおそれがある。それを避けるためには、236行と237行の間か、又は264行と265行の間にDISABLEを入れ、265行と266行の間にENABLEを入れておけば、この間に割り込みが発生しない。ただし、CCOM、INBUFFER(第2表)の値をより大きくしておく必要がある。

実際に9845T上で実験してみると、160文字のRips\$(I)とS\$(I)を100組(I=1から100まで、Mxrips=100)定義するのに1秒もかからない。IF文を使って比較検討しても1秒を超えなかった。そこで1,000組で実験したところ、さすがに10秒近くかかっている。主記憶容量が足りないのでは、1,000組もできるのは実験の間だけである。識別ルーチンを通して分岐した先での作業が単なる送信にとどまらず時間のかかる作業である場合には、ENABLEにして割り込みが可能な状態にする方がよい。

地質調査所月報(第34巻 第12号)

```

272 Key: ! Key operation routines.
273 Key0: BS=""
274 INPUT "ENTER ASCII number for CHR$:",Chr
275 IF Chr<1 THEN Key0
276 Chr=INT(Chr)
277 BS=CHR$(Chr)
278 RETURN
279 Key1: BS=BS$(2)
280 GOSUB Sending
281 BS=""
282 RETURN
283 Key2: BS=""
284 DISP "EDIT LINE ON"
285 Edit=1
286 WAIT 400
287 INPUT "ENTER editing line?",$
288 WS=L "&BS&Cr$"
289 GOSUB Write
290 BS=BS&Cr$
291 RETURN
292 Key2end: IF POS(Y$,Cr$) THEN BS=Y$/1,POS(Y$,Cr$)-1/
293 FOR I=1 TO LEN(B$)
294 BS/I,1/=BS/I,1/
295 IF NUM(B$/I,1/)>256 THEN BS/I,1/-"
296 NEXT I
297 PRINT USING "#,K";CHR$(27)&"&a-lyOC"
298 Write=2
299 PS=BS&" "
300 WS=BS
301 GOSUB Write
302 PRINT USING "#,K";CHR$(27)&"&aOC",P$
303 PS=""
304 PRINT USING "#,K";CHR$(27)&"&a-ly"
305 ENABLE
306 DISP "PLEASE EDIT"
307 Edit=0
308 BEEP
309 Y$=""
310 BS=""
311 WAIT 200
312 BEEP
313 RETURN
314 Key3: BS=""
315 IF Graphics=1 THEN Key3end
316 DISP "GRAPHICS ON"
317 Graphics=1
318 PRINTER IS 0
319 WAIT 400
320 GOTO Graphics
321 Key3end: EXIT GRAPHICS
322 ENABLE
323 DISP "GRAPHICS OFF"
324 Graphics=0
325 WAIT 400
326 RETURN
327 Key4: BS=""
328 IF Rips=1 THEN Key4end
329 Rips=1
330 DISP "AUTO RIPS ON"
331 WAIT 400
332 RETURN
333 Key4end: Rips=0
334 DISP "AUTO RIPS OFF"
335 WAIT 500
336 RETURN
337 Key5: BS=""
338 FS(10)="TEST1.TEXT"
339 LINPUT "ENTER allocate file name;",$FS(10)
340 Track=10
341 INPUT "ENTER track size;",$Track
342 Addtrack=5
343 INPUT "ENTER add track size;",$Addtrack
344 PS="File="&FS(10)&"&"; Track="&VAL$(Track)&" (with "&VAL$(Addtrack)" )
345 GOSUB Print
346 GOSUB If
347 IF If$. "Y" THEN Key5
348 BS="ALLOC DS("&FS(10)&" ) SP("&VAL$(Track)&"&","&VAL$(Addtrack)&" ) T CA
UNIT(USER) REL"&Cr$
349 RETURN
350 Logon: BS=$$(1)
351 RETURN
352 Logoff: BS="LOGOFF"&Eof$
353 RETURN
354 Backspace: BS=BS$
355 PRINT USING "#,K";CHR$(27)&"&a-1C", " "
356 PRINT USING "#,K";CHR$(27)&"&a-2C",CHR$(Cursor)
357 RETURN
358 Hardcopy: BS=""
359 IF Hardcopy=0 THEN Hard 16
360 IF Hardcopy=16 THEN Hard 0
361 Hardcopy=16
362 RETURN
363 Hard 0: Hardcopy=0
364 DISP "Hardcopy ON"
365 WAIT 400
366 DISP
367 RETURN
368 Hard 16: Hardcopy=16
369 DISP "Hardcopy OFF"
370 WAIT 400
371 DISP
372 RETURN
373 Key10: BS=""
374 GOSUB Upload
375 RETURN
376 Key11: BS=""
377 GOSUB Record
378 RETURN
379 Key12: BS="END"&Eof$
380 RETURN
381 Expln: PRINTER IS 0
382 PRINT
383 PRINT " INSTRUCTION"
384 PRINT
385 PRINT " (Key definitions.)"
386 PRINT
387 FOR Ik=0 TO 31
388 IF Ik<10 THEN PRINT " KEY ";Ik;": ";K$(Ik)

```

```

389 IF Ik 9 THEN PRINT " KEY";Ik;": ";K$(Ik)
390 NEXT Ik
391 PRINT "End mark on tape:"&Eof$
392 IF I$="" THEN
393 PRINT " ----- TAPE CARTRIDGE -----
394 PRINT CHR$(248)&" :T14 for data transmitting(UPLOAD) / :T15 for
data receiving(RECORD) "&CHR$(240)
395 PRINT
396 PRINT " 6 66 6 66 6 6 66 6
397 PRINT
398 PRINT
399 PRINTER IS 16
400 BEEP
401 WAIT 120
402 BEEP
403 RETURN
404 List: BS="LISTC"&Eof$
405 RETURN
406 Break: CCONTROL Sc;BREAK
407 ENABLE
408 BEEP
409 WAIT 200
410 BEEP
411 BS=""
412 RETURN
413 Key16: BS=""
414 RETURN
415 Key17: BS=""
416 RETURN
417 Key18: BS=""
418 RETURN
419 Key19: BS=""
420 DUMP GRAPHICS
421 RETURN
422 Key20: BS=""
423 GCLEAR
424 RETURN
425 Key21: BS=""
426 GOTO Revmt
427 RETURN
428 Key22: BS=""
429 GOTO Readal
430 Cat: BS=""
431 ON ERROR GOSUB Cat_chg
432 If$="R"
433 INPUT "Tape index catalog:(Left(:T14)=L/ Right(:T15)=R)",If$
434 IF (If$="R") AND (If$="L") THEN Cat
435 IF If$="R" THEN CAT
436 IF If$="L" THEN CAT ":T14"
437 ON ERROR GOSUB Error
438 IF Hardcopy=16 THEN RETURN
439 PRINTER IS Hardcopy
440 IF If$="R" THEN CAT
441 IF If$="L" THEN CAT ":T15"
442 PRINTER IS 16
443 RETURN
444 Cat_chg: BEEP
445 PRINT " CONFIRM your tape cartridge"
446 GOTO Cat
447 Key24: BS=""
448 RETURN
449 Key25: BS=""
450 RETURN
451 Key26: BS=""
452 RETURN
453 Key27: BS=""
454 RETURN
455 Key28: BS=""
456 RETURN
457 Key29: BS=""
458 GOTO Backup
459 Key30: BS=""
460 GOSUB Create
461 RETURN
462 Key31: BS=""
463 GOSUB Chg
464 RETURN
465 Keyend: ! End of key operations.
466 !

```

第8図 特殊キーの定義

特殊キーによる分岐ルーチンの作業内容のソース・リスト。

6. ユーザー定義による特殊キー操作

第3図における特殊キー操作は、実際にどのように定義され実行されるのだろうか。第8図(A)(B)は各キーを押した時に分岐する先のサブルーチン群である。特殊キーの例として、第3図のK7を押すと、第6図の139行に行く分岐がおこり、KBD\$に入っているK7キーの内容がBSに移される。KBD\$に入っていたのは1コードのみではなく、第1文字目にはASCIIコードの255番が入り、第2文字目には7番が入っている。

そこで第6図の156行から166行が作動して、ラベル Tkey 1(164行)に分岐し、次いでラベル Logoff に分岐してのち167行にひき返して、ラベル Logoff での処理により BS を送信して待機ループに戻る。

ラベル Logoff は第8図の352行であり、KBD\$ のかわりに LOGOFF と CRコードが代入される。つまり、LOGOFF により、M200のオンライン TSS 処理を終了させたい時には、キーを7回(LOGOFF と RETURNキー)たたくのではなくK7キーを押すだけで用が足りる。これは一刻も早く回線のオンライン処理を終了させたい時に用いる。通信実験中は、しばしば TSS オンライン処理と無関係な操作を施すので CRT 表示やキーボードからの入力が異常になることがあった。その時手早くオンライン処理を終了させるのに、K7キーはきわめて便利であった。

定義していないキーが打たれたときは BS に ASCII の Null を入れて分岐から167行にひき返させる。すると167行が実行されて、何もせずに待機ループに戻るのである。

各特殊キーの役割を紹介しよう。

0 ASCII コードの送信

BS に代入して送信したい ASCIIコードがキーボード上にない時は、その文字の ASCII 番号を入力することで1文字づつ送信することができる。たとえばエスケープ・コードを送る時はまず K0キーを押して数字の27を入力し、CRコードを送信(CONT キー、STORE キー、EXECUTE キー、RUN キー、STEP キーのいずれか一つ)すると、M200は ASCII 27番を受け取る。

① 送信データの再送信

K1キーを押すことによって、Wb\$(2)に入っていた送信データが再び送信され、CRT 画面上に表示される。この時 CRコードが送られないため、M200側は入力待ちの状態のままである。その間に送信行を修正・追加して送信できる。

これは、いったん CRコード相当キー(CONT, STORE, EXECUTE, STEP, RUN)を押してしまったあとでは、Backspace による修正が即時実行できなくなる欠点を補うためにつけた。

Wb\$(I)の数を増やせば、その分だけ、それ以前の送信行にさかのぼらせることができる。9845Tでは、RECALL キーとして知られている。データ通信開設時に RECALL バッファが働かなくなるのでこの措置が必要になった。

② 受信データの再呼び出し

M200の TSS 会話中の、エディット状態の時に使う。

K2キーを押すと、求める行番号を聞いてくるので、必要な数字を入力する。その番号の内容が CR を含まずに受信されて表示される。これは、プログラム・リストなどの修正編集に使う。

③ グラフィックス

9845Tをグラフィックスのモードにするために使用する。又押しすれば抜ける。つまり ON/OFF キーである。

④ 自動応答

ユーザーが第2図の61行から78行で定義した自動応答モードに入りたい時に使う。特にユーザーが何も定義していなければ何の働きもしない ON/OFF キーである。

⑤ アロケーション

M200のユーザー・ファイルにデータセットを開設する時、いろいろな修飾子を付け加えて送信する必要がある。しかし、一般に必要なものはファイル名とそのサイズだけなので、あとは暫定値を採用してくれるのが望ましい。ところがM200では、全てユーザーが定義するまでは受容してくれない。このような相手には、こちらの計算機で勝手に暫定値を与えてやることのできる。

K5キーはその一つであり、まずデータセット名とその修飾子を会話式に設定し、次にそのサイズを会話式に決める。ユーザーが動かすのはここまでで、あとの必要な値は9845Tが適当に判断して送信する。こうすると、いちいちマニュアルを参照してはその複雑さに悩む必要がなくなる。この会話式指定と暫定値採用が可能になることは、インテリジェント端末ならではの芸当である。

⑥ LOGON

K6キーを押すことで、LOGON TSS ID/PASS WORD が送信され、一挙に READY 状態に入るパーソナルな仕様。専用オンライン端末ではスイッチを入れると直ちに LOGON 可能になるが LOGON 情報はキーボードから入力しなければいけない。異なる OS の流れるインテリジェント端末では、いったんターミナル・エミュレータを流すために各種情報がまとめてメモリーに入力されて使用に供される。専用オンライン端末同様に不特定多数が LOGON することもできるし、K6キーを使うパーソナルな自動入力もできる。“TSS 45”では、電源スイッチを入れたあとは自動的に LOGON 情報も送信される。

⑦ LOGOFF

前段で述べた。

⑧ Backspace

K8キーと、キーボード上の Backspace キー、及びカーソルの左移動キーを Backspace に使う。

⑨ プリンタ出力

K 9 キーを押すと感熱プリンターにも CRT 画面表示内容が出力される。ただし、K 9 キーを押したあとの表示内容しか表示されない。また押しすると OFF になる ON/OFF キーである。9845 T では、CRT 画面表示バッファの内容をプリンタ出力する機能が標準装備されていないので K 9 キーを押す前の情報をプリンタ出力することはできない。

⑩ テープ・データの送信

外部磁気記憶装置(標準はカセット磁気テープ装置)のデータを送信するには、まず K 10 キーを押す。ファイルと装置名を指定すると送信可能となる。そのまますぐ送信することもできるが、いったん TSS モードに戻ってから、必要な時にまた K 10 キーを押すことで送信開始することもできる。送信中に K 10 キーを押すと、送信を途中で解除することができる。

⑪ テープ・データへの受信

⑩ と逆で、全受信データのテープ上への書き込みは、K 11 キーを押すことで可能になる。受信中に K 11 キーをまた押すと強制解除になる。

⑫ END

K 12 キーを押すと、END が M200 に送信される。なお 379 行の BS= の右辺 “END” の中身を直すと、その文字列が送信される。ユーザーの好みに合わせればよい。

⑬ キーの説明

各キーの内容を知りたい時には K 13 キーを押して、プリンタ出力する。

⑭ カタログ・リスト

K 14 キーを押すと、M200 にカタログされているデータセットの表をみることができる。

⑮ Break 信号

K 15 キーと、! キーは、Break 信号を M200 に送るキーである。なお、テープデータ上の ! などは Break 信号とはみなされない。あくまでキー操作上の送信である。

⑯ 未定義

⑰ 未定義

⑱ 未定義

⑲ グラフィックスのプリント

K 19 キーにより、9845 T のグラフィックス・メモリーにバッファされている内容をプリンタ出力する。

⑳ グラフィックスのクリア

K 20 キーは、9845 T のグラフィックス・メモリーをクリアする。

㉑ テープ・データの修正

K 21 キーを押すことで、テープ・データ修正ルーチン

に分岐する。これは ON/OFF キーであり強制的に解除できる。

㉒ テープ・データの内容の表示

9845 T の外部記憶装置にあるファイルの内容を表示できる。

㉓ カタログ

9845 T の外部記憶装置のファイルのカタログを参照できる。

㉔ 未定義

㉕ 未定義

㉖ 未定義

㉗ 未定義

㉘ 未定義

㉙ 送受信内容のバック・アップ

バック・アップ用のルーチンに入る ON/OFF キーである。

㉚ データ・ファイルの初期作成

あらかじめ、データ・ファイルを初期化作成する時に使用する。

㉛ 記憶媒体の初期化

外部記憶装置上の媒体を使用可能な状態に初期化する。

以上、32種のキーは自在に定義して使用できる。特殊キーがこれだけでは足りないときは別のキーや記号を使う。!はその例である。カタカナを全部特殊キーに用いると、さらに50種を超える特殊キー操作ができる。これは第3図(B)で定義する。また、たとえばK 0 キーに代入する ASCIIコードの番号で定義しておくこともできるが、この場合は、キーを1回押すだけで自動操作できるわけではなくなる。

このように、ユーザーは“TSS 45”にあらかじめ好みの機能を付け加えて、次々に拡張していくことができる。なお、“TSS 45”で頻繁に使うユーティリティ・サブルーチンを第9図に示す。

7. 外部記憶装置を使うデータの送受信

外部記憶装置には標準装備のカセット磁気テープ(CMT)の他に、フレキシブル磁気ディスク(FD)や磁気ディスク装置などがある。ここでは標準のCMTを中心に解説する。なお、外部記憶装置の設定は、第2図の39行から44行にかけて初期設定してある。

7.1 データの転送

前節第3図のK 10キーを押すと PC (プログラム・カウンタ)は、ラベル Upload(第10図 567行)に分岐する。まず転送すべきファイル名を、次にその装置名を聞いて

ターミナル・エミュレータ“TSS45”と分散処理データ通信(古宇田亮一)

```

467 Utlty: ! Utility subroutines.
468 Error: PRINTER IS 0
469 PRINT ERRM$
470 PRINTER IS 16
471 ENABLE
472 ! GOSUB Break
473 ! GOSUB Sc4
474 RETURN
475 !
476 Write: CWRITE Sc;W$
477 IF Backup=1 THEN GOSUB Bu_snd
478 W$=""
479 RETURN
480 !
481 Print: PRINT P$
482 IF Hardcopy=16 THEN Print1
483 PRINTER IS 0
484 PRINT P$
485 PRINTER IS 16
486 Print1: P$=""
487 RETURN
488 !
489 Cr_lf: W$=CHR$(13)+CHR$(10)
490 GOSUB Write
491 RETURN
492 !
493 End1: CWRITE Sc;"END"&eof$
494 RETURN
495 !
496 Pause: B$=""
497 PRINTER IS 16
498 PRINT PAGE
499 DISP "PROGRAM IS PAUSING:Press CONT, then restart."
500 BEEP
501 PAUSE
502 DISP "PROGRAM RESTARTS"
503 BEEP
504 WAIT 120
505 BEEP
506 PRINTER IS Hardcopy
507 RETURN
508 !
509 If: If$="N"
510 BEEP
511 WAIT 130
512 BEEP
513 INPUT "Is this O.K.?(YES=Y)/(NO=N)/(END=END)",If$
514 RETURN
515 !

```

第9図 ユーティリティ

いくつかのユーティリティ分岐ルーチンを示す。

くるので、CRT 画面への入力例を見ながら正しく入力する。ここでは CMT と FD の装置を1つ選択できる。この他磁気ディスク・ドライブなどを付加する時には、第11図の526行から532行の内容にその旨付加する。

このように、ファイル名と機器の設定などは、マニュアルに書いてあることを一々覚えてやるよりも、会話式に CRT 画面の指示に従って入力するので、慣れている人の誤りも少なくできるし、初心者にもわかり易い。特殊キーを押すことは覚えなれないといけないが、特殊キー上には説明がついているので、一目瞭然といえる。

もし、まちがって入力した場合には、第10図の591行の割り込みが働いて第12図のサブルーチンに分岐する。ここでは、索引カタログを調べて、CMT を交換するか、初期化をおこなうか、又はファイル名と機器名を変更するかを選択する。初期化により、CMT 上のデータはすべて破壊されるので、慎重におこなわれるように会話形式を複雑にしている。

さて、第3図のK10キーを押して、いったんデータ転送モードに入ったとする。しかし、実は転送したくないか、あとで転送したいことがある。キャンセルするに

```

567 Upload: B$=""
568 IF Upload=0 THEN DISP "Upload ON"
569 IF Upload=1 THEN Up_end
570 IF Upload=-1 THEN Up_begin
571 Upload=-1
572 GOSUB Print
573 P$="" Upload file name/ default="&FS(2)
574 GOSUB Print
575 Up_file: FS(0)=FS(2)
576 INPUT "ENTER upload file name; Default=CONT; End=END; Line num
ber cut mode=LNC;"FS(0)
577 IF FS(0)="END" THEN Up_end
578 IF FS(0)="LNC" THEN Num_cut=1
579 IF FS(0)="LNC" THEN Up_file
if$="L"
580 GOSUB F dev
581 IF (if$. "L") AND (if$._"R") AND (if$._"F") THEN Up_file
582 PS(2)=FS(0)
583 PS=" Present file name(upload)="&FS(2)
584 GOSUB Print
585 GOSUB If
586 IF if$="END" THEN Up_end1
587 IF if$="Y" THEN Up_file
588 Up_dev: OFF ERROR
589 ON END #2 GOTO Up_end
590 ON ERROR GOSUB Chg
591 Chg=1
592 ASSIGN #2 TO FS(2)
593 IF Chg=0 THEN Up_asg
594 READ #2,1
595 X$=""
596 BEEP
597 PRINT "Upload-mode is READY."
598 WAIT 160
599 BEEP
600 Up_begin: If$="Y"
601 INPUT "Are you O.K. to upload? (YES=Y/ NO=N/ END=END)",If$
602 IF If$="Y" THEN Upload=1
603 IF If$="N" THEN Uploading
604 IF If$="N" THEN RETURN
605 IF If$="END" THEN Up_end
606 BEEP
607 GOTO Up_begin
608 Uploading: IF (NUM(AS).48) OR (NUM(AS)_58) THEN RETURN
609 IF LEN(X$) 0 THEN Up_cnt
610 ON ERROR GOTO Up_nmc
611 IF Datatype=1 THEN Up_readx
612 READ #2;X$
613 IF (NUM(X$).42) OR (NUM(X$)_58) THEN Up_data ! +,-./0123456789
614 IF Num_cut=0 THEN Up_data
615 P1=POS(X$, " ")
616 X$=X$/P1+1/
617 Up_data: Datatype=0
618 Up_cnt: B$=X$
619 IF X$. "" THEN BS=TRIMS(X$)
620 IF LEN(X$) 72 THEN BS=TRIMS(X$/1,72/)
621 GOSUB Sending
622 IF LEN(X$) 73 THEN Up_nxt
623 X$=X$/73/
624 RETURN
625 Up_nxt: X$=""
626 RETURN
627 Up_end: OFF END #2
628 ASSIGN #2 TO *
629 Upload=0
630 Num_cut=0
631 DISP "Upload OFF"
632 ENABLE
633 BEEP
634 WAIT 160
635 BEEP
636 X$=""
637 OFF ERROR
638 ON ERROR GOSUB Error
639 WAIT 700
640 RETURN
641 Up_nmc: IF ERRN=65 THEN Up_nml
642 IF (ERRN=59) OR (ERRN=60) OR (ERRN=84) OR (ERRN=87) THEN Up_end
643 IF ERRN=33 THEN X$=""
644 IF ERRN=33 THEN Up_data
645 GOTO Up_end
646 Up_nml: IF Datatype=1 THEN Up_read
647 Up_readx: READ #2;X$
648 Datatype=1
649 X$=VAL$(X$)
650 GOTO Up_cnt
651 !
652

```

第10図 ホスト M1200 へのデータ転送

外部記憶装置のデータをホスト計算機(M200)に転送するサブルーチンのソース・リスト。第3図のK10で起動する。

は、ファイル名に END を入力するだけでよい。この場合、END というファイルは採用されないので注意を要する。K10キーはデータ転送中の強制解除キーの役割もしているの、まちがった所で押すと、第9図のエラー・ルーチンに飛んでしまう。この場合は、キーボードから RETURN を入力して、EXECUTE キーを押せば回復

```

516 File: INPUT "ENTER file name; End=END; Default=OK;",&F$(0)
517 IF F$(0)="END" THEN RETURN
518 IF F$(0)="OK" THEN RETURN
519 P$=" Present file="&F$(0)
520 GOSUB Print
521 GOSUB F_dev
522 IF (If$="_") AND (If$_="R") AND (If$_="F") THEN File
523 P$="Present file & device="&F$(0)
524 GOSUB Print
525 RETURN
526 F_dev: INPUT "ENTER Left(:T14)=L; Right(:T15)=R; Floppy(:F$)=F;(File
error=E);",&If$
527 IF If$="R" THEN Device=Dev1
528 IF If$="L" THEN Device=Dev2
529 IF If$="F" THEN Device=Dev3
530 IF POS(F$(0),":") THEN F$(0)=F$(0)/1,POS(F$(0),":")-1/
531 F$(0)-F$(0)&B$(Device)
532 RETURN
533 !

```

第11図 ファイル指定

外部記憶装置上のファイル名とそのファイルが存在する装置名を指定する。

```

534 Chg: DISP "T14 "4CHRS(24$)&" CONFIRM T14(STORE)/T15(READ) tape drive,
then press CONT. "4CHRS(24$)&" T15"
535 PAUSE
536 OFF ERROR
537 ON ERROR GOTO Chg_err
538 Chg_cat: P$=" Present catalog"
539 GOSUB Print
540 CAT DS(Device)
541 OFF ERROR
542 ON ERROR GOTO Chg
Chg=0
543
544 Job$="R"
545 INPUT "ENTER job; CONT(no job) C(change tape), I(INITIALIZE)
-BE careful !",&Job$
546 IF Job$="R" THEN ON ERROR GOSUB Error
547 IF Job$="R" THEN RETURN
548 IF Job$="I" THEN Chg_ini
549 GOTO Chg
550 Chg_ini: IF=1
551 INPUT "This is INITIALIZE. Enter -1 and wait. If not initialize, p
ress CONT.",&If
552 IF If. -1 THEN Chg
553 DISP "INITIALIZING; Please wait."
554 @WRITE Sc:Eof$
555 INITIALIZE "-T14"
556 DISP
557 ON ERROR GOSUB Error
558 RETURN
559 Chg_err: BEEP
560 PRINTER IS 0
561 PRINT ERR$
562 PRINTER IS 16
563 DISP "ERROR OCCURS; "&ERR$&" Confirm and press CONT."
564 PAUSE
565 GOTO Chg_cat
566 !

```

第12図 ファイル確認と修正

外部記憶装置名やファイル名が合わない時には、このサブルーチンに分岐する。ここで磁気記憶媒体 (CMT/FD) の初期化もおこなえる。

する。

転送するデータが行番号付きの場合には、行番号を消して送りたいことがあるので、ファイル名入力の際に、あらかじめ LNC を入力する。これは対話式になされるが、LNC という名のファイルは使えない。これらのファイル設定が終わると、本当にそれでよいのかどうか、もう一度聞いてくる。第10図の 586行から 588行がそれで、第9図の509行から514行へ分岐する。

ここで YES か NO を答える。NO の場合は、また第10図の 575行に戻るから、入力まちがいをここでチェックする。データ転送にすぐ入ってあわてないように、Y 以外を入力すると、575行に戻るようになっていく。ただし、ENDを入力すると、転送モードから転放される。

これらの操作をしている間に何か受信されると、割り込み機能が働いて、5.の後半で記述した操作がおこな

```

653 Create: P$=" Create "&F$(0)&"/ default RECORDS="&VAL$(C(1))&"; BYTES="&VA
LS$(C(2))
654 GOSUB Print
655 Cr_n: INPUT "ENTER number of RECORDS="&C(1)
656 P$=" Number of records="&VAL$(C(1))
657 GOSUB Print
658 INPUT "ENTER number of BYTES="&C(2)
659 P$=" Number of bytes ="&VAL$(C(2))
660 GOSUB Print
661 If$="N"
662 INPUT "Create=Y; Error=N; End=END; Ready-made-tape=OK;",&If$
663 IF If$="END" THEN ON ERROR GOSUB Error
664 IF If$="END" THEN RETURN
665 IF If$="OK" THEN RETURN
666 IF If$="Y" THEN Cr_n
667 CREATE F$(0),C(1),C(2)
668 RETURN
669 !

```

第13図 データ・ファイルの初期化

必要なレコード数とレコード長を会話式に設定して、データ・ファイルの初期化をおこなう。

れる。

これで準備ができた。この状態を Upload=-1 としておく。601行から 608行までは、本当に転送を開始してよいかどうかを聞くルーチンである。すぐに転送しなければ、N を入力して待機ループに戻る。

今度は M200 側からながめてみよう。“TSS 45”の送信仕様に従うと、M200 には送信がすべてキーボードから入力が続いているだけに見えてしまう。CMT データなどの転送は、今のところ M200の編集モードの中で、磁気ディスク上にデータセットを作成する場合に有効である。直接磁気テープに転送することもできるがタイミングのとり方がむずかしい。モニターするには編集モード上でおこなった方がよい。

M200をデータ入力待ちの状態にしてから、ふたたび K10キーを押す。この場合は OFF にならず、第10図の601行から始まるルーチンに復帰する。転送可能か否かを聞いてくるので、キャンセルの場合には END を入力し、転送の場合には、Y を入力する。

データを転送し終わったら、自動的に転送モードが終了される。途中でやめたい時には K10キーを押せばよい。データは、9845 T 側で文字列であるか、数値データかに関係なく、すべて文字列として転送される。もし文字列が空だった時は、9845 T 側でエラーが発生する。この場合は何も送らないことにしてもよいが、あとで並べ換えをおこなうのに便利のように、ASCII の126番(記号~)を送っている。

M200は1行に72文字を超えるデータを受けとれない(行番号が組合わさるため)。これを考慮して、第10図では、610行と 613行から 625行にかけて、テープなどの磁気媒体の一行データが72文字を超える時は72文字ずつ切断して転送するように設定している。これなども、経験的な対策の一つである。

7.2 受信データの収納

第3図の KIIキーは、M200 から受信されたデータを

て、望みのファイルに接続する方法をとる。そのためには、第14図の716行以下に必要な修正を加える。

7.3 交信バック・アップ

上述したデータ収納の考え方を拡大すると、TSSで交信中の全記録を外部記憶装置にとっておくことができる。ソース・リストは第15図にある通りで、第14図に若干の修正を加えたものである。第3図のK29キーを押すことによりバック・アップのモード(Backup=1)に入る。これは、ON/OFF キーなので、強制的に解除できる。

ただし、解除したあと、又バック・アップに戻りたい時には、別のデータ・ファイルを作成するか、第15図の760行を削除しておく必要がある。そうでないと、前のデータ・ファイルの上に重複して新たな交信記録が書き込まれてしまい、前の記録が失われてしまうからである。

データ・ファイルはバック・アップ専用のアサイン・コード3番(#3)に接続される。第15図の739行から、749行までは、ファイル名と機器の設定を行なう。753行では、データ・テープの初期化をおこなう。756行から766行までで、アサイン・コード3番にファイルを開放して準備を完了する。

7.2のデータ収納モードとのちがいは、受信データのみでなく、送信データもバック・アップしてファイルに書き込むこと、受信データに送信データを加えて一行にする場合もできる点にある。送信データは、767行から773行へ、受信したデータは767行から773行へPCが動いて処理される。終了は、782行から784行までである。7.1のデータ転送モードと7.2のデータ収納モードが同時にかかっている時でも、バック・アップ・モードを実行することができる。

7.4 データのモニター

第16図のソース・リストは、もともと“TSS 45”とは無関係に開発したプログラムを切断して挿入したものである。データ形式の9845Tファイルのをのぞき見ることができる。挿入の手口は、配列宣言部分を除き、ラベルと変数に整合性を持たせて、“TSS 45”のサブルーチンとして順応するように切替える自動変換プログラムを用いた(この自動変換プログラムについては別稿で述べたい)。これにより、“TSS 45”の変数やラベルと矛盾なく、独立のプログラムが“TSS 45”内に付加できる。

第3図のK22キーを押すと、第16図798行から始まる“READAL”サブルーチンにPCが移動する。まずファイル名とその装置名を入力したのち、そのファイルの全データを読みとって、プリンタ出力する。9845TのBASICインタプリタでは、データのタイプが文字列型か数値型かで区別されており、混用するとエラーが発生す

る。この場合、通常PCは動かなくなるが、ここでは工夫して第16図の868行に分岐して、もし文字列型なのに数値データを読みってしまった場合には、さらに833行に分岐させて、数値型で改めてデータを読みとる。これらの経過と出力様式は、第16図の811行から825行にてプリンタ出力し、見てすぐわかるようにしてある。

モニターで使われるアサイン・コードは4番(#4)である。これも、データ転送・収納モードなどと共存できる。

7.5 ファイル・データの修正

第17図のソース・リストも、もともと“TSS 45”と無

```

798 Readal: ! READ ALL DATA PROGRAM "READAL"
799 Datatype=0
800 Device=Dev1
801 X$=""
802 PRINTER IS 16
803 ON ERROR GOTO Al_err
804 PRINT " PRINT ALL ROUTIN"
805 Al_file: F$(0)=F$(4)
806 IF$="L"
807 GOSUB File
808 IF F$(0)="END" THEN Al_end
809 IF F$(0)="OK" THEN F$(0)=F$(4)
810 F$(4)=F$(0)
811 PRINT " Readed file=";F$(4)
812 ON ERROR GOTO Al_err
813 ASSIGN #4 TO F$(4)
814 PRINTER IS 0
815 PRINT "#####"
816 PRINT
817 PRINT " PRINT ALL DATA of file=";F$(4)
818 PRINT
819 PRINT " ABRREVIATIONS"
820 PRINT "$ , : strings data."
821 PRINT " " numeric data."
822 PRINT " %CHR$(231)&" : change the data type."
823 PRINT
824 PRINT "-----"
825 PRINT
826 PRINT "FILE="&CHR$(132)&F$(4)&"
827 READ #4,1
828 Al_red: X$=""
829 READ #4;X$
830 PRINT X$ %CX$
831 IF X$ ="/END/" THEN Al_red
832 GOTO Al_cls
833 Al_nmc: X=0
834 READ #4;X
835 PRINT " ";X
836 GOTO Al_nmc
837 Al_cls: ASSIGN #4 TO *
838 PRINT "(END OF DATA)"
839 PRINT
840 PRINT "-----"
841 PRINT
842 PRINT "#####"
843 Al_end: PRINTER IS 16
844 PRINT " END of the data reading of file ";F$(4);""
845 BEEP
846 WAIT 160
847 BEEP
848 OFF ERROR
849 ON ERROR GOSUB Error
850 Datatype=0
851 RETURN
852 !
853 Al_err: IF (ERRN=10) OR (ERRN=32) THEN Al_chr
854 IF (ERRN=11) THEN Al_chg
855 IF (ERRN=59) OR (ERRN=60) OR (ERRN=84) OR (ERRN=87) THEN Al_cls
856 IF (ERRN=05) THEN Al_typ
857 BEEP
858 PRINTER IS 16
859 PRINT PAGE
860 CAT D$(Device)
861 GOTO Al_file
862 Al_chr: PRINT " %CHR$(231)&"
863 Datatype=0
864 GOTO Al_red
865 Al_chg: PRINT " %CHR$(231)&"
866 Datatype=1
867 GOTO Al_nmc
868 Al_typ: IF Datatype=0 THEN Al_chg
869 IF Datatype=1 THEN Al_chr
870 !

```

第16図 磁気媒体データのモニター

データ通信回線を開設中でも、外部磁気記憶装置上のデータ・ファイルの中身をモニターできる。

```

871 Revmt: F$(0)=F$(5)
872 If$="L"
873 GOSUB File
874 IF F$(0)="END" THEN Rv_end
875 IF F$(0)="OK" THEN F$(0)=F$(5)
876 F$(5)=F$(0)
877 IF NOT POS(F$(5),".") THEN BEEP
878 IF NOT POS(F$(5),".") THEN Revmt
879 P$=" Revise editing file name="&F$(5)
880 GOSUB Print
881 GOSUB If
882 IF If$="END" THEN Rv_end
883 IF If$.="" THEN Revmt
884 ON ERROR GOSUB Chg
885 Rv_asg: ASSIGN #5 TO F$(5)
886 READ #5,1
887 I=1
888 ON END #5 GOTO Rv_cls
889 Rv_red: READ #5,V$(I)
890 I=I+1
891 GOTO Rv_red
892 Rv_cls: OFF END #5
893 ASSIGN #5 TO *
894 Irev=I-1
895 PRINTER IS 0
896 PRINT "Maximum number=";Irev
897 PRINTER IS 16
898 ON ERROR GOSUB Error
899 Rv_wrk: W$="
900 LINPUT "ENTER Print=Pn,m; Edit=En; End=END",W$
901 IF W$="END" THEN Rv_new
902 IF W$/1,1="P" THEN GOSUB Rv_prt
903 IF W$/1,1="E" THEN GOSUB Rv_edt
904 BEEP
905 GOTO Rv_wrk
906 !
907 Rv_prt: P1=POS(W$,",")
908 IF P1=0 THEN RETURN
909 ON ERROR GOTO Rv_wrk
910 Wm=VAL(TRIM$(W$/Z,P1-1))
911 IF P1+1 LEN(W$) THEN Rv_prt1
912 Wm=VAL(TRIM$(W$/P1+1))
913 GOTO Rv_prt2
914 Rv_prt1:Wm=Wm
915 Rv_prt2: IF Wm Irev THEN Wm=Irev
916 ON ERROR GOSUB Error
917 IF (Wm.0) OR (Wm.0) THEN RETURN
918 IF Wm Wm THEN Step=-1
919 PRINTER IS Hardcopy
920 FOR I=Wm TO Wm STEP Step
921 PRINT I;": ";V$(I)
922 NEXT I
923 Step=1
924 PRINTER IS 16
925 RETURN
926 !
927 Rv6edt: IF W$/2/=" THEN RETURN
928 ON ERROR GOTO Rv_wrk
929 I=VAL(TRIM$(W$/2/))
930 ON ERROR GOSUB Error
931 IF I Irev THEN RETURN
932 IF I=0 THEN RETURN
933 Rv_edt1:PRINT PAGE
934 EDIT "EDIT /If delete, enter DELETE or NULL;";V$(I)
935 PRINT V$(I)
936 GOSUB If
937 IF If$.="" THEN BEEP
938 IF If$. "Y" THEN Rv_edt
939 RETURN
940 !
941 Rv_new: W=0
942 FOR I=1 TO Irev
943 V(I)=0
944 IF (V$(I)="" ) OR (V$(I)="DELETE") OR (V$(I)="NULL") THEN V$(I)="_"
945 IF V$(I)="_" THEN V(I)=I
946 IF V$(I)="_" THEN W=W+1
947 NEXT I
948 Inew=Irev-W
949 C(1)=Inew
950 C(2)=30
951 F$(0)=F$(5)
952 GOSUB File
953 IF F$(0)="END" THEN BEEP
954 IF (F$(0)="OK") OR (F$(0)="END") THEN F$(0)=F$(5)
955 F$(5)=F$(0)
956 ON ERROR GOSUB Chg
957 GOSUB Create
958 IF If$="END" THEN Rv_end
959 ASSIGN #5 TO F$(5)
960 READ #5,1
961 FOR I=1 TO Irev
962 IF V(I) 0 THEN Rv_nxt
963 PRINT #5;V$(I)
964 Rv_nxt: NEXT I
965 ASSIGN #5 TO *
966 Rv_end: PRINT PAGE
967 DEFP "Revise MT OFF"
968 ON ERROR GOSUB Error
969 BEEP
970 WAIT 160
971 BEEP
972 P$=" Session ended."
973 GOSUB Print
974 RETURN
975 !
976 Link: !

```

第17図 磁気媒体データの修正

データ通信回線を開設中でも、外部磁気記憶装置上のデータ・ファイルの中身を修正できる。

関係に開発してあったものを挿入したものである。第3図のK21キーによって起動する。9845Tのファイル・データをモニターして必要な修正を施し、新たにデータ・ファイルを自動的に作成する。これに要する配列容量が大きすぎるので、特に使わないときには、第17図の始めで、

```
871 Revmt: RETURN
```

とし、872行以下976行までを削除してしまえばよい。この時、配列宣言のVとV\$の次元をなくすか、要素数を1にする。

このプログラムでは、モニターすべきファイルをアサイン・コード5番(#5)に開放する。そして、ファイルの中のデータをすべて読み出してV\$(I)の配列に入れる(第17図889行から891行)。ここで、Iの最大値は、9845Tの都合で800までしかとれない。これ以上多くすると、9845Tの残存容量が少なくなりすぎるからである。

V\$(I)に入ったデータを加工するため第17図の900行でメニュー選択を行なう。モニターすべき順番番号が、何番(n)から何番(m)までかを、Pn, mの仕様で入力すると、RV-prt(907行)に分岐する。ここで、nからmまでのV\$(I)の内容をプリンタ出力又はCRT画面表示する。

次に修正すべき番号(n)をEnの仕様で入力すると、Rv-edt(927行)に分岐する。第17図の934行に必要な修正を加える。もし削除したい時は、空白文字にするか、DELETEを代入する。空白文字は削除ではなく、空白として認識される。

修正作業がことごとく済んだら、ENDを入力すると、ラベルRv-new(941行)に分岐して、新たなファイルを作成する。ここでは、ファイル名を入力すれば、初期化からデータ・ファイルの作成まで自動的に進む(953行から964行まで)。

こうして、アサイン・コード5番を閉じれば、この作業は終了して、再び待機ループに戻る。修正したファイルは、もっぱら7.1の手順に使われたり、7.2の保存用ファイルに使われることになるだろう。

7.6 非同期式 RJE

インテリジェント端末の性質を上手に使うことで、電源スイッチのON/OFFのようなごく一部の手順を除いて、RJE(Remote Job Entry)を自動的に進めることができる。“TSS 45”を使った、一つの拡張問題としてこれをさぐってみよう。

第18図は、M200のTSS編集モードでFORTRANプログラムを作成し、実行(コンパイル、オブジェクト・モジュールとロード・モジュールの作成を含む)して結

```

00010 DIMENSION A(10)
00020 A(1)=1
00030 DO 1 I=2,10
00040 A(I)=A(I-1)*9
00050 WRITE(6,2) A(I)
00060 1 CONTINUE
00070 2 FORMAT(F15.0)
00080 STOP
00090 END
END OF DATA SET

R
GE COMPILER ENTERED
END OF COMPILATION
  9.
  81.
  729.
  6561.
  59049.
  531441.
  4782969.
  43046720.
  387420416.
END OF GO,SEVERITY CODE= 0
E

END

READY
    
```

第18図 テスト用 FORTRAN プログラム

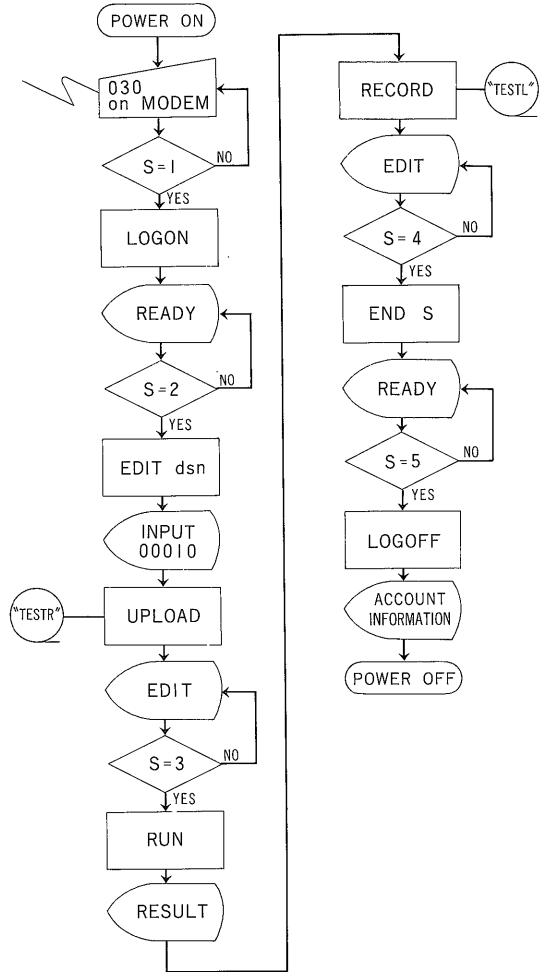
本文7.6 で使用する FORTRAN テスト・プログラム，“TSS45” で M200 と交信中に作成し，実行してバグのないことを確認した手順。第3図の K9 キーを押してプリンタ出力してみたもの。これと同じ手順を自動的に実行することが本文7.6 の内容となる。なお，このプログラムは，オフラインで使用した 9845T のデータ・ファイル上にも作成されている。

果を得るまでの簡単なプログラムとその流れである。9 の1乗から9乗までを計算して出力するだけである。第18図は TSS の編集モードで作業しているので，そのコマンドも合わせて表示されている。

[課題]：第18図の上半のプログラム(デバッグ済)を 9845T 上で作成し，M200 に自動転送して自動的にバッチ処理を実行させ，自動的に結果を9845T のデータ・ファイルに収納して終了させるように，“TSS 45” を変更する。

ここで，FORTRAN のプログラムは，9845T 上のデータ・ファイル “TESTR” に入力し，M200 上では，データセット “TEST. FORT” に転送することにする。結果は 9845T 上のデータ・ファイル “TESTL” に収納することにして，まず “TESTR” と “TESTL” を作成しておく。9845T 上でのこの操作はオフラインでできる。

RJE 手順の進行具合を計数 S の増大であらわすことにする。必要な作業手順のフロー・チャートを示せば，



第19図 非同期式 RJE の作業手順例

本文7.6 のフローチャート。判定を示す菱形は，すぐ上の表示が計数 S の示す内容に一致するまで，表示を待つことを示す。

第19図のようになる。これは，非同期式ターミナル・エミュレータ “TSS 45” を使う TSS 環境下に行う RJE なので，非同期式 RJE と呼ぼう。第19図は，双方のプログラムが微妙に交錯するので，理解の便のために作った。

実は，第3図の K4 キーを押すことによって，この非同期式 RJE のサブ・ルーチンを単独に作ってしまう方が簡単であるが，これはすでに本節の各記述で示してきた方法なので，送受信の割り込みの理解を深めるため，別の方法を採用してみよう。第20図はその修正全リストであり，行のうしろに！***で示したところがそうである。(A)は第2図と，(B)は第7図と比較されたい。

9845T 上でのキー操作をおこなわないことを自動化と


```
(A)
58      !
59      ! User definition of RIPS message for AUTO-RIPS operation.
60      Mxrips=2          ! Maximum number of operation.
61      Rips#(1)="KE054100I"      ! Rips#(1):RIPS message.
62      Rips#(2)="READY"          !***
63      S*(2)="EDIT TEST NEW FORT" !***
64      Rips#(3)="EDIT"          !***
65      S*(3)="RUN"              !***
66      Rips#(4)="EDIT"          !***
67      S*(4)="END S"            !***
68      Rips#(5)="READY"        !***
69      S*(5)="LOGOFF"          !***
70      Rips#(6)="^"
71      S*(6)="^"
72      Rips#(7)="^"
73      S*(7)="^"
74      Rips#(8)="^"
75      S*(8)="^"
76      Rips#(9)="^"
77      S*(9)="^"
78      Rips#(10)="^"
79      S*(10)="^"
80      !

(B)
232     !
233 Rcv_in: IF LEN(Y#)+LEN(A#)>159 THEN A#=Cr#
234     Y#=Y#&A#
235     IF NOT POS(A#,Cr#) AND NOT POS(A#,Lf#) THEN RETURN !***
236     V#(0)=Y#
237     IF Edit=1 THEN GOSUB Key2end
238     IF Graphics=1 THEN GOSUB Plot
239     Y#=""
240     GOSUB Rcv_op
241     IF S>6 THEN RETURN !***
242     IF Rips>=0 THEN Rcv_user !***
243     RETURN
244     !
245 Rcv_op: IF NOT POS(V#(0),Rips#(0)) THEN RETURN
246     B#=Cr#
247     GOSUB Sending
248     P#="DATA ENTRY DELAYED"
249     BEEP
250     GOSUB Print
251     RETURN
252     !
253 Rcv_user: IF POS(V#(0),Rips#(Rips)) THEN Rcv_uop !***
254     IF S>5 THEN RETURN !***
255     Rips=Rips+1
256     IF Rips<=Mxrips THEN Rcv_user
257     Rips=S !***
258     RETURN
259 Rcv_uop: IF Rips=1 THEN Rcv_pas
260     PRINTER IS 0 !***
261     PRINT "***** Receive=";V#(0);" *****" !***
262     PRINT "***** Rips#(Rips);" );Rips#(Rips);" *****" !***
263     PRINTER IS 16 !***
264     WAIT 2000 !***
265     IF S>5 THEN RETURN !***
266     IF S=2 THEN ON END #2 GOTO Up_end !***
267     IF S=2 THEN ASSIGN #2 TO F#(2) !***
268     IF S=3 THEN GOSUB Rc_asg !***
269     B#=$#(Rips)&Cr#
270     GOSUB Sending
271     IF S=3 THEN Record=1 !***
272     IF S=4 THEN Record=Upload=0 !***
273     IF S=4 THEN GOSUB Rc_end !***
274     IF S=2 THEN Upload=1 !***
275     IF S=2 THEN Mxrips=5 !***
276     PRINTER IS 0
277     PRINT "***** S=";S;" ";S#(S);" *****" !***
278     PRINTER IS 16
279     S=S+1 !***
280     Rips=S
281     RETURN
282     !
```

第20図 RJE のための付加リスト

(A)は M200 との応答用リストである。(B)は、受信データの表示ルーチンに、RJE 手順進行に伴って次々にデータが送受信されるための手順リスト。!***が新たに付加された部分。

しているので、Rips は 1 以上になる。つまり送信促進警報メッセージには応答しない。待つ必要がないからである。第 3 図の K 4 キーを押す操作もいらない。S の進行に伴うメッセージは第20図(A)にある通りである。

第19図の流れを走らせるだけでは、第18図と同じ出力結果しか得られない。通常業務ではその方が好ましいが、ここでは第20図の 260行から 263行と、276行から278行により、プログラムの進行をプリンタに出力させるこ

地質調査所月報(第34巻 第12号)

```

G0445 LOGON IN PROGRESS AT 09:58:59 ON AUGUST 3, 1982
### RIPS CENTER FACOM M200 TSS SERVICE ###
READY
***** Receive=
*****      READY
***** Rips$( 2 )=READY *****
E TESTX NEW FORT
***** S= 2 E TESTX NEW FORT *****

INPUT
00010 DIMENSION A(10)
00020 A(1)=1
00030 DO 1 I=2,10
00040 A(I)=A(I-1)*9
00050 WRITE(6,2) A(I)
00060 1 CONTINUE
00070 2 FORMAT(F15.0)
00080 STOP
00090 END
00100
E
***** Receive=
          00010
                00020
                        00030
                                00040
                                        00050
                                                00060
                                                        00070
                                                                00080
                                                                        00090
0
00100
*****E***
***** Rips$( 3 )=E *****
R
***** S= 3 R *****

GE#COMPILE#ERRE#ERRE#
*
*      GE COMPILER ENTEGE COMPILER ENTERED
*****
***** Rips$( 5 )=E *****
END S
***** S= 4 END *****
END#OF#COMPILE#DI#ION#
*****      END OF COMPILATION
***** Rips$( 5 )=E *****
END S
          9.
         81.
        729.
       6561.
      59049.
     531441.
    4782969.
   43046720.

***** S= 5 END S *****
387420416.
END OF GO,SEVERITY CODE= 0
E
END

NOTHING SAVED YET
ENTER SAVE OR END-

***** Receive=
*****      READY
***** Rips$( 6 )=READY *****
***** S= 6 LOGOFF *****

```

第21図 テスト結果(1)

LFコードの処理を特に考えていなかった時の処理結果。第20図(B)の260-263行と、276-278行がなければ、第18図と同じ結果が得られる。

とにする。これらを、RJE 進行モニター行と称しよう。RJE 進行モニター行は、RJE 進行係数Sの表示と、受信データの判断作業が実行されていることを表示する。

はじめに、実験的な実行をおこなった結果が第21図である。これでも一応動いているが、いくつかおかしな点

がある。特に、編集モードの入力において、行番号のあとにはLEコードが続いている。これでは、多量のプログラムを転送した時に、V\$(0)の内容が正確に反映されなくなる。これを避けるため、第20図の235行ではCRコードと共にLFコードも同じ扱いを受けるように変更

ターミナル・エミュレータ“TSS45”と分散処理データ通信（古宇田亮一）

```
G0445 LOGON IN PROGRESS AT 11:33:02 ON AUGUST 3, 1982
### RIPS CENTER FACOM M200 TSS SERVICE ###
READY
*****Receive=READY
***** Rips$( 2 )=READY *****
EDIT TEST NEW FORT
***** S= 2 EDIT TEST NEW FORT *****

INPUT
00010 DIMENSION A(10)
00020 A(1)=1
00030 DO 1 I=2,10
00040 A(I)=A(I-1)*9
00050 WRITE(6,2) A(I)
00060 1 CONTINUE
00070 2 FORMAT(F15.0)
00080 STOP
00090 END
00100
EDIT
*****Receive=EDIT
***** Rips$( 3 )=EDIT *****
RUN
***** S= 3 RUN *****

GE COMPILER ENTERED
END OF COMPILATION
      9.
      81.
      729.
      6561.
      59049.
      531441.
      4782969.
      43046720.
      387420416.
END OF GO,SEVERITY CODE= 0
EDIT
*****Receive=EDIT
***** Rips$( 4 )=EDIT *****
END S

SAVED IN DATA SET 'G0445.TEST.FORT'
READY
***** Receive=READY

*****
***** Rips$( 5 )=READY *****
***** LOGOFF ***** S= 4 END S *****

***** RIPS TSS ACCOUNT INFORMATION 82.08.03 11:33:57 *****
***** S
= 5 LOGOFF *****
* USER-ID ( G0445 ) REGION-SIZE ( 768 KB ) *
* CPU-TIME ( .44 S) SESSION-TIME ( 00:00:56 ) *
* TPUT-COUNT ( 18 ) TGET-COUNT ( 14 ) *
* EXECUTION CHARGE ..... 60 YEN *
*-----*
* BUDGET (GROUP) | CALCULATION-CHARGE | UTILIZATION-CHARGE *
* USED (GROUP) | 3,000,000 YEN | 200,000 YEN *
* BUDGET (INDIVIDUAL) | 727,984 YEN | 23,229 YEN *
* USED (INDIVIDUAL) | 3,000,000 YEN | 200,000 YEN *
* USED (INDIVIDUAL) | 94,751 YEN | 223 YEN *
*****
G0445 LOGGED OFF AT 11:33:59 ON AUGUST 3, 1982+
```

第22図 テスト結果(2)

第20図(B)の235行のように修正した時、多少見栄えがよくなる。第20図(B)の260-263行のように通常は使わない出力手順を加えることで、受信内容の複雑さを解析できる。

して、再び実行したものが、第22図である。

割り込みが予想よりも少し遅れて作動することがあるので、たとえば第22図の4行目は、第20図(B)の261行が反映されていない。9845Tのみで使用している時にはこの現象はあらわれないが、M200と交信している時には、少し遅れたタイミングでCRコードを送ってくるのでそれが働いて重ね打ちが生じる。

第21図と第22図の実験結果は、M200から送られてく

る信号の中には、9845Tで不要な多くの画面操作信号を含むことをおしえてくれる。これは、マニュアルには出していない。

この通信実験のもとになった“TESTR”の内容を“TSS 45”のREADALルーチンで読み出したのが第23図(A)に、又、受信されたM200による計算結果を収納した“TESTL”が第26図(B)に示されている。この(B)には途中の余計な受信データも入っているが、“TSS 45”の

KEQ54100I SESSION ENDED

(A)

PRINT ALL DATA of file=TESTR:T15

```

ABBEVIATIONS
$      : strings data.
      : numeric data.
|---  : change the data type.
    
```

```

-----
FILE='TESTR:T15'
$ DIMENSION A(10)
$ A(1)=1
$ DO 1 I=2,10
$ A(I)=A(I-1)*9
$ WRITE(6,2) A(I)
$ 1 CONTINUE
$ 2 FORMAT(F15.0)
$ STOP
$ END
<END OF DATA>
    
```

(B)

PRINT ALL DATA of file=TESTL:T15

```

ABBEVIATIONS
$      : strings data.
      : numeric data.
|---  : change the data type.
    
```

```

-----
FILE='TESTL:T15'
$ RED
$ END OF COMPILATION
$      9.
$      81.
$      729.
$      6561.
$      59049.
$      531441.
$      4782969.
$      43046720.
$      387420416.
$ END OF GO,SEVERITY CODE= 0
$ SAVE OR END-
<END OF DATA>
    
```

第23図 テスト・データ・ファイルの中身

送信用“TESTR”と受信用“TESTL”の中身を、それぞれ交信中に第3図 K22 キーを押してモニターした。(B)は、第22図と比較すると、タイミングの実際のとられ方がわかる。なお、何度実験しても(B)の結果しか得られていない。

REVMT ルーチンで改良できる(7.5)。

第20図(B)の264行では、受信から送信まで2秒の間隔をあけている。この行を欠いた実験では、送信データが正しくM200に拾われなかったことがあった。おそらくタイミングの問題、入力待ちに至るまでのタイム・ギャップの問題であろう。

先の実験で、1.5秒置いたことを考え、それより長い2秒にしてみたところ、うまくいった(1秒でも問題なかった)。これは、考えてみると少しおかし。M200が

信号を送ってくるのは、そこまではM200側も作業を完了しているはずなので、9845Tから直ちに次の作業指示を送れば、受けつけなければならない。ところが、少し待たねばならないのは、2つの理由がある。

たとえば、第1図のポーリングにかかる問題が夾在して、端末からの信号をM200がとり落とすことが考えられる。

もう一つは、M200がまだ処理していないのに先取りして次のメッセージを送ってしまう可能性である。これ

らは、M200自身をモニターしなければならないので、分散処理のため前からはみ出してくる。ただし、このようなタイミングの問題が生じて、インテリジェント端末なら、WAIT 命令を置くことによって経験的に切り抜けることが可能である。

第22図の下段の料金情報によると、この RJE にかかった費用は、わずか60円にすぎない。通常の TSS 処理にかかる費用からみれば、自動化によって、キーボード操作などに関わる労力と費用が節約される。なお、CPU 処理時間は0.44秒であった。

8. 考察とまとめ

① 分散処理ネットワークには、星形・ループ形・リング形・バス形などがあるが、工業技術院筑波研究センターの FACOM M200は、このうちのリング形に相当する。しかし、ユーザーから見た場合には、オンライン端末用ネットワークの機能については良く支援してくれるものの、ひとたび分散処理に使用しようとする時、必ずしも十分な対応がなされるものではなく、未知の部分が多数ある。小論で報告したターミナル・エミュレータは、テレタイプ型オンライン端末向きに作成したプロトコル・送受信処理用の小規模ルーチンを基礎にしている。つまり、ホスト計算機M200から見れば、オンライン端末のひとつにすぎないが、端末側の 9845T から見ると、その処理機能の一部に M200 を参加させているという特徴もっている。この意味での分散処理端末であり、きたるべき本格的な分散処理ネットワークを先取りするために作成された。現在のネットワークに合わせた通信実験をおこなうために、自作のターミナル・エミュレータにする必要があった。

② ネットワークを利用した場合は、ハードウェアとソフトウェアの共同利用化が促進され、その利用効率が高まる。地質分野では十分なソフトウェアが整備されているとはいいがたいが、特にユーザーと計算機管理者との間のギャップが大きいので、集中処理だけでは、需要をまかないきれない。ユーザー側に立つソフトウェアをユーザーが開発する必要はあるが、それがネットワークに結合される時、効果が相当期待できる。

③ シティ銀行の事例を検討すると、各計算機が独立対等であるような分散処理が理想的である。組織全体にとっては、オンライン完全自動処理に伴う技術的困難さよりも、通常はオフラインで使われ、必要に応じて通信回線で持続される方が効率的である。しかし、汎用データベースやネットワークの制御、各計算機のプロトコル・インタフェースなどをどうすべきかで、解決しなけれ

ばいけない問題が多い。既に実用化されてその有効性の証明されたネットワークもある。しかし、完全な分散処理に移行する前に、ホスト計算機のもつオンライン通信回線を利用した実験的な分散処理で問題点を抽出すると共に、実用化を先取りすることには意味がある。

④ オンライン回線に持続する一つの標準として、RS-232C シリアル・インタフェースがある。9845Tは、これを使って M200 に接続された。9845T 自身も、もともとデータ通信向きには作られていない。そこで、通信実験による処理に入る前に、オンライン端末として働くプログラムも作成する必要があった。通常のデータ通信機能を内蔵したマイコンなら、この程度のファームウェアを持っているのが普通なので、この手間はいらぬ。しかし、端末がオフラインで動くようにオンライン処理中も動かせるようになるかどうかは別である。9845T では、通信機能が割り込み処理として働くようになっていたため、オフラインで動く時とほぼ同様の仕事がこなせる。そのため、TSS オンライン状態で各種の通信実験ができる。

⑤ 9845T の拡張 BASIC インタプリタは、ファイル管理能力がすぐれているが、十分に構造化できるまでには至っていない。しかしラベル分岐機能を使用して、フローチャートに頼らずに書き下す形で、機能が独立したプログラムに作成することができた。

もし多重処理可能なミニコンなら、さらに上手に問題を解決することができる。つまり、各プログラムは独立させることができ、M200との通信のやりとりを管理するプログラムと、本来のミニコンの機能を使った業務が常に同時並行的になされる。この場合にプログラム相互の割り込み処理が自在に扱えるかどうかは鍵となる。

ターミナル・エミュレータ“TSS 45”では、各プログラムがサブルーチンとして1つのプログラム・ファイル中に結合されているので、割り込み処理は論理的に行なえた。そして、いったん通信回線を開くと、常時待機ループにプログラム・カウンタを走らせながら、必要が生じた時のみ、割り込み処理に必要なサブルーチンに分岐する。

これがマルチジョブで可能なら、割り込みが加わらない限り、オフライン仕様で使用できるし、ファイルが各ジョブで分割使用できるなら、オンラインとオフラインが同居できることになるだろう。9845Tは、マルチジョブ機能を持たないので、これは無理にせよ、パーソナルな使用では、オンラインもオフラインも自在に扱える。

⑥ TTY 手順が内蔵されている類のマイコンでは、オンライン端末としては申し分ないけれども、マイコン

独自のオペレーティング・システムがどの程度有効に活用できるかどうかは、どのようなターミナル・エミュレータを作成するかによるだろう。

“TSS 45”で用いた送受信処理の考え方は、計算機の機種が異なれば内容変更しなければいけない。しかし、割り込み構造は転用して一般化できる。マイコンをインテリジェント端末として使うとき、自身がオペレーティング・システムを持ち、又、周辺装置を有しているので、単なるオンライン TSS 端末や RJE 端末としてだけで使うのは効率が低い。インテリジェント端末自身も負荷分散をはかって、データ通信処理業務の費用対効果を高めなければいけない。M200 と 9845 T の双方が独自の処理をおこないつつ、データ通信回線を通じた分散処理をすすめることが、インテリジェント端末化の目的である。

⑦ “TSS 45”では、豊富な特殊キー操作の他に、ユーザー定義による特殊キー操作も加えられるように作成されている。M200 のオペレーティング・システムは、大きな意味では会話式であるが、ファイル(データセット)の定義やその管理、各種の命令セットには細かい点で会話式になっていないところがある。また暫定値の採用も全面的におこなっているわけではない。

このような点を、ユーザー側から穴埋めするには“TSS 45”が有利である。会話式でない部分や暫定値設定などは、9845 T を使う時と同じ形で入力できる。これは、マイコンのユーザーにとって、異なるオペレーティング・システムを持つ M200 のような異なる計算機を動かす時、マイコンで習い憶えた方法でなめらかに入門できる利点がある。ネットワークを構成する時、もしローカル・ネットワークにこのような機能が付いているなら、ユーザーは異なる計算機であることをあまり意識しないで使えるかもしれない。

⑧ 9845 T の外部記憶装置のデータ通信への参加について、“TSS 45”では、ファイルデータの転送と収納、交信のバックアップ、ファイルデータのモニターと修正の機能を持っている。必要に応じて、さらに機能を付加することもできる。

これらの応用として、非同期-TSS 環境における自動 RJE を試行した。これは Edit モードで行なったため、どうしても速度が遅くなる難点はあるが、通常の手操作に比べれば圧倒的な早さを持つので、ルーチンに使うには便利であろう。

9845 T の主記憶容量が限られていて、しかもマルチジョブ機能を持たないので、プログラム・ライン数の多い場合は、プログラム・ファイルを外部記憶装置の助けを

かりて分割して入力・実行しないと行けない。多人数で同時に使うわけではないから、この問題は少ないだろう。ただし、ローカル・ネットワークを組む時には問題になる。

膨大で時間のかかる計算処理や、M200 のもつユーティリティを利用したい時などに“TSS 45”を介在させるなら、9845 T の本来持つ機能以上のものが通信回線を通して得られる。また、入力処理などに M200 のオンライン TSS 機能を直接使って待ち時間を費すよりも、“TSS 45”を介在させて、必要な入力はあらかじめすませておいて、あとで TSS を利用するなら、M200 の負荷も軽くなるであろう。簡単な計算処理は、各端末レベルで解決すべきである。

YHP 社が供給している IBM2780/3780 仕様の RJE ターミナル・エミュレータの場合には、より高速(2倍から4倍)でデータを送信できる。しかし、会話形で RJE をするのはなく、旧来のパッチ処理に相当しているのみなので、会話形に用いる場合には“TSS 45”の考え方が有利である。7.6 で試みた方法は、M200 と 9845 T のデータ通信をテストすることに主眼があったが、会話形で処理するのではない RJE には、いっそ徹底的に自動化した方がよいのではないかと提起でもある。

通信実験中に問題になった送受信のタイミングの問題は、方式が非同期式のためでもある。同期信号を常時とっているわけではないため時間的ロスをもたらす上、1,200 bps 以上に高速化するには、ハードウェア上の桎梏がある。フルスクリーン編集には、9,600 bps 程度に速度をあげ、TTY よりも、同期をとる BSC 手順や、さらに進んで SDLC, HDLC のような手順が望ましい。その場合でも“TSS 45”のようにユーザー指向の高いターミナル・エミュレータの思想は生かされるだろう。

最後の問題は、ANSI 拡張 BASIC が依然としてハードウェア手順を反映した言語(データ/プロシージャ・オリエンティド)であることによる制約である。インタプリタ型のため実行速度が遅い点もある。今後はオブジェクト・オリエンティド言語上で作成していくことが必要であろう。

“TSS 45”は、あくまで端末からみた分散処理にすぎない。自身が通信回線網の制御に同等に参加し、M200 にかかる負担を積極的に分担できるものではない。そこまで分散処理化を進め、ネットワークが持つ有効性を発揮できるようになって、はじめて真の分散処理と呼べるにちがいない。

謝辞

データ通信回線への接続について富士通㈱社の RIPS 担当係の方々から、初期の御便宜を得た。横河ヒューレット・パッカート㈱社から、マニュアル図の一部の転載許可をいただいた。

FACOM-M200に接続するデータ通信回線の問題点とターミナル・エミュレータの現状について、計量研究所・豊田幸司氏、機械技術研究所・筒井康賢氏、電子技術総合研究所・田中隆氏から有益な御指摘をいただいた。リモート・ステーションとデータ通信回線について、また、FACOM-M200上で動くソフトウェアについては、地質調査所物理探査部・中塚正氏に有益な御指導をいただいた。YHP 9845 T の問題点について、地質調査所鉦床部・金沢康夫氏、月村勝宏氏から御指摘いただいた。

草稿の査読と懇切で有益な御指摘を、化学技術研究所・杉江正昭氏、地質調査所環境地質部・小出仁氏、同所鉦床部吉井守正氏、佐藤岱生氏からいただいた。

小論を草するにあたり、地質調査所企画室、石原舜三室長、鉦床部嶋崎吉彦部長、佐藤壮郎課長地殻熱部小川克郎課長から御世話いただいた。

以上の方々に厚く感謝申し上げる次第です。

(受付：1982年8月23日；受理：1983年7月12日)

付録 RS-232C シリアル・インタフェース

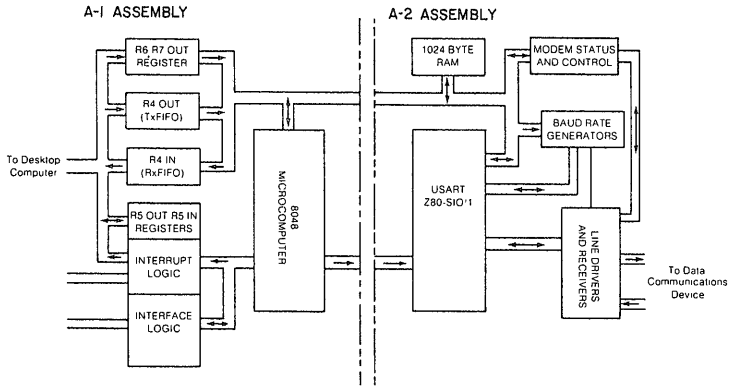
1963年、EIA(米国電子工業協会)はデータ通信のための標準シリアル・インタフェース RS-232C を設定した。RS-232C とは、Recommended Standard-232(コード番号) Connector の略語である。現在使われている仕様は1969年に改訂された仕様である。入出力装置が遠隔にあった場合、回線制御装置を経由して入出力装置と接続する必要がある。RS-232C シリアル・インタフェースは、この回線制御装置とデータ通信回線網を接続する機械的特徴を示すものであり、信号回線数とコネクタの形状規模の明細を規定し、さらに信号回線に流れる電気信号の仕様も規定している。

単なる入力装置の制御ということであれば、IBM 2780/3780という端末仕様がある。IBM 2780はバッファを持ったデータ転送用端末の仕様であり、回線速度は4,800 bps (bits per second)まで、制御する入出力装置の組み合わせによって少しづつ内容が異っている。IBM3780は、IBM 2780 より大きなバッファを持つ端末で回線速度7,200 bps まで、ラインプリンタなどの制御に用いら

付表 RS-232C に使用する各ピンの定義

Pin	Circuit Reference		Description
	RS-232C	CCITT	
*1	AA	101	EARTH or Frame Ground
*2	BA	103	Transmitted Data
*3	BB	104	Received Data
*4	CA	105	Request to Send
*5	CB	106	Clear to Send
*6	CC	107	Data Set Ready
*7	AB	102	Signal Ground
*8	CF	109	Received Line Signal Detector
9	+P		Positive DC Test Voltage
10	-P		Negative DC Test Voltage
11			Unassigned
*12	SCF	122	Secondary Received Line Detector
13	SCB	121	Secondary Clear to Send
14	SBA	118	Secondary Transmitted Data
*15	DB	114	Transmitter Clock
16	SBB	119	Secondary Received Data
*17	DD	115	Received Clock
18			Unassigned
*19	SCA	120	Secondary Request to Send
*20	CD	108 / 2	Data Terminal Ready
21	CG	110	Signal Quality Detector
*22	CE	125	Ring Indicator
*23	CH / CI	112, 111	Data Rate Selector
*24	DA	113	External Transmitter Clock
25			Unassigned

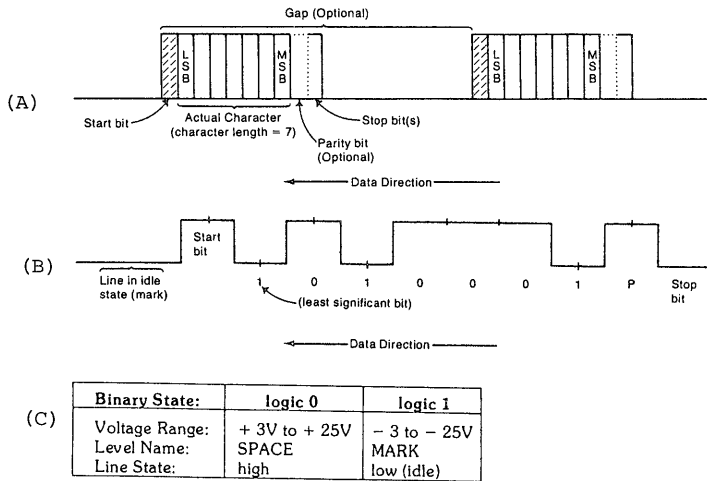
*印は YHP98046 シリアル・インタフェースで使用しているピンである。CCITT はフランス国の仕様。(図は横河ヒューレット・パッカート㈱社の許可を得て同社マニュアル No. 98046-90030 p. A-5 から転載した)。



付図1 YHP98046 シリアル・インタフェースの内部構造ブロック・ダイアグラム

A-1 アセンブリは YHP 9845T システムのシステム・バスに接続され、送受信信号は各レジスタを通じて 8048 MCPU で制御・変換されて USART に送られる。

A-2 アセンブリは USART(本文参照)の働きで、モデムとの送受信信号を変換する。(図は横河ヒューレット・パッカード 社の許可を得て、同社マニュアル No. 98046-90030 p. 3-2 から転載した)。

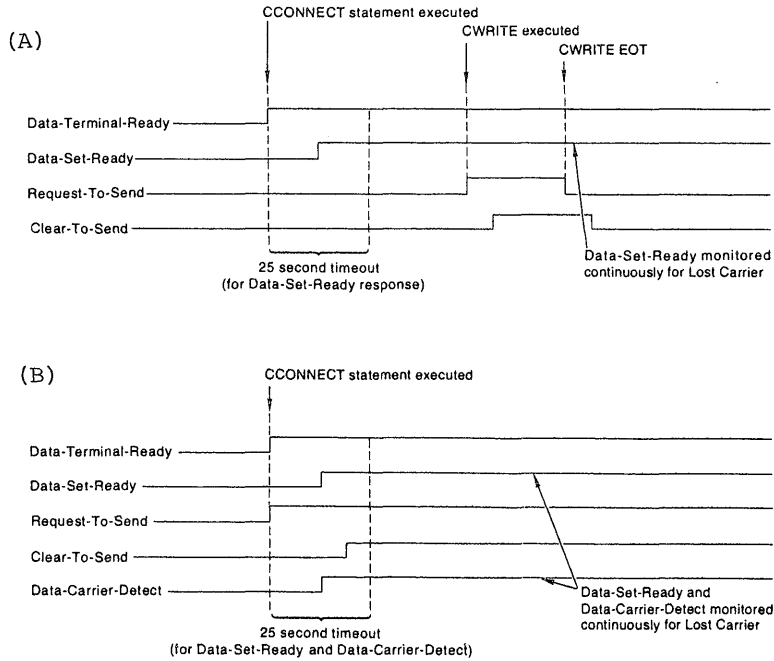


付図2 文字データの流れと定義

送受信データは ASCII コードが使われる。(A)は、各コードの内容・順序と連なりを示した。(B)は回線を流れるデータの電圧の起伏を示す。各コードは、すべて2進数に直され、1つのコードは7桁の2進数で表わされる。(C)は、各ビット(bit: 2進数)で使われる電圧とその意味を示している。正電位の +3V から +25V までの間は、0と考えられ、負電位の -3V から -25V までの間は、1と考えられる。データが流れずに、アイドルの状態にいる時には、1の状態をとる。0が来た時(負から正へ励起)、次の bit から ASCII コードが始まり、7桁で ASCII コード1つの伝送が完了する。ここでパリティ・チェック(偶数か奇数か)をおこない、最後に1つ以上の bit(s) で文字コードの終了を伝える。各コードの間の gap は、あらかじめ仕様に従ってきめておく。(図は横河ヒューレット・パッカード 社の許可を得て同社マニュアル No. 98046-90010 p.1-7 と 1-8 から転載した)。

れている。IBM 2780/3780と同じ仕様は、ホスト M-200の外部入出力装置として9845Tをデータ通信回線につないでいるだけなので、既に完成されたプログラムを使ったデータの送受信だけに使うには便利である。しかし、

同時に 9845T 側でも別のプログラムを使った処理手順を埋め込んで走らせるという芸当はできない。というのは、入出力機器がホスト計算機の支配下におかれているから、その支配の条件下で9845T自身の処理を走らせ



付図3 半二重方式と全二重方式

(A)は半二重式の電位変化様式で、折線の各々は、上から第20、6、4、5番のRS-232C各ピンに相当している。(B)は全二重方式の場合で、最下線のdata-carrier-detectは第8番ピンに相当する。(B)の方が両方向独立に送受信制御信号を送れるので、応答が早い。(図は横河ヒューレット・パッカード株式の許可を得て、同社マニュアル No. 98046-90010 p.2-15から転載した)。

る自由度がきわめて少ないからである（もともと、やってもやれない事はないはずである）。IBM 2780/3780 仕様のエミュレータは、既に YHP 社が供給しているものがあるので、これを用いることができる。これを RJE (Remote Job Entry)ターミナル・エミュレータと呼ぶ。

プログラムの作成と編集やその他の操作を会話式におこなう場合は、TSS (Time-Sharing System: 時分割システム)手順で行なう。TSS 環境における通信手順は、二進同期法 BSC などを用いる単なる RJE と異なる。ただし回線速度は一般に遅く、9845T と M200 の間では 1,200 bps でデータ通信がおこなわれている。プログラムを不断に開発・改良する方に重点がおかれている場合には、TSS 環境下で作業をすすめる必要がある。また、後述するように、TSS 環境下でも RJE (又はバッチ処理)ができるのが M200 の特長であり、これを利用すれば、作業の大半を TSS で完了できる。TSS で用いられている回線制御規約には各種のものがあるが、9845T では RS-232C を用いている(なお、回線制御規約は IBM 2780/3780 仕様以外にも各種各様存在しているが省略する)。

RS-232C は、25ピン構成のコネクター端子を持つ。各ピンの内容は付表に示した通りである。このうち送受信データそのものは第2ピンと第3ピンを流れ、第1ピンはアースとなっている。その他のピンはデータ通信の制御に使われている。二次送受信データ用のピン(13, 14, 16, 19)も存在して複雑な伝送制御に供している。9845T で使われる RS-232C は HP 98046B シリアル・インタフェースと呼ばれ、25本のピンのうち使われるものは、1-8, 12, 15, 17, 19, 20, 22-24の16本である。

信号はすべて2進数で表現される。第7ピンが信号のゼロ電位を示し、+3~+15ボルトを0、-3~-15ボルトを1と認識することになっている。ただし、+3~+5及び-3~-5ボルトは遷移領域のノイズ端にあてているので、0と1に変えるドライバーが起動するのは±5ボルトからとされている。HP 98046B はテラタイプ型のインタフェース(TTY)であるため、20mAの電流ループが流れており、非同期で通信が実行される。この通信では、開始ビット(0, space)で定義される先頭メッセージによって1信号が送られ、そのあとに1個以上の終端ビット(1, mark)がある。

付図1にHP 98046B シリアル・インタフェースの構造を示した。9845Tのシステム I/O バスに流れるデータは、レジスターによる制御のもとに、A-1 アセンブリ中の6 MHzで動く8048 μ CPUを介して、A-2 アセンブリのバスに流れる。 μ CPU 8048は1024 B(バイト)のROM(Read Only Memory)と64 BのRAM(Random Access Memory)を用意しており、9845Tのデータを適切に変換する。逆にA-2 アセンブリのバスから流れてきたデータは、9845Tに変換されて出て行く。

A-2 アセンブリのUSART (Universal Synchronous/Asynchronous Receiver/Transmitter)は、Z 80 serial Input/Output ICチップを使って、A-1 アセンブリから流れてきた並列データを直列データに変換する。又は、直列データを並列データに変換してA-1 アセンブリに送り込む。9845TやM200のデータ・バスを流れるのは並列データであるが、データ通信回線の中は直列データが往来するので、この処理が必要である。HP 98046Bは、1文字を7ビットのASCIIコードで表現して直列型に直す働き(又は並列型に直す)をしているわけである。付図2は1文字のデータが直列型に変換されてデータ通信回線を流れる様子を概念的に示したものである。まず start bit が伝送コードの開始を告げる。続いて2進数が7桁流れて、コード番号を認識する。次のビットはパリティ・チェックのためであり、通信回線を通る間に誤りがあったかどうか調べる。これはオプションのため、はじめに定義しないとつかないことがある。最後に

1つ以上の stop bit(s)がきて終了する。この終端ビットから次の開始ビットまでの間の最小間隔時間が何ミリ秒になるかは、あらかじめ定義しておく必要がある。9845TやM200のタイミングがうまく合わないと開始ビットの読み落としが起る。本文中述べた通信実験では、いずれかの応答が早すぎるために片方がついていけない事象がしばしば起こった。

M200の持つネットワークの各ノードにはモデム(復変調装置)が接続されている。RS-232Cは、このモデムに接続される。モデムを使う接続には全二重式(同時送受信)と半二重式(交互に送受信)の他に単向式(送信又は受信のみ)がある。ここでは全二重式が使われ、第2, 3ピンを通じるデータの送受信の制御ピンのデータ往来を同時双方向に独立伝送できる。単に二重式とも呼ぶ。付図3にモデム接続にした時の全二重、半二重それぞれの信号の往来の違いを示した。どのような制御形式をとるべきか、また何が最適であるかは、回線網制御の方式によって決まる。最近ではRS-232Cよりも、同じEIA標準のRS-422やRS-423に置き換える例が増えている。相手が別の計算機でなく、実験装置の計測制御端子であるような場合には、GP-IB(General Purpose-Interface Bus)とも呼ばれるIEEE-488インタフェースが多く使われる。あるいは外部記憶装置の制御にはDMA(Direct Memory Access)と呼ばれる高速伝送チャンネルを使うこともある。