

```

00010 00001          OPT   REL,O,S,LLEN=120
00050 00005          OPT   LIST

00070 00007          *****
00080 00008          *
00090 00009          *   XCOM.SA      REV. DATE 02/10/81   *
00100 00010          *
00110 00011          *           REAL TIME EXECUTIVE COMMON   *
00120 00012          *           EQUATE AND MACRO FILE - 6809   *
00130 00013          *           CUSTOMISED FOR RANK 1200   *
00140 00014          *           LICENSED MATERIAL - PROPERTY OF   *
00150 00015          *           CREATIVE STRATEGIES PTY. LTD.   *
00160 00016          *****

00180 00018          *   CONDITIONAL ASSEMBLY CONTROL SWITCHES

00200 00020          0001  A IETRAP SET    ON      INTERNAL SYSTEM ERROR TRAPS (ON/OF)
00210 00021          0001  A DMEM  SET    ON      DYNAMIC MEMORY FUNCTIONS - XMEM - (ON/OF)
00220 00022          0001  A XSTAK SET    ON      STACK MANIPULATION FUNCTIONS - XSTACK - (ON/OF)
00230 00023          0000  A TRAP  SET    OF      SYSTEM TRAP FUNCTIONS (ON/OF)
00240 00024          0001  A DISPLY SET   ON      SYSTEM STATUS DISPLAY TASK (ON/OF)
00250 00025          0001  A UBUG  SET    ON      USER DEBUG CODE (ON/OF)

00270 00027          *   CLOCK TICK RATE.

00290 00029          000F  A TICK  EQU    15      15 MILLISECS PER CLOCK TICK

07940 00794          OPT   LIST

00010 00796          *****
00015 00797          *
00020 00798          *   TCOM.SA      REV. DATE 02/10/81   *
00025 00799          *
00030 00800          *           REAL TIME EXECUTIVE 6809   *
00035 00801          *           TASK RELATED MACRO FILE   *
00040 00802          *
00045 00803          *           LICENSED MATERIAL - PROPERTY OF   *
00050 00804          *           CREATIVE STRATEGIES PTY. LTD.   *
00055 00805          *****

06455 02085          OPT   LIST

00020 02087          *****
00030 02088          *
00040 02089          *   DOSEQU.SA  REV. DATE 06/05/81   *
00050 02090          *
00060 02091          *           REAL TIME EXECUTIVE   *
00070 02092          *           ADOS EQUATE FILE   *
00080 02093          *
00090 02094          *           LICENSED MATERIAL - PROPERTY OF   *
00100 02095          *           CREATIVE STRATEGIES PTY. LTD.   *
00110 02096          *****

03380 02423          OPT   LIST

```

00015 02426 NAM SCRDRV
00020 02427 TTL INSTANTIATABLE SCREEN DRIVER - CUSTOM FOR RNK1200
00025 02428 IDNT ADOS INSTANTIATABLE SCREEN DRIVER TASK

00035 02430 *****
00040 02431 * *
00045 02432 * SCRDRV.SA REV. DATE 06/05/81 *
00050 02433 * *
00055 02434 * REAL TIME EXECUTIVE *
00060 02435 * ADOS SCREEN DRIVER TASK *
00065 02436 * CUSTOMISED FOR RANK 1200 SYSTEM *
00070 02437 * LICENSED MATERIAL - PROPERTY OF *
00075 02438 * CREATIVE STRATEGIES PTY. LTD. *
00080 02439 *****

00090 02441 * THIS MODULE CONTAINS CODE CONDITIONAL UPON THE "SHORT"
00095 02442 * SWITCH.

00105 02444 * SETTING THIS SWITCH TO "OF" ENABLES ALL PROCESSES REQUIRING
00110 02445 * LONG PACKETS (READ,WRITE,AND FLAVOURS OF IL,DL,AND POS).

00120 02447 * SETTING THIS SWITCH TO "ON" DISABLES AND REMOVES ALL CODE
00125 02448 * REQUIRING LONG PACKETS THUS PRODUCING A MORE COMPACT VERSION
00130 02449 * OF THE SCREEN DRIVER.

00140 02451 0000 A SHORT SET OF

00155 02454 * DESCRIPTION OF COMMAND PACKET FORMATS FOR DIFFERENT FUNCTIONS.

00165 02456 * CURSOR UP:

00175 02458 * ISSUED PACKET: VD\$CMD = CRUP

00185 02460 * CURSOR DOWN:

00195 02462 * ISSUED PACKET: VD\$CMD = CDOWN

00205 02464 * CURSOR LEFT:

00215 02466 * ISSUED PACKET: VD\$CMD = CLEFT

00225 02468 * CURSOR RIGHT:

00235 02470 * ISSUED PACKET: VD\$CMD = CRIGHT

00245 02472 * CURSOR HOME: (TOP LEFT CORNER OF SCREEN)

00255 02474 * ISSUED PACKET: VD\$CMD = CHOME

00265 02476 * INSERT CHARACTER:

00275 02478 * ISSUED PACKET: VD\$CMD = IC

00280 02479 * VD\$POS = \$FFFF OR POSITION

00285 02480 * VD\$CHR = 0 OR AN ASCII CHARACTER

```

00300 02483      *  DELETE CHARACTER:

00310 02485      *    ISSUED PACKET: VD$CMD = DC
00315 02486      *                VD$POS = $FFFF OR POSITION (V,H)

00325 02488      *  INSERT LINE:

00335 02490      *    ISSUED PACKET: VD$CMD = IL
00340 02491      *                VD$POS = $FFFF OR POSITION
00350 02493      *                VD$CHR = LONG PACKET SELECTOR ($FF = SHORT)

00360 02495      *    IF THE LONG PACKET IS SELECTED (VD$CHR <> $FF) THE FOLLOWING APPLIES:-

00370 02497      *    RETURN PACKET: VD$1ST = OFFSET INTO PACKET OF 1ST NON-SPACE
00375 02498      *                VD$LST = OFFSET INTO PACKET OF LAST NON-SPACE
00380 02499      *                VD$MSK = BIT MASK OF DATA TYPES READ FROM SCREEN
00385 02500      *                VD$BUF = TOP LINE OF SCREEN BEFORE INSERT

00400 02503      *  DELETE LINE:

00410 02505      *    ISSUED PACKET: VD$CMD = DL
00415 02506      *                VD$POS = $FFFF OR POSITION
00425 02508      *                VD$CHR = LONG PACKET SELECTOR ($FF = SHORT)

00435 02510      *    IF LONG PACKET SELECTED (VD$CHR <> $FF) THE FOLLOWING APPLIES:-

00445 02512      *                VD$BUF = NEW BOTTOM LINE FOR SCREEN AFTER DL

00455 02514      *    RETURN PACKET: VD$1ST = OFFSET INTO PACKET OF 1ST NON-SPACE
00460 02515      *                VD$LST = OFFSET INTO PACKET OF LAST NON-SPACE
00465 02516      *                VD$MSK = BIT MASK OF DATA TYPES WRITTEN TO SCREEN

00480 02519      *  ERASE TO END OF LINE:

00490 02521      *    ISSUED PACKET: VD$CMD = EEOL
00495 02522      *                VD$POS = $FFFF OR POSITION

```

```

00510 02525      * POSITION CURSOR AND ISSUE CHARACTER

00520 02527      * ISSUED PACKET: VD$CMD = POS
00525 02528      * VD$POS = $FFFF OR POSITION
00530 02529      * VD$CHR = 0 OR CHARACTER
00540 02531      * VD$1ST = LONG PACKET SELECTOR ($FF = SHORT)

00550 02533      * IF LONG PACKET SELECTED (VD$CHR <> $FF) AND THE CHARACTER
00555 02534      * IS A <LF> AT THE BOTTOM OF THE SCREEN, THE FOLLOWING APPLIES:-

00565 02536      * RETURN PACKET: VD$1ST = OFFSET INTO PACKET OF 1ST NON-SPACE
00570 02537      * VD$LST = OFFSET INTO PACKET OF LAST NON-SPACE
00575 02538      * VD$MSK = BIT MASK OF DATA TYPES READ FROM SCREEN
00580 02539      * VD$BUF = TOP LINE OF SCREEN BEFORE SCROLL

00595 02542      * GET CO-ORDINATES OF CURRENT CURSOR:

00605 02544      * ISSUED PACKET: VD$CMD = GETC

00615 02546      * RETURN PACKET: VD$POS = CO-ORDINATES OF CURRENT CURSOR

00625 02548      * RESTORE CURSOR TO LAST POSITION

00635 02550      * ISSUED PACKET: VD$CMD = RESTOR

00645 02552      * INVERT AN AREA OF THE SCREEN:

00655 02554      * ISSUED PACKET: VD$CMD = INVERT
00660 02555      * VD$POS = $FFFF OR POSITION
00665 02556      * VD$CHR = BYTE COUNT FOR INVERSION

00675 02558      * SET THE MARGINS OF THE SCREEN:

00685 02560      * ISSUED PACKET: VD$CMD = SETMRG
00690 02561      * VD$POS = TOP OF SCREEN CO-ORDINATES
00695 02562      * VD$1ST = BOTTOM OF SCREEN VERTICAL CO-ORD.
00700 02563      * VD$LST = BOTTOM OF SCREEN HORIZONTAL CO-ORD.

```

```

00715 02566      *   FLASH AN AREA OF THE SCREEN

00725 02568      *   ISSUED PACKET: VD$CMD = FLASH
00730 02569      *                   VD$POS = $FFFF OR POSITION
00735 02570      *                   VD$CHR = FIELD SIZE COUNT

00745 02572      *   STOP AN AREA OF THE SCREEN FROM FLASHING

00755 02574      *   ISSUED PACKET: VD$CMD = STOPFL
00760 02575      *                   VD$POS = $FFFF OR POSITION

00775 02578      *   NOTE:  THE FOLLOWING CALLS ASSUME A LONG (92 BYTE) PACKET

00785 02580      *   READ A LINE FROM THE SCREEN:

00795 02582      *   ISSUED PACKET: VD$CMD = READ
00800 02583      *                   VD$POS = POSITION FOR READ
00805 02584      *                   VD$CHR = BYTE COUNT FOR READ

00815 02586      *   RETURN PACKET: VD$1ST = OFFSET INTO PACKET OF 1ST NON-SPACE
00820 02587      *                   VD$LST = OFFSET INTO PACKET OF LAST NON-SPACE
00825 02588      *                   VD$MSK = BIT MASK OF DATA TYPES READ FROM SCREEN
00830 02589      *                   VD$BUF = DATA READ FROM SCREEN

00840 02591      *   WRITE A LINE TO THE SCREEN:

00850 02593      *   ISSUED PACKET: VD$CMD = WRITE
00855 02594      *                   VD$POS = POSITION FOR WRITE
00860 02595      *                   VD$CHR = BYTE COUNT FOR WRITE
00865 02596      *                   VD$BUF = DATA TO BE WRITTEN TO SCREEN

00880 02599      *   RETURN PACKET: VD$1ST = OFFSET INTO PACKET OF 1ST NON-SPACE
00885 02600      *                   VD$LST = OFFSET INTO PACKET OF LAST NON-SPACE
00890 02601      *                   VD$MSK = BIT MASK OF DATA TYPES WRITTEN TO SCREEN

```

```

00910 02605          ***** NOTES

00920 02607          * VD$POS REPRESENTS THE VERTICAL AND HORIZONTAL CO-ORDINATES ON
00925 02608          * THE SCREEN AT WHICH THE FUNCTION IS TO BE PERFORMED. IF THIS IS
00930 02609          * SET TO $FFFF THE FUNCTION WILL USE THE POSITION OF THE CURRENT
00935 02610          * CURSOR LOCATION.

00945 02612          * THE CO-ORDINATES TAKE THE FORM (V,H) WHERE V=LINE NUMBER ON
00950 02613          * SCREEN (0 THRU (LINES ON SCREEN - 1)) AND H= POSITION IN LINE
00955 02614          * (0 THRU 79).

00965 02616          * ALL FUNCTIONS RETURN IN VD$POS THE RESULTING CO-ORDINATES ON THE
00970 02617          * SCREEN OF THE CURSOR AT THE COMPLETION OF THE FUNCTION

00980 02619          * IF AN INVALID COMMAND IS RECIEVED, THE PACKET IS RELEASED WITH
00985 02620          * THE COMMAND BYTE (VD$CMD) SET TO $FF.

00995 02622          * ANY STRING READ OR WRITE TO THE SCREEN USING A LONG PACKET
01000 02623          * RETURNS A BIT MASK OF THE DATA TYPES ENCOUNTERED DURING THE
01005 02624          * PROCESS. THE BIT MASK DEFINITIONS ARE AS FOLLOWS:-

01015 02626          *          $80 - NUMERICS 0,1,2,3,4,5,6,7,8,9
01020 02627          *          $40 - HEX. CHARACTERS 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
01025 02628          *          $20 - UPPER CASE A,B,,,,,,,,,X,Y,Z
01030 02629          *          $10 - LOWER CASE a,b,,,,,,,,,x,y,z
01035 02630          *          $08 - NUMERIC OPERATORS + - .
01040 02631          *          $04 - SPECIAL OPERATORS ( ) * / < = > [ ] ^
01045 02632          *          $02 - SPECIAL CHARACTERS
01050 02633          *          $01 - SPACE

```

01065 02636

* EXTERNAL REFERENCES AND DEFINITIONS

01075 02638

* EXTERNAL DEFINITIONS

01085 02640

* EXTERNAL REFERENCES


```

01095 02642D 0000          DSCT

01105 02644          *    TEMPORARIES

01115 02646          *    ALL SAVE LOCATIONS DEFINED IN TASK'S DIRECT PAGE

01125 02648D 0000          SDIR .          DEFINE START OF DIRECT PAGE

01135 02650D 0000          DRMB  PACKET,2 STORAGE LOCATION FOR ADDRESS OF CURRENT PACKET
01140 02651D 0000          DRMB  LINE,2  ADDRESS OF THE CURRENT LINE ON SCREEN
01145 02652D 0000          DRMB  POSN,2  POINTER TO THE SCREEN POSITION FOR A TRANSACTION
01150 02653D 0000          DRMB  OLDCUR,2 BACKTRACK COPY OF CURSOR
01155 02654D 0000          DRMB  FLSHBF,2 ADDRESS OF PACKET BUFFER FOR FLASH ZONES
01160 02655D 0000          DRMB  PHASE,1 FLASH PHASE INDICATOR
01165 02656D 0000          DRMB  TOPSCR,2 CURRENT TOP OF SCROLL
01170 02657D 0000          DRMB  BOTSCR,2 CURRENT BOTTOM OF SCROLL
01175 02658D 0000          DRMB  LINES,1 LINES ON SCREEN
01180 02659D 0000          DRMB  TLC,2   ADDRESS OF FIRST BYTE ON VRAM
01185 02660D 0000          DRMB  BOTLIN,2 ADDRESS OF LAST LINE ON SCREEN
01190 02661D 0000          DRMB  DIRFLG,1 DIRECTON FLAG - USED DURING TEXT MOVES
01195 02662D 0000          DRMB  WRTFLG,1 SCREEN STATUS DATA (<>0 = SCREEN ALTERED)
01200 02663D 0000          DRMB  CT$SCR,4 INPUT PACKET CHAIN
01205 02664D 0000          DRMB  CT$FLS,4 PACKET CHAIN CONTROLLING SCREEN FLASH ZONES
01210 02665D 0000          DRMB  SCREND,2 END OF AREA TO BE SCROLLED
01215 02666D 0000          DRMB  STMP,2  TEMPORARY FOR USE BY SCROLL
01220 02667D 0000          DRMB  ELAPCH,2 POINTER TO ELAPSED TIME CHAIN
01225 02668D 0000          DRMB  DCSB,DV$SIZ DEVICE CONTROL/STATUS BLOCK FOR SCREEN

01235 02670D 0000          DEND
01235          OPT    MEX
01235          005C  A BPSIZE EQU  BASEPC

01235          *    SHARED DIRECT PAGE MAY BE POSSIBLE

```

```

01250 02673          * STRUCTURES UNIQUE TO THIS TASK

01260 02675          *  STRUCTURE OF INPUT SCREEN COMMAND PACKETS

01270 02677D 0000          STRUC

01280 02679D 0000          SENT  VD$LNK,P.SIZE PACKET LINKAGES
01285 02680D 0000          SENT  VD$POS,2 COLUMN/ROW CO-ORDINATES FOR I/O
01290 02681D 0000          SENT  VD$1ST,1 FIRST INSTANCE OF NON-BLANK CHARACTER IN READ
01295 02682D 0000          SENT  VD$LST,1 LAST INSTANCE OF NON-BLANK CHARACTER IN READ
01300 02683D 0000          SENT  VD$CMD,1 COMMAND CODE
01305 02684D 0000          SENT  VD$MSK,0 BIT MASK SHOWING DATA TYPES ENCOUNTERED IN READ
01310 02685D 0000          SENT  VD$CHR,1 CHARACTER FOR DISPLAYING
01315 02686D 0000          SENT  VD$SSZ,0 SIZE OF A SHORT COMMAND PACKET
01320 02687D 0000          SENT  VD$BUF,80 LONG PACKET DATA BUFFER
01325 02688D 0000          SENT  VD$LSZ,0 SIZE OF A LONG COMMAND PACKET

01335 02690          005C  A VD$SIZ EQU  LOC

```

01350 02693P 0000

PSCT

01360 02695

* TASK CONTROL BLOCK FOR SCREEN DRIVER

01370 02697P 0000

XTASK SCRN,8,B TASK HAS A BASE PAGE

01370

OPT MEX,NOCL

01370

0000 A DCNT SET 0 *** DEFINE OFFSETS FOR DYNAMICALLY BUILT T.C.B. AND STACK ***

01370

0000 A B\$SCRN EQU DCNT DEFINE BASE OF TASK DIRECT PAGE

01370

00D5 A DCNT SET DCNT+(\$100-(T.SIZE+S.SIZE+2)) SET RAM SIZE TO 2**8 BYTES

01370

0088 A RSIZ SET 8!+\$80

01370

00D5 A S\$SCRN EQU DCNT DEFINE INITIAL STACK FRAME

01370

00E3 A DCNT SET DCNT+S.SIZE+2

01370

00E3 A T\$SCRN EQU DCNT DEFINE START OF TCB

01370

XDEF T\$SCRN

01370

XDEF B\$SCRN

01370 P 0000

PSCT ***** TASK ROM ***** (TASK DESCRIPTOR BLOCK)(T.D.B)

01370 P 0000

0000 A P\$SCRN FDB 0 (R.LPTR) DUMMY LINK TO NEXT TASK

01370

XDEF P\$SCRN

01370 P 0002

00D5 A FDB S\$SCRN (R.SP) SP POSITION FOR INITIAL INSTANCE OF TASK

01370 P 0004

53 A FCC 4,SCRN (R.NAME) TASK NAME

01370 P 0008

00E3 A FDB T\$SCRN (R.T.PT) POINTER TO TCB

01370 P 000A

054D A FDB P\$SIZE (R.SIZ) EXTERNAL TASK SIZE

01370 P 000C

88 A FCB RSIZ (R.SSIZ) SET SIZE OF STACK AND RAM FOR AN INSTANCE OF THIS TASK

01370 P 000D

0000 A FDB 0 (R.MAPS) ROOM FOR TASK MAP SIZE

01370 P 000F

0600 A FDB T\$SIZE R.TOTS TOTAL SIZE OF TASKS PSCT+DSCT

01370 P 0011

0000 A FDB B\$SCRN (R.BASE) POINTER TO START OF TASK RAM/STACK/BSCT

01375 02698P 0013

ASSOC I,SCRNIN INITIALISATION ROUTINE

01380 02699P 0016

ASSOC K,ARKILL DYNAMIC MEMORY RELEASE ROUTINE

01385 02700P 0019

ASSOC M,SCRDRV MAINLINE EXECUTION ENTRY POINT

01395 02702P 001C

ETASK

```

01410 02705          * MACRO DEFINITIONS UNIQUE TO THIS TASK

01420 02707          * MACRO TO FORM COMMAND CODE JUMP TABLE

01430 02709          * \0 = NAME OF APPROPRIATE ROUTINE
01435 02710          * \1 = COMMAND NAME

01445 02712          SCRCAL MACR
      02713          1450 FDB \0
      02714          1455 \1 EQU SLOC
      02715          1460 XDEF \1
      02716          1465 SLOC SET SLOC+1
      02717          1470 ENDM

01480 02719          * TABLE OF COMMAND ROUTINES

01490 02721          0000  A SLOC  SET   0          INITIALISE COMMAND CODE COUNTER

01500 02723P 001D          TBL  SCRCAL CURUP,CRUP CURSOR UP FUNCTION
01505 02724P 001F          SCRCAL CURDWN,CDOWN CURSOR DOWN FUNCTION
01510 02725P 0021          SCRCAL CURLFT,CLEFT CURSOR LEFT (<-) FUNCTION
01515 02726P 0023          SCRCAL CURRYT,CRIGHT CURSOR RIGHT FUNCTION (->)
01520 02727P 0025          SCRCAL CURHOM,CHOME CURSOR HOME FUNCTION
01525 02728P 0027          SCRCAL DOIC,IC  INSERT CHARACTER
01530 02729P 0029          SCRCAL DODC,DC  DELETE CHARACTER
01535 02730P 002B          SCRCAL DOIL,IL  INSERT LINE FUNCTION
01540 02731P 002D          SCRCAL DODL,DL  DELETE LINE FUNCTION
01545 02732P 002F          SCRCAL DOEEOLE,EEOL ERASE TO END OF LINE FUNCTION
01550 02733P 0031          SCRCAL DOPOS,POS POSITION CURSOR FUNCTION
01555 02734P 0033          SCRCAL DOGETC,GETC RETURN CURSOR FUNCTION
01560 02735P 0035          SCRCAL DOREST,RESTOR RESTORE CURSOR FUNCTIN
01565 02736P 0037          SCRCAL DOINV,INVERT INVERT A LINE FUNCTION
01570 02737P 0039          SCRCAL DOSETM,SETMRG SET TOP AND BOTTOM MARGINS FOR SCREEN
01575 02738P 003B          SCRCAL DOFLSH,FLASH SET AN AREA OF THE SCREEN TO FLASHING MODE
01580 02739P 003D          SCRCAL DOSTFL,STOPFL STOP AN AREA OF THE SCREEN FROM FLASHING
01585 02740          0000  A          IFEQ  SHORT
01590 02741P 003F          SCRCAL DOREAD,READ DO SCREEN READ FUNCTION
01595 02742P 0041          SCRCAL DOWRIT,WRITE WRITE TO SCREEN WITHOUT ALTERING CURSOR
01600 02743          ENDC

```

```

01615 02746          *  INITIALISATION ENTRY POINT

01625 02748          *  ON ENTRY TASK'S X REGISTER POINTS TO A FOUR BYTE INITIALISATION
01630 02749          *  PACKET OF THE FOLLOWING FORMAT:-

01640 02751          *          BYTES 0-1 = ADDRESS OF TOP LINE OF SCREEN TO BE USED
01645 02752          *          BYTES 2-3 = ADDRESS OF A 16-BIT SAVE LOCATION FOR THE
01650 02753          *                   ADDRESS OF THE SCREEN DRIVER'S D.C.S.B.
01655 02754          *          BYTE    4 = NUMBER OF LINES ON SCREEN TO BE USED

01665 02756P 0043          SCRININ LADP  DCSB,U   POINT TO D.C.S.B.
01670 02757P 004B EF  98 02  A          STU    [2,X]   AND ISSUE ITS ADDRESS
01675 02758P 004E E6  04          A          LDB    4,X     LOAD NUMBER OF LINES IN SCREEN
01680 02759P 0050 D7  15          A          STB    LINES   AND SAVE IT
01685 02760P 0052 AE  84          A          LDX    0,X     POINT TO SCREEN
01690 02761P 0054 9F  16          A          STX    TLC     ISSUE ADDRESS OF TOP LINE
01695 02762P 0056 30  88 50  A          LEAX   80,X    POINT TO FIRST NON-RESERVED LINE
01700 02763P 0059 9F  11          A          STX    TOPSCR  AND ISSUE IT AS TOP OF SCROLL
01705 02764P 005B C0  02          A          SUBB   #2     CALC. NO. OF LINES IN MAIN BODY OF SCREEN
01710 02765P 005D 86  50          A          LDA    #80    80 CHARACTERS PER LINE
01715 02766P 005F 3D                MUL
01720 02767P 0060 31  8B          A          LEAY   D,X    POINT TO BOTTOM LINE OF SCREEN
01725 02768P 0062 109F 18          A          STY    BOTLIN  ISSUE IT
01730 02769P 0065 109F 0C          A          STY    OLDCUR  MAKE IT OLD CURSOR POSITION
01735 02770P 0068 30  A8 50  A          LEAX   80,Y    POINT TO END OF SCREEN
01740 02771P 006B 9F  13          A          STX    BOTSCR  AND ISSUE IT AS END OF SCROLL
01745 02772P 006D 9E  16          A          LDX    TLC     POINT TO TOP OF SCREEN
01750 02773P 006F C6  20          A          LDB    #$20   LOAD A SPACE

01760 02775P 0071 E7  80          A CLRSCR STB  0,X+   AND CLEAR SCREEN
01765 02776P 0073 9C  13          A          CMPX   BOTSCR  FINISHED?
01770 02777P 0075 26  FA  0071  BNE    CLRSCR  LOOP IF NOT
01775 02778P 0077                DABB   #,7,#,9  GRAB A 128 BYTE BUFFER
01780 02779P 0080 9F  0E          A          STX    FLSHBF  AND MAKE IT BUFFER FOR FLASH PACKETS
01785 02780P 0082                LADP   CT$SCR,X POINT TO INPUT CHAIN
01790 02781P 008A                LADP   CT$FLS,U AND FLASH CHAIN
01795 02782P 0092 AF  84          A          STX    L.PREV,X INITIALISE THEM
01800 02783P 0094 AF  02          A          STX    L.NEXT,X
01805 02784P 0096 EF  C4          A          STU    L.PREV,U
01810 02785P 0098 EF  42          A          STU    L.NEXT,U
01815 02786P 009A 0F  10          A          CLR    PHASE   RESET FLASH PHASE
01820 02787P 009C 0F  1B          A          CLR    WRTFLG  SHOW SCREEN NOT ALTERED
01825 02788P 009E                INFO   .         POINT TO SYSTEM CONFIGURATION TABLE
01830 02789P 00A0 EE  0A          A          LDU    I.ELAP,X POINT TO SYSTEM ELAPSED TIME WAIT CHAIN
01835 02790P 00A2 DF  28          A          STU    ELAPCH  AND SAVE ITS ADDRESS
01840 02791P 00A4 E6  88 1D  A          LDB    I.CHNS,X LOAD NUMBER OF SCHEDULED CHAINS IN SYSTEM
01845 02792P 00A7 5A                DECB                          WE WANT THE LOWEST PRIORITY CHAIN
01850 02793P 00A8 58                ASLB
01855 02794P 00A9 CB  1F          A          AADB   #I.SCHD SO FORM OFFSET TO ADDRESS OF CHAIN IN TABLE
01860 02795P 00AB                HOMX   ,(B,X)  AND ISSUE IT TO TCB
01865 02796P 00AF 8D  02  00B3  BSR    BLDCSB  INITIALISE D.C.S.B.
01870 02797P 00B1                EXAR   .         AND EXIT

```

01885 02800

* ROUTINE TO INITIALISE THE D.C.S.B.

```
01895 02802P 00B3          BLDCSB LADP  DCSB,U  POINT TO THIS INSTANCE'S COPY
01900 02803P 00BB C6   32   A          LDB   #DV$SIZ  LOAD BYTE COUNT

01910 02805P 00BD 6F   C0          A BLDC10 CLR   0,U+   CLEAR THIS BYTE
01915 02806P 00BF 5A          DECB          FINISHED?
01920 02807P 00C0 26   FB   00BD          BNE   BLDC10  LOOP IF NOT
01925 02808P 00C2          LADP  DCSB,U  POINT TO DCSB JUST BUILT
01930 02809P 00CA EF   42          A   STU   L.NEXT,U AND INITIALISE LINKAGES
01935 02810P 00CC EF   C4          A   STU   L.PREV,U
01940 02811P 00CE          CLDB   (M.RES,U)
01945 02812P 00D2 8E   0000   P   LDX   #P$SCRN  LOAD A(TDB)
01950 02813P 00D5 AF   48          A   STX   DV$TDB,U  ISSUE IT
01955 02814P 00D7 9E   07          A   LDX   T$.FIR  LOAD ADDRESS OF THIS INSTANCE
01960 02815P 00D9 AF   46          A   STX   DV$TCB,U AND ISSUE IT
01965 02816P 00DB 86   01          A   LDA   #1      LOAD MAXIMUM NUMBER OF LOGICAL UNITS
01970 02817P 00DD A7   4D          A   STA   DV$NLU,U  ISSUE IT
01975 02818P 00DF 1F   B8          A   TFR   DP,A
01980 02819P 00E1 C6   1C          A   LDB   #CT$SCR  FORM ADDRESS OF INPUT CHAIN
01985 02820P 00E3 ED   C8 2E   A   STD   DV$IPC,U AND ISSUE AS INPUT PACKET CHAIN
01990 02821P 00E6 39          RTS          RETURN
```

02000 02823

* K ROUTINE TO RELEASE PACKET BUFFERS WHEN THIS INSTANCE DELETED

```
02010 02825P 00E7          ARKILL RABX  ,FLSHBF,#,7  RELEASE FLASH ZONE PACKET BUFFER
02015 02826P 00ED          EXAR   .          AND EXIT
```

```

02030 02829                *   MAINLINE ENTRY POINT

02040 02831P 00EF                SCRDRV TDEL  18,N   SET UP FOR 3 FLASHES PER SECOND

02050 02833P 00F4                SCR10  LADP  CT$SCR,X POINT TO INPUT CHAIN
02055 02834P 00FC                RFCX    , ,R    AND SEE IF ANY PACKETS THERE
02060 02835P 0100 26  0F  0111    BNE    INPHAS  BRANCH IF SO
02065 02836P 0102 BD  0457  P     JSR    INVCUR  ELSE, FLASH THE CURSOR
02070 02837P 0105 03  10    A     COM    PHASE    SHOW NEW PHASE
02075 02838P 0107 8D  41    014A   BSR    FLSHSC  AND ANY ZONES ON THE SCREEN
02080 02839P 0109                MOVE   ,ME, ,ELAPCH AND WAIT FOR A TIMEOUT TO WAKE US UP
02085 02840P 010F 20  E3    00F4   BRA    SCR10  THEN GO CHECK THE CHAIN AGAIN

02095 02842P 0111 9F  06    A INPHAS STX  PACKET  SAVE THE PACKET ADDRESS
02100 02843P 0113 0D  10    A     TST    PHASE    TEST THE CURSOR PHASE
02105 02844P 0115 27  06    011D   BEQ    NOINV  IF CLEAR, INVERT FLASH ZONES
02110 02845P 0117 0F  10    A     CLR    PHASE    ELSE, INVERT CURSOR
02115 02846P 0119 BD  0457  P     JSR    INVCUR
02120 02847P 011C    8C    A     FCB    $8C    AND SKIP OVER THE BSR

02130 02849P 011D 8D  2B    014A NOINV BSR    FLSHSC  BRING FLASH ZONES INTO PHASE WITH CURSOR
02135 02850P 011F 9E  06    A     LDX    PACKET  POINT TO COMMAND PACKET

02145 02852P 0121 9F  06    A GOTONE STX  PACKET  SAVE ADDRESS OF PACKET
02150 02853P 0123 E6  0A    A     LDB    VD$CMD,X PICKUP THE COMMAND
02155 02854P 0125 C1  13    A     CMPB  #SLOC  IS IT VALID?
02160 02855P 0127 24  1B    0144   BHS    BADPKT IGNORE THIS PACKET IF NOT
02165 02856P 0129 58                ASLB                   ELSE, FORM OFFSET INTO TABLE
02170 02857P 012A CE  001D  P     LDU    #TBL  POINT TO TABLE
02175 02858P 012D AD  D5    A     JSR    [B,U] AND GO DO ROUTINE
02180 02859P 012F BD  027A  P     JSR    DOGETC RETURN THE CURSOR POSITION IN PACKET

02190 02861P 0132                RELPKT LADP  CT$SCR,X POINT TO PACKET CHAIN
02195 02862P 013A                RRNX    ,    AND RELEASE THE PROCESSED PACKET
02200 02863P 013C 24  E3    0121   BCC    GOTONE IF MORE THERE, PROCESS THEM
02205 02864P 013E 8D  0A    014A   BSR    FLSHSC  ELSE, ENSURE THAT CURSOR AND ZONES ARE OUT OF PHASE
02210 02865P 0140                NEXT   .    WAIT A WHILE
02215 02866P 0142 20  B0    00F4   BRA    SCR10  AND GO CHECK THE CHAIN AGAIN

02225 02868P 0144 C6  FF    A BADPKT LDB  #$FF  SHOW COMMAND ILLEGAL
02230 02869P 0146 E7  0A    A     STB    VD$CMD,X
02235 02870P 0148 20  E8    0132   BRA    RELPKT AND GO RELEASE THE PACKET

```

```

02250 02873          *      TASK  SUBROUTINES.

02260 02875          *  ROUTINE TO FLASH ANY ZONES PRESENT ON THE SCREEN

02270 02877          *  PACKETS ON CT$FLS DESCRIBE THE ZONE AS FOLLOWS

02280 02879          *      BYTES 6-7 = ADDRESS OF FIRST CHARACTER IN ZONE
02285 02880          *      BYTE   8 = NUMBER OF CHARACTERS IN ZONE

02295 02882P 014A          FLSHSC LADP   CT$FLS,U POINT TO CHAIN
02300 02883P 0152 DF  26    A          STU   STMP   AND SAVE ADDRESS

02310 02885P 0154 EE  42    A FLSH10 LDU   L.NEXT,U POINT TO NEXT PACKET
02315 02886P 0156 1193 26    A          CMPU  STMP   IS IT CHAIN HEADER?
02320 02887P 0159 27  09  0164        BEQ   FLSH20  EXIT IF SO
02325 02888P 015B AE  46    A          LDX   6,U   ELSE, POINT TO ZONE
02330 02889P 015D E6  48    A          LDB   8,U   GET CHARACTER COUNT
02335 02890P 015F BD  0328  P          JSR   DGIN10  AND GO INVERT THE ZONE
02340 02891P 0162 20  F0  0154        BRA   FLSH10  AND LOOP

02350 02893P 0164 39          FLSH20 RTS          AND RETURN

```



```

02365 02896                * CURSOR OPERATIONS

02375 02898                * CURSOR UP

02385 02900P 0165 BD  044B  P CURUP JSR   CHKSIZ  CHECK CURRENT SCREEN SIZE
02390 02901P 0168 25  11  017B      BLO   CURU10  EXIT IF LESS THAN ONE LINE
02395 02902P 016A 109F 0C   A       STY   OLDCUR  ELSE, UPDATE BACKTRACK CURSOR
02400 02903P 016D 31  A8 B0  A       LEAY  -80,Y  MOVE CURSOR UP A LINE
02405 02904P 0170 109C 11   A       CMPY  TOPSCR  STILL ON SCREEN?
02410 02905P 0173 24  06  017B      BHS   CURU10  IF SO, EXIT
02415 02906P 0175 DC  13   A       LDD   BOTSCR  ELSE, FORM OFFSET INTO BOTTOM LINE
02420 02907P 0177 93  11   A       SUBD  TOPSCR
02425 02908P 0179 31  AB   A       LEAY  D,Y    AND WRAP AROUND

02435 02910P 017B 39                CURU10 RTS          AND RETURN

02445 02912                * CURSOR DOWN

02455 02914P 017C BD  044B  P CURDWN JSR   CHKSIZ  CHECK CURRENT SIZE OF SCREEN
02460 02915P 017F 25  11  0192      BLO   CURD10  EXIT IF LESS THAN ONE LINE
02465 02916P 0181 109F 0C   A       STY   OLDCUR  ELSE, UPDATE OLD CURSOR
02470 02917P 0184 31  A8 50  A       LEAY  80,Y  AND MOVE CURSOR DOWN ONE LINE
02475 02918P 0187 109C 13   A       CMPY  BOTSCR  OFF SCREEN?
02480 02919P 018A 25  06  0192      BLO   CURD10  IF NOT, OK
02485 02920P 018C DC  11   A       LDD   TOPSCR
02490 02921P 018E 93  13   A       SUBD  BOTSCR  ELSE, FORM OFFSET INTO TOP LINE
02495 02922P 0190 31  AB   A       LEAY  D,Y    AND WRAP AROUND

02505 02924P 0192 39                CURD10 RTS          RETURN

02515 02926                * CURSOR LEFT

02525 02928P 0193 109F 0C   A  CURLFT STY   OLDCUR  UPDATE COPY
02530 02929P 0196 109C 11   A       CMPY  TOPSCR  OFF SCREEN?
02535 02930P 0199 26  03  019E      BNE   CURL10  IF NOT, OK
02540 02931P 019B 109E 13   A       LDY   BOTSCR  ELSE, WRAP IT AROUND

02550 02933P 019E 31  3F   A  CURL10 LEAY  -1,Y  MOVE CURSOR LEFT ONE POSITION
02555 02934P 01A0 39                RTS          AND RETURN

```

```

02570 02937                *  CURSOR RIGHT

02580 02939P 01A1 109F 0C    A CURRYT STY    OLDCUR    UPDATE COPY
02585 02940P 01A4 31  21    A          LEAY  1,Y      MOVE CURSOR RIGHT ONE
02590 02941P 01A6 109C 13    A          CMPY  BOTSCR
02595 02942P 01A9 25  03    01AE       BLO   CURR10    EXIT IF NOT OFF SCREEN
02600 02943P 01AB 109E 11    A          LDY   TOPSCR    ELSE, WRAP AROUND

02610 02945P 01AE 39                CURR10 RTS                AND RETURN

02620 02947                *  SEND CURSOR HOME

02630 02949P 01AF 109F 0C    A CURHOM STY    OLDCUR    UPDATE COPY
02635 02950P 01B2 109E 11    A          LDY   TOPSCR    AND POINT TO "HOME" POSITION
02640 02951P 01B5 39                RTS                AND RETURN

02650 02953                *  RESTORE CURSOR TO PREVIOUS POSITION

02660 02955P 01B6 109E 0C    A DOREST LDY    OLDCUR    PICK UP BACKTRACK COPY
02665 02956P 01B9 39                RTS                AND RETURN

```

02680 02959 * CHARACTER/LINE INSERTION AND DELETION FUNCTIONS

02690 02961 * INSERT A CHARACTER

02700	02963P	01BA	BD	0374	P	DOIC	JSR	NEWPOS	GET THE INSERTION POSITION
02705	02964P	01BD	9E	08	A	LDX	LINE	POINT TO LINE FOR INSERTION	
02710	02965P	01BF	30	88 4E	A	LEAX	78,X	POINT TO END OF LINE	
02715	02966P	01C2	9C	13	A	CMPX	BOTSCR	ELSE, ARE WE STILL ON SCREEN?	
02720	02967P	01C4	23	04	01CA	BLS	DOIC10	BRANCH IF SO	
02725	02968P	01C6	9E	13	A	LDX	BOTSCR	ELSE, POINT TO END OF SCREEN	
02730	02969P	01C8	30	1E	A	LEAX	-2,X		
02740	02971P	01CA	9C	0A	A	DOIC10	CMPX	POSN	INSERTION AT END OF LINE?
02745	02972P	01CC	25	14	01E2	BLO	DOIC40	IF SO, NO MOVES ARE NECESSARY	
02755	02974P	01CE	A6	84	A	DOIC20	LDA	0,X	GET A CHARACTER
02760	02975P	01D0	84	7F	A	ANDA	#\$7F	STRIP OFF THE INVERSION	
02765	02976P	01D2	6D	01	A	TST	1,X	ARE WE MOVING TO AN INVERTED SCREEN?	
02770	02977P	01D4	2A	02	01D8	BPL	DOIC30	BRANCH IF NOT	
02775	02978P	01D6	8A	80	A	ORA	#\$80	ELSE, INVERT THE CHARACTER	
02785	02980P	01D8	A7	01	A	DOIC30	STA	1,X	ISSUE CHARACTER
02790	02981P	01DA	9C	0A	A	CMPX	POSN	FINISHED?	
02795	02982P	01DC	27	04	01E2	BEQ	DOIC40	EXIT IF SO	
02800	02983P	01DE	30	1F	A	LEAX	-1,X	ELSE, BACKSPACE	
02805	02984P	01E0	20	EC	01CE	BRA	DOIC20	AND LOOP	
02815	02986P	01E2	9E	06	A	DOIC40	LDX	PACKET	POINT TO PACKET
02820	02987P	01E4	A6	0B	A	LDA	VD\$CHR,X	GET CHARACTER TO BE INSERTED	
02825	02988P	01E6	26	02	01EA	BNE	DOIC50	BRANCH IF ONE THERE	
02830	02989P	01E8	86	20	A	LDA	#\$20	ELSE, ISSUE A SPACE	
02840	02991P	01EA	9E	0A	A	DOIC50	LDX	POSN	POINT TO POSITION FOR INSERT
02845	02992P	01EC	BD	03E1	P	JSR	ISCHAR	AND ISSUE THE CHARACTER	
02850	02993P	01EF	109E	0A	A	LDY	POSN	AND SET THE CURSOR ON IT	
02855	02994P	01F2	39			RTS			

02870 02997

* DELETE A CHARACTER

02880	02999P	01F3	BD	0374	P	DODC	JSR	NEWPOS	POINT TO POSITION FOR DELETION
02885	03000P	01F6	BD	03A1	P		JSR	TOPOS	ADVANCE TO POSITION FOR DELETE
02890	03001P	01F9	5D				TSTB		DELETE LAST CHARACTER IN LINE?
02895	03002P	01FA	27	15	0211		BEQ	DODC30	BRANCH IF SO
02905	03004P	01FC	5A			DODC10	DECB		ELSE, COUNT OF A CHARACTER
02910	03005P	01FD	27	12	0211		BEQ	DODC30	EXIT IF DONE
02915	03006P	01FF	A6	01	A		LDA	1,X	GET A CHARACTER
02920	03007P	0201	84	7F	A		ANDA	#\$7F	REMOVE INVERSION
02925	03008P	0203	6D	84	A		TST	0,X	IS DESTINATION INVERTED?
02930	03009P	0205	2A	02	0209		BPL	DODC20	BRANCH IF NOT
02935	03010P	0207	8A	80	A		ORA	#\$80	ELSE, INVERT THE CHARACTER
02945	03012P	0209	A7	80	A	DODC20	STA	0,X+	
02950	03013P	020B	9C	13	A		CMPX	BOTSCR	OFF SCREEN?
02955	03014P	020D	26	ED	01FC		BNE	DODC10	LOOP IF NOT
02960	03015P	020F	30	1F	A		LEAX	-1,X	ELSE, BACKSPACE
02970	03017P	0211	86	20	A	DODC30	LDA	#\$20	ISSUE A SPACE TO END OF LINE
02975	03018P	0213	BD	03E1	P		JSR	ISCHAR	
02980	03019P	0216	109E	0A	A		LDY	POSN	PICK UP NEW CURSOR
02985	03020P	0219	39				RTS		AND RETURN

03000 03023

* INSERT A LINE

```
03010 03025P 021A BD 044B P DOIL JSR CHKSIZ CHECK THE SCREEN SIZE
03015 03026P 021D 23 32 0251 BLS DOIL20 IF LESS THAN 2 LINES, ILLEGAL
03020 03027P 021F BD 0374 P JSR NEWPOS ELSE, FORM ADDRESS OF LINE
03025 03028 0000 A IFEQ SHORT
03030 03029P 0222 DE 06 A LDU PACKET POINT TO PACKET
03035 03030P 022A E6 4B A LDB VD$CHR,U IS THERE A BUFFER FOR TOP LINE?
03040 03031P 0226 C1 FF A CMPB #-1
03045 03032P 0228 27 18 0242 BEQ DOIL10 BRANCH IF NOT
03050 03033P 022A 9E 11 A LDX TOPSCR ELSE, POINT TO TOP LINE
03055 03034P 022C DC 13 A LDD BOTSCR
03060 03035P 022E 34 06 A PSHS D SAVE CURRENT END OF SCROLL
03065 03036P 0230 30 88 50 A LEAX 80,X AND SET IT AS END OF LINE
03070 03037P 0233 9F 13 A STX BOTSCR
03075 03038P 0235 9E 11 A LDX TOPSCR THEN POINT TO TOP LINE OF SCREEN
03080 03039P 0237 33 4C A LEAU VD$BUF,U POINT TO START OF READ BUFFER
03085 03040P 0239 C6 50 A LDB #80 MOVE 80 CHARACTERS
03090 03041P 023B BD 045E P JSR SCREAD AND MOVE THE BUFFER TO THE SPACE WE'VE MADE
03095 03042P 023E 35 06 A PULS D
03100 03043P 0240 DD 13 A STD BOTSCR RESTORE END OF SCROLL
03105 03044 ENDC

03115 03046P 0242 9E 11 A DOIL10 LDX TOPSCR POINT TO TOP LINE
03120 03047P 0244 DE 08 A LDU LINE POINT TO LINE
03125 03048P 0246 33 C8 50 A LEAU 80,U
03130 03049P 0249 DF 24 A STU SCREND ISSUE END OF SCROLL
03135 03050P 024B BD 0529 P JSR SCROLL AND SCROLL THE SCREEN
03140 03051P 024E 109E 0A A LDY POSN UPDATE CURSOR

03150 03053P 0251 39 DOIL20 RTS AND RETURN
```

03165 03056

* DELETE A LINE

```
03175 03058P 0252 BD 044B P DODL JSR CHKSIZ CHECK THE SCREEN SIZE
03180 03059P 0255 23 22 0279 BLS DODL10 EXIT IF LESS THAN 2 LINES
03185 03060P 0257 BD 0374 P JSR NEWPOS ELSE, FORM POINTERS TO LINE
03190 03061P 025A 109E 0A A LDY POSN UPDATE CURSOR
03195 03062P 025D 9E 08 A LDX LINE POINT TO LINE TO BE DELETED
03200 03063P 025F DE 13 A LDU BOTSCR POINT TO BOTTOM OF SCROLL
03205 03064P 0261 DF 24 A STU SCREND ISSUE END OF SCROLL
03210 03065P 0263 BD 0529 P JSR SCROLL AND DELETE THE LINE
03215 03066 0000 A IFEQ SHORT
03220 03067P 0266 DE 06 A LDU PACKET POINT TO STRING
03225 03068P 0268 E6 4B A LDB VD$CHR,U IS THERE A NEW LINE TO GO ON THE BOTTOM?
03230 03069P 026A C1 FF A CMPB #-1
03235 03070P 026C 27 0B 0279 BEQ DODL10 EXIT IF NOT
03240 03071P 026E 33 4C A LEAU VD$BUF,U POINT TO BUFFER
03245 03072P 0270 9E 18 A LDX BOTLIN ELSE, POINT TO BOTTOM LINE
03250 03073P 0272 C6 50 A LDB #80 MOVING 80 CHARACTERS
03255 03074P 0274 D7 1B A STB WRTFLG SHOW SCREEN ALTERED
03260 03075P 0276 BD 048D P JSR SCRVRT AND MOVE THE NEW LINE IN
03265 03076 ENDC
```

03275 03078P 0279 39

DODL10 RTS

```

03290 03081                * RETURN CURRENT CURSOR AS COLUMN/ROW CO-ORDINATES

03300 03083P 027A 1F 20    A DOGETC TFR   Y,D    PLACE CURSOR IN D
03305 03084P 027C 9E 06    A          LDX   PACKET  POINT TO THE PACKET
03310 03085P 027E 6F 06    A          CLR   VD$POS,X CLEAR ROW CO-ORDINATE (LINE NUMBER)
03315 03086P 0280 93 16    A          SUBD  TLC     AND CHANGE CURSOR TO OFFSET INTO SCREEN

03325 03088P 0282 83 0050  A DOGE10 SUBD  #80     SUBTRACT A LINE FROM OFFSET
03330 03089P 0285 25 04 028B BCS     DOGE20  IF FINISHED, BRANCH
03335 03090P 0287 6C 06    A          INC   VD$POS,X ELSE, ADVANCE LINE NUMBER
03340 03091P 0289 20 F7 0282  BRA     DOGE10  AND LOOP

03350 03093P 028B C3 0050  A DOGE20 ADDD  #80     RESTORE FROM EXTRA SUB
03355 03094P 028E E7 07    A          STB   VD$POS+1,X AND ISSUE OFFSET INTO LINE
03360 03095P 0290 39                RTS

03370 03097                * ERASE TO END OF LINE

03380 03099P 0291 BD 0374  P DOEEOL JSR   NEWPOS  FORM POSITION ON SCREEN
03385 03100P 0294 9E 08    A          LDX   LINE    POINT TO LINE
03390 03101P 0296 30 88 50  A          LEAX  80,X    POINT TO NEXT LINE
03395 03102P 0299 9C 13    A          CMPX  BOTSCR  OF SCREEN?
03400 03103P 029B 23 02 029F BLS     DOEE10  BRANCH IF NOT
03405 03104P 029D 9E 13    A          LDX   BOTSCR  ELSE, USE END OF SCREEN

03415 03106P 029F 9F 08    A DOEE10 STX   LINE    ISSUE LINE ADDRESS
03420 03107P 02A1 9E 0A    A          LDX   POSN    POINT TO START POSITION
03425 03108P 02A3 86 20    A          LDA   #$20   LOAD A SPACE
03430 03109P 02A5 97 1B    A          STA   WRNFLG  AND SHOW SCREEN ALTERED

03440 03111P 02A7 A7 80    A DOEE20 STA   0,X+   ISSUE A SPACE
03445 03112P 02A9 9C 08    A          CMPX  LINE    FINISHED?
03450 03113P 02AB 25 FA 02A7 BLO     DOEE20  LOOP IF NOT
03455 03114P 02AD 109E 0A  A          LDY   POSN    ELSE, UPDATE CURSOR
03460 03115P 02B0 39                RTS     AND RETURN

```

03475 03118

* POSITION THE CURSOR ON A CHARACTER

03485 03120P 02B1 BD 0374 P DOPOS JSR NEWPOS FORM THE POINTERS
03490 03121P 02B4 9E 06 A LDX PACKET POINT TO PACKET
03495 03122P 02B6 A6 0B A LDA VD\$CHR,X AND GET THE CHARACTER TO BE ISSUED
03500 03123P 02B8 27 05 02BF BEQ DOPO10 BRANCH IF NONE THERE
03505 03124P 02BA 9E 0A A LDX POSN ELSE, POINT TO THE POSITION
03510 03125P 02BC BD 03AF P JSR ISSUE AND ISSUE THE CHARACTER

03520 03127P 02BF 109E 0A A DOPO10 LDY POSN UPDATE THE CURSOR
03525 03128P 02C2 0D 1A A TST DIRFLG INCREMENT THE CURSOR?
03530 03129P 02C4 27 0A 02D0 BEQ DOPO20 EXIT IF NOT
03535 03130P 02C6 31 21 A LEAY 1,Y ELSE, ADVANCE THE CURSOR
03540 03131P 02C8 109C 13 A CMPY BOTSCR OFF SCREEN?
03545 03132P 02CB 26 03 02D0 BNE DOPO20 EXIT, IF NOT
03550 03133P 02CD 109E 0A A LDY POSN ELSE, RESTORE CURSOR

03560 03135P 02D0 39 DOPO20 RTS AND RETURN

03570 03137

* SET THE TOP AND BOTTOM SCROLL LIMITS TO NEW MARGINS

03580 03139P 02D1 BD 0374 P DOSETM JSR NEWPOS CALCULATE THE POSITION FOR TOP OF SCROLL
03585 03140P 02D4 9E 0A A LDX POSN POINT TO TOP OF SCROLL
03590 03141P 02D6 9F 11 A STX TOPSCR AND ISSUE IT
03595 03142P 02D8 9E 06 A LDX PACKET
03600 03143P 02DA EE 08 A LDU VD\$1ST,X GET CO-ORDINATES FOR BOTTOM OF SCROLL
03605 03144P 02DC EF 06 A STU VD\$POS,X
03610 03145P 02DE BD 0374 P JSR NEWPOS CALCULATE BOTTOM OF SCROLL
03615 03146P 02E1 9E 0A A LDX POSN
03620 03147P 02E3 9F 13 A STX BOTSCR ISSUE IT
03625 03148P 02E5 9E 08 A LDX LINE POINT TO START OF LAST LINE
03630 03149P 02E7 30 88 B0 A LEAX -80,X
03635 03150P 02EA 9C 11 A CMPX TOPSCR OF SCREEN?
03640 03151P 02EC 24 02 02F0 BHS DOSE10 BRANCH IF NOT
03645 03152P 02EE 9E 11 A LDX TOPSCR ELSE, USE TOP OF SCROLL

03655 03154P 02F0 9F 18 A DOSE10 STX BOTLIN ISSUE ADDRESS OF BOTTOM LINE
03660 03155P 02F2 109E 11 A LDY TOPSCR SET CURSOR TO HOME
03665 03156P 02F5 39 RTS AND RETURN
03670 03157 0000 A IFEQ SHORT

03685 03160

* READ A NUMBER OF CHARACTERS FROM THE SCREEN COMPRESSING SPACES

03695 03162P 02F6 8D 0F 0307 DOREAD BSR SETIO SET UP FOR THE READ
03700 03163P 02F8 BD 045E P JSR SCREAD DO THE READ
03705 03164P 02FB 20 05 0302 BRA DOWR10 AND EXIT

03715 03166

* WRITE A STRING TO THE SCREEN DECOMPRESSING SPACES

03725 03168P 02FD 8D 08 0307 DOWRIT BSR SETIO SET UP FOR THE WRITE
03730 03169P 02FF BD 048D P JSR SCRVRT DO THE WRITE
03740 03171P 0302 35 10 A DOWR10 PULS X
03745 03172P 0304 9F 13 A STX BOTSCR RESTORE THE BOTTOM OF SCROLL LIMIT
03750 03173P 0306 39 RTS AND EXIT

03760 03175

* ROUTINE TO SET UP FOR A READ OR WRITE ON SCREEN

03770 03177P 0307 BD 038F P SETIO JSR GETPOS FORM POSITIONS ON SCREEN
03775 03178P 030A 9E 13 A LDX BOTSCR
03780 03179P 030C 35 06 A PULS D GET THE RETURN ADDRESS
03785 03180P 030E 34 16 A PSHS X,D AND SAVE BOTTOM OF SCROLL ABOVE RETURN ADDRESS
03790 03181P 0310 9E 08 A LDX LINE POINT TO LINE FOR FUNCTION
03795 03182P 0312 30 88 50 A LEAX 80,X POINT TO NEXT LINE
03800 03183P 0315 9F 13 A STX BOTSCR AND MAKE IT END OF SCREEN
03805 03184P 0317 DE 06 A LDU PACKET
03810 03185P 0319 E6 4B A LDB VD\$CHR,U LOAD THE NUMBER OF CHARACTERS FOR READ
03815 03186P 031B 33 4C A LEAU VD\$BUF,U POINT TO LINE BUFFER
03820 03187P 031D 9E 0A A LDX POSN POINT TO POSITION FOR READ OR WRITE
03825 03188P 031F 39 RTS AND RETURN WITH OLD BOTSCR ON STACK
03830 03189 ENDC

03845 03192

* INVERT A STRING OF CHARACTERS

03855 03194P 0320 8D 52 0374 DOINV BSR NEWPOS FORM POSITION ON SCREEN
03860 03195P 0322 9E 06 A LDX PACKET POINT TO PACKET
03865 03196P 0324 E6 0B A LDB VD\$CHR,X GET CHARACTER COUNT
03870 03197P 0326 9E 0A A LDX POSN POINT TO FIRST CHARACTER

03880 03199P 0328 A6 84 A DOIN10 LDA 0,X GET A CHARACTER
03885 03200P 032A 88 80 A EORA #\$80 INVERT IT
03890 03201P 032C A7 80 A STA 0,X+ AND RE-ISSUE IT
03895 03202P 032E 9C 13 A CMPX BOTSCR OF SCREEN?
03900 03203P 0330 27 03 0335 BEQ DOIN20 EXIT IF SO
03905 03204P 0332 5A DECB ELSE, DEC THE COUNT
03910 03205P 0333 26 F3 0328 BNE DOIN10 AND LOOP

03920 03207P 0335 39 DOIN20 RTS

03930 03209

* MARK A ZONE OF THE SCREEN FOR FLASHING

03940 03211P 0336 8D 3C 0374 DOFLSH BSR NEWPOS CALCULATE POSITION OF ZONE
03945 03212P 0338 GPBX ,FLSHBF GET A FLASH PACKET
03950 03213P 033C 29 16 0354 BVS DOFL10 EXIT IF NONE THERE
03955 03214P 033E DC 0A A LDD POSN ELSE, PICKUP THE POSITION OF THE ZONE
03960 03215P 0340 ED 06 A STD 6,X AND ISSUE IT TO PACKET
03965 03216P 0342 DE 06 A LDU PACKET POINT TO INPUT PACKET
03970 03217P 0344 E6 4B A LDB VD\$CHR,U AND PICK UP THE CHARACTER COUNT
03975 03218P 0346 E7 08 A STB 8,X ISSUE IT
03980 03219P 0348 1F 10 A TFR X,D PLACE ADDRESS OF FLASH PACKET INTO D
03985 03220P 034A LADP CT\$FLS,X POINT TO FLASH CHAIN
03990 03221P 0352 LSCX ,,, AND LINK THE PACKET ONTO IT

04000 03223P 0354 39 DOFL10 RTS

04010 03225

* STOP A ZONE FLASHING

04020 03227P 0355 8D 1D 0374 DOSTFL BSR NEWPOS POINT TO POSITION ON SCREEN
04025 03228P 0357 LADP CT\$FLS,X POINT TO FLASH CHAIN
04030 03229P 035F 34 10 A PSHS X
04035 03230P 0361 DC 0A A LDD POSN GET POSITION

04045 03232P 0363 AE 02 A DOST10 LDX L.NEXT,X POINT TO NEXT PACKET
04050 03233P 0365 AC E4 A CMPX 0,S BACK TO CHAIN HEADER
04055 03234P 0367 27 09 0372 BEQ DOST20 EXIT IF SO
04060 03235P 0369 10A3 06 A CMPD 6,X ELSE, IS THIS THE ZONE WE WANT?
04065 03236P 036C 26 F5 0363 BNE DOST10 LOOP IF NOT
04070 03237P 036E RRNX ,(L.PREV,X) ELSE, UNLINK THE PACKET

04080 03239P 0372 35 90 A DOST20 PULS X,PC AND RETURN

```

04095 03242                * DRIVER UTILITY ROUTINES

04105 03244                * ROUTINE TO CONVERT TO ROW/COLUMN CO-ORDINATES TO
04110 03245                * ACTUAL ADDRESSES ON THE SCREEN. IF THE CO-ORDINATES ARE
04115 03246                * $FFFF, THIS ROUTINE USES THE CURRENT CURSOR VALUE

04125 03248P 0374 9E 06    A NEWPOS LDX   PACKET   POINT TO PACKET
04130 03249P 0376 0F 1A    A           CLR   DIRFLG   ASSUME NEW POSITION
04135 03250P 0378 EE 06    A           LDU   VD$POS,X GET CO-ORDINATES
04140 03251P 037A 1183 FFFF A           CMPU  #-1     CURRENT CURSOR?
04145 03252P 037E 26 05 0385 BNE    NEWP10  BRANCH IF NOT
04150 03253P 0380 0A 1A    A           DEC   DIRFLG   ELSE, SHOW USING CURSOR
04155 03254P 0382 BD 027A P           JSR   DOGETC   AND CONVERT THE CURSOR TO CO-ORDINATES

04165 03256P 0385 8D 08 038F NEWP10 BSR   GETPOS   FORM POSITION AND LINE ADDRESSES
04170 03257P 0387 0D 1A    A           TST   DIRFLG   USING CURRENT CURSOR?
04175 03258P 0389 26 03 038E BNE    NEWP20  EXIT IF SO
04180 03259P 038B 109F 0C    A           STY   OLDCUR   ELSE, UPDATE TRACKING CURSOR

04190 03261P 038E 39                NEWP20 RTS

04200 03263                * ROUTINE TO CONVERT CO-ORDINATES TO SCREEN ADDRESSES
04205 03264                * PRODUCES START OF LINE IN WHICH CO-ORDINATES ARE FOUND
04210 03265                * ADDRESS OF THE POSITION POINTED TO BY CO-ORDINATES

04220 03267P 038F 9E 06    A GETPOS LDX   PACKET   POINT TO PACKET
04225 03268P 0391 E6 06    A           LDB   VD$POS,X GET LINE NUMBER
04230 03269P 0393 86 50    A           LDA   #80     80 CHARACTERS PER LINE
04235 03270P 0395 3D                MUL                FORM OFFSET INTO SCREEN
04240 03271P 0396 D3 16    A           ADDD  TLC     FORM ADDRESS OF LINE IN SCREEN
04245 03272P 0398 DD 08    A           STD   LINE    AND ISSUE IT
04250 03273P 039A EB 07    A           ADDB  VD$POS+1,X ADD COLUMN NUMBER
04255 03274P 039C 89 00    A           ADCA  #0
04260 03275P 039E DD 0A    A           STD   POSN    AND ISSUE IT AS POSITION ON SCREEN
04265 03276P 03A0 39                RTS

04275 03278                * ROUTINE TO STEP ALONG A LINE TO THE POSN REQUIRED, RETURNING
04280 03279                * THE NUMBER OF CHARACTERS LEFT IN THAT LINE

04290 03281P 03A1 9E 08    A TOPOS  LDX   LINE    POINT TO START OF LINE
04295 03282P 03A3 C6 50    A           LDB   #80     80 CHARACTERS IN LINE

04305 03284P 03A5 9C 0A    A TOPO10 CMPX  POSN    THERE YET?
04310 03285P 03A7 27 05 03AE           BEQ   TOPO20  EXIT IF SO
04315 03286P 03A9 30 01    A           LEAX  1,X     ELSE, ADVANCE POINTER
04320 03287P 03AB 5A                DECB                SHOW ONE LESS
04325 03288P 03AC 26 F7 03A5           BNE   TOPO10  AND LOOP

04335 03290P 03AE 39                TOPO20 RTS                RETURN

```

```

04350 03293                *  ROUTINE TO ISSUE A CHARACTER TO THE SCREEN

04360 03295P 03AF 81  7F      A  ISSUE  CMPA  #$7F      ELSE, IS IT A DEL?
04365 03296P 03B1 27  1F      03D2      BEQ  RUB      IF SO, HANDLE RUBOUT
04370 03297P 03B3 81  20      A          CMPA  #$20      ELSE, IS IT A CONTROL CHARACTER
04375 03298P 03B5 25  0C      03C3      BCS  SPEC      IF SO, HANDLE IT
04380 03299P 03B7 4D                TSTA                COMPRESSED SPACES?
04385 03300P 03B8 2B  03      03BD      BMI  CMPSPC     IF SO, EXPAND THEM
04390 03301P 03BA 8D  25      03E1      BSR  ISCHAR     ELSE, ISSUE THE CHARACTER
04395 03302P 03BC 39                RTS                RETURN

04405 03304P 03BD BD  04DC     P  CMPSPC  JSR  EXPAND     EXPAND COMPRESSED SPACES
04410 03305P 03C0 9F  0A      A          STX  POSN      UPDATE POINTER
04415 03306P 03C2 39                RTS                AND RETURN

04425 03308P 03C3 0F  1A      A  SPEC  CLR  DIRFLG     SHOW SPECIAL CHARACTER RECIEVED
04430 03309P 03C5 81  0D      A          CMPA  #$0D      IS IT CR?
04435 03310P 03C7 27  23      03EC      BEQ  SPCCR
04440 03311P 03C9 81  0A      A          CMPA  #$0A
04445 03312P 03CB 27  2A      03F7      BEQ  SPCRLF     HANDLE LINE FEED
04450 03313P 03CD 81  0C      A          CMPA  #$0C
04455 03314P 03CF 27  5F      0430      BEQ  SPCFF      HANDLE FORM FEED

04465 03316P 03D1 39                SPEC10 RTS        ELSE, IGNOR IT

04475 03318P 03D2 0F  1A      A  RUB  CLR  DIRFLG     SHOW RUBOUT OCCURRED
04480 03319P 03D4 9C  11      A          CMPX  TOPSCR     AT TOP OF SCREEN?
04485 03320P 03D6 27  F9      03D1      BEQ  SPEC10     IF SO, CAN'T DO RUB
04490 03321P 03D8 30  1F      A          LEAX  -1,X      ELSE, BACKSPACE CURSOR
04495 03322P 03DA 86  20      A          LDA  #$20      PICKUP A SPACE
04500 03323P 03DC 8D  03      03E1      BSR  ISCHAR     AND ISSUE IT
04505 03324P 03DE 9F  0A      A          STX  POSN      UPDATE POINTER
04510 03325P 03E0 39                RTS                AND RETURN

```

```

04520 03327                * ISSUE CHARACTER IN A TO SCREEN POSITION IN X

04530 03329P 03E1 6D 84      A ISCHAR TST    0,X      IS DESTINATION INVERTED?
04535 03330P 03E3 2A 02 03E7      BPL ISCHAL  BRANCH IF NOT
04540 03331P 03E5 88 80      A      EORA  #$80    ELSE, INVERT CHARACTER

04550 03333P 03E7 A7 84      A ISCHAL STA    0,X      ISSUE THE CHARACTER
04555 03334P 03E9 97 1B      A      STA  WRTFLG  AND SHOW THAT SCREEN ALTERED
04560 03335P 03EB 39          RTS

04570 03337                * ISSUE <CR>,<LF>,<<LF><CR>>,<FF> CHARACTERS TO SCREEN

04580 03339P 03EC 9E 08      A SPCCR LDX    LINE    POINT TO START OF LINE
04585 03340P 03EE 8D 5B 044B      BSR  CHKSIZ  CHECK SCREEN SIZE
04590 03341P 03F0 24 02 03F4      BHS  SPCC10  BRANCH IF ONE LINE OR MORE
04595 03342P 03F2 9E 11      A      LDX    TOPSCR  ELSE, USE HOME POSITION

04605 03344P 03F4 9F 0A      A SPCC10 STX    POSN    UPDATE POINTER
04610 03345P 03F6 39          RTS      AND RETURN

04620 03347P 03F7 8D F3 03EC SPCRLF BSR    SPCCR  DO A <CR> FIRST

04630 03349P 03F9 8D 50 044B SPCLF  BSR    CHKSIZ  IS SCREEN LESS THAN 2 LINES?
04635 03350P 03FB 23 09 0406      BLS  SPCL10  EXIT IF NOT
04640 03351P 03FD 30 88 50  A      LEAX  80,X
04645 03352P 0400 9C 13      A      CMPX  BOTSCR  ELSE, ARE WE IN BOTTOM LINE?
04650 03353P 0402 24 03 0407      BHS  SPCL20  IF SO, SCROLL
04655 03354P 0404 9F 0A      A      STX    POSN    ELSE, ISSUE NEW POSITION

04665 03356P 0406 39          SPCL10 RTS      AND RETURN

04675 03358P 0407 9E 11      A SPCL20 LDX    TOPSCR  POINT TO TOP OF SCROLL
04680 03359          0000  A      IFEQ  SHORT
04685 03360P 0409 DE 06      A      LDU    PACKET  POINT TO PACKET
04690 03361P 040B E6 48      A      LDB  VD$1ST,U SAVE TOP LINE?
04695 03362P 040D C1 FF      A      CMPB  #-1
04700 03363P 040F 27 17 0428      BEQ  SPCL30  BRANCH IF NOT
04705 03364P 0411 DC 13      A      LDD  BOTSCR
04710 03365P 0413 34 06      A      PSHS  D      SAVE END OF SCROLL
04715 03366P 0415 30 88 50  A      LEAX  80,X
04720 03367P 0418 9F 13      A      STX  BOTSCR  AND SET IS AS NEXT LINE
04725 03368P 041A 9E 11      A      LDX  TOPSCR  POINT TO TOP LINE
04730 03369P 041C C6 50      A      LDB  #80    READ 80 CHARACTERS
04735 03370P 041E 33 4C      A      LEAU  VD$BUF,U POINT TO BUFFER
04740 03371P 0420 8D 3C 045E      BSR  SCREAD  AND DO READ
04745 03372P 0422 35 06      A      PULS  D
04750 03373P 0424 DD 13      A      STD  BOTSCR  RESTORE END OF SCROLL
04755 03374P 0426 9E 11      A      LDX  TOPSCR  POINT TO TOP OF SCROLL

04765 03376          0428  P SPCL30 EQU    *
04770 03377          ENDC
04775 03378P 0428 DE 13      A      LDU  BOTSCR  POINT TO BOTTOM OF SCREEN
04780 03379P 042A DF 24      A      STU  SCREND  ISSUE END OF SCROLL
04785 03380P 042C BD 0529  P      JSR  SCROLL  AND SCROLL THE SCREEN
04790 03381P 042F 39          RTS

```

```

04805 03384P 0430 9E 11      A SPCFF LDX  TOPSCR  POINT TO TOP OF SCREEN
04810 03385P 0432 86 20      A      LDA  #$20   LOAD A SPACE
04815 03386P 0434 97 1B      A      STA  WRTFLG  SHOW SCREEN ALTERED
04820 03387P 0436 9F 0A      A      STX  POSN    ISSUE NEW POSITION

04830 03389P 0438 A7 80      A SPCF10 STA  0,X+   ISSUE A SPACE
04835 03390P 043A 9C 13      A      CMPX  BOTSCR  FINISHED?
04840 03391P 043C 26 FA 0438      BNE  SPCF10  LOOP IF NOT

04850 03393P 043E                SPCF20 LADP  CT$FLS,X POINT TO FLASH CHAIN
04855 03394P 0446                RRNX  ,      RELEASE A FLASH PACKET
04860 03395P 0448 28 F4 043E      BVC  SPCF20  AND LOOP IF MORE THERE
04865 03396P 044A 39                RTS      ELSE, RETURN

```

```

04875 03398                * ROUTINE TO CHECK THE CURRENT SIZE OF THE SCREEN

```

```

04885 03400P 044B 34 06      A CHKSIZ PSHS  D      SAVE D
04890 03401P 044D DC 13      A      LDD  BOTSCR
04895 03402P 044F 93 11      A      SUBD  TOPSCR  CALCULATE SIZE OF SCREEN
04900 03403P 0451 1083 0050  A      CMPD  #80
04905 03404P 0455 35 86      A      PULS  D,PC   RESTORE D AND RETURN

```

```

04915 03406                * ROUTINE TO INVERT THE CURSOR POSITION

```

```

04925 03408P 0457 A6 A4      A INVCUR LDA  0,Y    GET THE CURSOR CHARACTER
04930 03409P 0459 88 80      A      EORA  #$80   INVERT IT
04935 03410P 045B A7 A4      A      STA  0,Y    AND ISSUE IT
04940 03411P 045D 39                RTS

```

```

04955 03414          0000  A      IFEQ  SHORT
04960 03415          *  ROUTINE TO READ FROM THE SCREEN

04970 03417          *  X= ADDRESS ON SCREEN U=ADDRESS OF BUFFER B=NUMBER OF BYTES

04980 03419P 045E 34   20      A SCREAD PSHS  Y      SAVE CURSOR FOR NOW
04985 03420P 0460 32   7B      A      LEAS  -5,S  MAKE SOME WORK SPACE ON STACK
04990 03421P 0462          CLDB  (2,S)  CLEAR ADDRESS OF FIRST NON-BLANK
04995 03422P 0466 6F   64      A      CLR   4,S  CLEAR DATA TYPE MASK
05000 03423P 0468 108E 052D    P      LDY   #ASCTBL-$20 LOAD ADDRESS OF MASK TABLE (OFFSET FOR CONTROLS)

05010 03425P 046C A6   80      A SCRE10 LDA   0,X+  GET A CHARACTER
05015 03426P 046E 84   7F      A      ANDA  #$7F  STRIP OFF ANY INVERSION
05020 03427P 0470 8D   37      04A9  BSR   CHKASC  GO CHECK ITS TYPE
05025 03428P 0472 A7   C0      A      STA   0,U+  ISSUE IT TO BUFFER
05030 03429P 0474 9C   13      A      CMPX  BOTSCR  OFF SCREEN?
05035 03430P 0476 27   03      047B  BEQ   SCRE20  EXIT IF SO
05040 03431P 0478 5A          DECB          ELSE, DEC THE COUNT
05045 03432P 0479 26   F1      046C  BNE   SCRE10  AND LOOP TILL DONE

05055 03434P 047B 9E   06      A SCRE20 LDX   PACKET  POINT TO PACKET
05060 03435P 047D EC   E4      A      LDD   0,S  LOAD A(LAST NON-BLANK CHARACTER)
05065 03436P 047F E7   09      A      STB  VD$1ST,X AND ISSUE IT
05070 03437P 0481 EC   62      A      LDD   2,S  LOAD A(FIRST NON-BLANK CHARACTER)
05075 03438P 0483 E7   08      A      STB  VD$1ST,X AND ISSUE IT
05080 03439P 0485 E6   64      A      LDB  4,S  GET DATA TYPE BIT MASK
05085 03440P 0487 E7   0B      A      STB  VD$MSK,X AND ISSUE IT
05090 03441P 0489 32   65      A      LEAS  5,S  RELEASE WORKSPACE
05095 03442P 048B 35   A0      A      PULS  Y,PC  RESTORE CURSOR AND EXIT

05105 03444          *  ROUTINE TO PERFORM A WRITE TO THE SCREEN

05115 03446P 048D A6   C0      A SCRWR1 LDA   0,U+  GET A CHARACTER
05120 03447P 048F 2A   09      049A  BPL   SCRW10  BRANCH IF NOT COMPRESSED SPACES
05125 03448P 0491 34   04      A      PSHS  B      ELSE, SAVE COUNT
05130 03449P 0493 8D   47      04DC  BSR   EXPAND  EXPAND THEM
05135 03450P 0495 35   04      A      PULS  B      RESTORE COUNT
05140 03451P 0497 EB   5F      A      ADDB  -1,U  AND ADJUST IT FOR THE NUMBER OF SPACES WRITTEN
05145 03452P 0499 5C          INCB

05155 03454P 049A BD   03E1  P SCRW10 JSR   ISCHAR  ISSUE THE CHARACTER TO THE SCREEN
05160 03455P 049D 30   01      A      LEAX  1,X  ADVANCE THE POINTER
05165 03456P 049F 9C   13      A      CMPX  BOTSCR  OFF SCREEN?
05170 03457P 04A1 26   02      04A5  BNE   SCRW20  BRANCH IF NOT
05175 03458P 04A3 30   1F      A      LEAX  -1,X  ELSE, WRITE TO LAST POSITION ON SCREEN

05185 03460P 04A5 5A          SCRW20 DECB          ELSE, DEC THE COUNT
05190 03461P 04A6 26   E5      048D  BNE   SCRWR1  AND LOOP
05195 03462P 04A8 39          RTS          RETURN

```

05210 03465

* ROUTINE TO CHECK DATA FROM SCREEN.

```
05220 03467P 04A9 34 04 A CHKASC PSHS B SAVE THE CHARACTER COUNT
05225 03468P 04AB E6 A6 A LDB A,Y GET THE CHARACTER'S BIT MASK
05230 03469P 04AD EA 67 A ORB 7,S ADJUST MASK
05235 03470P 04AF E7 67 A STB 7,S AND ISSUE IT
05240 03471P 04B1 81 20 A CMPA #$20 IS CHARACTER A SPACE
05245 03472P 04B3 27 0A 04BF BEQ CHKA20 IF SO, BRANCH
05250 03473P 04B5 6D 65 A TST 5,S ELSE, IS THIS FIRST NON-BLANK?
05255 03474P 04B7 26 02 04BB BNE CHKA10 IF NOT, BRANCH
05260 03475P 04B9 AF 65 A STX 5,S ELSE, ISSUE ITS ADDRESS

05270 03477P 04BB AF 63 A CHKA10 STX 3,S UPDATE LAST INSTANCE POINTER
05275 03478P 04BD 35 84 A PULS B,PC AND EXIT

05285 03480P 04BF 8D 02 04C3 CHKA20 BSR CMPRES COMPRESS SPACES
05290 03481P 04C1 35 84 A PULS B,PC RESTORE COUNT AND RETURN

05300 03483P 04C3 86 01 A CMPRES LDAA #1 INITIALISE COUNT

05310 03485P 04C5 9C 13 A CMPR10 CMPX BOTSCR OFF SCREEN?
05315 03486P 04C7 27 0F 04D8 BEQ CMPR20 EXIT IF SO
05320 03487P 04C9 E6 84 A LDB 0,X ELSE, GET THE NEXT CHARACTER
05325 03488P 04CB C4 7F A ANDB #$7F STRIP OFF THE INVERSION
05330 03489P 04CD C1 20 A CMPB #$20 SPACE?
05335 03490P 04CF 26 07 04D8 BNE CMPR20 EXIT IF NOT
05340 03491P 04D1 4C INCA ELSE, ADVANCE SPACE COUNT
05345 03492P 04D2 30 01 A LEAX 1,X AND POINTER
05350 03493P 04D4 6A 62 A DEC 2,S DEC THE COUNT
05355 03494P 04D6 26 ED 04C5 BNE CMPR10 AND LOOP

05365 03496P 04D8 40 CMPR20 NEGA FORM COMPRESSED SPACE
05370 03497P 04D9 6C 62 A INC 2,S COMPENSATE FOR DEC IN LOOP
05375 03498P 04DB 39 RTS AND RETURN

05385 03500 ENDC
05390 03501P 04DC 1F 89 A EXPAND TFR A,B

05400 03503P 04DE 86 20 A EXPA10 LDA #$20 ISSUE A SPACE
05405 03504P 04E0 BD 03E1 P JSR ISCHAR
05410 03505P 04E3 30 01 A LEAX 1,X ADVANCE POINTER
05415 03506P 04E5 9C 13 A CMPX BOTSCR OFF SCREEN?
05420 03507P 04E7 27 03 04EC BEQ EXPA20 EXIT IF SO
05425 03508P 04E9 5C INCB ELSE, COUNT THIS ONE OFF
05430 03509P 04EA 26 F2 04DE BNE EXPA10 AND LOOP

05440 03511P 04EC 30 1F A EXPA20 LEAX -1,X BACKSPACE TO LAST SPACE ISSUED
05445 03512P 04EE 39 RTS AND RETURN
```



```

05455 03514          *   THIS TURNS OUT TO BE VERY TIME CONSUMING SO IT IS WRITTEN FOR
05460 03515          *   SPEED.
05465 03516          *
05470 03517          *   SINCE INDEXING HAS A RANGE OF 255 MAX, WE MOVE THREE
05475 03518          *   LINES AT A TIME, USING OFFSETS UP TO 240

05485 03520          *   ON ENTRY X=TOP LINE OF SCROLL, U=SCREND=LAST BYTE OF SCROLL

05495 03522P 04EF C6 14      A MOV3L  LDB      #20      NO OF TIMES AROUND LOOP
05500 03523P 04F1 34 04      A          PSHS     B          SAVE IT
05505 03524P 04F3 EE 88 50  A BYTE3  LDU      80,X
05510 03525P 04F6 EC 88 52  A          LDD      82,X
05515 03526P 04F9 EF 84      A          STU      0,X
05520 03527P 04FB ED 02      A          STD      2,X
05525 03528P 04FD EE 89 00A0 A          LDU      160,X
05530 03529P 0501 EC 89 00A2 A          LDD      162,X
05535 03530P 0505 EF 88 50  A          STU      80,X
05540 03531P 0508 ED 88 52  A          STD      82,X
05545 03532P 050B EE 89 00F0 A          LDU      240,X
05550 03533P 050F EC 89 00F2 A          LDD      242,X
05555 03534P 0513 EF 89 00A0 A          STU      160,X
05560 03535P 0517 ED 89 00A2 A          STD      162,X
05565 03536P 051B 30 04      A          LEAX    4,X      BUMP TO NEXT 4 BYTES
05570 03537P 051D 6A E4      A          DEC      0,S
05575 03538P 051F 26 D2 04F3 BNE     BYTE3
05580 03539P 0521 32 61      A          LEAS    1,S
05585 03540          *
05590 03541          *   NOW ADVANCE TO NEXT BLOCK AND MAKE SURE THAT WE'RE STILL ON THE SCREEN
05595 03542          *
05600 03543P 0523 30 89 00A0 A          LEAX    160,X  ADVANCE TO NEXT BLOCK
05605 03544P 0527 DE 24      A          LDU      SCREND  POINT TO END OF SCROLL

05615 03546P 0529 33 C9 FF10 A SCROLL LEAU   -240,U
05620 03547P 052D DF 26      A          STU      STMP
05625 03548P 052F 9C 26      A          CMPX    STMP     MORE THAN THREE LINES LEFT?
05630 03549P 0531 25 BC 04EF BLO     MOV3L  MOVE THEM IF SO
05635 03550P 0533 33 C9 00A0 A          LEAU    160,U  POINT TO LAST LINE
05640 03551P 0537 DF 26      A          STU      STMP
05645 03552P 0539 20 05 0540 BRA     BITP20
05650 03553          *
05655 03554          *   MOVE THE REMAINING BITS AND PIECES
05660 03555          *
05665 03556          *
05670 03557P 053B A6 88 50  A BITP10 LDA     80,X
05675 03558P 053E A7 80      A          STA     0,X+
05680 03559P 0540 9C 26      A BITP20 CMPX    STMP     END OF SCROLL?
05685 03560P 0542 26 F7 053B BNE     BITP10
05690 03561P 0544 86 20      A          LDA     #$20
05695 03562P 0546 A7 80      A LINTWO STA     0,X+  CLEAR THE BOTTOM LINE
05700 03563P 0548 9C 24      A          CMPX    SCREND  END OF SCROLL?
05705 03564P 054A 26 FA 0546 BNE     LINTWO
05710 03565P 054C 39          RTS

05720 03567P 054D          PEND    SCRN   END OF PSCT
05725 03568          0000  A          IFEQ    SHORT

```

05740 03571 * THE FOLLOWING BIT DEFINITIONS APPLY TO THE RETURNED DATA TYPE
 05745 03572 * BIT TABLE.

05755 03574 * \$80 - NUMERICS 0,1,2,3,4,5,6,7,8,9
 05760 03575 * \$40 - HEX. CHARACTERS 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
 05765 03576 * \$20 - UPPER CASE A,B,,,,,,,,,X,Y,Z
 05770 03577 * \$10 - LOWER CASE a,b,,,,,,,,,x,y,z
 05775 03578 * \$08 - NUMERIC OPERATORS + - .
 05780 03579 * \$04 - SPECIAL OPERATORS () * / < = > [] ^
 05785 03580 * \$02 - SPECIAL CHARACTERS
 05790 03581 * \$01 - SPACE

05800 03583 * TABLE OF BIT MASK VALUES FOR ASCII CHARACTERS \$20-\$7F

05810 03585P 054D	01	A	ASCTBL FCB	1,2,2,2,2,2,2,2,6,6,6,10,2,10,10,6
05815 03586		*		! " # \$ % & ' () * + , - . /
05825 03588P 055D	C0	A	FCB	\$C0,\$C0,\$C0,\$C0,\$C0,\$C0,\$C0,\$C0,\$C0,\$C0,\$C0,2,2,6,6,6,2
05830 03589		*		0 1 2 3 4 5 6 7 8 9 : ; < = > ?
05840 03591P 056D	02	A	FCB	2,\$60,\$60,\$60,\$60,\$60,\$60,\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20
05845 03592		*		@ A B C D E F G H I J K L M N O
05855 03594P 057D	20	A	FCB	\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20,\$20,6,2,6,6,2
05860 03595		*		P Q R S T U V W X Y Z [\] ^ _
05870 03597P 058D	02	A	FCB	2,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16
05875 03598		*		` a b c d e f g h i j k l m n o
05885 03600P 059D	10	A	FCB	16,16,16,16,16,16,16,16,16,16,16,16,2,2,2,2
05890 03601		*		p q r s t u v w x y z { } ~

05900 03603 ENDC

05910 03605P 05AC XEND SCRNM

05920 03607

END

TOTAL ERRORS 00000--00000

TOTAL WARNINGS 00000--00000

A\$CLOS	0000	A\$DEOC	0008	A\$OPEN	0008	A\$R	0080	A\$RXW	00B0	A\$W	0020	A\$XR	00C0	A\$XRW	00E0	A\$XRW	00F0
A\$XW	0030	A.CCNT	0000	A.COLN	0003	A.DAY\$	0007	A.DAYS	000B	A.DEL	0013	A.HRS	0001	A.MINS	0004	A.MTHS	000E
A.SIZE	0014	A.YEAR	0011	A0000	0000	A4000	0001	A8000	0002	AC000	0003	ARKILL	00E7	ASCTBL	054D	ASFLAG	000F
B\$SCRN	0000	B.ERAM	0008	B.ESIZ	000A	B.HEAD	000C	B.LINK	0000	B.SIZE	0010	B.SRAM	0006	B.TAIL	000E	BADPKT	0144
BASEPC	005C	BD1	0000	BD2	0010	BD3	0020	BD4	0030	BD5	0040	BD6	0050	BD7	0060	BIGJ	0081
BIT	0000	BITP10	053B	BITP20	0540	BLDC10	00BD	BLDCSB	00B3	BLINK	0000	BLK0	0001	BLK1	0003	BLK2	0005
BLK3	0007	BLOCK	007B	BOTLIN	0018	BOTSCR	0013	BPADD	0006	BPSIZE	005C	BUDMSK	007D	BYTE3	04F3	C\$.CUR	0009
C.LINK	0000	C.LOCK	000C	C.NEXT	000A	C.SIZE	0010	C.TTCB	000E	C.WAIT	000D	CALRUN	001A	CBIT	0001	CC\$AGC	0002
CC\$LGC	0003	CC\$UBC	0001	CC\$UGC	0000	CDOWN	0001	CHKA10	04BB	CHKA20	04BF	CHKASC	04A9	CHKSIZ	044B	CHOME	0004
CLEFT	0002	CLRSCR	0071	CMPR10	04C5	CMPR20	04D8	CMPRES	04C3	CMPSPC	03BD	CRIGHT	0003	CRUP	0000	CT\$FLS	0020
CT\$SCR	001C	CURD10	0192	CURDWN	017C	CURHOM	01AF	CURL10	019E	CURLFT	0193	CURR10	01AE	CURRYT	01A1	CURU10	017B
CURUP	0165	D.SIZE	0086	DA\$.ST	0018	DAY	0014	DAYNO	0017	DC	0006	DCNT	00E3	DCSB	002A	DE\$ACC	000D
DE\$FDF	000C	DE\$NAM	0000	DE\$OGP	000E	DE\$PFU	000F	DE\$RIB	000A	DE\$SIZ	0010	DE\$SUF	0008	DIRECT	0000	DIRFLG	001A
DISPLY	0001	DIV	000C	DL	0008	DMEM	0001	DODC	01F3	DODC10	01FC	DODC20	0209	DODC30	0211	DODL	0252
DODL10	0279	DOEE10	029F	DOEE20	02A7	DOEEOL	0291	DOFL10	0354	DOFLSH	0336	DOGE10	0282	DOGE20	028B	DOGETC	027A
DOIC	01BA	DOIC10	01CA	DOIC20	01CE	DOIC30	01D8	DOIC40	01E2	DOIC50	01EA	DOIL	021A	DOIL10	0242	DOIL20	0251
DOIN10	0328	DOIN20	0335	DOINV	0320	DOP010	02BF	DOP020	02D0	DOPOS	02B1	DOREAD	02F6	DOREST	01B6	DOSE10	02F0
DOSETM	02D1	DOST10	0363	DOST20	0372	DOSTFL	0355	DOWR10	0302	DOWRIT	02FD	DT\$FHC	0001	DT\$NEW	0008	DT\$SIO	0002
DT\$TRU	0004	DV\$ATR	000E	DV\$DCH	0020	DV\$DDF	0014	DV\$DDI	0016	DV\$DNA	0018	DV\$DNI	001A	DV\$DSA	001C	DV\$DSI	001E
DV\$IDB	0012	DV\$IPC	002E	DV\$IQU	0026	DV\$IRF	002A	DV\$LNK	0000	DV\$NLU	000D	DV\$OPC	0030	DV\$OQU	0028	DV\$ORF	002C
DV\$PGE	000C	DV\$REC	000B	DV\$SEC	000A	DV\$SIZ	0032	DV\$STA	0010	DV\$TCB	0006	DV\$TDB	0008	DV\$UIB	0022	DV\$UOB	0024
EEOL	0009	ELAPCH	0028	EMCOM	0086	ENDRAM	0079	ESCOM	0010	EUCOM	002F	EXPA10	04DE	EXPA20	04EC	EXPAND	04DC
FD\$CMP	0008	FD\$CON	0010	FD\$DEL	0040	FD\$FMA	0005	FD\$FMB	0003	FD\$FMC	0007	FD\$FMD	0001	FD\$FML	0002	FD\$FMR	0004
FD\$FMU	0000	FD\$IND	0000	FD\$RND	0030	FD\$SEQ	0020	FD\$WRT	0080	FLASH	000F	FLSH10	0154	FLSH20	0164	FLSHBF	000E
FLSHSC	014A	FREMEM	002F	FRI	0005	GETC	000B	GETPOS	038F	GOTONE	0121	I.ALLM	0012	I.AVTB	0016	I.CHNS	001D
I.DATE	000E	I.DORM	0010	I.ELAP	000A	I.EXEC	0002	I.INT	0006	I.SBLK	0018	I.SCAA	0019	I.SCHD	001F	I.SIZE	001F
I.TICK	001E	I.TOD	000C	I.TRAP	0014	I.TRVT	0004	I.TSHR	0008	I.USER	0000	I.XXAA	001B	IBIT	0010	IC	0005
ID	0000	ID\$000	0048	ID\$BOT	0046	ID\$CAT	0040	ID\$CCT	0044	ID\$DAT	000C	ID\$DIR	003E	ID\$LCT	0042	ID\$NAM	0000
ID\$OVL	0026	ID\$REV	000A	ID\$SIZ	0080	ID\$USR	0012	ID\$VER	0008	ID\$VOL	003A	IE	FFF0	IETRAP	0001	IL	0007
INPHAS	0111	INVCUR	0457	INVERT	000D	IOCAC	003A	IOCCMD	0007	IOCDBE	0018	IOCDBP	0014	IOCDBS	0016	IOCDCB	0022
IOCDF	003E	IOC DEN	003D	IOC DTT	0009	IOCEOF	000C	IOCFDF	0039	IOCLNK	0000	IOCLOK	0024	IOCLSN	000A	IOCLUN	0028
IOCLMS	000E	IOCNAM	002D	IOCOGP	003B	IOCOPT	0008	IOCPFU	003C	IOCRIB	0037	IOCSBE	001E	IOCSBI	0020	IOCSBP	001A
IOCSBS	001C	IOCSDW	0010	IOCSIZ	0040	IOCSLS	0012	IOCSSTA	0006	IOCSUF	0035	IOCTMP	003F	IOCVOL	0029	ISCHAL	03E7
ISCHAR	03E1	ISSUE	03AF	ITMP	0084	J	0082	J.NEXT	0002	J.PREV	0000	J.SIZ	0004	J.SIZE	0006	K.C.PT	0000
K.DCSB	0006	K.NAME	0002	K.NEXT	0008	K.SIZE	000A	L.NAME	0006	L.NEXT	0002	L.PARM	0004	L.PREV	0000	L.SIZE	000A
LINE	0008	LINES	0015	LINTWO	0546	LOC	0600	LOC1	05AC	M.FLAG	0005	M.NEXT	0002	M.PREV	0000	M.RES	0004
M.SIZE	0006	MAPCON	0014	MASK	00D0	MON	0001	MONTH	0015	MOV3L	04EF	NA	0002	NBIT	0008	NDY	0000
NEWP10	0385	NEWP20	038E	NEWPOS	0374	NOINV	011D	OF	0000	OG\$000	0000	OG\$001	0010	OG\$002	0020	OG\$003	0030
OG\$004	0040	OG\$005	0050	OG\$006	0060	OG\$007	0070	OG\$DIS	0006	OG\$DOR	0008	OG\$EXC	0005	OG\$NAC	0000	OG\$PRC	0007
OG\$R	0001	OG\$RA	0004	OG\$RW	0003	OG\$SYS	0080	OG\$W	0002	OLDCUR	000C	ON	0001	OP\$LOK	0080	OP\$N_U	0038
OP\$ULK	0040	P\$SCRN	0000	P.HOME	0004	P.NEXT	0002	P.PREV	0000	P.SIZE	0006	P1	0000	P2	0004	PACKET	0006
PF\$000	0001	PF\$001	0002	PF\$002	0004	PF\$003	0008	PF\$004	0010	PF\$005	0020	PF\$006	0040	PF\$007	0080	PHASE	0010
PM\$000	0000	PM\$001	0008	PM\$002	0010	PM\$003	0018	PM\$004	0020	PM\$005	0028	PM\$006	0030	PM\$007	0038	PM\$OPM	0080
PM\$P00	0000	PM\$P01	0001	PM\$P02	0002	PM\$P03	0003	PM\$P04	0004	PM\$P05	0005	PM\$P06	0006	PM\$P07	0007	PM\$RSB	0040
POS	000A	POSN	000A	P.SIZE	054D	R.BASE	0011	R.LPTR	0000	R.MAPS	000D	R.NAME	0004	R.SIZ	000A	R.SIZE	0013
R.SP	0002	R.SSIZ	000C	R.T.PT	0008	R.TOTS	000F	RABX	0017	RB\$	0080	RB\$CMO	0073	RB\$CRD	0072	RB\$CRE	0070
RB\$CRH	0070	RB\$CRM	0071	RB\$CRY	0074	RB\$EIF	006B	RB\$LA	0078	RB\$LAC	006D	RB\$LAD	006D	RB\$LAM	006E	RB\$LAY	006F
RB\$LB	0075	RB\$OT	007C	RB\$PIC	006A	RB\$SA	007A	RB\$SDW	0000	RB\$SL	0076	RB\$SZ	007E	READ	0011	RELPKT	0132
RESTOR	000C	RSIZ	0088	RUB	03D2	S\$.INT	0000	S\$SCRN	00D5	S.A	0001	S.B	0002	S.CC	0000	S.DP	0003
S.PC	000A	S.SIZE	000C	S.U	0008	S.X	0004	S.Y	0006	SAT	0006	SCOM	0000	SCR10	00F4	SCRDRV	00EF
SCRE10	046C	SCRE20	047B	SCREAD	045E	SCREND	0024	SCRNIN	0043	SCROLL	0529	SCRW10	049A	SCRW20	04A5	SCRWRT	048D
SETIO	0307	SETMRG	000E	SHORT	0000	SLOC	0013	SPCC10	03F4	SPCCR	03EC	SPCF10	0438	SPCF20	043E	SPCFF	0430

SPCL10 0406	SPCL20 0407	SPCL30 0428	SPCLF 03F9	SPCRLF 03F7	SPEC 03C3	SPEC10 03D1	SPLIT 0083	ST\$BSY 0040
ST\$DON 0080	STMP 0026	STOPFL 0010	STRAM 0077	SUN 0007	SYSTAK 000D	T\$.FIN 0002	T\$.FIR 0007	T\$.SCH 0005
T\$SCRN 00E3	T.CHAN 0011	T.CHNS 000F	T.DYN 000D	T.GRP 000E	T.HOME 0017	T.KEY 0013	T.LINK 0000	T.MARK 0019
T.PARM 000A	T.R.PT 0015	T.SIZE 001D	T.SP 001B	T.WAIT 000C	TABEND 0077	TABPTR 007F	TBL 001D	TEMP 000B
THU 0004	TICK 000F	TIMSTR 001B	TLC 0016	TOPO10 03A5	TOPO20 03AE	TOPOS 03A1	TOPSCR 0011	TRAP 0000
TSIZE 0600	TUE 0002	UBUG 0001	VBIT 0002	VD\$1ST 0008	VD\$BUF 000C	VD\$CHR 000B	VD\$CMD 000A	VD\$LNK 0000
VD\$LST 0009	VD\$LSZ 005C	VD\$MSK 000B	VD\$POS 0006	VD\$SIZ 005C	VD\$SSZ 000C	WED 0003	WRITE 0012	WRTFLG 001B
X.DAY 0004	X.DAY\$ 0008	X.DYNO 0007	X.MNTH 0005	X.RUNC 000A	X.SECS 0001	X.SIZE 001F	X.TICK 0000	X.TIME 0002
X.TSTR 000B	X.YEAR 0006	XSTAK 0001	Y.ARFX 001E	Y.BITX 001F	Y.CMRK 0020	Y.CPAD 0021	Y.CPAX 0022	Y.CTAX 0023
Y.CTDA 0024	Y.DABB 0025	Y.DSAI 0026	Y.ERMK 0027	Y.EXAR 0028	Y.GPBX 0029	Y.GQXB 002A	Y.HOMX 002B	Y.IHTX 002C
Y.INFO 002D	Y.LCSX 002E	Y.LSCX 002F	Y.LTDX 0030	Y.MOVE 0031	Y.NEXT 0032	Y.NIFC 0033	Y.NIFE 0034	Y.NIFN 0035
Y.NIFS 0036	Y.NPAD 0037	Y.NPAX 0038	Y.PQXB 0039	Y.RABX 003A	Y.RCFB 003B	Y.RCFW 003C	Y.RCKX 003D	Y.RCTD 003E
Y.RCTX 003F	Y.RDAB 0040	Y.RDDX 0041	Y.REMK 0042	Y.RFCX 0043	Y.RHDX 0044	Y.RRFX 0045	Y.RRNX 0046	Y.RSFX 0047
Y.RSTX 0048	Y.RTAD 0049	Y.RTXB 004A	Y.SDCX 004B	Y.SMRK 004C	Y.STAX 004D	Y.TELL 004E	Y.TLDX 004F	Y.TSFB 0050
Y.TSFW 0051	Y.TSMX 0052	Y.TTLK 0053	Y.TTUL 0054	Y.USXC 0055	Y.UTSX 0056	YEAR 0016	Z.SIZE 0008	ZARFX 0057
ZBIT 0004	ZDABB 0058	ZDSAI 0059	ZGPBX 005A	ZGQXB 005B	ZLSCX 005C	ZPQXB 005D	ZRAWT 0003	ZRRFX 005E
ZRRNX 005F	ZRSFX 0060	ZSECS 0011	ZSFLAG 0004	ZTICK 0010	ZTIME 0012	ZUSXC 0061		