5. 開発報告

ＰＣ９８シリーズユーザのためのＴＳＳ通信制御プログラム

工学部　構造工学科
修行　稔

１．まえがき

ＭＳ－ＤＯＳ上で動くスクリーン・エディターが最近いろいろと市販されているが、実際に使ってみるとその性能の良さに驚かされる。筆者の手元にあるパソコンＰＣ９８ＸＬとそのハイレゾリューション・モード用スクリーン・エディターＭＩＦＥＳＸを使用した感じでは、総合的に見るとセンターの専用端末を用いたデータセットの編集機能を凌ぐのではないかとさえ思われるほどである。パソコンをＴＳＳ端末として使う場合いろいろな形態が考えられる訳であるが、上記のような最近の事情を考慮すると「ＦＯＲＴＲＡＮソース・プログラムやデータなどのファイルをパソコン側で作成し、これを一括してホストに転送して計算を依頼する。エラーが生じてプログラムやデータの修正が必要な場合には、修正に必要な情報をホストからパソコンに転送し、これをもとにパソコンのファイルを修正して再び一括してホストに転送して計算を依頼する。」という形を中心とした使い方があってもよいように思われる。

ただ、このような使い方が実用的であるためには少なくとも次の二つの条件が満たされなければならない。

１．パソコンとホスト間の転送速度が速いこと。

２．スクリーン・エディターが使えるＭＳ－ＤＯＳモードと、ファイル転送や計算依頼のできるＴＳＳモードとの切り替えが簡単なキー操作で瞬時に行え、転送したファイルがすぐ計算に使用できること。

条件１の転送速度に関しては、２４００ｂｐｓのモデムを用いれば充分実用になる可能性があり、条件２についてもＭＳ－ＤＯＳ上で走る通信制御プログラムを作製することで容易に実現できる。そこで、筆者は次のような方針のもとにプログラムを作ってみることにした。

１．できるだけ転送速度を上げるため、アセンブリ言語で組む。

２．機能は必要最小限度のものに絞り、通常のＴＳＳとファイルの転送のみとしてホストのファイル（データセット）の編集は富士通提供のＰＦＤＥ　ＴＴＹＴＹＰＥ［９］に頼る。そのかわり、前記の使用方法に対してはできるだけ使い易くなるよう工夫する。

３．日本語の受信をサポートする。（注１）

試作したものをこれまで約５ケ月間使用してきたが、不備なところの手直しもほぼ終わり、現在筆者の手元で快調に動いている。性能としては、ＰＣ９８ＸＬのハイレゾリューション・モード（クロック１０ＭＨｚ）で２４００ｂｐｓのときファイル転送速度が受信の場合毎秒約２

１０字（ＴＴＹ　Ｔ４０１０　コマンド入力時（注２））、送信の場合毎秒約２００字である。また、信頼性については、１３００行のＦＯＲＴＲＡＮソース・プログラムを交換回線を用いて５回往復転送し、原プログラムと比較するという操作をこれまで何度か繰り返し、文字の欠落がないことを確認している。使ってみると案外便利なものであり、本センターのユーザー諸氏の中にもあるいは使ってみたい方もおられるのではないかと思い、ここに紹介することにした。

２．必要な機器

２．１　ハードウェア

まず、中心となるパソコンであるが、今のところ正しく動くことを確認しているのはＰＣ９８ＸＬ、ＰＣ９８０１ＶＸ、ＰＣ９８０１ＶＭの３機種だけである。基本的にはＭＳ－ＤＯＳの走る１６ビット・マシンであれば使用可能であると思われるが、ＰＣ９８シリーズ以外の機種の場合は，ＲＳ２３２ＣのＩ／Ｏポート・アドレスや拡張システムコールなどが違う可能性があるから、プログラムを若干変更し、再度アセンブルしなければならない。なお、ディスク・ドライブが本体内蔵のものを含めて１ドライブ以上必要である。実際には２ドライブないと、いろいろと不便を生じる。

次に、センターとの接続に用いる音響カプラまたはモデムとＲＳ２３２Ｃケーブルがいる。本プログラムの性格上モデムは１２００ｂｐｓ以上のものが望ましい。最近のモデムの値下がりは著しく、２４００ｂｐｓのものが５万円以下で買える。

ディスプレイは高解像度のものが必要で、できればカラー・ディスプレイがよいが、モノクロでも使える。プリンターはなくてもよい。

２．２　ソフトウェア

ＭＳ－ＤＯＳ　Ｖ．２．１１以上のシステム・ディスクとＭＳ－ＤＯＳ上で動くスクリーン・エディターが必要である。筆者はＭＳ－ＤＯＳ　Ｖ．３．１０とメガソフト社のＭＩＦＥＳ－９８を使用している。スクリーン・エディターはこれがないと仕事ができないという訳ではないが、ないと効率が著しく悪くなる。

３．　準備

ＭＳ－ＤＯＳのシステム・ディスクには、ふだんあまり使わないファイルもたくさん入っている。そこで、まずシステム・ディスクのバックアップ・コピーを取ったあと、これとは別に、必要なファイルだけをコピーしたディスクを作り、本プログラムなどはこれに納めて使うほうがよい。何を残すかは各人の事情によって異なるが、本プログラムの実行に際して最小限度必

要なファイルは以下の通りである。

システム・ファイル　COMMAMD. COM　RSDRV. SYS

SPEED. COM　CONFIG. SYS　AUTOEXEC. BAT

NECREN. DRV (またはNECDIC. DRV)　KEY. COM

これに、スクリーン・エディター関係のファイルと本プログラムの実行形式ファイルTSS. COMが加わることになる。MS－DOSのバージョンによってはRSDRV. SYSがついていないものもあるが、この場合はRSDRV. SYSは必要ない。

上記のディスクができたら、まずスクリーン・エディターでAUTOEXEC. BATを呼び出し、前の内容を消して代わりに次のように入力してディスクに格納する。

```
DATE⇩
TIME⇩
SPEED RS232C-0 2400 BITS-7 PARITY-EVEN STOP-2 XON⇩
```

3行目はSPEEDコマンドであり、1200.bpsのときは2400を1200に変えねばならない。センターの設定ではストップビットは1であるが2とした方がファイル送信に用いているTRANSFERコマンドのエラーが少いようである［3］。次にCONFIG. SYSを呼び出し、次の文のうちまだ書き込まれていないものがあれば、それを追加してディスクに格納する。

```
DEVICE=RSDRV.SYS⇩
DEVICE=NECREN.DRV⇩ (またはNECDIC.DRV⇩)
BUFFERS=30⇩
FILES=20⇩
```

RSDRV. SYSが付いていないバージョンの場合は1行目は不要である。CONFIG. SYSには、ラムディスクやプリンター（98XLのハイレゾリューション・モードのとき）を使う場合にはそれぞれのディバイス・ドライバーを登録しなければならないが、これについてはMS－DOSのマニュアルを参照されたい。

4.　使い方

4. 1　基本

（1）モデムとパソコンの電源スイッチをいれ、前記3. で作製したディスクを用いてパソコンを立ち上げる。

（2）　A>TSS⇩

と入力すると（A＞　はパソコンのプロンプター）本プログラムが走り出すから、モデムの

マニュアルに従ってホストとモデムを接続し、ＣＴＲＬ－Ｂを押す（ＣＴＲＬ－ＢはＣＴＲＬキーを押しながらＢキーを押すことを意味する。以下同じ。）。ベルコード（ピッという音）が返ってきたら、ＬＯＧＯＮすることによって通常のＴＳＳモードにはいることができる。ＴＳＳモードにはいったら、

READY

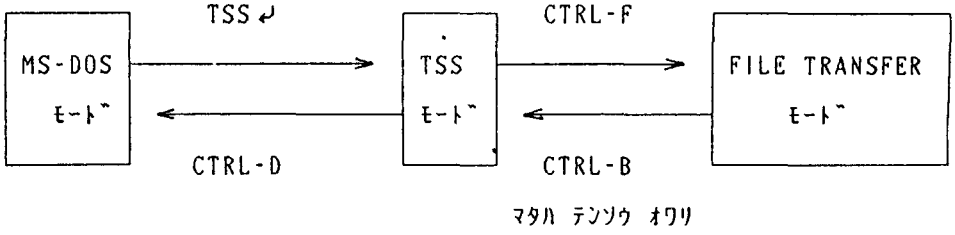
TTY T4010⇩　　（注２）

READY

TERM LINESIZE(136)⇩　　（注３）

と入力しておく。ＴＳＳモードではＰＦＤＥ　ＴＴＹＴＹＰＥを用いてホストのデータセットのスクリーン・エディットが行える（注４）。

（３）ＴＳＳモードからのモード変換は、次のようにして行う。



前述の通り、例えばＣＴＲＬ－ＦはＣＴＲＬキーを押しながらＦキーを押すことを意味する。ＢはＢｒｅａｋ、ＦはＦｉｌｅ　ｔｒａｎｓｆｅｒ、ＤはＤｏｓの意である。

（４）ＴＳＳモードでＲＥＡＤＹ状態のときＬＯＧＯＦＦと入力すれば、ホストからの終了メッセージを出力して自動的にＭＳ－ＤＯＳモードに復帰する。この後２、３秒してホストとモデムとの接続が切れ、モデムのキャリア検出ランプが消えるから、再度ＴＳＳを行いたければ、このランプの消灯を確認してから上記（２）の要領でもう一度ＬＯＧＯＮしなければならない。

４．２　ファイルの転送

まず、ＴＳＳモードにおいてＣＴＲＬ－Ｆを押し、ファイル転送モードに入る。このモードでは為すべき仕事が全てディスプレイ上に黄色の文字で表示されるから、その通りに入力していけばよい。本プログラムではファイルの受信にはホストのＥＤＩＴモードのＬＩＳＴコマンドを用い、送信にはＴＲＡＮＳＦＥＲコマンド（［６］、［８］）を利用しているから、このことを念頭に置いておかれると使い易くなるのではないかと思う。ファイルの送信について注意点を述べると、送信終了後１０秒位待つと転送がうまくいっていれば、

KEQ530581 DATA SET F1234.XXX.FORT END PROCESSED

READY

と返ってくるから、このデータセットをＴＲＡＮＳＦＥＲコマンドから解放するため、

FREE ALL⇩

と入力する。転送がうまくいかなかった時はその旨のメッセージが返ってくるから、再度送信を試みる（注５）。なお、受信はアスキー・ファイル、日本語ファイルともにできるが、送信できるファイルはアスキー・ファイルに限られる。受信送信ともパソコン側のファイル名にはパス名がついていてもよい。

４．３　その他

（１）ＣＴＲＬ－Ｂは通常のＢｒｅａｋキーと同じと考えてよく、いつ使用してもよい。ＴＳＳモードおよびＦＩＬＥ　ＴＲＡＮＳＦＥＲモードで作業ミスをしたときやファイル転送を中断したいとき、あるいはどうしたらよいか分からなくなった時などは、ＣＴＲＬ－Ｂを押せばＲＥＡＤＹ状態に復帰する。

（２）画面の文字は、ＭＳ－ＤＯＳモードのとき白、ＴＳＳモードのとき緑で表示され、ＦＩＬＥ　ＴＲＡＮＳＦＥＲモードのときはユーザーが為すべき仕事が黄色、転送中のファイルの内容が青色で表示される。

（３）本プログラムによるファイルの転送では、ファイルの内容に余分なものが一切付加されないから、ホストあるいはパソコン側の計算にそのまますぐ使うことができる。

（４）本プログラムでは，ファンクション・キーを全く使用していない。ＭＳ－ＤＯＳのＫＥＹコマンドを使ってＴＢＬという拡張子を持つファイルを作ることによって、いろいろな組合せのファンクション・キーの設定が可能であるから、好みに応じて設定されるとよい（注４）。

（５）ホストがフロー制御をサポートしていれば、モードに拘らず、画面への出力を一時停止させるキーＣＴＲＬ－Ｓが使用できる。再開させたいときには任意のキーを押せばよい。ただし、ＴＳＳモードおよびＦＩＬＥ　ＴＲＡＮＳＦＥＲモードのファイル受信時には画面表示の再開のためには任意のキーを２回押す必要があり、かつ画面にその文字が出力されるが、受信中のファイルの内容には何の影響もないから心配ない。

（６）現在の画面をプリンターに出力させたければＣＯＰＹキーを押す。これから画面に出力させるものをプリンターにも出力させたいときにはＣＴＲＬ－Ｐを押す。プリンターへの出力を中止させたいときは再度ＣＴＲＬ－Ｐを押せばよい。ＣＴＲＬ－Ｐはトグルスイッチになっている。プリンターの速度は普通極めて遅いため、ＣＴＲＬ－Ｐの押下とプリンターの反応とにタイムラグを生ずるから注意を要する。なお、これらのキーはどのモードでも使え

るが、ＴＳＳモードとＦＩＬＥ　ＴＲＡＮＳＦＥＲモードで画面への出力を同時にプリンターに出力させると、プリンターのバッファがすぐオーバーフローを起こしてトラブルの原因になるから、同時出力をさせずに一度ファイルに落としたものをＭＳ－ＤＯＳモードでプリントさせる方がよい。

（７）以上の各種スイッチの意味はＨＥＬＰキーで参照できる。

5.　あとがき

本プログラムは、画面表示の高速化コマンドＴＴＹ　Ｔ４０１０の入力後であれば、ＰＣ９８ＸＬとＨＩ－ＭＯＤＥＭ２４００を用い、２４００ｂｐｓで筆者の行番号なしの１３００行のＦＯＲＴＲＡＮソース・プログラムを３分３０秒で送信し、３分２０秒で受信する。また、ＰＦＤＥ　ＴＴＹＴＹＰＥコマンドを用いれば、ホストのデータセットのフルスクリーン・エディットが専用端末とほぼ同じ感覚で行える。ただ、研究室からホストに接続した場合ひとつしかない電話がふさがってしまうという難点がある。現在既に、既設の電話線のみで電話とモデムとが同時に使用可能となる通信制御機器が市販されていることでもあり、センターのご努力でこの問題点の早期解決がなされるよう希望したい。なお、本プログラムはセンターに置いてあるので、使用ご希望の方はセンターでコピーして一応使ってみて頂き、何か不都合が起こるようであれば筆者にご連絡たまわりたい。

最後に、本プログラムの開発に際していろいろとお世話になった情報処理センターの教職員の方々に心からお礼を申しあげる。

（注１）

ホストが交換回線からの日本語の入力をサポートするようになった時点で、日本語ファイルの送信機能を追加する予定である。

（注２）

ＴＴＹ　Ｔ４０１０コマンドは、ホストから端末への転送速度を高速化させるコマンドである。このコマンドを発しないとファイルの受信速度が１／３程度に低下する。なお、転送速度の低速化はＴＴＹ　ＴＷコマンドによって可能である。

（注３）

転送するファイルのレコード長が１３６（１３６はプリンタのラインサイズ）以上の場合は、例えば２５５バイトであれば、

```
TERM LINESIZE(255)⇩
```

と入力する。レコード長が標準の８０バイトでありプリンタも使用しないのであれば、このコマンドは必要ない。

(注4)

ＰＦＤＥ　ＴＴＹＴＹＰＥは専用端末のＰＦキーの代替機能をサポートしており、ヘルプ情報のＰＦ１、ＥＮＤのＰＦ３、画面送りのＰＦ７とＰＦ８、修正画面の再表示のＰＡ２キーの機能が、パソコンではそれぞれＨＯＭＥキー（またはＣＴＲＬ－＾）を押した後、１⇩、３⇩、７⇩、８⇩、ＰＡ２⇩と入力することで代替できる。ＭＳ－ＤＯＳのＫＥＹコマンドを用いてこれらの文字列をそれぞれｆ・１やｆ・３などに割り当てておけば、データセットの編集が専用端末とほぼ同じ感覚で行える。なお、九大の大型計算機のＰＦＤＥは制御コードが長大と若干異なるので、ＮＶＴを介して九大の計算機を使用する際には、ＰＦＤＥの使用直前と直後にＣＴＲＬ－Ｅを押してコード変換機能の組み込みと切り離しを行う必要がある。切り離しを忘れると、日本語がほかの文字に化けるから注意を要する。

(注5)

ファイルの送信において送信エラーが頻発するようであれば、次のことを確認もしくは試行されたい。

a．ストップビットは2になっているか。

b．送信先のデータセットを一度ＤＥＬＥＴＥし、再度ＥＤＩＴコマンドで作成してみる。その際、一行目には何か文字を入力しておく。

c．ＣＴＲＬ－Ａを押し、使用中の機器に転送速度を合わせる。

参考文献

1）泉、小山、山田：マイクロコンピュータによるＦＡＣＯＭ　Ｍ－１８０ⅡＡＤへのファイルの送信及び受信について、長崎大学情報処理センターレポート第１号（１９８０）

2）清木、芳本：ＰＣ－８８００／ＰＣ－８０００シリーズによるテキスト編集および、ＴＳＳインテリジェントターミナルのためのプログラム、長崎大学情報処理センターレポート第3号（１９８２）

3）金丸：ＰＣ－９８００ＴＳＳインテリジェントターミナル用プログラム、長崎大学情報処理センターレポート第4号（１９８３）

4）金丸、杉本、内田：ＰＣ－９８００ＴＳＳインテリジェントターミナル用プログラム（ＭＳ－ＤＯＳ版）への移植と２４００ｂｐｓモデムの使用感、長崎大学情報処理センターレポート第6号（１９８５）

5）木須：オンラインＴＳＳ（ＦＡＣＯＭ用）のためのフルスクリーンエディター（ＰＣ－９８０１版）、長崎大学情報処理センターレポート第7号（１９８７）

6）長谷部、吉井、古金：ホスト（ＦＡＣＯＭ　Ｍ－１８０，２００，３８２）とＴＴＹ手順端末間でのファイル転送、九州大学大型計算機センター広報　Ｖｏｌ．１８　Ｎｏ．５

（１９８５）
7）武政：Ｃ言語による日本語ＴＳＳ端末エミュレータ　－ＰＣ－９８０１シリーズ対応－、九州大学大型計算機センター広報　Ｖｏｌ．２０　Ｎｏ．３（１９８７）
8）富士通（株）：ＦＡＣＯＭ　ＯＳ　ＩＶ／Ｆ４　ＭＳＰ　ＴＳＳコマンドセットＴＴＹ（ＣＳ／ＴＴＹ）使用手引書
9）富士通（株）：ＰＦＤ使用手引書プログラム開発機能編
１０）入江、永井、篠原、松尾：ＪＯＩＳ型漢字端末エミュレータの作成について、九州大学大型計算機センター広報　Ｖｏｌ．１　Ｎｏ．３（１９８５）
１１）日本電気（株）：ＭＳ－ＤＯＳ　ＴＭ３．１　プログラマーズリファレンスマニュアル
１２）日本電気（株）：ＭＳ－ＤＯＳ　ＴＭ３．１　マクロアセンブラマニュアル

```
;       ************************************************************
;       *            NEC PC98 - FACOM M360 communication          v.1.0 *
;       *                  by m.shugyo  April, 1987                    *
;       ************************************************************
;
code    segment
        assume  cs:code,ds:code,es:code,ss:code
;
        org 100h
start:  jmp lbl
;
msg1    db  1ah,13
        db  "   ************************************************************",13,10
        db  "   *   NEC PC98 series - FACOM M360  communication   v.1.0 *",13,10
        db  "   *                                                          *",13,10
        db  "   *          CTRL-B : ブレイク信号／仕事の中断               *",13,10
        db  "   *          CTRL-D : ＭＳ－ＤＯＳへ                         *",13,10
        db  "   *          CTRL-F : ファイルの転送                         *",13,10
        db  "   ************************************************************",13,10
        db  "                                        copyright  shugyo  '87",13,10
        db  "$"
msg1a   db  "   ************************************************************",13,10
        db  "   *          CTRL-A : 使用機器の設定                         *",13,10
        db  "   *          CTRL-B : ブレイク信号／仕事の中断               *",13,10
        db  "   *          CTRL-D : ＭＳ－ＤＯＳへ                         *",13,10
        db  "   *          CTRL-E : 九大ＰＦＤのための特殊コード系         *",13,10
        db  "   *          CTRL-F : ファイルの転送                         *",13,10
        db  "   *          CTRL-P : プリンターのスイッチ                   *",13,10
        db  "   *          CTRL-S : 画面出力の一時停止                     *",13,10
        db  "   *          COPY   : ハードコピー                           *",13,10
        db  "   ************************************************************",13,10
        db  "$"
msg2    db  1ah,1bh,"[42m",13
        db  1bh,"[12C",1bh,"[42m"," ************************** ",13,10
        db  1bh,"[12C"," *      ファイルの転送      * ",13,10
        db  1bh,"[12C"," ************************** ",13,10,10
        db  1bh,"[21m",13,10
        db  "       Ｒ（受信）またはＳ（送信）を押して下さい．",13,10,10,"$"
msg3    db  1ah,1bh,"[22m",13
        db  "              ***************************",13,10
        db  "              *      ファイルの受信       *",13,10
        db  "              ***************************",13,10,13,10
        db  1bh,"[21m",13
        db  "受信するデータセットをＥＤＩＴし，ＵＮＮコマンドで行番号",13,10,10
        db  "を消して下さい．その後，ＣＴＲＬ－Ｇを押して下さい．",13,10,10
        db  "受信するものがデータセットでない場合は，ＣＴＲＬ－Ｇのみ",13,10,10
        db  "を押して下さい．",13,10,10,1bh,"[22m$"
msg4    db  1bh,"[21m",13
        db  "受信したデータを格納するファイルの名前を入力して下さい．",13,10,10
        db  "          （例：  Ａ：ＮＡＭＥ．ＤＡＴ）",13,10,10,1bh,"[22m$"
msg5    db  1bh,"[21m",13
        db  "同じ名前のファイルが既に存在します．",13,10,10
        db  "Ｇ（上から書き込む）またはＲ（再入力）を入力して下さい．",13,10,10
        db  1bh,"[22m$"
msg5a   db  1bh,"[21m",13
        db  "ファイル名が不当です．",1bh,"[22m$"
msg6    db  1bh,"[21m",13
        db  "ＬＩＳＴコマンドその他の出力命令を入力して下さい．",13,10,10
        db  "受信はＣＴＲＬ－Ｂで中止できます．",13,10,10,1bh,"[22m$"
msg7    db  1ah,1bh,"[22m",13
```

```
        db  "            ****************************",13,10
        db  "            *      ファイルの送信       *",13,10
        db  "            ****************************",13,10,10,1bh,"[21m",13
        db  "ＥＤＩＴコマンドを用いて送信先のデータセットを作成し，",13,10,10
        db  "ＳＡＶＥしてＲＥＡＤＹモードに戻って下さい．その後，",13,10,10
        db  "ＣＴＲＬ－Ｇを押して下さい．",13,10,10
        db  1bh,"[22m$"
msg7a   db  1bh,"[21m",13
        db  "送信先のデータセットの名前を入力して下さい．",13,10,10
        db  "      （例：  ＮＡＭＥ．ＦＯＲＴ）",13,10,10
        db  1bh,"[22m$"
msg8    db  1bh,"[21m",13
        db  "送信するファイルの名前を入力して下さい．",13,10,10
        db  "    （例：  Ｂ：ＮＡＭＥ．ＦＯＲ）",13,10,10
        db  "送信はＣＴＲＬ－Ｂで中止できます．",13,10,10,1bh,"[22m$"
msg9    db  1bh,"[21m",13
        db  "ファイルが見つかりません．$"
msg10   db  1bh,"[21m",13
        db  "受信したデータは下記のファイルに格納されました．",13,10,10,"$"
msg11   db  1bh,"[21m",13
        db  "送信したデータは下記のデータセットに格納されました．",13,10,10,"$"
msg11a  db  1bh,"[21m",13
        db  "１０秒ほどお待ち下さい．",13,10,"$"
msg12   db  1bh,"[17m",13
        db  "file_write error happened!",13,10,"$"
msg13   db  1bh,"[17m",13
        db  "file_read error happened!",13,10,"$"
msg14   db  13,10,10
        db  "九州大学の大型計算機でのＰＦＤ（ＰＦＤＥ）にのみ対応",13,10,10
        db  "する特殊なコード系になりました．ＰＦＤの終了後は必ず",13,10,10
        db  "ＣＴＲＬ－Ｅを押して下さい．",13,10,10,"$"
msg15   db  13,10,10
        db  "普通のコード系に戻りました．",13,10,10,"$"
msg16   db  13,10,10
        db  "     使用中の機器に合わせて下さい．ファイル送信 ",13,10,10
        db  "     エラーが起こらないようなら，５にして下さい．",13,10,10
        db  "                1 : cpu-V30,   1200bps",13,10
        db  "                2 : cpu-V30,   2400bps",13,10
        db  "                3 : cpu-80286, 2400bps",13,10
        db  "                    (normal)           ",13,10
        db  "                4 : cpu-80286, 2400bps",13,10
        db  "                    (high resolution) ",13,10
        db  "                5 : ?                  ",13,10,10
        db  "                  あなたの機器は =$"
crlf    db  13,10,13,10,"$"
crt1    db  1bh,"[>1h$"
crt2    db  1bh,"[>11$"
crt3    db  1bh,"[20m$"
crt4    db  1bh,"[m$"
crt5    db  1bh,"[0K$"
;
aux_buffer      db  300 dup(?)
abi             dw  ?
nbytes_d        dw  ?
file_name       db  80 dup(?)
center_f_name   db  40 dup(?)
handle          dw  ?
buffer          db  300 dup(?)
nbytes_u        dw  ?
flg_aok         db  ?
flg_wb          db  ?
```

```
flg_n_k         db  ?
;
m_attrb         db  "5"
intvl           dw  1
;
keypfd_ins    db  1ch,0,0,0,0,0
keypfd_del    db  1dh,0,0,0,0,0
keypfd_hlp    db  09h,0,0,0,0,0
key_ins       db  1bh,"P",0,0,0,0
key_del       db  1bh,"D",0,0,0,0
;
nssn          dw  9
ssn_end       db  "KEQ56470I"
ntrans1       dw  9
trans1        db  "TRANSFER "
ntrans2       dw  27
trans2        db  " FROM AU(TRANSFIL) TYPE(3)",13
nhdr          dw  23
hdr           db  "HDR=TRANSFIL080FB03120",13
neof1         dw  9
eof1          db  "KEQ52500I"
neof2         dw  6
eof2          db  "EOF=",3,13
nrts          dw  2
rts           db  ".S"
;
;
set_key_code  macro  key,buffer
         mov ax,key
         mov dx,offset buffer
         mov cl,0dh
         int 0dch
         endm
;
create_new_file  macro  path,attrib
         mov dx,offset path
         mov cx,attrib
         mov ah,5bh
         int 21h
         endm
;
create_handle  macro  path,attrib
         mov dx,offset path
         mov cx,attrib
         mov ah,3ch
         int 21h
         endm
;
open_handle  macro  path,access
         mov dx,offset path
         mov al,access
         mov ah,3dh
         int 21h
         endm
;
close_handle  macro  handle
         mov bx,handle
         mov ah,3eh
         int 21h
         endm
```

```
;
read_handle  macro  handle,buffer,bytes
         mov bx,handle
         mov dx,offset buffer
         mov cx,bytes
         mov ah,3fh
         int 21h
         endm
;
write_handle  macro  handle,data,bytes
         mov bx,handle
         mov dx,offset data
         mov cx,bytes
         mov ah,40h
         int 21h
         endm
;
check_kbd_status  macro
         mov ah,0bh
         int 21h
         endm
;
cnsl_in  macro
         mov ah,08h
         int 21h
         endm
;
cnsl_out  macro  character
         mov dl,character
         mov ah,02h
         int 21h
         endm
;
display  macro  string
         mov dx,offset string
         mov ah,09h
         int 21h
         endm
;
bs  macro
         push ax
         cnsl_out 08h
         cnsl_out 20h
         pop ax
         endm
;
aux_out  macro  character
         mov dl,character
         mov ah,04h
         int 21h
         endm
;
cmp_string  macro  n,stg1,stg2
         push si
         push di
         mov cx,n
         mov si,offset stg1
         mov di,offset stg2
         repe cmpsb
         pop di
         pop si
```

```
          endm
;
aux_out_string  macro  n,string,intvl
          local a,b
          push si
          xor si,si
        a:mov cx,intvl
          b:
          loop b
          aux_out string[si]
          inc si
          cmp si,n
          jne a
          pop si
          endm
;
time_intvl  macro  n
          local a,b
          xor si,si
        a:mov cx,1000h
          b:
          loop b
          inc si
          cmp si,n
          jne a
          endm
;
send_break  macro
          mov dx,32h                        ;i/o address of RS-232C
          mov ax,3fh                        ;break command
          out dx,ax
          time_intvl 8
          mov ax,37h                        ;command byte
          out dx,ax
          time_intvl 1
          call read_232c                    ;clear RS-233C
          endm
;
read_232c  proc near
          mov nbytes_d,0
          mov cl,0eh                        ;data length on RS-232C
          mov dl,00h
          int 0dch
          cmp ax,0                          ;empty?
          jne r232c
          ret
    r232c:read_handle 3,aux_buffer,ax
          mov nbytes_d,ax
          mov abi,0                         ;reset aux_buffer index
          ret
read_232c  endp
;
aux_in  proc near
          mov bx,abi
          cmp bx,nbytes_d
          jl auin
          call read_232c
          mov bx,abi
          cmp nbytes_d,0
          jne auin
```

```
          ret
     auin:mov al,aux_buffer[bx]
          inc abi
          cmp flg_n_k,"N"                    ;nagasaki univ?
          je cc3
;convert code in PFD of kyushu univ.
          cmp al,16h                         ;^V?
          jne cc1
          mov al,56h                         ;[
          ret
      cc1:cmp al,7ch                         ;|?
          jne cc2
          mov al,21h                         ;!
          ret
      cc2:cmp al,7eh                         ;~?
          jne cc3
          mov al,5eh                         ;^
      cc3:ret
aux_in  endp
;
aux_in_esc  proc near
          call aux_in
          cmp nbytes_d,0                     ;empty?
          je aux_in_esc                      ;retry aux_in
          ret
aux_in_esc  endp
;
code_chng  proc near
          cmp flg_n_k,"N"
          je hc
          mov flg_n_k,"N"
          display msg15
          ret
       hc:mov flg_n_k,"K"
          display msg14
          ret
code_chng  endp
;
esc_seq  proc near
          mov flg_aok,"A"
          call aux_in_esc
          push ax
          cmp al,28h                         ;ANK?
          je ank
          cmp al,0a8h                        ;28+80(shift)?
          je ank
          cmp al,24h                         ;KANJI?
          je kanji
          cmp al,0a4h                        ;24+80?
          je kanji
          cmp al,0ch                         ;^L?
          je pfd
          cmp al,"X"
          je clr
          cnsl_out 1bh
          mov buffer[si],1bh                 ;other ESC sequence
          inc si
          pop ax
          ret
      clr:display crt5                       ;erase line
```

```
          pop ax
          call aux_in_esc
          jmp end
      pfd:cnsl_out 1ah                    ;clear screen
          pop ax
          call aux_in_esc
          jmp end
      ank:pop ax
          call aux_in_esc
          call aux_in_esc
          jmp end
    kanji:pop ax
          call aux_in_esc
          call aux_in_esc
          mov flg_aok,"K"
      end:cmp al,1bh
          je esc_seq
          ret
esc_seq   endp
;
jis_to_sjis  proc near                    ;convert kanji code
     jts1:cmp al,20h
          jge jts2
          cnsl_out al
          mov buffer[si],al
          inc si
          call aux_in_esc
          jmp jts1
     jts2:push ax
     jts3:call aux_in_esc
          cmp al,20h
          jge jts4
          cnsl_out al
          mov buffer[si],al
          inc si
          jmp jts3
     jts4:pop bx
          mov ah,bl
          mov cl,0f3h
          int 0dch
          push ax
          cnsl_out ah
          pop ax
          mov buffer[si],ah
          inc si
          ret
jis_to_sjis  endp
;
write_on_buffer  proc near
          cmp al,1bh                      ;ESC sequence?
          jne wob1
          call esc_seq
    wob1:cmp flg_aok,"A"                  ;ANK?
          je wob2
          call jis_to_sjis
    wob2:cmp al,00h                       ;NULL?
          je wob3
          cnsl_out al
    wob3:mov buffer[si],al
          inc si
          cmp al,0ah                      ;LF?
```

```
          je wob4
          cmp si,255                    ;buffer full?
          jge wob4
          mov flg_wb,"C"                ;continue
          ret
     wob4:mov flg_wb,"N"                ;next string
          ret
write_on_buffer  endp
;
mchn_attrb  proc near
          display msg16
          cnsl_out m_attrb
          cnsl_out 08h
          cnsl_in
          cnsl_out al
          cmp al,"1"
          jne ma1
          mov m_attrb,"1"
          mov intvl,4400
          jmp ma5
      ma1:cmp al,"2"
          jne ma2
          mov m_attrb,"2"
          mov intvl,1500
          jmp ma5
      ma2:cmp al,"3"
          jne ma3
          mov m_attrb,"3"
          mov intvl,2600
          jmp ma5
      ma3:cmp al,"4"
          jne ma4
          mov m_attrb,"4"
          mov intvl,2900
          jmp ma5
      ma4:cmp al,"5"
          jne ma5
          mov m_attrb,"5"
          mov intvl,1
      ma5:display crlf
          ret
mchn_attrb  endp
;
;***************
;*   terminal   *
;***************
;
lbl:      display crt1                  ;enable bottom line
          display msg1
          set_key_code 17h,keypfd_ins
          set_key_code 18h,keypfd_del
          set_key_code 1eh,keypfd_hlp
          mov cl,0e0h                   ;release nihongo kinou
          mov ax,00h
          int 0dch
;
          mov flg_n_k,"N"               ;normal host code
lb2:      send_break
lb2a:     mov flg_aok,"A"               ;reset to ANK mode
          display crt3
          mov nbytes_d,0                ;data length on aux_buffer
```

```
        mov abi,0                       ;reset aux_buffer index
lb2b:   xor si,si                       ;reset buffer index
;
;*** key in ***
;
lb3:    check_kbd_status
        cmp al,00h                      ;empty?
        je lb5
        cnsl_in
        cmp al,02h                      ;break command?
        je lb2
        cmp al,04h                      ;quit command?
        jne lb4
        jmp lb100a
lb4:    cmp al,06h                      ;file transfer command?
        je lb8
        cmp al,05h                      ;host code change?
        jne lb4a
        call code_chng
        jmp lb3
lb4a:   cmp al,01h                      ;mchn_attrb change?
        jne lb4b
        call mchn_attrb
        jmp lb3
lb4b:   cmp al,09h                      ;help_key?
        jne lb4c
        display msg1a
        jmp lb3
;
;*** send character to RS-232C ***
;
lb4c:   aux_out al
        cnsl_out al
;
;*** get character from RS-232C ***
;
lb5:    call aux_in
        cmp nbytes_d,0                  ;empty?
        je lb3
        call write_on_buffer
        cmp flg_wb,"N"
        je lb5a
        jmp lb5
lb5a:   cmp_string nssn,ssn_end,buffer  ;"KEQ56470I"
        jnz lb6
        jmp lb100                       ;end process
lb6:    jmp lb2b
;
;*************************
;*  file transfer routine  *
;*************************
;
lb8:    display crlf
        display msg2
;
        cnsl_in
        cmp al,"R"                      ;receive?
        je lb20
        cmp al,"r"
        je lb20
        cmp al,"S"                      ;send?
```

```
        je lb8a
        cmp al,"s"
        je lb8a
        cmp al,02h                      ;break command?
        jne lb8
        display crlf
        jmp lb2
lb8a:   jmp lb60
;
;******************
;*  receive file  *
;******************
;
;*** open center file ***
;
lb20:   display crlf
        display msg3
;
        mov nbytes_d,0                  ;data length on aux_buffer
        mov abi,0                       ;reset aux_buffer index
lb20a:  xor si,si                       ;reset buffer index
;
lb21:   check_kbd_status
        cmp al,00h                      ;empty?
        je lb24
        cnsl_in
        cmp al,02h                      ;break command?
        jne lb22
        display crlf
        jmp lb2
lb22:   cmp al,07h                      ;go command?
        je lb30
        cmp al,09h                      ;help_key?
        jne lb22a
        display msg1a
        jmp lb21
;
lb22a:  aux_out al
        cmp al,08h                      ;BS?
        jne lb23
        bs
lb23:   cnsl_out al
;
lb24:   call aux_in
        cmp nbytes_d,0                  ;empty?
        je lb21
        call write_on_buffer
        cmp flg_wb,"N"
        je lb25
        jmp lb24
lb25:   jmp lb20a
;
;*** create handle ***
;
lb30:   display crlf
        display msg4
;
        xor si,si                       ;reset file_name index
;
lb31:   cnsl_in
        cmp al,02h                      ;break command?
```

```
        jnc lb31a
        display crlf
        jmp lb2
lb31a:  cmp al,09h                      ;help_key?
        jne lb32
        display msg1a
        jmp lb31
;
lb32:   mov file_name[si],al
        cmp al,0dh                      ;CR?
        je lb34
        inc si
        cmp al,08h                      ;BS?
        jne lb33
        dec si
        cmp si,00h
        je lb31
        dec si
        bs
lb33:   cnsl_out al
        jmp lb31
;
lb34:   cmp si,00h                      ;CR only?
        je lb30
        mov file_name[si],0             ;put NUL at data end
        create_new_file file_name,0
        jnc lb37
        cmp ax,50h                      ;if file already exists
        je lb35
        display crlf
        display msg5a                   ;illegal file name
        jmp lb30
;
lb35:   display crlf
        display msg5
;
        cnsl_in
        cmp al,"G"                      ;go?
        je lb36
        cmp al,"g"
        je lb36
        cmp al,"R"                      ;reenter?
        je lb35a
        cmp al,"r"
        je lb35a
        jmp lb35
lb35a:  jmp lb30
;
lb36:   create_handle file_name,0
lb37:   mov handle,ax
        mov file_name[si],"$"
;
;*** send LIST command and receive center file ***
;
lb40:   display crlf
        display msg6
;
        mov nbytes_d,0                  ;data length on aux_buffer
        mov abi,0                       ;reset aux_buffer index
lb41:   xor si,si                       ;reset buffer index
```

```
;
lb42:    check_kbd_status
         cmp al,00h                          ;empty?
         je lb45
         cnsl_in
         cmp al,02h                          ;break command?
         jne lb42a
         jmp lb50
lb42a:   cmp al,09h                          ;help_key?
         jne lb43
         display msg1a
         jmp lb42
;
lb43:    aux_out al
         cmp al,08h                          ;BS?
         jne lb44
         bs
lb44:    cnsl_out al
;
lb45:    call aux_in
         cmp nbytes_d,0                      ;empty?
         je lb42
         call write_on_buffer
         cmp flg_wb,"N"
         je lb46
         jmp lb45
;
lb46:    cmp buffer,0ah                      ;LF only?
         je lb41
         cmp_string neof1,eof1,buffer        ;"KEQ525001"
         jnz lb47
         jmp lb50
;
lb47:    write_handle handle,buffer,si
         jc lb48                             ;write error?
lb47a:   jmp lb41
lb48:    display crlf
         display msg12
;
;*** close handle ***
;
lb50:    send_break
         close_handle  handle
         display crlf
         display msg10
         display file_name
         display crlf
         time_intvl 80
         cnsl_out 07h
         jmp lb2
;
;***************
;*  send file  *
;***************
;
;*** create center file of destination ***
;
lb60:    display crlf
         display msg7
;
         mov nbytes_d,0                      ;data length on aux_buffer
```

```
        mov abi,0                           ;reset aux_buffer index
lb60a:  xor si,si                           ;reset buffer index
;
lb61:   check_kbd_status
        cmp al,00h                          ;empty?
        je lb62
        cnsl_in
        cmp al,02h                          ;break command?
        jne lb61a
        jmp lb90a
lb61a:  cmp al,07h                          ;go command?
        jne lb61b
        jmp lb66
lb61b:  cmp al,09h                          ;help_key?
        jne lb61c
        display msg1a
        jmp lb61
lb61c:  cmp al,01h                          ;mchn_attrb change?
        jne lb61d
        call mchn_attrb
        jmp lb61
lb61d:  aux_out al
        cmp al,08h                          ;BS?
        jne lb61e
        bs
lb61e:  cnsl_out al
;
lb62:   call aux_in
        cmp nbytes_d,0                      ;empty?
        je lb61
        call write_on_buffer
        cmp flg_wb,"N"
        je lb62a
        jmp lb62
lb62a:  cmp_string nrts,rts,buffer[3]       ;".S"
        jnz lb63
        jmp lb70
lb63:   jmp lb60a
;
;*** send transfer command ***
;
lb66:   display crlf
        display msg7a
;
        xor si,si                           ;reset center_f_name index
;
lb67:   cnsl_in
        cmp al,02h                          ;break command?
        jne lb67a
        jmp lb90a
lb67a:  cmp al,09h                          ;help_key?
        jne lb67b
        display msg1a
        jmp lb67
lb67b:  cmp al,01h                          ;mchn_attrb change?
        jne lb68
        call mchn_attrb
        jmp lb67
;
lb68:   mov center_f_name[si],al
        cmp al,0dh                          ;CR?
```

```
        je lb69
        inc si
        cmp al,08h                          ;BS?
        jne lb68a
        dec si
        cmp si,00h
        je lb67
        dec si
        bs
lb68a:  cnsl_out al
        jmp lb67
;
lb69:   cmp si,00h                          ;CR only?
        jne lb69a
        jmp lb66
lb69a:  mov center_f_name[si],"$"
        aux_out_string ntrans1,trans1,intv1
        mov bx,si
        aux_out_string bx,center_f_name,intv1
        aux_out_string ntrans2,trans2,intv1
        display crlf
        jmp lb60a
;
;*** open handle ***
;
lb70:   display crlf
        display msg8
;
        xor si,si                           ;reset file_name index
;
lb71:   cnsl_in
        cmp al,02h                          ;break command?
        jne lb71a
        jmp lb90a
lb71a:  cmp al,09h                          ;help_key?
        jne lb71b
        display msg1a
        jmp lb71
lb71b:  cmp al,01h                          ;mchn_attrb change?
        jne lb72
        call mchn_attrb
        jmp lb71
;
lb72:   mov file_name[si],al
        cmp al,0dh                          ;CR?
        je lb74
        inc si
        cmp al,08h                          ;BS?
        jne lb73
        dec si
        cmp si,00h
        je lb71
        dec si
        bs
lb73:   cnsl_out al
        jmp lb71
;
lb74:   cmp si,00h                          ;CR only?
        je lb74a
        mov file_name[si],0                 ;put NULL at data end
        open_handle file_name,0
```

```
        mov handle,ax
        jnc lb80                        ;if file not found
        display crlf
        display msg9
lb74a:  jmp lb70
;
;*** send file ***
;
lb80:   display crlf
        aux_out_string nhdr,hdr,intvl   ;file header
lb81:   read_handle handle,buffer,255
        jnc lb82                        ;read error?
        display crlf
        display msg13
        jmp lb90
lb82:   cmp ax,00h                      ;if end of file
        jne lb82a
        jmp lb90
;
lb82a:  mov nbytes_u,ax                 ;data length on buffer
        xor si,si
;
lb83:   mov cx,intvl                    ;time interval
        lb83a:
        loop lb83a
        cmp buffer[si],1ah              ;if end of file (^z)
        jne lb83b
        jmp lb90
lb83b:  cmp buffer[si],0ah              ;LF?
        je lb84
        aux_out buffer[si]
        cnsl_out buffer[si]
        inc si
        cmp si,nbytes_u
        jne lb83
        jmp lb81
;
lb84:   cnsl_out buffer[si]
        inc si
;
        check_kbd_status
        cmp al,00h                      ;empty?
        je lb85
        cnsl_in
        cmp al,02h                      ;break command?
        je lb90
        cmp al,09h                      ;help_key?
        jne lb84a
        display msg1a
lb84a:  cmp al,01h                      ;mchn_attrb change?
        jne lb85
        call mchn_attrb
;
lb85:   call read_232c                  ;clear RS-232C
;
        cmp si,nbytes_u
        jne lb86
        jmp lb81
lb86:   jmp lb83
;
```

```
;*** close handle ***
;
lb90:     close_handle  handle
          display crlf
          display msg11
          display center_f_name
          cnsl_out 07h
lb90a:    display crlf
          display msg11a
          display crlf
          aux_out_string neof2,eof2,intvl
          time_intvl 30
          aux_out_string neof2,eof2,intvl
          jmp lb2a
;
;*******************
;*   end process   *
;*******************
;
lb100:    mov dx,32h                          ;shut off RS-232C
          mov ax,00h
          out dx,ax
;
lb100a:   mov cl,0e1h                         ;disable nihongo kinou
          int 0dch
          set_key_code 17h,key_ins
          set_key_code 18h,key_del
          display crt4
          display crt2                        ;disable bottom line
          mov ah,4ch                          ;end procedure
          mov al,00h
          int 21h
;
code      ends
          end  start
```