ST-2900 FLEX CONVERSION PACKAGE

User's Manual

Sardis Technologies
2261 East 11th Avenue
Vancouver, B.C.
Canada   V5N 1Z7

Although much effort has been made to ensure the accuracy of the software and documentation, David C. Wiens and Sardis Technologies disclaim any and all liability for consequential damages, economic loss, or any other injury arising from or on account of the use of, possession of, defect in, or failure of the supplied material.

This manual last revised September 7, 1984.

Adapting FLEX for the ST-2900

Although Sardis Technologies is not presently selling a configured version of FLEX that will boot directly, you can create such a bootable disk in a matter of minutes, with no programming whatsoever. The ST-2900 FLEX Conversion Package does the work for you. (FLEX is a trademark of Technical Systems Consultants, hereinafter referred to as TSC.)

Besides the ST-2900 FLEX Conversion disk, you need a copy of FLEX. The versions that have so far been verified to be compatible with the ST-2900 conversion package are:

1) General Version of 6809 FLEX, version 3.01, from TSC
2) FLEX 9.0 for the SWTPc system, from TSC - no version given, but was purchased in March 1980.

Many other versions should also run without changes. If your version of FLEX has problems running on the ST-2900 system, contact us and we will try to help (but no guarantees).

The ST-2900 Conversion package supplies its own console, printer, and disk driver routines, so it doesn't matter if the FLEX version you choose does or doesn't support the disk configuration you need.

Booting Up For The First Time

1) Write protect the distribution diskette containing the FLEX operating system, as well as the one with the ST-2900 FLEX Conversion Package.
2) After the system is powered up, insert the Conversion Package disk into drive 0, then type "D F". Follow the prompts (respond by typing a carriage return). When asked to choose between FLEX.COR and FLEX.SYS, specify "1" for FLEX.COR. For now, the "Patches" disk is the same as the "Conversion" disk. If you get the "BOOT ERR" or "ERROR DETECTED" message, try again from the beginning of this step.
3) When you get the "Date?" prompt, enter the date (eg. 06,30,84), press the carriage return key, and you should soon see the "+++" FLEX-is-ready prompt.
4) Execute the "DSKSET" command to configure the disk drivers for your particular configuration. (DSKSET is explained elsewhere in this manual)
5) If you haven't already studied the FORMAT command (explained elsewhere in this manual), do so first.
6) Insert the FLEX Conversion disk into drive 0, then type "FORMAT d", where "d" is the drive number where you want to format a blank disk. Specify DOUBLE-DENSITY=N, DOUBLE-SIDED=N, LOWER-DENSITY=N. After you have responded to all the prompts, the system will take approx. 1/2 minute to format the disk.
7) Use FLEX's COPY command to copy the entire Conversion disk to the newly formatted disk. Now put the original Conversion disk away into a safe place.
8) Use the FLEX "LINK" command to link to XLOADER.SYS on the new conversion disk.

Creating A New System Disk

1) In order to speed up the booting process next time, as well as to reduce wear and tear on the distribution disks, you will want to create a system disk. This disk will contain a FLEX.SYS file configured for the ST-2900, a STARTUP file, and various utilities. The next 6 steps describe how to create this disk.
2) Use the FORMAT command to format a blank disk. Specify LOWER-DENSITY=N.
3) Execute the "SYSGEN" command from the conversion disk and follow its prompts. SYSGEN is described elsewhere in this manual.
4) Copy the desired utilities from the FLEX distribution disk to the new system disk.
   a) the most important utilities to copy over at this time are: COPY, CAT, LIST, DELETE, RENAME, EXEC, ASN, TTYSET, P.
   b) several files on the FLEX disk are not useable on the ST-2900 system and should NOT be copied over: FLEX.SYS, NEWDISK.TXT, NEWDISK.CMD, PRINT.SYS.
5) Copy the desired files from the Conversion distribution disk to the new system disk. The most important files to copy over at this time are: DSKSET.CMD, PRTSET.CMD, MCPRINT.SYS, FORMAT.CMD, DCHECK.CMD.
6) Study the "BUILD" and "STARTUP" commands described in the FLEX Users Manual, then create a STARTUP and CONFIG file on the new system disk. For example:

```
 STARTUP:
   EXEC CONFIG
 CONFIG:
   DSKSET DR=0,ST=20,DS=N,MT=40,DR=1,VL=Y,ST=6,MT=80
   TTYSET BE=08
   ASN W=1
   GET MCPRINT.SYS
   PRTSET LM=10,BR=1200
```

7) To test the new system disk, press the reset button, then follow the instructions in the following section.

Booting From A Configured System Disk

1) You must have already created a system disk as described in the previous section. Note -- the system disk must have been formatted by the ST-2900 "FORMAT" command, not by any other format utility or on another brand of computer.
2) Insert the system disk into drive 0, then type "D F". After a few seconds you should get the "Date?" prompt. When you have entered the date, the STARTUP file will automatically be executed, followed by the display of the FLEX "+++" prompt.
3) If you get the "BOOT ERR" or "ERROR DETECTED" message, try again from the beginning of step #2. If you get the "NOT LINKED" message, run the TSC "LINK" command to link to the "FLEX.SYS" file, then rerun step #2.

New Utility Commands

The next few pages describe additional commands included on the FLEX Conversion disk.

A few will run on any FLEX system; the rest will only work on the ST-2900.

BLIST

The BLIST command is similar to the LIST command, in that it lists the contents of text files to the screen or printer. However, it uses a very large buffer to reduce wear and tear on the disk drives, especially valuable when a slow printer is being used. Pageination and line numbers are not supported in this version. Its syntax is:

```
    BLIST,<file spec>
```

where <file spec> is the name of the text type file to list (an extension of .OUT is assumed unless otherwise specified). BLIST may be preceded with the "P" command. The "escape character" feature of FLEX (refer to TTYSET) is active. BLIST and its buffer occupy most of $0100-$7FFF of memory.

CHECKSUM

The CHECKSUM command reads any disk file and computes a checksum on it. This is a quick way of determining whether or not two copies of a program or text file are identical versions.

Another way of using it is, every time you make changes to a program or other file, to calculate the checksum of the new version and write it in a log. Later, if you want to determine which version a particular copy is, calculate the checksum on it and look it up in the log.

The syntax of the command is:

```
    CHECKSUM,<file spec>
```

where <file spec> is the name of the file on which to calculate the checksum (the extension defaults to .BIN). The 2 byte checksum is displayed as a four digit hexadecimal number.

Another possible use for this command is to read a file to see if any sectors in it have read errors.

DCHECK

The DCHECK command lets you check 3 disk related values -- motor-on hold time, delay from "motor-on" to "ready" signal, and disk drive rotational speed -- so you can determine if any adjustments are necessary.

Its syntax is:

```
    DCHECK
```

When called, it prompts "ENTER COMMAND - ". Four sub-commands are available:

a) "H" - starts the drive motors so you can manually (with a wristwatch) time how long the motors remain on before they turn off. The motors must be off when you call this command, otherwise you will get the "*NOT READY" message. Trimpot R2 on the FDC board sets this value from approx. 1 to 15 seconds.

b) "D" - measures the delay from the time the "motor-on" signal is activated until the "ready" signal to the 1793 becomes true. The time is displayed in milliseconds (assumes that crystal Y2 on the CPU board is 16 MHz.) The drive motors must be off when you call this command, otherwise you will get the "*NOT READY" message. Trimpot R1 on the FDC board sets this value from approx. 0.1 to 1.25 seconds. Adjust this value to the longest motor start time of any of your drives.

c) "S d" - displays the rotational speed of drive "d" (assumes that Y2 on the CPU board is 16 MHz). A soft sectored disk must be in the selected drive, and the drive door closed, otherwise you will receive a "*TIME-OUT ERROR". The data on the disk is not affected. The speed is displayed in rpm (eg. 299.5) and the display continues to be updated with new readings until any key is pressed on the keyboard. Acceptable values are between 297.0 and 304.0 rpm but the closer to 300.0 the better. Refer to the OEM or maintenance manual for your particular disk drive on how to adjust its speed.

d) "Q" - quits DCHECK and returns to FLEX.

DSKERLOG.OVR

The DSKERLOG.OVR overlay is not a regular command. When activated, it adds detailed error logging facilities to the disk driver routines. To activate or reset it:

```
    +++GET DSKERLOG.OVR
```

Three memory locations are of interest. The address of the beginning of the error log table is at $F00C-$F00D. The pointer to the next available entry in the table is at $F00E-$F00F. When the table becomes full, additional errors are not logged. Reloading DSKERLOG.OVR will reset the pointer. To completely disable DSKERLOG so the memory space it occupies ($E000-$E34F) can be used for other purposes, set the value in $F00A-$F00B to 0000.

Each entry in the log consists of 8 bytes:

Byte 0 - type of error. The high order bit (bit 7) indicates whether the error was detected by the 1793 chip ("0") or whether it was a time out error ("1"). The 3 low order bits indicate which routine received the error.
1 - read sector
2 - write sector
3 - verify sector
4 - (reserved for future use)
5 - seek
Byte 1 - disk drive number
Byte 2 - track number
Byte 3 - sector number
Bytes 4 & 5 - buffer address (not always valid for SEEK)
Byte 6 - status code from 1793 (bit 2 is always set if time-out error, regardless of what the 1793 status was)
Byte 7 - (reserved for future use)

DSKSET

The DSKSET utility lets you set up the disk drivers to handle various disk drive configurations, without machine language patching. It can also display the current configuration. Its syntax is:

```
   DSKSET[,<parameter list>]
```

where <parameter list> is a list of 2 character parameter codes, each followed by an equals sign "=" and by the value being assigned. Each parameter should be separated by a comma. All parameters following a DR=d parameter apply only to that specified drive, until the next DR=d parameter. If no DR parameter is specified, drive 0 is assumed. If no parameter list is specified at all, the existing values of all parameters will be displayed. Some examples are:

```
  +++DSKSET
  +++DSKSET ST=12,DS=Y,RE=0
  +++DSKSET DR=1,VL=Y,ST=20,DR=2,VL=Y,MT=35
```

The first example displays the existing values of all parameters. The next line sets drive number 0's stepping rate to 12 msec. and the double-sided flag to "yes", and resets the read-sector error counter to zero. The last example flags drive 1 as valid with a stepping rate of 20 msec., and also flags drive 2 as valid with only 35 tracks.

Note -- FORMAT also updates some of these values. More specifically, the media values for the drive containing the disk being formatted are updated as per the responses to FORMAT's prompts re double/single density, lower density Y/N, and the TM value is set to the TD value.

The DSKSET command is serially re-usable and position independent, so could be made memory resident, if desired.

Here is a list and description of all DSKSET parameters:

DR=d   select DRive

Specifies to which drive the following parameters apply.

VL=Y or VL=N   VaLid drive

This sets the drive to be active or not active. All drive numbers that have no corresponding disk drive attached should be specified as invalid. Immediately after FLEX has been booted up, only drive 0 is flagged as valid.

ST=dd   STepping rate (msec.)

This parameter specifies the stepping rate for that drive. The four values allowed are 6, 12, 20, 30.

DS=Y or DS=N   Double Sided drive

This specifies whether the disk drive itself is capable of double-sided operation, or not.

TD=48 or TD=96   Track density of Drive (tpi)

Specifies whether the disk drive steps at 48 tpi (tracks per inch) or 96 tpi.

MT=ddd   Maximum Track allowed

The decimal value "ddd" may be a number between 1 and 255, representing the maximum number of tracks the disk drive supports. Typical values are 35, 40, or 80.

PT=ddd   first write Precompensated Track

The decimal value "ddd" may be any number from 0 to 127, representing the lowest numbered physical track number that is to be write precompensated. If this value is greater than the highest track number of the drive, then write precompensation is in effect disabled for that drive.

PS=N or PS=Y   Precompensate Single density tracks

This determines whether a single density track in the above specified range is also to be write precompensated or whether only double density tracks are.

TM=48 or TM=96   Track density of Media (tpi)

Specifies whether the diskette currently inserted is formatted with 48 or 96 tpi.

Density

The density of a disk is automatically determined by the system so cannot be updated by DSKSET. When a new diskette is inserted, this value will not change until a track other than track 0 is read or written.

SS=dd   Sectors per side of Single density track

This parameter specifies how many sectors are on one side of one single density track of the diskette currently inserted. Valid values are 9 or 10. The TSC standard is 10. If this value does not match the actual format of the diskette in that drive, the system will look for some sectors on the wrong side of the disk and will never find them.

SD=dd   Sectors per side of Double density track

This parameter specifies how many sectors are on one side of one double density track of the diskette currently inserted. Valid values are 16 to 18. The TSC standard is 18. If this value does not match the actual format of the diskette in that drive, the system will look for some sectors on the wrong side of the disk and will never find them.

RE=ddddd   Read Error count

This parameter sets the read error counter to a specific value, usually back to zero. The counter is incremented every time an error is detected when attempting to read a sector from disk. Switching back and forth between single density and double density disks (in the same drive) will result in many "soft" read errors.

WE=ddddd   Write Error count

This parameter sets the write error counter to a specific value, usually back to zero. The counter is incremented every time an error is detected when attempting to write a sector to disk.

VE=ddddd   Verify Error count

This parameter sets the verify error counter to a specific value, usually back to zero. The counter is incremented every time an error is detected when attempting to verify a sector just written.

A sample display is:

```
DRIVE CHARACTERISTICS       DR      0     1     2     3
                                    --    --    --    --
 VALID (Y/N)                VL      Y     Y     Y     N
 STEP RATE (msec.)          ST     12     6    30
 DOUBLE SIDED (Y/N)         DS      N     Y     N
 48/96 TPI                  TD     48    96    48
 MAXIMUM TRACKS             MT     40    80    35
 FIRST PRECOMP TRACK        PT     80    43    80
 PRECOMP SD TRACK           PS      N     N     N

MEDIA CURRENTLY INSERTED

 48/96 TPI                  TM     48    96    48
 DENSITY (S/D)                      D     D     S
 SECTORS/SIDE (SD TRACK)    SS     10    10     9
 SECTORS/SIDE (DD TRACK)    SD     18    18    16

ERROR LOGGING

 READ ERRORS                RE        12
 WRITE ERRORS               WE         0
 VERIFY ERRORS              VE         1
```

FORMAT

FORMAT is a replacement for the FLEX "NEWDISK" command. Its syntax is similar:

```
FORMAT,<drive>
```

where <drive> is a single digit decimal number (from 0 to 3) specifying which drive will contain the disk to be formatted.

Most of the prompts are self-explanatory and are answered with either a "Y" or "N". At the end of the prompting session, but before formatting begins, you will be given another chance to abort or continue. Several prompts that need more explanation are:

a) LOWER-DENSITY FORMAT? - Usually you will respond with "N" to select the TSC standard of 10 sectors per track for single density and 18 sectors per track for double density. If you respond with "Y", you will get fewer sectors per track (9 for single, 16 for double density) with longer gaps between each, similar to the IBM 3740 standard.

b) SWTPC FORMAT? - This prompt only has significance if the disk is to be read or written on another system. Most systems that use 1791 or 1793 controller chips (such as the ST-2900) can read disks formatted with a response of "Y" as well as those formatted with "N" (as long as the disk driver routines disable the side flag compare). Systems using the 1795 or 1797 are more fussy. Manufacturers such as GIMIX properly implemented the side flag -- use a response of "N" for disks to be read on their systems. Others, such as Southwest Technical Products Corp. (SWTPc) used the side flag to select single vs double density -- respond with "Y" to the prompt for disks to be read on their systems.

c) VOLUME NAME - the name must follow standard file name rules.

d) VOLUME NUMBER - any number from 0 through 65535.

After the last prompt, the formatting process will take approximately 34 seconds for a single sided, single density 40 track disk, to approx. 58 seconds for double sided, double density 40 tracks. Defective sectors will slow down the process. Any sector that is deemed "bad" will be reported as "BAD SECTOR AT TT SS" where "TT SS" is the track/sector number in hexadecimal.

If there any bad sectors on track zero, or if the very first or very last sector in the data chain is bad, or if 50 or more sectors have errors (soft or hard) in the verify pass, the formatting will be aborted.

If you receive the message "ERROR ON WRITE TRACK", check that drive's rotational speed with the DCHECK command -- it is probably spinning too fast.

NOTE:

- FORMAT erases the contents of FLEX's command line buffer, so any commands specified after FORMAT, on the same line, will be ignored.
- FORMAT does not support double stepping, so 96 tpi drives will format disks only at 96 tpi.
- 35, 40, and 80 tracks can be formatted.
- FORMAT may not be run while the print spooler is active.

A sample session would look like this:

```
+++FORMAT 1
ARE YOU SURE?Y
35 TRACKS?N
40 TRACKS?N
80 TRACKS?Y
LOWER-DENSITY FORMAT?N
DOUBLE DENSITY?Y
DOUBLE SIDED?Y
SWTPC FORMAT?Y
VOLUME NAME - TEST
VOLUME NUMBER - 1
INSERT DISK TO BE FORMATTED INTO DRIVE 1
 (C)ontinue, or (A)bort:C
WRITE PASS DONE
VERIFY PASS DONE
FORMATTING COMPLETE
SECTORS FORMATTED = 2844
```

LOADO

The LOADO command is used to load a binary file from disk into memory. An optional address offset lets you load the program into a different location than specified by the file. Its syntax is:

```
    LOADO,<file spec>[,<offset>]
```

where <file spec> is the name of a binary file to be loaded (extension defaults to .BIN), and <offset> is an optional 4 digit hexadecimal address displacement. Refer to the FLEX Programmer's Manual for a description of the offset field (location $CC1B-$CC1C) and the LOAD routine (address $CD30), as these are used by LOADO.

Some examples are:

```
+++LOADO ABCDE.CMD
+++LOADO WXYZ.1,E000
```

The first example loads binary file "ABCDE.CMD" from the work drive into those memory locations specified in the file. The second example loads binary file "WXYZ.BIN" from drive 1 into memory, but at locations $2000 lower than specified. (Eg., a byte destined for $9374 is loaded into $7374 instead, because $9374 + $E000 = $7374 when the result is truncated to 16 bits.)

Note -- LOADO resides in FLEX's Utility Command Space (starting at $C100); don't use it to load another program into this area.

PATCHES.OVR

The PATCHES.OVR overlay is not a regular command. It provides a way to apply changes to the FLEX operating system and its driver routines at initial boot time. Patches cannot be made to the FLEX.SYS file by appending to it another file containing the patches (the boot process ignores all code after the first transfer address encountered). The PATCHES.OVR file provides a means of inserting user provided code in front of that transfer address.

The PATCHES.OVR file is accessed at two different times. One is when you boot up from the ST-2900 FLEX Conversion disk. The other is during the SYSGEN command when the FLEX.SYS file is being created.

To add your own custom overlay code, edit the supplied PATCHES.TXT file, then use an assembler to create a new PATCHES.OVR file, which can reside on any disk. Next time you boot up from the Conversion disk or use the SYSGEN command, and you get the "INSERT PATCHES DISK INTO DRIVE 0" prompt, insert the disk containing the new PATCHES.OVR file.

PRTSET

The PRTSET utility command lets you change several parameters affecting the printer driver (especially for serial printers), without any machine language patching. It can also display the current configuration. Its syntax is:

```
    PRTSET[,<parameter list>]
```

where <parameter list> is a list of 2 character parameter codes, each followed by an equals sign "=" and by the value being assigned. Each parameter should be separated by a comma. If no parameter list is specified at all, the existing values of all parameters will be displayed. Some examples:

```
   +++PRTSET
   +++PRTSET LM=12,BR=300
```

The first example displays the existing values of all parameters. The second example sets the left margin width to 12, and the baud rate of the serial printer port to 300 baud.

The PRTSET command is serially re-usable and position independent, so could be made memory resident if desired.

WARNING - before sending graphics or multi-byte command codes to the printer, the NC, NL, NF, and LM values should be set to zeroes, otherwise the printer driver will insert nulls and spaces that will mess up the data.

Note - PRTSET's update mode is not allowed while the printer spooler is active.

Here is a list and description of all PRTSET parameters:

Driver Name

This value is updated by the printer driver itself, and cannot be updated by PRTSET. To change the active printer driver, use the GET command to load the new driver (eg. "+++GET MSPRINT.SYS").

NC=ddd   Nulls after Carriage return

The decimal value "ddd" specifies the number of nulls (0-255) to be sent to the printer after each carriage return ($0D) character. This parameter is ignored by MPPRINT.SYS and MSPRINT.SYS.

NL=ddd   Nulls after Line feed

The decimal value "ddd" specifies the number of nulls (0-255) to be sent to the printer after each line feed ($0A) character. This parameter is ignored by MPPRINT.SYS and MSPRINT.SYS.

NF=ddd   Nulls after Form feed or vertical tab

The decimal value "ddd" specifies the number of nulls (0-255) to be sent to the printer after each form feed ($0C) or vertical tab ($0B) character. This parameter is ignored by MPPRINT.SYS and MSPRINT.SYS.

LM=ddd   Left Margin width

The decimal value "ddd" (0-255) specifies how many "space" ($20) characters are to be sent to the printer after each carriage return ($0D) character. Since many printers don't allow paper positioning or settable margins, this feature lets you create a left margin under software control. Note - if the print head is sitting to the left of this new margin and the first printable character is not preceded by a carriage return character, then the first line will not obey the LM value. The BLIST command handles first lines properly, however.

BR=ddddd   Baud Rate of printer port

This parameter sets the baud rate of the serial printer port (Port B). The seven values allowed are 110, 300, 600, 1200, 2400, 4800, 9600. This parameter is, of course, ignored by MPPRINT.SYS.

A sample display is:

```
               PRTSET

PRINTER PARAMETERS

 DRIVER NAME                MCPRINT.SYS
 NULLS AFTER CR       NC      5
 NULLS AFTER LF       NL      0
 NULLS AFTER FF/VT    NF     40
 LEFT MARGIN WIDTH    LM     10
 BAUD RATE            BR   1200
```

PRINTER DRIVERS - MSPRINT.SYS, MCPRINT.SYS, MPPRINT.SYS

This package includes three printer drivers, two for printers connected to the second serial port on the CPU board, and one for a printer hooked up to one of the parallel ports on the FDC board. (Note, the parallel ports are currently not completely compatible with Centronics parallel interface type printers, as no buffers are provided on the FDC board.)

You can either rename the printer driver you will be using to "PRINT.SYS" so the "P" command can find it, or leave the name as is and load it via the "GET" command before the first time you use the "P" command. (If "P" detects that a printer driver is already loaded, it will not look for a "PRINT.SYS" file.)

The source code for all three drivers is provided via the MSPRINT.TXT, MCPRINT.TXT, and MPPRINT.TXT files.

Also refer to the "PRTSET" command.

SYSGEN

SYSGEN is used to create a system disk containing a FLEX.SYS file that is fully configured for the ST-2900 system. Its syntax is:

```
  SYSGEN,<drive>
```

where <drive> is a single digit decimal number specifying which drive will contain the new system disk. SYSGEN's prompts are similar to what you got when you first booted up the system from the Conversion disk.

The separate prompt for a "PATCHES" disk lets you make changes to the PATCHES.TXT file and create a new "PATCHES.OVR" file on another disk.

The FLEX "LINK" command does not need to be used on the new system disk as SYSGEN performs that function as well.

CHANGES TO TSC'S FLEX MANUALS

1) The FLEX manuals state that, after using the MON command, or after pressing the reset button, to re-enter FLEX by jumping to location $CD03. Do NOT do this on the ST-2900. Refer to the descriptions of the "J", "F W", and "F U hhhh" commands in the ST-MON manual.
2) FLEX manuals talk about "automatic drive searching" in several places:
   a) the description of the "ASN" command in the FLEX User's Manual
   b) the description of FMS function 20 "Find Next Drive" in the FLEX Programmer's Manual
   c) the description of the "Check Drive Ready" disk driver routine in the FLEX Programmer's Manual

   The ST-2900 system, with 5" drives attached, functions (more or less) like TSC's description of a system with 8" drives. All four drives will be searched, and the system will not hang up if a drive is empty.
3) If automatic drive searching is not invoked (ie. ASN has set the system and/or work drive to a specific drive) or an explicit drive number is specified in a command, operation is slightly different. Attempting to to access a drive not containing a disk will NOT hang up the system forever (or until a disk is inserted and the door closed) as on most other 5" systems, so do NOT press the reset button. However, the system will take as long as one minute before timing out with an error message, so be patient!
4) There are three routines defined in newer versions of FLEX that were missing in the earliest versions:
   a) $D3E5-$D3E6 contains an address pointing to a routine that inputs a character from the console keyboard without echoing it. It was the omission of this routine that cause the present "mess" with so many software packages requiring patching to run on systems that don't have an ACIA at $E004.
   b) $DE18-$DE1A has a jump instruction to a "warm start" disk driver subroutine that is called during the FLEX warmstart procedure.
   c) $DE1B-$DE1D has a jump instruction to a "SEEK to track" disk driver subroutine.

   The ST-2900 FLEX Conversion Package supports all three routines, even for old versions of FLEX. Note however, if you are using one of these early versions of FLEX that the "warmstart" disk driver routine will never be called by FLEX.
5) The disk driver jump table at $DE00-$DE1D must never be changed on the ST-2900. If you want to add another set of disk drivers (for hard disk, RAM-disk, etc.), instructions are given in the FLEX I/O source code package, available from us at modest cost.

CONTENTS OF THE ST-2900 FLEX CONVERSION DISKETTE

XLOADER.SYS	Bootstrap loader for non-system disk (linked)
S29FLXTM.BIN	Console and timer I/O routines
S29FLXEX.BIN	Disk driver executive routine
S29FLXDK.BIN	Disk driver routines
DSKERLOG.OVR	Disk driver error logging overlay
PATCHES.OVR	Patches overlay
BLIST.CMD	Buffered List command
CHECKSUM.CMD	Calculate checksum on file
DCHECK.CMD	Check disk drive settings
DSKSET.CMD	Set or display disk parameters
FORMAT.CMD	Format a diskette
LOADO.CMD	Load a binary file with offset
PRTSET.CMD	Set or display printer parameters
SYSGEN.CMD	Generate a new system diskette
MSPRINT.SYS	Printer driver for serial printer
MCPRINT.SYS	Printer driver for serial printer
MPPRINT.SYS	Printer driver for parallel printer
PATCHES.TXT	Source code for PATCHES.OVR
MSPRINT.TXT	Source code for MSPRINT.SYS
MCPRINT.TXT	Source code for MCPRINT.SYS
MPPRINT.TXT	Source code for MPPRINT.SYS

GENERAL INFORMATION FOR ADAPTING SOFTWARE PACKAGES TO RUN ON THE ST-2900

Programs that use only standard documented FLEX calls should run on the ST-2900 without any changes whatsoever. But programs that bypass FLEX for such purposes as getting a character from the keyboard will need modifications.

Some software packages, in their "installation" mode, ask you if you have a standard SS-50 bus configuration (an ACIA addressed at $E004). If you say "no", they will ask for the address of one or more routines, mentioned below. For example, for input-character-no-echo, display the value contained in $D3E5-$D3E6, and respond to the package's prompt with that value (ie., don't respond with "$D3E5", but with whatever is contained in it.)

Other packages require you to write a small I/O driver. The code for input-character-no-echo would be as simple as:

```
        INCHNE JMP   [$D3E5]
```

The following are standard FLEX address vectors:

1) $D3E5-$D3E6 contains the address of a routine that inputs a character from the console without echoing it. Upon return the character is in Register A.
2) $D3F7-$D3F8 contains the address of a routine that checks the console port to see if a character has been keyed on the keyboard. It returns condition code NE if a character has been keyed and is waiting to be read, or EQ if none is waiting.
3) $D3F9-$D3FA contains the address of a routine that outputs a character to the console. Before calling the routine, load the character into Register A.

ADAPTING THE "DYNACALC" SPREADSHEET TO THE ST-2900

During the running of the INSTALL command there are three prompts that need special consideration:

a) DOES YOUR COMPUTER USE S-BUG (OR USE SAME ACIA ADDRES)?
   Respond with "N".
b) DO YOU -
    1) HAVE TERMINAL ADDRESS
    2) HAVE ADDRESS OF ADDRESS OF TERMINAL
    3) HAVE OWN INPUT CHARAC ROUTINE
    4) QUIT
   Respond with "3".
c) YOUR ADDRESS?
   Look at the contents of $D3E5-$D3E6 and key in that address.

ADAPTING THE "SUPER-SLEUTH" DISASSEMBLER TO THE ST-2900

Change the extension on the CSSINCHR.SWT file to .TXT, then modify it to look like this:

```
* INPUT CHARACTER
INCHRT  STX    BADDR
        JSR    [$D3E5]
        LDX    BADDR
        RTS
```

Then follow the rest of the manual's instructions to generate an executable program.

ADAPTING THE "SCREDITOR III" WORD PROCESSOR TO THE ST-2900

Answer the following prompts in the CONGEN command as indicated:

a) "SYSTEM EXIT VECTOR" - display the contents of $D3F3-$D3F4 and key in that address. Note -- after using Screditor's EX command, use ST-MON's "F U hhhh" command to return to Screditor.
b) "SCREEN TYPE" - use "2" for external character display routine
c) "SCREEN DRIVERS COMPATIBLE" - answer with "Y"
d) "ADDRESS OF SCREEN OUTPUT" - display the contents of $D3F9-$D3FA and key in that address.
e) "ADDRESS OF SCREEN INIT" - display the contents of $D3F5-$D3F6 and key in that address.

Answer the following prompts in the KEYGEN command as indicated:

a) "KEYBOARD TYPE" - use "5" for external routine using NE
b) "META KEYS" - answer with "N"
c) "ADDRESS OF KEYBOARD CHECK" - display the contents of $D3F7-$D3F8 and key in that address.
d) "ADDRESS OF GET CHARACTER" - display the contents of $D3E5-$D3E6 and key in that address.
e) "MUST ECHO BE CONTROLLED" - answer with "N"
f) "KEYBOARD DELAY CONSTANT" - experiment with values between 10 and 30.

Answer the following prompts in the PRTGEN command as indicated:

a) "PRINTER OUTPUT ADDRESS" - respond with "$CCE4"
b) "PRINTER INIT ADDRESS" - respond with "$CCC0"
c) "PRINTER LEFT MARGIN" - respond with 000 (use PRTSET instead)
d) "NULLS AFTER CR-LF (NEW LINE)" - respond with 000 (use PRTSET instead)

ADAPTING THE "STYLOGRAPH" WORD PROCESSOR TO THE ST-2900

Carefully study the section towards the end of the Stylograph manual entitled "Using Stylograph with the FLEX disk operating system". Follow its instructions on modifying the STYIO file. Use the following code as a guide:

```
        ORG    IOBEG

*CONSTANTS
PORT    FDB    $FF20       DUART ADDRESS
RAMLIM  FDB    $FFFF       HIGHEST RAM USED IF LOWER THAN MEMEND
SIMFLG  FCB    0           NON-ZERO IF NO INTERRUPTS
PRTFLG  FCB    1           ZERO IF OUTPUT THROUGH FLEX, ELSE PRTOUT
INTVEC  FDB    $F058       INTERRUPT VECTOR LOCATION

DINTON  EQU    $D3CD       ENABLE SPECIFIED INTERRUPTS - ST-2900
DINTOF  EQU    $D3CF       DISABLE SPECIFIED INTERRUPTS - ST-2900
INCHEK  EQU    $D3F7       FLEX - CHECK STATUS OF KEYBOARD PORT
INCHN   EQU    $D3E5       FLEX - INPUT CHARACTER, NO ECHO
OUTCH   EQU    $D3F9       FLEX - OUTPUT CHARACTER TO CONSOLE

        ORG    IOBEG+$10 SET BRANCH TABLE
*BRANCH TABLE
INTON   BRA    JINTON
INTOFF  BRA    JINTOFF
PINIT   BRA    JPINIT
PCHECK  BRA    JPCHECK
GETCH   BRA    JGETCH
PRTOUT  EQU    *

*OUTPUT CHARACTER THROUGH DUART
        JMP    [OUTCH]

*GET A CHARACTER
JGETCH  JMP    [INCHN]

*CHECK FOR CHARACTER AT INPUT
JPCHECK JMP    [INCHEK]

*ALLOW DUART TO GENERATE INTERRUPTS ON RxRDYA
JINTON  PSHS   A
        LDA    #$02
        JSR    [DINTON]
        PULS   A,PC

*INITIALIZE PORT
JPINIT  PSHS   A            FLUSH FIFO
JP10    JSR    [INCHEK]
        BEQ    JI10
        JSR    [INCHN]
        BRA    JP10

JINTOFF PSHS   A            DISABLE RxRDYA INTERRUPT
JI10    LDA    #$02
        JSR    [DINTOF]
        PULS   A,PC
```

ADAPTING THE "RMS" DATA MANAGEMENT PACKAGE TO THE ST-2900

1) Load RMS.CMD into memory with the FLEX "GET" command, then use the "MON" command to exit FLEX and enter ST-MON.
2) Look at the following locations and verify that they contain the values in the "OLD" column:

address	old	new	description
0022	E0	don't care	address of console port MSB
0023	04	" "	" " " " LSB
1B87	34	6E	routine to input character from console port
1B88	10	9F	
1B89	9E	D3	
1B8A	22	E5	
1B94	35	don't care	
1B95	10	" "	
1B96	39	" "	
1B97	34	6E	routine to output character to console port
1B98	14	9F	
1B99	9E	D3	
1B9A	22	F9	
1BA3	35	don't care	
1BA4	14	" "	
1BA5	39	" "	

3) If the "old" values agree, patch RMS with the "new" values at the locations indicated.
4) If the "old" values don't agree, you have a different version of RMS than the one we tested. Use ST-MON's "S" command to search for the same code residing in a different location.

SOURCES OF SOFTWARE FOR FLEX

- Frank Hogg Laboratory
  The Regency Tower, Suite 215
  770 James St.
  Syracuse, NY 13203
  U.S.A.
  (315) 474-7856

  Ask for their "Serious Users Software Catalog" that is chock full of assemblers, compilers, disassemblers, debuggers, editors, word processors, spelling checkers, data base managers, spread sheets, and accounting packages. Just too many to list here. This catalogue is one of those "must have" items.

- Southeast Media
  5900 Cassandra Smith Rd.
  Hixson, Tennessee 37343
  U.S.A.
  (615) 842-4601 for information
  1-800-338-6800 (toll free) to order

  Southeast Media now claim to be the largest 68XX software distributor in the world, with over 300 programs available for a wide variety of systems. Recent issues of "'68' Micro Journal" magazine, published by a different division of the same company (and same address as Southeast Media) contain a 6 page mini-catalogue of software, most of it for FLEX. Besides their software sales, this is one magazine you can't afford not to read.

- AAA Chicago Computer Center
  120 Chestnut Lane
  Wheeling, Illinois 60090
  U.S.A.
  (312) 459-0450

  Their two page "catalogue" is updated monthly and appears as an ad in '68' Micro Journal.

- Many other suppliers of FLEX software also advertise in '68' Micro Journal.

INTERRUPTS FOR ADVANCED PROGRAMMERS

Of the 6 possible types of interrupts, FLEX normally uses only IRQ and SWI3. The other four (FIRQ, NMI, SWI, SWI2) are completely user defineable. See the description below of the hardware interrupt jump table.

The ST-2900 FLEX Conversion Package includes an interrupt polling routine for IRQ interrupts from the 2681 DUART and 6522 VIA that easily lets you set up multiple interrupt sources. The polling system identifies the source of the IRQ interrupt and calls the appropriate service routine. Note -- the 6522 must be installed for interrupt polling to work.

Two interrupts are predefined. The 6522's Timer 1 is used to generate an interrupt every 10 msec. when the FLEX print spooling routine is active. The DUART's "counter ready" interrupt is used to implement a time-out timer interrupt for the floppy disk drivers.

The other 6 possible interrupts from the 6522, and the other 7 possible interrupts from the DUART, are available for your own purposes. Refer to the description of the "interrupt flag register" in the 6522 data sheet, and the description of the "interrupt status register" in the 2681 data sheet.

Each interrupt service routine has three main requirements:

a) It should reset the associated interrupt status flag in the device. For example, reading the character in the DUART's RHRA (receive data) register will automatically also clear the matching RxRDYA flag (bit 1) of the DUART's interrupt status register (ISR).

b) Until the routine is done, no further interrupts will be acknowledged (the "I" flag in the condition code register was set when the current interrupt was acknowledged and should NOT be cleared by the service routine). Therefore, the service routine should "do its thing" and exit as quickly as possible.

c) Exit the routine with an "RTI" instruction.

To activate the interrupt and its routine you must:

a) Store the address of the service routine in the VIA or DUART interrupt jump table described below.

b) Enable the interrupt by calling the DINTON or VINTON routine described below.

Let's look at a simplified example that would receive data from the second serial port on an interrupt driven basis and store it into a buffer:

```
 DUARTB  EQU    $FF20+8
 DINTON  EQU    $D3CD
 DINTOF  EQU    $D3CF
 DRXRYB  EQU    $F04B
 BUFPTR  RMB    2
 *
 * INITIALIZE AND ENABLE THE INTERRUPT
 *
 ENABL   LDX    #$7000     INIT BUFFER POINTER
         STX    BUFPTR
         LDX    #SERVCE    SET UP JUMP TO SERVICE ROUTINE
         STX    IRXDYB+1
         LDA    #$20       ENABLE RXRDYB INTERRUPT
         JSR    [DINTON]
         RTS
*
* DISABLE THE INTERRUPT
*
DISABL   LDA    #$20       DISABLE RXRDYB INTERRUPT
         JSR    [DINTOF]
         RTS
*
* INTERRUPT SERVICE ROUTINE
*
SERVCE   LDX    BUFPTR     GET BUFFER POINTER
         LDA    DUARTB+3   GET REC'D CHARAC AND RESET INTERRUPT
         STA    ,X+        STORE CHARAC IN BUFFER
         STX    BUFPTR     UPDATE BUFFER POINTER
         RTI               RETURN
```

If you receive an "INTERRUPT ERROR X" message where X is a character between "A" and "F", it means either that an interrupt was received but the polling routine could not determine which source caused it (maybe the interrupt was mistakenly reset before if could be processed), or the vector (in the table at $F030-$F05C) associated with the interrupting device does not point to an appropriate service routine.

There are three device registers you should only update via the 6 routines mentioned below. Do NOT modify these registers directly. They are the DUART's "ACR" and "IMR" registers, and the VIA's "IER" register.

Addr	Name	Description
D3C9	DACRON	set on bits in DUART ACR register
D3CB	DACROF	set off bits in DUART ACR register
D3CD	DINTON	set on bits in DUART IMR register
D3CF	DINTOF	set off bits in DUART IMR register
D3D1	VINTON	set on bits in VIA IER register
D3D3	VINTOF	set off bits in VIA IER register

To call these routines, load Register A with a value where bits to be turned on or off are 1's, bits not to be affected are 0's, then do an indirect subroutine call. For example, to turn off bits 4 & 5 in the DUART's IMR register:

```
      LDA    #$30      disable RxRDYB & TxRDYB interrupts
      JSR    [$D3CF]
```

and to turn on bit 2 in the VIA's IER register:

```
      LDA    #$04      enable shift register interrupt
      JSR    [$D3D1]
```

HARDWARE INTERRUPT JUMP TABLE

Note - 4 bytes are allowed for each entry, so you could use an indirect jump such as JMP [ZZZZZ] if desired.
- the interrupt service routine jumped to should reset the source of the interrupt (if a hardware interrupt - FIRQ, IRQ, NMI), do the necessary processing, and exit with an RTI instruction.

```
                        ORG    $FEE7
FEE7          JSWI3     JMP    XXXXX     (reserved for use by FLEX - do not change
                        NOP
FEEB          JSWI2     JMP    AAAAA
                        NOP
FEEF          JFIRQ     JMP    BBBBB
                        NOP
FEF3          JIRQ      JMP    YYYYY     (reserved for use by FLEX - do not change
                        NOP
FEF7          JSWI      JMP    CCCCC
                        NOP
FEFB          JNMI      JMP    DDDDD
                        NOP
```

VIA AND DUART IRQ JUMP TABLES

```
                *
                * VIA (6522) IRQ JUMP TABLE
                *
                        ORG    $F030
F030            VTIMR1  JMP    XXXXX   TIMER 1 (do not change - used by FLEX)
F033            VTIMR2  JMP    AAAAA   TIMER 2
F036            VCB1    JMP    BBBBB   CB1
F039            VCB2    JMP    CCCCC   CB2
F03C            VSR     JMP    DDDDD   SHIFT REGISTER
F03F            VCA1    JMP    EEEEE   CA1
F042            VCA2    JMP    FFFFF   CA2
                *
                * DUART (2681) IRQ JUMP TABLE
                *
F045            DICHNG  JMP    GGGGG   INPUT PORT CHANGE
F048            DBRKB   JMP    HHHHH   DELTA BREAK B
F04B            DRXRYB  JMP    IIIII   RxRDYB
F04E            DTXRYB  JMP    JJJJJ   TxRDYB
F051            DCTRDY  JMP    YYYYY   COUNTER READY (do not change -used by FLEX
F054            DBRKA   JMP    KKKKK   DELTA BREAK A
F057            DRXRYA  JMP    LLLLL   RxRDYA
F05A            DTXRYA  JMP    MMMMM   TxRDYA
```

SYSTEM MEMORY MAP

0000 - BFFF		User memory
C000 - D36F	*	Various components of FLEX
D370 - D3AA		Console and timer drivers
D3AB - D3E4		Internally used address vectors
D3E5 - D3FC	*	Console and timer address vectors
D3FD - DDFF	*	Various components of FLEX
DE00 - DE1D	*	Disk driver jump table
DE1E - DEFF		Disk driver routines
DF00 - DFFF		Overflow area for printer drivers
E000 - E34F		DSKERLOG.OVR
E350 - E7FF		(not used)
E800 - EFFF		Utility Command Area overflow (used by DSKSET)
F000 - F0FF		Various vectors and variables and tables
F100 - F37F		Overflow area for console & timer drivers
F380 - F893		Disk driver routines
F894 - FEE6		ST-MON 1.01
FEE7 - FEFF		Hardware interrupt vectors
FF00 - FF5F		I/O devices
FF60 - FFFF		6883 (SAM) registers, etc.

* These areas are standard FLEX -- refer to the FLEX User's Manual and FLEX Programmer's Manual for more details.

....