RCA

DESIGN IDEAS BOOK CDP1802

BMP
802

RCA MICROPROCESSORS 1800

# DESIGN IDEAS BOOK

## for the CDP1802

## COSMAC Microprocessor

BMP 802

CONTENTS (cont'd) - (2)



## INTRODUCTION

New and integrated circuits, particularly microprocessors, are often thought difficult to understand. This is a myth, not true, they are complex but not difficult to understand as they operate in a logical way. Understanding microprocessors does require study, explanations and guidelines and the purpose of this book is to provide such information to help you become really familiar with these devices and especially the CDP1802, a CMOS microprocessor.

Happy reading and good luck.

RCA Applications Team - Europe.

# UT4 CMOS ROM MONITOR PROGRAM

A CMOS ROM monitor program is necessary for a prototyping system to have an easy possibility to load programs, verify and start execution.

One external flag and the Q output are used for this interface. For additional hardware, see Figure 1 and Figure 2 , it assumes a clock frequency of 2MHz for 100 and 300bd. This monitor program does not need any RAM.

It starts at 8000 (highest address bit high) means after reset this line has to be set high externally to start the monitor program. (See evaluation kit manual MPM203 or MPM 224, as well Microboard brochures).

After being entered it initializes itself and waits for CR or LF. Carriage return sets it to full duplex and line feed to half duplex. The answer is the UT4 prompt. This utility program can even work together with a paper tape reader or punch. For additional information, see MPM224, MPM203.

## ?M Command

To interrogate memory, the user types a command such as

?MF5 3

and terminates it with CR(carriage return). UT4 responds by printing out the contents of memory beginning at location 00F5: three bytes are printed out as two hex digits each. Each line of output begins with the address, and data is grouped in 2-byte (4-digit) blocks. When necessary, new lines are begun every 16 bytes, with the previous lines ending in semicolons. The user may enter any number of digits to specify the beginning location (leading zeroes are implied, if necessary). If more than four digits are entered, only the last four are used. The number of bytes to be typed out should be in hex.



Again, if more than four digits are entered, only the last four are used. This feature allows the user to correct a mistake. He simply keeps typing, putting in the correct 4-digit values (230024 is effectively 0024).

## !M Command

In general, data is entered into memory by means of a command such as

!M2F 434F534D4143

This command enters six bytes (two hex digits each) into memory beginning at location 2F. It is normally terminated by a CR. Once again, the starting location is determined by the last four digits entered. Data is entered into memory after each two hex digits are typed. If the user types an odd number of digits, the last digit is ignored, and the error message ('?') is typed out.

The !M command provides two options that facilitate memory loading. First, a string of data can be extended from line to line by typing in a comma just before the normal CR. (In this case the user must type CR-LF (carriage return-line feed) before he can begin a new line). For example :

!M23 56789ABC,(CR) (LF)

DEF0123456,(CR) (LF)

3047 (CR)

enters 11 successive bytes beginning at location 0023. Between successive hex pairs while data is being entered, any non-hex character except the comma (and semicolon, as will be discussed) is ignored. This arrangement permits arbitrary LF's, spaces (for readability), nulls (generated

arbitrary LF's, spaces (for readability), nulls (generated

by the utility program or by a time-share system to give the carriage time to return), etc...

As a second optional form of data entry, a string of input data can be terminated by a semicolon (and a CR). The utility program then expects more data to follow on the next line, but preceded by a new beginning address. The line must have the format of an !M command, but with the initial !M omitted. (The utility program ignores all non-hex characters following !M, which allows the CR, LF, and nulls to be input from the Teletype without disturbing the !M command). Note also that the semicolon feature on input allows non-contiguous memory to be loaded.

## $P Command

A third utility command is $P. For example

$P6C

Starts execution at location 6C with RØ as the program counter (after the user presses CR and the utility program provides a LF). The last-four-digits-in rule applies to the address typed in.

$P always begins with RØ as program counter and X = 0. This arrangement is consistent with the fact that P = 0 and X = 0 after the CPU is RESET. Refer to the CDP1802 data sheet for other actions of RESET.

## Summary of command usage

In summary, after receiving the prompt character, '*' the user may type

?M (address) Δ (count) CR

!M (non-hex) (address) Δ (data) (optional, or ;) CR

(Where the data may have non-hex digits between each hex pair)



or

$P (address) CR

UT4 ignores initial characters until it detects ?, !, or $. Then, inputs which are not compatible with the above formats cause an error message.

## Summary of UT4 operating instructions

A further detailed summary of these basic operating instruction is given below, repeating the information just given in a more concise form.

1. After pressing "RUN UTILITY" (start at 8000), the user should press either CR or LF: LF for half duplex, CR for full duplex. This instruction sets up the bit-serial timing and specifies echo or not.

2. UT4 will return * as a prompt.

3. Following *; UT4 ignores all characters until one of ?, $, or ! is typed in.

4. Following ?M or !M, UT4 waits for a hex character. It then assembles an address. If more than four hex digits are typed, only the last four are used. Next, a space is required. Note : Δ denotes a space.

   a. For ?M addr Δ a hex count must follow (again, only the last four digits are kept), and the command is terminated by CR.

   b. For !M addr Δ data must follow. An even number of

   b. For !M addr Δ data must follow. An even number of hex digits is required. Before each hex pair arbitrary filler, except for a CR, comma, or semicolon, is allowed. CR terminates the command, unless it is immediately preceded by a comma or, as is generally the case, by a semicolon.

i. In case of comma CR the user must insert an LF for UT4 to continue to accept data. This procedure is a form of line continuation.

ii. In case of a semicolon all following characters are ignored until the CR is typed. Then the user must again provide an LF, and UT4 continues as if it had received optional filler, then a starting address, then a space, and then data.

iii. The !M command can be followed by as many continuation lines as needed, mixed between the two types if desired, and is finally terminated with a CR not preceded by a comma or semicolon.

5. Command ßP must be followed by starting address (last four digits used if more than four are typed in). If no address is entered, 0 is assumed. Program execution begins at this location with RØ as program counter with X set to 0.

6. When a !M or ?M command is accepted and compeleted, UT4 types another prompt character.

7. When UT4 detects bad syntax, it types out a ? and returns the carriage. If a mistake is made when data is entered (by typing in an odd number of digits), all data will have been entered except the last hex digit. Note that the "only-last-four-digits" rule in the address field allows the user to correct an error without retyping the whole command. For example, a mistaken 234 can be corrected by continuing 2340235=0235. A bad command can be aborted by typing in any illegal character except can be aborted by typing in any illegal character except after !M or ?M or between input hex data pairs. In these cases, the user should type any digit and then, for example, a period.



## UT4 register storage feature

UT4 provides for storing in RAM 13 1/2 of the 16 CDP1802 scratch-pad registers. A CDP1824 32-byte RAM has to be provided for this function. The RAM occupies addresses 8C00 - 8C1F. By pressing RESET followed by RUN U, registers R0 - RF are automatically stored in the CDP1824, in numerical order, most significant byte first. R0, R1, and R4.1 are altered in the process.

By using the command

?M8C00 20

The register contents which existed in the Microprocessor at the instant that RESET was pressed preceding the depression of RUN U can be examined. It should be remembered that UT4 uses registers R0, R1, R3, R4.1, R5, and RC - RF. These registers, therefore, will be modified. Should the user wish to continue program execution, he must initialize these registers, by program if necessary. A sample listing is given in Figure 1. It should be recalled that R0, R1, and R4.1 are not correct.

## The bit serial terminal interface

The serial terminal interface is an example of minimizing hardware complexity by the use of software. Further, it illustrates the increased flexibility that can be more readily achieved by software. The CPU receives serial data by sampling $\overline{EF4}$. It transmits serial data via its Q output. Details on the electrical I/O interface are given in the Application Note entitled "Data Terminal Interface Considerations for RCA Microprocessor Evaluation Kit CDP18S020.

The sample character waveform in Figure 4 helps to show what the interface software must do. Each character is framed by a START bit and one or two STOP bits. On input, this signal is tied to $\overline{EF4}$ which is sensed by UT4 at the midpoints of each of the bits. Software assembles the resultant ASCII character. On output, the character is transmitted one bit at a time through the Q output of the CDP1802. (See Figures 2 and 3).

The flexibility obtainable with software is demonstrated by the ability of the program UT4 to sample a character string and adjust its timing so as to cope with terminals of different, even non-standard, character rates. However, it should be noted that while a program is timing either input or output in this manner (i.e., by counting instruction executions), it is completely dedicated to that task and cannot be interrupted except for an occasional DMA service.

```
            RØ   R1            R4.1
?M8C00 20  /   /              /
8C00 D0D0 8202 2222 3333 9444 5555 6666 7777;
8C10 8888 9999 AAAA BBBB CCCC DDDD EEEE FFFF
```

Figure 1 : Sample Listing Illustrating Register Storage

This utility program can even work together with a paper tape reader or punch. For additional information see MPM 224, MPM203.



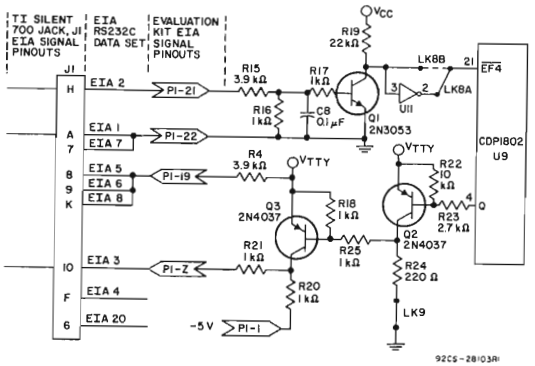


Figure 2 : The EIA RS232 Serial Data Interface For Connecting TI Silent 700 Data Terminal

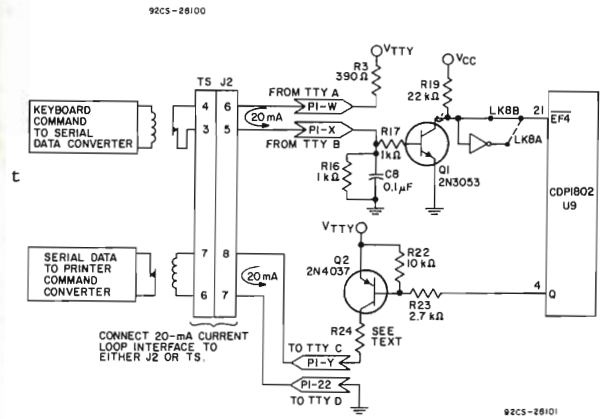


Figure 3 : The 20mA Current Loop Interface To Connect a Teletype In Full Duplex Mode

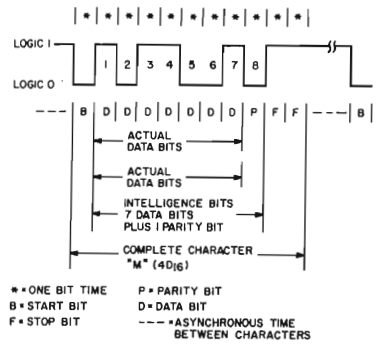


Figure 4 : Data Terminal Bit Serial Output For The Character "M"

UTILITY PROGRAM UT4 LISTING

```
!M
0000 ;                0001              ORG #8000
8000 ;                0002       .. UT4 IS A UTILITY PROGRAM TO ALTER
8000 ;                0003       .. MEMORY, DUMP MEMORY, AND BEGIN PROGRAM
8000 ;                0004       .. EXECUTION AT A GIVEN LOCATION. THE COMMANDS
8000 ;                0005       .. ACCEPTED ARE $PHHHH (BEGIN EXECUTION AT THE
8000 ;                0006       .. SPECIFIED LOCATION WITH R0 AS PROGRAM
8000 ;                0007       .. COUNTER), !MHHHH DATA (PUT DATA AT SPECIFIED
.
8000 ;                0008       .. LOCATION), AND ?MHHHH HHHH (OUTPUT DATA
8000 ;                0009       .. FROM SPECIFIED LOCATION FOR SPECIFIC COUNT)
8000 ;                0010       .. AT THE BEGINNING OF A COMMAND ALL CHARACTERS
.
8000 ;                0011       .. ARE IGNORED UNTIL A ?,!,OR $ IS
8000 ;                0012       .. ENCOUNTERED. IN THE ?M AND !M COMMANDS NON
8000 ;                0013       .. HEX CHARACTERS ARE IGNORED AFTER M UNTIL A
8000 ;                0014       .. HEX IS READ, THEN THE FIRST NON HEX
8000 ;                0015       .. CHARACTER MUST BE A SPACE . NON HEX
8000 ;                0016       .. CHARACTERS BETWEEN HEX PAIRS OF THE DATA IN
8000 ;                0017       .. THE !M COMMAND ARE IGNORED EXCEPT FOR CR,
8000 ;                0018       .. SEMICOLON, AND COMMA.
8000 ;                0019       .. THE BAUD RATE OF UT4 IS DEPENDENT UPON THE
8000 ;                0020       .. TERMINAL BEING USED. A CR OR LF IS ENTERED
8000 ;                0021       .. AT THE BEGINNING TO SPECIFY THE APPROPRIATE
8000 ;                0022       .. DELAY BETWEEN BITS. UT4 WILL ECHO
8000 ;                0023       .. CHARACTERS IF A CR IS CHOOSEN AS THE
8000 ;                0024       .. TIMING CHARACTER. ECHOING WILL NOT TAKE
8000 ;                0025       .. PLACE IF A LF IS INPUT AS THE TIMING
8000 ;                0026       .. CHARACTER.
8000 ;                0027       .. UT4, AT INITIATION, STORES, ALL REGISTERS
8000 ;                0028       .. BETWEEN 8C00 AND 8C1F IF IT FINDS RAM THERE
8000 ;                0029       .. (BUT R0, R1, AND R4.1 ARE CLOBBERED).
8000 ;                0030          PTER=#00    ..AUXILIARY FOR MAIN ROUTINE
8000 ;                0031            CL=#01    ..CLOBBERED
8000 ;                0032            ST=#02    ..STACK POINTER-ONLY
8000 ;                0033                      ..REFERENCE TO RAM
8000 ;                0034          SUB=#03       ..SUBROUTINE PC
8000 ;                0035            PC=#05    ..MAIN PROGRAM COUNTER
8000 ;                0036        SWITCH=CL     ..DISTINGUSHES BETWEEN ?M AND !M
8000 ;                0037         DELAY=#0C    ..DELAY ROUTINE PROGRAM COUNTER
8000 ;                0038           ASL=#0D    ..HEX ASSYMBLY REG ON INPUT,
8000 ;                0039                      ..AUX FOR HEX OUTPUT
8000 ;                0040        CENTER=ASL    ..USED TO COUNT OUTPUT BYTES
8000 ;                0041           AUX=#0E    ..AUX.1 HOLDS BIT-TIME CONSTANT
8000 ;                0042          CHAR=#0F    ..CHAR.1 HOLDS I/O BYTE
8000 ;                0043       ..
8000 ;                0044       .. ENTER IN R0
8000 C4;              0045          NOP
8001 F880B0;          0046          LDI A.1(UT4) ;PHI R0 ..SET PC WHILE
8004 ;                0047                               ..FINGER IS ON
```



```
8004 ;                0048       ..
8004 ;                0049       .. THE FOLLOWING WRITES REGISTER CONTENTS INTO
8004 ;                0050       .. 8C00-8C1F IF IT EXISTS. 8BFE IS ASSUMED NOT
.
8004 ;                0051       .. TO BE RAM (ELSE ROUTINE OVERRUNS).
8004 F88CB1;          0052          LDI #8C ;PHI CL    ..CL IS CLOBBERED
8007 ;                0053                             ..BY THIS ROUTIINE
8007 F81EA1;          0054          LDI #1E ;PLO CL    ..SET UP WHERE RF.0
800A ;                0055                             ..IS TO GO, MINUS 1
800A F8A0B4;          0056          LDI #A0 ;PHI R4    ..R4.1 STORES A MODIFIED
800D ;                0057                             ..INSTRUCTION
800D E1;              0058          SEX CL
800E F8D051;          0059  LOOP2: LDI #D0 ;STR CL     ..SET UP SEP INSTRUCTION
.
8011 ;                0060                             ..FOR RETURN
8011 F3;              0061          XOR                ..CHECK THAT IT WROTE
8012 3A29;            0062          BNZ UT4            ..
8014 21;              0063          DEC CL             ..PREPARE FOR MODIFIED
8015 ;                0064                             ..INSTRUCTION
8015 94FC70;          0065          GHI R4  ;ADI#70    ..SEE IF IT IS IN THE 90'S
.
8018 331C;            0066          BDF *+#04
801A FC21;            0067          ADI#21             ..IF NO,8N BECOMES 9N
801C FC7F;            0068          ADI#7F             ..IF YES, 9N BECOMES 8(N-1)
.
801E B451;            0069          PHI R4    ;STR CL     ..SET MODIFIED INSTR
8020 ;                0070                              ..INTO RAM
8020 F3;              0071          XOR               ..CK THAT IT WROTE
8021 3A29;            0072          BNZ UT4
8023 D1;              0073          SEP CL              ..GO TO EXECUTE INSTR
8024 ;                0074                              ..(80-9F)
8024 51;              0075          STR CL              ..STORE RESULT IN RAM
8025 2121;            0076          DEC CL  ;DEC CL     ..BACK UP FOR NEXT BYTE
8027 300E;            0077          BR LOOP2
8029 ;                0078       ..
8029 90B5B3;          0079  UT4:GHI R0      ;PHI PC    ;PHI SUB  ..#80-†PC.1
802C ;                0080                                 ..AND SUB.1
802C F830A5;          0081          LDI A.0(UT4A)      ;PLO PC   ..
802F D5;              0082          SEP PC
8030 E5;              0083  UT4A:SEX PC
8031 7155;            0084          DIS,#55          ..NOTE PC=5 ASSUMED
8033 ;                0085                           ..HERE!
8033 6101;            0086          OUT 1,#01      ..SELECT RCA GROUP
8035 F8FEA3;          0087          LDI A.0(TIMALC)    ;PLO SUB      ..READ ONE
8038 ;                0088                                 ..TO SET TIMER
8038 D3;              0089          SEP SUB
8039 ;                0090       ..
8039 ;                0091       ... INITIATION NOW DONE
8039 ;                0092       ..
8039 F89CA3;          0093  START:LDI A.0(TYPE5D) ;PLO SUB
```

```
803C D30D;          0094          SEP SUB; ,#0D        ..CR=CARRIAGE RETURN
803E D30A;          0095  ST2:SEP SUB; ,#0A   ..LF=LINE FEED
8040 D32A;          0096          SEP SUB; ,#2A      ..* AS PROMPT CHARACTER
8042 F800ADBD;      0097  IGNORE:LDI #00;PLO ASL;PHI ASL      ..PREPARE TO
8046 ;              0098                                       ..INPUT HEX
8046 ;              0099                              ..DIGITS,CLEAR ASL
8046 F83BA3;        0100          LDI A.0(READAH) ;PLO SUB
8049 D3;            0101          SEP SUB         ..INPUT COMMAND
804A FB24;          0102          XRI #24                  ..IS IT $ ?
804C 32D6;          0103          BZ DOLLAR
804E FB05;          0104          XRI #05         ..IS IT ! ? (TEST WITH $.XOR.!)

8050 A1;            0105          PLO SWITCH      ..AND SAVE RESULT
8051 CE;            0106          LSZ             ..EQIV. TO BR RDARGS
8052 FB1E;          0107          XRI #1E         ..IS IT ?
8054 ;              0108                            ..?(TEST WITH $.XOR.!.XOR.?)

8054 3A42;          0109          BNZ IGNORE ..IGNORE ALL UNTIL A COMMAND IS
8056 ;              0110                     ..READ
8056 ;              0111      ..
8056 ;              0112      ..THE FOLLOWING IS COMMON FOR ?M AND !M
8056 ;              0113      ..(SWITCH.0 =0 FOR THE LATTER)
8056 ;              0114      ..
8056 D3;            0115  RDARGS:SEP SUB          ..NOTE SUB AT READAH. NOW
8057 ;              0116                          ..READ HEX ARGS
8057 FB4D;          0117          XRI #4D         ..SHOULD BE M
8059 3ACA;          0118          BNZ SYNERR
805B D3;            0119  RD1:SEP SUB
805C 3B5B;          0120          BNF * -#01      ..IGNORE NON HEX CHARS.
805E ;              0121                          ..AFTER M.
805E D3;            0122          SEP SUB
805F 335E;          0123          BDF *-#01       ..READ IN FIRST ARG
8061 ;              0124                        ..(LOCATIONN IN MEMORY)
8061 FB20;          0125          XRI #20         ..NEXT CHAR SHOULD BE A SPACE

8063 3ACA;          0126          BNZ SYNERR
8065 9DB0;          0127          GHI ASL ;PHI PTER
8067 8DA0;          0128          GLO ASL ;PLO PTER   ..PTER NOW POINTS INTO
8069 ;              0129                              ..USER MEMORY
8069 81;            0130          GLO SWITCH          ..LOOK AT SWITCH
806A 32B4;          0131          BZ EX1              ..IF 0 IT WAS !
806C ;              0132                              ..OTHERWISE IT WAS ?
806C ;              0133      ..THE FOLLOWING DOES (?M LOC COUNT) COMMAND
806C ;              0134  ..
806C F800ADBD;      0135          LDI #00 ;PLO ASL ;PHI ASL  ..CLEAR ASL
8070 D3;            0136  RD2:SEP SUB
8071 3370;          0137          BDF RD2             ..READ IN SECOND ARG
8073 ;              0138                              ..(NUMBER OF BYTES)
8073 FB0D;          0139          XRI #0D             ..NEXT CK FOR CR
8075 3ACA;          0140          BNZ SYNERR

8075 3ACA;          0140          BNZ SYNERR
```



```
8077 F89CA3;        0141          LDI A.0(TYPE5D) ;PLO SUB     ..TYPE
807A 8DA1;          0142          GLO ASL         ;PLO SWITCH
807C 9DB1;          0143          GHI ASL         ;PHI SWITCH
807E D30A;          0144  LINE:SEP SUB; ,#0A                  ..LF
8080 90BF;          0145  LINE1:GHI PTER          ;PHI CHAR  ..PREPARE LINE
8082 ;              0146                                     ..HEADING
8082 F8AEA3;        0147          LDI A.0(TYPE2)  ;PLO SUB
8085 D3;            0148          SEP SUB                    ..TYPE 2 HEX DIGIT
.S
8086 80BF;          0149          GLO PTER   ;PHI CHAR
8088 F8AEA3;        0150          LDI A.0(TYPE2)  ;PLO SUB
808B D3;            0151          SEP SUB                    ..TYPE
808C D320;          0152          SEP SUB; ,#20              ..SPACE
808E ;              0153      ..
808E 40BF;          0154  TLOOP:LDA PTER          ;PHI CHAR  ..FETCH 1 BYTE FOR

8090 ;              0155                                     ..TYPING
8090 F8AEA3;        0156          LDI A.0(TYPE2)  ;PLO SUB
8093 D3;            0157          SEP SUB        ..TYPE 2 HEX
8094 21;            0158          DEC SWITCH
8095 81;            0159          GLO SWITCH
8096 3A9B;          0160          BNZ TL3        ..BRANCH IF NOT DONE YET
8098 91;            0161          GHI SWITCH
8099 3239;          0162          BZ START       ..BRANCH IF DONE
809B 80FA0F;        0163  TL3:GLO PTER ;ANI#0F  ..IS PTER DIV BY 16
809E 3AA6;          0164          BNZ TL2
80A0 D33B;          0165          SEP SUB; ,#3B ..IF YES TYPE ; THEN
80A2 D30D;          0166          SEP SUB; ,#0D ..CR AND
80A4 307E;          0167          BR LINE
80A6 F6;            0168  TL2:SHR                ..DIV BY 2?
80A7 338E;          0169          BDF TLOOP      ..IF NO LOOP BACK, ELSE
80A9 308C;          0170          BR TLOOP -#02 ..AND THEN LOOP BACK
80AB ;              0171      ..
80AB ;              0172      ..THE FOLLOWING DOES(!M LOC DATA) COMMAND
80AB ;              0173      ..ENTER AT EX1
80AB ;              0174      ..
80AB ;              0175      ..EFFECT OF THE FOLLOWING IS TO READ IN HEX
80AB ;              0176      ..TERMINATING WITH A CR,IGNORING NON-HEX CHARS.
.
80AB ;              0177      ..PAIRS; EXCEPTIONS: A COMMA BEFORE A CR ALLOWS
.
80AB ;              0178      ..THE INPUT TO CONTINUE ON THE NEXT LINE AND A
80AB ;              0179      ..SEMICOLON ALLOWS AN !M COMMAND TO
80AB ;              0180      ..BE ASSUMED.
80AB ;              0181      ..
80AB D3;            0182  EX3:SEP SUB              ..INPUT UNTIL A HEX IS READ
.
80AC 3BAB;          0183          BNF EX3
80AE ;              0184      ..
80AE D3;            0185  EX2:SEP SUB              ..LOOKING FOR SECOND HEX

80AE D3;            0185  EX2:SEP SUB              ..LOOKING FOR SECOND HEX
```

```
80AF ;              0186                            ..DIGIT
80AF 3BCA;          0187          BNF SYNERR        ..BR IF NOT HEX
80B1 8D50;          0188          GLO ASL    ;STR PTER       ..**SET BYTE**
80B3 10;            0189          INC PTER
80B4 D3;            0190    EX1:SEP SUB                     ..NOTE SUB AT
80B5 ;              0191                                    ..READAH
80B5 33AE;          0192          BDF EX2                   ..BR IF HEX
80B7 FB0D;          0193          XRI #0D                   ..CHECK IF CR
80B9 3239;          0194          BZ START
80BB FB21;          0195    EX4:XRI #21                     ..ELSE CK IF COMMA
.
80BD ;              0196                            ..(TEST WITH CR.XOR.,)
80BD 32AB;          0197          BZ EX3      ..IF ELSE BRANCH
80BF FB17;          0198          XRI #17           ..CK FOR SEMICOLON(TEST WITH
.
80C1 ;              0199                            ..CR.XOR.,.XOR.;)
80C1 3AB4;          0200          BNZ EX1           ..IGNORE ALL ELSE
80C3 D3;            0201          SEP SUB           ..ON SEMI IGNORE AL UNTIL CR
.
80C4 ;              0202                            ..THEN LOOP BACK
80C4 FB0D;          0203          XRI #0D
80C6 3AC3;          0204          BNZ *-#03
80C8 3058;          0205          BR RD1            ..THEN BRANCH BACK
80CA ;              0206                            ..FOR !M COMMAND
80CA ;              0207        ..
80CA F89CA3;        0208    SYNERR:LDI A.0(TYPE5D) ;PLO SUB    ..GENERAL
80CD ;              0209                                  ..RESULT OF
80CD ;              0210                                  ..SYNTACTIC ERROR
80CD D30D;          0211          SEP SUB; ,#0D                   ..CR
80CF C081F8;        0212          LBR FSYNER                ..FINISH ERROR MSG
.
80D2 ;              0213      ..
80D2 ;              0214      ..THE FOLLOWING DOES $P HHHH
80D2 ;              0215      ORG #80D6
80D6 D3;            0216    DOLLAR:SEP SUB                  ..NOTE SUB.0=READAH
80D7 FB50;          0217          XRI #50                   ..SHOULD BE P
80D9 3ACA;          0218          BNZ SYNERR
80DB D3;            0219    D1:SEP SUB
80DC 33DB;          0220          BDF D1                    ..ASSEMBLE HEX
80DE ;              0221                                    ..STING INTO ASL
80DE FB0D;          0222          XRI #0D                   ..FIRST NONHEX
80E0 ;              0223                                  ..MUST BE CR
80E0 3ACA;          0224          BNZ SYNERR
80E2 9DB0;          0225          GHI ASL    ;PHI R0
80E4 8DA0;          0226          GLO ASL    ;PLO R0      ..SET UP NEXT PC
80E6 F89CA3;        0227          LDI A.0(TYPE5D)    ;PLO SUB
80E9 D30A;          0228          SEP SUB; ,#0A           ..LF
80EB E5;            0229          SEX PC
80EC 7000;          0230          RET,#00             ..AND USER PROGRAM
80EE ;              0231                              ..BEGINS (IN R0)
```



```
80EE ;              0232                                  ..EXIT TO UT4
80EE ;              0233      ..
80EE ;              0234      ..
80EE ;              0235      ..
80EE ;              0236      ..
80EE ;              0237      ..SUBROUTINES
80EE ;              0238      ..
80EE ;              0239      ..DELAY ROUTINE
80EE ;              0240      ..DELAY IS 2(1+AUX.1(3+@SUB))
80EE ;              0241      ..USED BY TYPE, READ, AND TIMALC.
80EE ;              0242      ..AUX.1 IS ASSUMED TO HOLD A DELAY CONSTANT
80EE ;              0243      ..=((BIT TIME OF TERMINAL)/
80EE ;              0244      ..(20*INSTR TIME OF COSMAC))-1.
80EE ;              0245      ..THIS CONSTANT CAN BE GENERATED
80EE ;              0246      ..AUTOMATICALLY BY THE TIMALC ROUTINE.
80EE ;              0247      ..
80EE D3;            0248    DEXIT:SEP SUB
80EF 9EF6AE;        0249    DELAY1:GHI AUX ;SHR  ;PLO AUX     ..SHIFT OUT
80F2 ;              0250                                      ..ECHO FLAG
80F2 2E;            0251    DELAY2:DEC AUX              ..AUX.0 HOLDS BASIC
80F3 ;              0252                                ..BIT DELAY
80F3 43FF01;        0253          LDA SUB     ;SMI #01    ..PICK UP A CONSTANT
80F6 3AF4;          0254          BNZ *-#02               ..LOOP AS SPECIFIED
80F8 ;              0255                                  ..BY CALL
80F8 8E;            0256          GLO AUX                       ..DONE YET ?
80F9 32EE;          0257          BZ DEXIT
80FB 23;            0258          DEC SUB                   ..POINTS SUB
80FC ;              0259                              ..AT DELAY POINTER
80FC 30F2;          0260          BR DELAY2
80FE ;              0261      ..
80FE ;              0262      ..ROUTINE TO CALUCULATE BYTE TIME AND ECHO
80FE ;              0263      ..FLAG. WAITS FOR LF (NO ECHO) OR CR(ECHO)
80FE ;              0264      ..BE TYPED IN. ALSO SETS UP POINTER TO
80FE ;              0265      ..DELAY ROUTINE.
80FE ;              0266      ..AUX.1 ENDS UP HOLDING, IN THE MOST
80FE ;              0267      ..SIGNIFICANT 7 BITS,THE DELAY CONSTANT.
80FE ;              0268      ..LEAST SIGNIFICANT BIT IS 0 FOR ECHO, 1 FOR
80FE ;              0269      ..NO ECHO
80FE ;              0270      ..
80FE 93BC;          0271    TIMALC:GHI SUB     ;PHI DELAY
8100 F800AEAF;      0272          LDI #00     ;PLO AUX          ;PLO CHAR
8104 F8EFAC;        0273          LDI A.0(DELAY1)               ;PLO DELAY
8107 ;              0274                                ..DELAY ROUTINE READY
8107 3707;          0275          B4 *                    ..WAIT FOR START BIT
8109 3F09;          0276          BN4 *                   ..WAIT FOR FIRST
810B ;              0277                            ..NON ZERO DATA BIT
810B F803;          0278          LDI #03                   ..SET UP FOR
810D ;              0279                              ..10 EXECUTIONS
810D ;              0280                              ..SO ROUND-OFF IS MINIMAL
810D FF01;          0281    TC2:SMI #01
```

```
810F 3A0D;          0282          BNZ *-#02
8111 8F;            0283          GLO CHAR                    ..LOOK TO SEE
8112 ;              0284                                      ..IF DATA
8112 ;              0285                                      ..CHANGED PREVIOUSLY
.
8112 3A17;          0286          BNZ ZRONE                   ..BR IF IT HAD
8114 3719;          0287          B4 INCR                     ..ELSE LOO FOR
8116 ;              0288                                    ..CHANGE TO 0 NOW
8116 ;              0289                                    ..BRANCH IF NO
8116 1F;            0290          INC CHAR                    .. IF YES SET SWITCH
.
8117 371E;          0291    ZRONE:B4 DAUX                     ..LOOK FOR CHANGE
8119 ;              0292                                      ..TO 1, BR IF YES
8119 1E;            0293    INCR:INC AUX
811A F807;          0294          LDI #07                     ..SET UP FOR
811C ;              0295                                       ..20 INSTR. LOOP
811C 300D;          0296          BR TC2
811E ;              0297       ..AUX.0 NOW HOLDS #LOOPS IN 2 BIT TIMES
811E 2E2E;          0298    DAUX:DEC AUX    ;DEC AUX          ..REDUCE COUNT
8120 ;              0299                                      ..TO BALANCE
8120 ;              0300                                    ..FIXED OVERLOAD
8120 ;              0301                                    ..IN CALLING DELAY
8120 8EF901BE;      0302          GLO AUX ;ORI #01 ;PHI AUX         ..LSB AUX.1=
.1
8124 DC0C;          0303          SEP RC; ,#0C                        ..1.5 BIT
8126 ;              0304                                      ..TIME DEAY
8126 3F2C;          0305          BN4 WAIT  ..BR IF LF=†NO ECHO, LSB AUX.1=1
8128 ;              0306
8128 9EFAFE;        0307          GHI AUX    ;ANI#FE
812B BE;            0308          PHI AUX      ..CR=†ECHO, LSB AUX.1=0
812C DC26;          0309    WAIT:SEP RC; ,#26
812E D5;            0310          SEP R5
812F ;              0311    ..
812F ;              0312    ..
812F ;              0313    ..
812F ;              0314    ..READ ROUTINE--READS 1 BYTE INTO CHAR.1.
812F ;              0315    ..WHEN ENTERED VIA READAH,THEN
812F ;              0316    ..IF INPUT IS A HEX DIGIT ITS HEX VALUE
812F ;              0317    ..IS SHIFTED INTO ASL FROM THE RIGHT
812F ;              0318    ..AND DF=1,ELSE DF=0; CLOBBERS CHAR, AUX.0,(ASL
812F ;              0319    ..ON READAH). LEAVES BYTE IN D (BUT CLOBBERED IF
812F ;              0320    ..SUBR LINKAGE IS USED). LEAVES PC AT READAH
812F ;              0321    ..ENTRY POINT; EXITS TO R5.
812F ;              0322    ..
812F ;              0323    ..WARNING:READ PROCESS HAS NOT FINISHED. DO
812F ;              0324    ..NOT TYPE IMMEDIATELY, OR ELSE ENTER TYPE VIA
812F ;              0325    ..TYPE5D
812F ;              0326    ..
812F ;              0327          ORG #812F
812F FC07;          0328    CKDEC:ADI #07       ..CK FOR ASCII DECIMAL INPUT
```



```
8131 3337;          0329          BDF NFND
8133 FC0A;          0330          ADI #0A
8135 3387;          0331          BDF FND       ..SUB NET 30
8137 FC00;          0332    NFND:ADI #00        ..SETS DF=0
8139 9F;            0333    REXIT:GHI CHAR      ..CHARACTER INTO D
813A D5;            0334          SEP R5
813B F800;          0335    READAH:LDI #00
813D 38;            0336          SKP         ..SKIP OVER TO READ1
813E 83;            0337    READ:GLO SUB        ..CONSTANT WITH A VALUE †0
813F C8;            0338          LSKP
8140 F801;          0339    TTYRED:LDI #01
8142 AF;            0340    READ1:PLO CHAR      ..SET ENTRY FLAG
8143 F880BF;        0341    READ2:LDI #80    ;PHI CHAR      ..INITIALIZE
8146 ;              0342                                   ..INPUT BYTE
8146 ;              0343                                     ..WHEN SHIFTED 80
8146 ;              0344                                     ..IS 1, WILL BE DONE
.
8146 E3;            0345          SEX SUB
8147 8FF6;          0346          GLO CHAR    ;SHR      ..DF=1 -†ENTRY VIA TTYRED
.
8149 3B4D;          0347          BNF TTY1-#02
814B 6780;          0348          OUT 7 ,#80      ..READER ON
814D 3F4D;          0349          BN4 *       ..WAIT FOR END OF LAST DATA BIT
814F 374F;          0350    TTY1:B4 *       ..WAIT FOR PRESENT START BIT
8151 DC02;          0351          SEP RC; ,#02      ..DELAY HALF BIT TIME
8153 374F;          0352          B4 TTY1       ..BR IF NO START BIT
8155 8FF6;          0353          GLO CHAR    ;SHR      ..ENTRY VIA TTYRED?
8157 3B5B;          0354          BNF NOBIT       ..BR IF NO
8159 6740;          0355          OUT 7 ,#40
815B ;              0356    ..
815B E2C4;          0357    NOBIT:SEX R2    ;NOP      ..RESET X, AND DELAY
815D 9EF6;          0358    BIT:GHI AUX    ;SHR      ..ECHO ?
815F 3368;          0359          BDF NOECHO      ..BR IF NO
8161 3766;          0360          B4 OUTBIT       ..IS THE BIT A 1 ?
8163 7B;            0361          SEQ       ..SET Q
8164 3068;          0362          BR NOECHO
8166 7A;            0363    OUTBIT:REQ      ..RESET Q
8167 C4;            0364          NOP       ..DELAY
8168 DC07;          0365    NOECHO:SEP RC; ,#07       ..WAIT ONE BIT TIME
816A C4C4;          0366          NOP    ;NOP      ..MORE DELAY
816C 9FF6BF;        0367          GHI CHAR    ;SHR    ;PHI CHAR      ..SHIFT
816F ;              0368                                     ..THE INPUT CHAR
816F 3378;          0369          BDF NEXT      ..BR IF INPUT FINISHED
8171 ;              0370                                   ..D=CHAR.1
8171 F980;          0371          ORI#80
8173 3F5B;          0372          BN4 NOBIT       ..BR IF INPUT WAS A ZERO
8175 BF;            0373          PHI CHAR
8176 305D;          0374          BR BIT      ..CONTINUE LOOP
8178 ;              0375    ..
8178 ;              0376    ..
```

```
8178 7A;          0377  NEXT:REQ     ..OUTPUT THE STOP BIT
8179 3243;        0378          BZ READ2      ..BR IF D=0, =↑CHAR.1
817B ;            0379                        ..IS A NULL
817B 8F;          0380          GLO CHAR      ..CK ENTRY FLAG
817C 3A39;        0381          BNZ REXIT      ..BR IF ENTRY WAS VIA READ
817E 9F;          0382          GHI CHAR
817F FF41;        0383          SMI#41       ..CK FOR ASCII HEX
8181 3B2F;        0384          BNF CKDEC      ..(AT TOP OF ROUTINE)
8183 FF06;        0385          SMI#06       ..CK FOR A THRU F
8185 3337;        0386          BDF NFND
8187 ;            0387  ..
8187 ;            0388  ..
8187 FEFEFEFE;    0389  FND:SHL    ;SHL    ;SHL    ;SHL
818B FC08FE;      0390          ADI#08    ;SHL
818E AE;          0391  FND1:PLO AUX      ..READY TO SHIFT INTO RD
818F 8D7EAD;      0392          GLO ASL     ;SHLC    ;PLO ASL      ..SHIFT
8192 ;            0393                                          ..LOW HALF
8192 9D7EBD;      0394          GHI ASL     ;SHLC    ;PHI ASL      ..SHIFT
8195 ;            0395                                          ..HIGH HALF
8195 8EFE;        0396          GLO AUX     ;SHL
8197 3A8E;        0397          BNZ FND1       ..BR IF NOT FINISHED
8199 3039;        0398          BR REXIT
819B ;            0399  ..TYPE ROUTINE--TYPES 1 BYTE FROM @R5!,@R6!,
819B ;            0400  ..OR CHAR.1,OR TYPES A BYTE AS TWO HEX DIGITS
819B ;            0401  ..FROM CHAR.1 FOLLOWS A LINE FEED BY SIX NULLS.
819B ;            0402  ..USES 2 AUXILIARY REGS-AUX AND CHAR-PLUS
819B ;            0403  ..RAM LOCATION @ST.EXITS READY TO TYPE 1 BYTE
819B ;            0404  ..FROM @R5!. EXITS TO R5
819B ;            0405  ..WHEN ENTERED AT TYPE5D,PAUSES TO ALLOW AN
819B ;            0406  ..EARLIER READ TO COMPLETE.
819B ;            0407  ..
819B ;            0408  ..AUX.0 HOLDS OUTPUT CHAR (AT FIRST), THEN
819B ;            0409  ..THE DELAY CONSTANT BETWEEN BITS. CHAR.0 HOLDS
819B ;            0410  ..THE NUMBER OF BITS (11) IN ITS LOWER DIGIT,
819B ;            0411  ..AND IN ITS UPPER DIGIT HOLDS A CODE--
819B ;            0412  ..       0 FOR BYTE OUTPUT
819B ;            0413  ..       1 FOR FIRST HEX OUTPUT
819B ;            0414  ..       2 FOR LST NULL OUTPUT
819B ;            0415  ..       8 FOR LF OUTPUT
819B ;            0416  ..
819B ;            0417          ORG #819C
819C DC17;        0418  TYPE5D:SEP RC; ,#17      ..3 BIT TIME DELAY
819E 38;          0419          SKP       ..SKP TO TYPE5D
819F D5;          0420  TEXIT:SEP R5
81A0 4538;        0421  TYPE5:LDA R5    ;SKP      ..ENTRY FOR UT4
81A2 ;            0422                            ..SKIP TO TYPE
81A2 4638;        0423  TYPE6:LDA R6    ;SKP      ..ENTRY FOR G.P.
81A4 ;            0424                            ..IMMED TH
81A4 9F;          0425  TYPE:GHI CHAR
81A5 AE;          0426  TY1:PLO AUX      ..SAVE BYTE FOR LATER

81A5 AE;          0426  TY1:PLO AUX      ..SAVE BYTE FOR LATER
```



```
81A6 FB0A;        0427          XRI#0A      ..IS IT LINE FEED ?
81A8 3ABF;        0428          BNZ TY2
81AA F88B;        0429          LDI#8B      ..(# OF BITS)+(#OF NULLS
81AC ;            0430                      ..TO FOLLOW LF+1)
81AC 30C1;        0431          BR TY3
81AE 9F;          0432  TYPE2:GHI CHAR      ..UT4 ENTRY
81AF F6F6F6F6;    0433  TY4:SHR    ;SHR    ;SHR    ;SHR      ..SHIFT FIRST
81B3 ;            0434                                        ..HEX TO RIGHT
81B3 FCF6;        0435          ADI#F6      ..CONVERT TO HEX
81B5 3BB9;        0436          BNF *+#04      ..IF A OR MORE
81B7 FC07;        0437          ADI#07      ..ADD NET 37
81B9 FFC6AE;      0438          SMI#C6    ;PLO AUX      ..ELSE ADD NET 30
81BC F81B;        0439          LDI#1B      ..10+(# OF BITS)
81BE C8;          0440          LSKP      ..EQUIV. TO BR TY3
81BF ;            0441  ..
81BF F80B;        0442  TY2:LDI#0B      ..(# OF BITS TO OUTPUT)
81C1 AF;          0443  TY3:PLO CHAR      ..SAVE MAIN TALLY VALUE
81C2 ;            0444  ..
81C2 ;            0445  ..
81C2 7B;          0446  BEGIN:SEQ      ..START BIT
81C3 8E;          0447          GLO AUX      ..GET CHAR TO BE TYPED
81C4 AD;          0448          PLO RD      ..SAVE THE CHAR.
81C5 ;            0449                      ..(AUX.0 CLOBBERED)
81C5 DC07;        0450  PREBIT:SEP RC; ,#07    ..WAIT ONE BIT TIME
81C7 ;            0451            ..RETURN FROM DELAY WITH D=0
81C7 2F;          0452               DEC CHAR      ..DEC THE BIT COUNTER
81C8 F5;          0453          SD      ..SET DF=1
81C9 8D76AD;      0454          GLO RD    ;SHRC    ;PLO RD      ..SHIFT
81CC ;            0455                                ..OUTPUT CHAR
81CC 33D1;        0456          BDF OUT1B      .. BR IF THE BIT IS A 1
81CE 7B;          0457          SEQ      ..ELSE SET Q TO ZERO
81CF 30D3;        0458          BR OUT1B+#02
81D1 7A;          0459  OUT1B:REQ      ..SET Q TO 1
81D2 C4;          0460          NOP      ..DELAY
81D3 8FFA0F;      0461          GLO CHAR    ;ANI#0F      ..FINISHED TYPING ?
81D6 C4C4;        0462          NOP    ;NOP      ..DELAY(14 INSTR.LOOP)
81D8 3AC5;        0463          BNZ PREBIT      ..BR IF NOT FINISHED
81DA 8FFCFB;      0464  NXCHAR:GLO CHAR    ;ADI#FB
81DD AF;          0465          PLO CHAR      ..SET UP FOR NEXT CHAR
81DE 3B9F;        0466          BNF TEXIT      ..BUT EXIT IF NO MORE
81E0 FF1B;        0467          SMI#1B      ..TEST FOR ALTERNATIVES
81E2 329F;        0468          BZ TEXIT      ..IF JUST TYPED LST NULL
81E4 3BEA;        0469          BNF HEX2      ..IF JUST TYPED FIRST HEX
81E6 ;            0470                      ..JUST TYPED LF OR NULL--
81E6 F800;        0471          LDI#00      ..PREPARE TO TYPE NULL
81E8 30F5;        0472          BR HX22
81EA ;            0473  ..
81EA 9FFA0F;      0474  HEX2:GHI CHAR    ;ANI#0F     ..GET 2ND HEX DIGIT
81ED FCF6;        0475          ADI#F6      ..CONVERT TO HEX
81EF 3BF3;        0476          BNF *+#04      ..IF A MORE

                                BNF *+#04      ..IF A MORE
```

```
81F1 FC07;          0477          ADI#07      ..ADD NET 37
81F3 FFC6;          0478          SMI#C6      ..ELSE ALL NET 30
81F5 AE;            0479  HX22:PLO AUX       ..STORE CHAR AWAY
81F6 30C2;          0480          BR BEGIN
81F8 ;              0481  ..
81F8 D30A;          0482  FSYNER:SEP SUB; ,#0A      ..LF
81FA D33F;          0483          SEP SUB; ,#3F      ..?
81FC C08039;        0484          LBR START
81FF ;              0485          END
0000
```

TABLE 1

UT4 REGISTER UTILIZATION

| REGISTER NAME | REGISTER NUMBER | FUNCTION and COMMENTS |
|---|---|---|
| PTER | R0 | Altered by UT4 while storing registers. |
| CL | R1 | |
| SUB | R3 | Program counter for all READ, all TYPE, and TIMALC routines. |
| PC | R5 | Program counter for UT4, which calls the routines above. |
| DELAY | RC | Program counter for the DELAY routine. Points to DELAY1 in memory. |
| ASL | RD | Assembled into by READAH (input hex digits). |
| AUX | RE | AUX.1 holds time constant and echo bit. AUX.0 is used by all READ and TYPE routines and by TIMALC. |
| CHAR | RF | CHAR.1 holds input/output ASCII character. CHAR.0 is used by all READ and TYPE routines |
| CHAR | RF | CHAR.1 holds input/output ASCII character. CHAR.0 is used by all READ and TYPE routines and by TIMALC. |



TABLE 2

ENTRY POINTS FOR UT4 SUBROUTINES

| ENTRY NAME | ABSOLUTE ADDRESS | FUNCTION and COMMENTS |
|---|---|---|
| READ | 813E | Input ASCII → CHAR.1, D (if non-standard linkage). |
| READAH | 813B | Same as READ. If hex character, DIGIT → ASL (see text). |
| TTYRED | 8140 | Same as READ. Controls paper tape reader (see text). |
| TYPE5D | 819C | 1.5-bit delay. Then TYPE5 function. |
| TYPE5 | 81A0 | Output ASCII character at M(R5). Then increment R5. |
| TYPE6 | 81A2 | Output ASCII character at M(R6). Then increment R6. |
| TYPE | 81A4 | Output ASCII character at CHAR.1. |
| TYPE2 | 81AE | Output hex digit pair in CHAR.1. |
| TIMALC | 80FE | Read input character and set up control byte in AUX.1. Initialize RC to point to DELAY1. |
| DELAY1 | 80EF | Delay, as function of M(R3) (see text). Then R3+1 → R3. |

NOTES:

1. All routines except DELAY use R3 as program counter, exit with SEP5, and alter registers X, D, DF, AUX, and CHAR.
2. DELAY routine uses RC as program counter, exits with SEP3 after incrementing R3, and alters register X, D, DF, and AUX.
3. READ and READAH exit with R3 pointing back at READAH.
4. All five TYPE routines exit with R3 pointing at TYPE5.
5. As indicated in Table 3-I, ASL = RD, AUX = RE, and CHAR = RF.

5. As indicated in Table 3-I, ASL = RD, AUX = RE, and CHAR = RF.

## OUTPUT ROUTINE USING UT4

The monitor program UT4 includes all the software to use the "software" UART with EF4 and Q line,not only for the monitor program, but as well these subroutines can be called by user programs. This is very useful,for example, to send messages to the terminal during program execution.

Description of the program :

1. At the first instruction,interrupts have to be disabled as the whole timing is via counters that cannot be interrupted (line 9).

2. The TYPE subroutine comes back with a SEP 5 instruction, which means it has to be called from a program running in R5 as PC (10-12).

3. A DELAY counter has to be initialized (14-15).

4. Timing constant and echo bit have to be loaded (17-18).

5. The text pointer has to be prepared, pointing to the text bytes at add #0036 (20-21).

6. A call of subroutine at #819C gets the timing right (23-25).

7. Now the output routine starts with preparing R3 to 81A4 (27).

8. Now the first ASCII character is loaded via R6 in D and then to RF.1, where the subroutine TYPE gets the byte from (28).



9. Branch now if this byte in D is #00, which means end of string (29,43).

10. If it is not #00, do a SEP 3 and call the TYPE routine (30).

11. TYPE exits with a SEP 5 which means the program continues at location #002B and branches back to OUTPUT (31).

12. If the string has been sent a delay routine is initialized and when it has finished, the same string is sent again.