FLEX™ NEWSLETTER NO. 3
March 1980

Copyright (c) 1980 by Technical Systems Consultants, Inc.
P.O. Box 2570, West Lafayette, Indiana 47906

Here we go with FLEX Newsletter No. 3. One of the original goals of this newsletter was to permit users to send in tips, suggestions, and ideas they have derived from using Technical Systems Consultants' FLEX based software. Although we have received very few, that goal is still in effect. If you have any ideas of general interest to FLEX users, feel free to submit them.

1) FLEX News

Our family of FLEX users continues to grow. We have just begun distribution of a version of FLEX for "General Use". This FLEX is the same as all other FLEX's except that it is setup and fully documented to allow the user to write his own disk driver routines and terminal I/O routines to adapt to custom hardware. There are three basic hardware requirements to support this version of FLEX as listed here:

1. At least 12K of user RAM at $0000.
2. 8K of RAM for FLEX at $C000 for 6809 or $A000 for 6800.
3. A floppy disk capable of running 256-byte, soft sectors.

The user must write the simple disk driver routines such as READ a single sector, WRITE a single sector, SEEK to a track, SELECT a drive, RESTORE to track 0, etc. He must also write the routines associated with terminal I/O such as INPUT a single character, OUTPUT a single character, exit to MONITOR, check STATUS of input, etc. These routines are all appended onto the core of FLEX to bring FLEX up on the custom hardware. By writing one's own drivers, it is possible to mix disk drive types (such as soft-sectored floppies and hard disks), to adapt to most any monitor, to adapt to a non-standard I/O device, etc. The supplied adaptation manual is almost 100 pages in length and includes sample source listings for all the user supplied routines. This package is available for both 6800 and 6809 on 5 or 8 inch disks at a cost of $150.00 and includes editor and assembler. Drop us a line for more details.

We also now have available a version of FLEX for the 6800 EXORciser with EXORdisk II or III. This means that a standard FLEX disk from an 8 inch SWTPc system can be read and executed on EXORciser hardware and vice versa. The package requires no hardware modifications (with the possible exception of memory readdressing) and uses the standard MDOS ROM boot for booting. The package includes editor and assembler and sells for $150.00. A 6809 EXORciser version will be available soon.

2) New Products

We are always working on new products and have found that most of you are interested in what is coming up.

We are nearing completion of our multi-user, multi-tasking operating system called "UniFLEX™". The initial version will be for the SWTPc S/09 computer system only (the 128K system). This is a full-blown multi-tasking operating system which is modeled very closely after the UNIX operating system developed by Bell Labs (UNIX is a trademark of Bell Labs). It is a very state-of-the-art operating system and is not intended for smaller microcomputer installations where FLEX should suffice. We expect to be shipping early versions by May or June. If you are interested in such a system, write for a four-page document describing some of its features.

Many of you already know that we are developing both a PASCAL compiler and a "C" compiler for the 6809. We expect them both to be completed sometime this summer, probably the PASCAL first. They will both be full, native compilers meaning they run on a 6809 machine and produce 6809 assembler language code as their output. The "C" will be a complete implementation as defined in the "bible", Kernighan and Ritchie's The C Programming Language. The PASCAL will be a standard PASCAL implementation with several extensions. We will attempt to make the PASCAL extensions include as many of the extensions in UCSD PASCAL as possible. Both of the compilers will be available for operation under 6809 FLEX and UniFLEX.

With PASCAL, C, and all UniFLEX software we will be starting a new policy in software pricing. The packages will be sold for a single-end-use purchase price as normal which will include 60 days maintenance. An additional maintenance contract will be available on a yearly basis to provide support, patches, updates, etc. After 60 days, those who have not purchased maintenance will receive no support whatsoever. This sort of arrangement has become essential in order that we may offer these packages at a reasonable purchase price and still be able to afford supporting them.

We are also getting into 68000 software. The initial 68000 software will be a cross-assembler which runs on a 6800 or 6809 FLEX system and produces 68000 object code. This package should be available in the next month or two at a cost of $250.00. We also plan a resident version of UniFLEX, PASCAL, C, and most of the software we currently offer under FLEX.

Other planned projects are a relocating assembler and linking loader for both 6800 and 6809 FLEX. We are working on a 6809 text processor which features proportional spacing and relative paper motion commands, but that package is many months away.

3) Patches and Fixes

Here are several patches for bugs or enhancements to several of our software packages.

a) ALTERING THE NUMBER OF TRACKS IN 6800 FLEX 2.0 NEWDISK

Many people have asked how to change the number of tracks which the 6800 FLEX 2.0 (5") NEWDISK command formats onto disks. In general, the desire is to extend it to 40 tracks for the Wangco or Siemens drives. These changes should be made in the NEWDISK code ONLY if the old values listed match those in your particular version.

Change the following bytes:

```
1. A1EC from 23    to  MAXTRK
2. A245 from 23    to  MAXTRK
3. A286 from 01 54 to  (MAXTRK-1)*10   [10 decimal]
4. A2EA from 23    to  MAXTRK
5. A3BF from 22    to  MAXTRK-1
```

Where MAXTRK is the maximum number of tracks to be formatted. When saving NEWDISK, you should save from $A100 to $A620 with a transfer address of $A100.

For example, a 40-track NEWDISK should be changed as follows:

```
1. A1EC from 23    to  28
2. A245 from 23    to  28
3. A286 from 01 54 to  01 86
4. A2EA from 23    to  28
5. A3BF from 22    to  27
```

SEE LETTERS IN
'68 MICRO RE THIS
PATCH. IT IS INCOMPLETE
SEE ALSO #4, pg. 3

b) 6809 FIND UTILITY

The 6809 FIND Utility (one of the additional utilities available in the 6809 FLEX Utilities package) will not work with consecutive spaces in the search string. This problem may be fixed by altering the source as shown and reassembling.

1. Add the following equates to the "System Interface Definitions":

```
BUFPNT  EQU   $CC14
LSTTRM  EQU   $CC11
```

2. Replace the seven lines of code bounded by FIND1 and FIND2 with:

```
        LDY   BUFPNT
FIND1   LDA   0,Y+    GET CHARACTER
        STA   0,X+    STORE IN BUFFER
        CMPA  #$D     CHECK CHARACTER
        BEQ   FIND2   IF END OF STRING
        CMPA  EOL
        BNE   FIND1   IF NOT END OF STRING
FIND2   LEAY  -1,Y
        STY   BUFPNT
        STA   LSTTRM
        LEAX  -1,X    SAVE LWA+1 OF TARGET
```

c) ILLEGAL LABELS IN 6809 ASSEMBLER

The register names in 6809 assembler language cannot be used as labels or symbols in Technical Systems Consultants' 6809 Assembler. This is in agreement with Motorola's 6809 assembler definition. In most cases there is no check for such an error, so it is up to the user not to use those names as labels. The register names which should not be used are A, B, CC, DP, X, Y, U, S, D, and PC. Early versions of the assembler manual omitted this point.

d) I/O ROUTINES FOR 6800 DEBUG RUNNING WITH SWTBUG

Those using SWTPc's SWTBUG with Technical Systems Consultants' 6800 DEBUG Package may experience problems with the ESCape character stopping output as described. This happens because SWTBUG re-initializes the ACIA after each character through it, thus destroying any characters which may be pending in its buffer. The suggested fix is to provide your own I/O routines instead of using SWTBUG's. The following patch does just that for an ACIA at location $8004.

```
Starting at address $5AE8 place the following bytes:
B6 80 04 46 24 FA B6 80 05 39 36 B6 80 04 85 02 27 F9 32 B7 80 05 39
Then at $4107 put:  5A E8
 and at $410A put:  5A F2
```

e) 6809 FLEX

Early copies of 6809 FLEX have a problem in the File Management System that can cause crashing of the directory in random access files of around 20,000 sectors or larger. Chances are this would never occur on a floppy disk, but if you are using hard disks it could be a problem. The fix is quite simple. There should be a "30 85" (two bytes) at $DCAD or close by. Change those two bytes to "3A 12".

f) 6809 TEXT EDITOR NOT FULLY RELOCATABLE

There are two points in the 6809 Text Editing System (disk version) which prevent it from being fully position independent. If you need to run the editor at some location other than $0000, make the following changes:

Address	Old code	New code
178C	BD 05 BD	17 EE 2E
1809	BD 05 BD	17 ED B1

4) Update Policy

A firm update policy has been established for Technical Systems Consultants, Inc, disk software. If a program was purchased within the past 60 days, an updated version will be supplied free of charge. The customer need only return the original disk and proof of purchase. If the customer has owned the program more than 60 days, an updated version may be obtained at a cost of $10.00. The customer should return the original disk, proof of purchase, and $10.00. In either case, the customer need not return the original disk if he wishes to purchase another at a cost of $10.00 (the program owned more than 60 days would require $20.00 total). The proof of purchase must still be supplied. This policy only applies to corrections to problems found in a program. Completely new versions of a program (such as for a different processor or operating system) must be purchased outright.

5) Version 2.4 of the 6800 Text Processing System

Several minor problems have been detected in version 2.3 of the 6800 Text Processing System for disk. These have been corrected in a new version just released. Owners of the 6800 Text Processing System may wish to exercise the update policy outlined in section 4 above and obtain a copy of this version. Alternatively, we have printed a list of the fixes and how to patch them, though many of the patches will not fit in place. This list of changes may be obtained by writing to us. The 6809 Text Processing System we currently sell is a re-done version of the 6800 one. It already has all the modifications in version 2.4.

6) Zero Sectors Left When Writing

SEE "WILTON HART" LETTER + COMMENTS RE THIS PROBLEM IN '68 MICRO. ALSO, SEE #4, pg. 5

Several people have experienced and called about a peculiarity in the operation of FLEX. It comes about during the writing of a file. The writing of the file is terminated prematurely and subsequent examination of the disk shows zero sectors left in the free chain even though it is known that the disk is not full. This is not really a problem in FLEX, but rather an actual error in reading the disk. The explanation is as follows. The sectors in a FLEX file or in the chain of free or available sectors on the disk are linked together by a pointer in the first two bytes of each sector. Following these pointers or links is how FLEX finds the next available sector in the free chain to be used for adding to a file. Now suppose FLEX is writing or creating a new file. Each time it needs to add another sector to the file, it must be obtained from the free chain. The only way FLEX knows where the next free chain sector is located is to read the link or pointer in the current sector. Now if the sector we are supposed to get from the free chain is unreadable, for whatever reason, we are stuck. We have lost our forward link. You might hastily say to simply skip the unreadable sector and get the next one, but if you think a second you will realize that is impossible. The only way we know where the sector after the unreadable one is located is to read the link or pointer stored in the unreadable sector. Therein lies the problem. There is really nothing that can be done about it since we have no idea where any more available sectors are on the disk. The recourse decided upon in the design of FLEX was to close the file being written and set the number of available sector left on the disk to zero. There's really not much else that can be reasonably done. If this happens to you, the best thing to do is to copy all the files from that disk to a new one and then reformat the disk.

7) Refute of Sort/Merge Review

It is not our custom to disagree with reviews of our products. Indeed, we are usually very pleased to see our software or any 6800/6809 based products get mention in the predominately 8080/Z80 oriented magazines. Our software is not perfect, and it is good if a review points out the weaker aspects of a package so that a prospective customer can be aware of things which might cause him problems. Now if all this sounds like a BUT is coming, you are right. We recently received a review which we felt was totally out of line and which we feel impelled to refute. The November 1979 issue of Interface Age carried an article by Jim Schreier titled, The Sport of Sorting. The article was a review of our 6800 Sort/Merge package. This article is very generous in its compliments of the software itself, but is quite negative in its opinion of the documentation. The latter is, of course, what we disagree with. A couple of quotes from the article follow:

> '...the documentation is written on a "higher level" than some users may feel comfortable with.'
> 'A neophyte to microcomputers may be justly confused by the jargon.'

Our basis for disputing these quotes is the fact that this particular manual was carefully written to avoid such "higher level jargon". Those quotations are probably quite appropriate to several of our other manuals, but the sort/merge manual carefully defines any terms before their first use. This is one of the few manuals which we ever tested on a total neophyte to computers. We handed it to a secretary and had her sit down at a terminal and work through it. Except for some questions about the operation of FLEX itself, there were very few points which she could not understand. Most of those points were changed to be more readable.

To our knowledge, these are the only negative comments ever received on the sort/merge documentation. For example, a review by Dale Puckett of the very same package was printed in the August 1979 issue of '68' Micro Journal. We quote from that article:

> 'DOCUMENTATION. It's outstanding. The manual states in the introduction that it was written with a non-computerist in mind. I agree and would go as far as to say that your secretary will be able to use the system...'

8) Single Drive Copy Utility

For practical use, it is recommended that FLEX (or any disk operating system) be run on at least a two drive system. This allows a user to easily back up his files and to easily create new disks for distribution. There is nothing, however, to keep FLEX from being used on a single drive system. In order to do so, one will need a "single drive copy" program which allows files to be copied from one disk to another with only one drive on the system. This involves alternatively inserting two disks into the drive until the entire file, which may not fit in memory, has been copied. The user can certainly develop his own single drive copy routine or can purchase one from Technical Systems Consultants for $15.00. This includes a two page manual and object code disk. Be sure to specify 8 or 5 inch disk, 6800 or 6809, and include 3% for postage and handling (10% outside US and Canada).

9) Setting Line Width in BASIC

Many users have expressed confusion in setting the maximum line width in our BASIC. There is no BASIC command to do such. BASIC obtains the line width and other environment parameters from FLEX. Thus, to change the line width while running BASIC, you should change the TTYSET line width in FLEX. That value CAN BE CHANGED WHILE RUNNING BASIC. To do so, use the "+" command or the "EXEC" statement to perform a TTYSET command to FLEX. For example, to set a line width of 132 in BASIC, the following statement might be used:

```
110 EXEC, "TTYSET WD=132"
```

Note that the TTYSET command can be used in this manner to set any environment parameter from BASIC.

10) SWTPc Versions of FLEX

While Technical Systems Consultants, Inc, is the author and owner of the FLEX Disk Operating System, we are not the only ones who distribute it. Several companies have been licensed to distribute versions of FLEX for their hardware. One in particular is Southwest Technical Products. The reason for this paragraph is to further point out that the version of FLEX they distribute may not, and in fact is not, identical to what we distribute. Now the actual FLEX program itself is the same except for necessary adaptation to hardware, but the supplied set of utilities is not. For example, SWTPc is supplying 6809 FLEX version 2.6 with their 8" hardware which has several utilities we did not write:

FIX.CMD
RM.CMD
SP.CMD
N.CMD
CLEAN.CMD
WRITPROM.CMD
NEWDISK.T10
P.COR
S.CMD
USEMF.CMD
Y.CMD
PSP.CMD
TOUCH.CMD
PO.CMD
SBOX.CMD
VER.CMD
C4MAT.CMD
READPROM.CMD
UCAL.CMD

The reason for pointing these routines out is that we did not write them and, therefore, cannot support them. If you have questions about these commands, or any others which licensed distributors of FLEX may be supplying, don't call us!

Here is a list of the commands we supply with FLEX. If you have questions about something not listed here, chances are it was written by the supplier of FLEX and not us. You should contact that supplier for advice or support of any such additional commands.

FLEX.SYS
COPY.CMD
DELETE.CMD
P.CMD
BUILD.CMD
DATE.CMD
VERSION.CMD
PRINT.CMD
XOUT.CMD
NEWDISK.CMD
ERRORS.SYS
LIST.CMD
RENAME.CMD
SAVE.CMD
EXEC.CMD
O.CMD
PROT.CMD
QCHECK.CMD
SAVE.LOW
CAT.CMD
ASN.CMD
TTYSET.CMD
APPEND.CMD
JUMP.CMD
LINK.CMD
VERIFY.CMD
I.CMD
PRINT.SYS

Note that certain version of FLEX we distribute (EXORciser FLEX for example) do not have printer spooling implemented and therefore do not include PRINT.CMD or QCHECK.CMD which are used only in conjunction with printer spooling. Also, most versions of FLEX distributed by Technical Systems Consultants include EDIT.CMD and ASMB.CMD.

11) Telex System Installed

To better serve our foreign customers and US industrial concerns, we have installed a Telex system at the company headquarters. Our number is "Telex 276143". The answer-back is "TSC WLAF".

12) MiniFLEX to 6809 FLEX Copy Utility

In this issue of the FLEX Newsletter we include the listing of a utility to copy files from an old 6800 miniFLEX disk directly to a 6809 FLEX disk on a SWTPc minifloppy system. The user simply boots up 6809 FLEX on his 5 inch SWTPc system, inserts a miniFLEX disk in drive one and a 6809 FLEX disk in drive 0, and types the command to copy a file. Complete instructions are given at the beginning of the listing printed later in this newsletter.

13) 6809 Diagnostics Package

We recently placed a nifty, new package of 6809 Diagnostics on the market. The package is comprised of two sets of programs: six memory diagnostics and ten disk repair utilities. The memory diagnostics include a "zeroes" and "ones" test, random pattern test, walking bit tests, dynamic RAM dropout test, and a memory convergence test. All these tests are position independent. The disk repair utilities are aimed at recovering data from or repairing the data on "structurally damaged" diskettes. They have nothing to do with the disk hardware, but allow you to work around problems like CRC errors, crashed directories, accidentally deleted files, etc. Included are 3 diagnostic utilities which report unreadable sectors and structural inconsistencies among the files on a disk, 2 utilities for recovering data when the directory on a diskette is not readable, a utility to remove bad or intermittant sectors from the free space, a program to retrieve deleted files from the diskette free chain, a single-sector read/write/modify routine, and a copy utility which ignores CRC errors.

The 100 page manual goes further than simply describing how to use the utilities. It describes the types of problems which might occur, why they occur, how to detect them, how to decide which of the diagnostic utilities to attack the problem with, how to use the utilities, and in some cases what to do if the diagnostics cannot help. Many examples are given, and a "Case Study" reviews 6 typical cases of disk problems and how they were solved.

This package is available only for 6809 FLEX. The price of $75.00 includes the user's manual and the object code on a 5 or 8 inch diskette.

```
                 *
                 * MINIFLEX TO FLEX 9.0 MOVE UTILITY
                 *
                 * COPYRIGHT (C) 1979 BY
                 * TECHNICAL SYSTEMS CONSULTANTS, INC.
                 * BOX 2570, W. LAFAYETTE, IN 47906
                 *
                 *
                 * INSTRUCTIONS:
                 *
                 * THIS UTILITY PERMITS THE USER TO COPY NAMED FILES
                 * FROM A 6800 MINIFLEX DISK TO A 6809 FLEX DISK.
                 * INSTRUCTIONS FOR USE ARE AS FOLLOWS:
                 *
                 * 1) BOOT UP FLEX 9.0.
                 * 2) INSERT A 6809 DISK WITH A COPY OF THIS UTILITY
                 *    ON IT INTO DRIVE #0.
                 * 3) INSERT THE MINIFLEX DISK IN DRIVE #1.
                 * 4) TYPE A COMMAND OF THE FOLLOWING FORM:
                 *         +++MOVE9,MINIFILE,NINEFILE
                 *    WHERE "MOVE9.CMD" IS THE NAME OF THIS UTILITY,
                 *    "MINIFILE" REPRESENTS THE NAME OF THE FILE ON
                 *    THE MINIFLEX DISK, AND "NINEFILE" REPRESENTS
                 *    THE NAME OF THE FILE TO BE CREATED ON THE
                 *    FLEX 9.0 DISK.  THE DEFAULT EXTENSION FOR BOTH
                 *    FILENAMES IS ".TXT".  DRIVE NUMBERS NEED NOT
                 *    BE SPECIFIED AND IF SPECIFIED WILL BE IGNORED
                 *    SINCE DRIVE #0 MUST BE FLEX 9.0 AND DRIVE #1
                 *    MUST BE MINIFLEX.
                 *
                 * IF IT BECOMES NECESSARY TO RESET THE CPU DURING
                 * OPERATION OF THIS UTILITY FOR ANY REASON, BE SURE
                 * TO RE-BOOT FLEX 9.0 BEFORE CONTINUING WITH ANY
                 * DISK OPERATIONS!  THIS IS DUE TO THE FACT THAT
                 * THE 6809 DRIVERS ARE ALTERED WHEN READING THE
                 * MINIFLEX DISK AND MUST NOT BE LEFT IN THAT STATE.
                 *


                 * EQUATES

           CD03  WARMS   EQU    $CD03
           D406  FMS     EQU    $D406
           D403  FMSCLS  EQU    $D403
           CD1E  PSTRNG  EQU    $CD1E
           CD24  PCRLF   EQU    $CD24
           CD2D  GETFIL  EQU    $CD2D
           CD33  SETEXT  EQU    $CD33
           CD3F  RPTERR  EQU    $CD3F
           CC2B  MEMEND  EQU    $CC2B
           C840  FCB     EQU    $C840
```

```
                          ****************************************
                          * START OF PROGRAM
                          ****************************************

 C100                             ORG     $C100

 C100 20    09            MOVE    BRA     MOVE1

 C102 01                          FCB     1         VERSION #1
 C103                     TRACK   RMB     1
 C104                     SECTOR  RMB     1
 C105                     TEMP    RMB     2
 C107                     SCTRL1  RMB     2
 C109                     SCTSID  RMB     2

                          * GET MINIFLEX FILENAME

 C10B 8E    C208          MOVE1   LDX     #MFCB
 C10E BD    CD2D                  JSR     GETFIL
 C111 1025 00B3                   LBCS    ERROR
 C115 86    01                    LDA     #1
 C117 BD    CD33                  JSR     SETEXT
 C11A 86    01                    LDA     #1
 C11C A7    03                    STA     3,X       FORCE DRIVE #1

                          * GET FLEX 9.0 FILENAME & OPEN IT

 C11E 8E    C840                  LDX     #FCB
 C121 BD    CD2D                  JSR     GETFIL
 C124 1025 00A0                   LBCS    ERROR
 C128 86    01                    LDA     #1
 C12A BD    CD33                  JSR     SETEXT
 C12D 6F    03                    CLR     3,X       FORCE DRIVE #0
 C12F 86    02                    LDA     #2
 C131 A7    84                    STA     0,X
 C133 BD    D406                  JSR     FMS
 C136 1026 008E                   LBNE    ERROR
 C13A 86    FF                    LDA     #$FF
 C13C A7    88 3B                 STA     59,X

                          * FIND SECTOR BYTES IN DRIVERS

 C13F CC    C600                  LDD     #$C600    GET "LDB #0" PATTERN
 C142 8E    DE20                  LDX     #$DE20    APPROX. START OF DRIVER CODI
 C145 10A3 80             FIND1   CMPD    0,X+
 C148 26    FB                    BNE     FIND1
 C14A BF    C107                  STX     SCTRL1    SAVE LOCATION OF 1ST ONE
 C14D CC    C10A                  LDD     #$C10A    GET "CMPB #10" PATTERN
 C150 8E    DE20                  LDX     #$DE20    APPROX. START OF DRIVER CODI
 C153 10A3 80             FIND2   CMPD    0,X+
 C156 26    FB                    BNE     FIND2
 C158 BF    C109                  STX     SCTSID    SAVE SECTORS/SIDE LOCATION
```

```
                        * FIND THE ADDRESS OF THE MINIFLEX FILE

C15B CC   0004                  LDD   #$0004     DIRECTORY STARTS AT 00-04
C15E FD   C103                  STD   TRACK
C161 FC   C103         MOVE2    LDD   TRACK
C164 1027 0097                  LBEQ  FNFERR     ERROR IF AT END OF DIRECTORY
C168 17   0066                  LBSR  READMS     READ A SECTOR
C16B 8E   C250                  LDX   #MFCB+72   POINT TO 1ST ENTRY
C16E 108E C20C         MOVE3    LDY   #MFCB+4    POINT TO NAME
C172 BF   C105                  STX   TEMP
C175 A6   84                    LDA   0,X        GET 1ST CHARACTER
C177 1027 0084                  LBEQ  FNFERR     IF 0, FILE NOT FOUND
C17B 2B   0D                    BMI   MOVE5      SKIP IF DELETED ENTRY
C17D C6   0B                    LDB   #11
C17F A6   80           MOVE4    LDA   0,X+       COMPARE NAMES
C181 A1   A0                    CMPA  0,Y+
C183 26   05                    BNE   MOVE5
C185 5A                         DECB
C186 26   F7                    BNE   MOVE4
C188 20   0D                    BRA   MOVE6
C18A BE   C105         MOVE5    LDX   TEMP
C18D 30   88 18                 LEAX  24,X       POINT TO NEXT ENTRY
C190 8C   C2C6                  CMPX  #MFCB+190 AT END OF SECTOR?
C193 25   D9                    BLO   MOVE3      NEXT ENTRY IF NOT
C195 20   CA                    BRA   MOVE2      NEXT SECTOR IF SO

                        * FOUND THE FILE ENTRY

C197 BE   C105         MOVE6    LDX   TEMP       POINT TO START OF ENTRY
C19A EC   0D                    LDD   13,X       GET STARTING DISK ADDRESS
C19C FD   C103                  STD   TRACK      SAVE IN TRACK & SECTOR

                        * READ MINIFLEX FILE INTO FLEX 9.0 FILE

C19F FC   C103         MOVE7    LDD   TRACK      GET TRACK & SECTOR
C1A2 27   18                    BEQ   GOTALL     SKIP IF END OF FILE
C1A4 17   002A                  LBSR  READMS     READ SECTOR
C1A7 108E C24C                  LDY   #MFCB+68   POINT TO DATA
C1AB C6   7C                    LDB   #124
C1AD 8E   C840                  LDX   #FCB
C1B0 A6   A0           MOVE8    LDA   0,Y+
C1B2 BD   D406                  JSR   FMS
C1B5 26   11                    BNE   ERROR
C1B7 5A                         DECB
C1B8 26   F6                    BNE   MOVE8
C1BA 20   E3                    BRA   MOVE7

                        * FILE IS TRANSFERRED; CLOSE IT AND EXIT

C1BC 8E   C840         GOTALL   LDX   #FCB
C1BF 86   04                    LDA   #4
C1C1 A7   84                    STA   0,X
C1C3 BD   D406                  JSR   FMS
C1C6 27   03                    BEQ   EXIT
```

```
                                * ERROR AND EXIT ROUTINES

 C1C8 BD    CD3F      ERROR    JSR      RPTERR

 C1CB BD    D403      EXIT     JSR      FMSCLS
 C1CE 7E    CD03               JMP      WARMS       BACK TO FLEX

                                * READ A SINGLE MINIFLEX SECTOR

 C1D1 8E    C208      READMS   LDX      #MFCB
 C1D4 ED    88 1E              STD      30,X        SET SECTOR FOR READING
 C1D7 86    80                 LDA      #128
 C1D9 A7    9F C107            STA      [SCTRL1]    ADJUST DRIVERS FOR 128
 C1DD A7    9F C109            STA      [SCTSID]    ENSURE SIDE 0
 C1E1 86    09                 LDA      #9          READ SINGLE SECTOR FUNCTION
 C1E3 A7    84                 STA      0,X
 C1E5 BD    D406               JSR      FMS         DO READ
 C1E8 34    01                 PSHS     CC          SAVE CONDITION CODES
 C1EA 6F    9F C107            CLR      [SCTRL1]    ADJUST DRIVERS FOR 256
 C1EE 86    0A                 LDA      #10         RESET SECTORS PER SIDE
 C1F0 A7    9F C109            STA      [SCTSID]
 C1F4 35    01                 PULS     CC
 C1F6 26    D0                 BNE      ERROR       SKIP IF AN ERROR
 C1F8 EC    88 40              LDD      64,X        GET FORWARD LINK
 C1FB FD    C103               STD      TRACK       SAVE IT
 C1FE 39                       RTS

                                * FILE NOT FOUND ERROR

 C1FF 8E    C840      FNFERR   LDX      #FCB
 C202 86    04                 LDA      #4          FILE NOT FOUND ERROR CODE
 C204 A7    01                 STA      1,X         INTO ERROR STATUS BYTE
 C206 20    C0                 BRA      ERROR       GO REPORT IT


                                * FCB FOR MINIFLEX FILE

 C208                 MFCB     RMB      192


                               END      MOVE

0 ERROR(S) DETECTED
```