Order this document
as AN459/D

AN459

A Monitor for the MC68HC05E0

Peter Topping,
MCU Applications Group,
Motorola Ltd., East Kilbride

INTRODUCTION

Development systems for single-chip MCUs can be complex and expensive. This can dissuade potential users of this type of microprocessor from designing them into new applications. This application note describes a simple "entry" development system suitable for debugging hardware and software for the M6805 range of microprocessors. The M6805 range includes both CMOS and NMOS parts, most of which are single-chip devices which include mask-programmable ROM, RAM, I/O and one or more timers. The exceptions are the CMOS MC146805E2 and MC68HC05E0 which have no on-chip ROM but instead have data and address buses which enables them to use external memories and peripherals.

The development system uses the MC68HC05E0 processor and can be used to develop applications intended for any M6805 but is particularly suitable for applications which will use the MC68HC05E0 itself. The MC68HC05E0 has a non-multiplexed bus which requires no additional hardware to interface with RAMs, ROMs and EPROMS. It has 480 bytes of RAM, 3 timers, 3 chip-select lines and 4 8-bit ports enabling it to meet many requirements with the addition of only an EPROM containing the application software.

Such a system will be most cost effective in small volume applications not justifying a mask programmed single-chip MCU and also in applications requiring larger programs than can be accommodated in ROMed MCU versions.

For software development, however, RAM and a monitor have to be incorporated so software can be loaded and de-bugged. An MCM6264 8Kbyte (or MCM60L256/MCM6206 32Kbyte) RAM is used. The EBUG05 monitor EPROM, listed at the end of this application note, fulfils the monitor requirement.

In an E0 application the development system can be used as an add-on to the target system hardware, as shown in Figure 1 (components to the left of the dotted line). The application hardware needs very little modification as it already contains the MPU, the interface to the development system being mostly via the socket for the EPROM. Apart from the keyboard and display connections which use port A, only four or five additional connections are required. These can be made available on a small connector on the application PCB. In one of the 8K applications, used to check out the development system (reference 1), a keyboard and display were required, so only four I/O lines (see below) were dedicated to the development system. In a 16K application, A13 (PD5) is also required, so a fifth I/O line would be used. The keyboard provides up to 32 keys using 7 I/O lines and an external 3-line to 8-line decoder. One additional I/O line is used for the display. If the keyboard and/or display is not required in the final application (e.g., reference 2) then the I/O lines, used by them during development, are available for other purposes but their use cannot readily be de-bugged. This is a disadvantage of such a simple development system. There is, however, a major advantage of this system compared with those which use the display and keyboard of a PC. In this system the PC can be used to display the program listing file, obviating the need for printouts during debug.

© Freescale Semiconductor, Inc., 2004. All rights reserved.

Freescale Semiconductor, Inc.

CIRCUIT

Figure 1 shows the circuit used. The 8K or 16K (half of an MCM60L256) RAM is placed between $0000 and $3FFF; this means that the 512 bytes from $0000 to $01FF are not used, as they are mapped over the E0's I/O and RAM space. Locations $0200 to $021F are used by the monitor, so use of the RAM should start at $0220 or above. Some RAM may be required to replace E0 page 0 RAM (16 bytes), used by the monitor or for other purposes during de-bug; the application used to check out the monitor used RAM from $0400. This provides 7K (15K using an MCM60L256) of RAM for the code being de-bugged. The start address chosen for the code being developed can be extended down to $0220 if this RAM is not required, providing a program space of nearly 15½Kbytes. The 16K limitation is caused by the split of the memory map into four (see Figure 3), to simplify address decoding and facilitate the ability to load code from an EPROM. If this capability is not required, more complex address decoding would allow up to 60K of RAM to be available for code; the monitor uses less than 4K.

The MC74HC138 provides the chip selects during de-bug as the on-chip chip selects signals are not suitable for the monitor's address map. The final system will usually not require an MC74HC138.

The 28-pin EPROM socket which will contain the final de-bugged code is used during development as the main interface to the monitor and its RAM. Only four or five other signals are required. These are the two or three most significant address lines, A15, A14 and, optionally, A13 and the clock (P02), which are required by the MC74HC138, the R/W line and a port line (PB2) for inputting or outputting code on an RS232 serial link. PB2 can be used in the application as it is only used by the monitor during serial transfers when the application is not running. Care should, however, be taken to avoid contention within the hardware connected to PB2. If only 8K bytes of external RAM are required, then A13 is not required and its pin, PD5, can be used by the application.

The hardware connected to the EPROM socket can also include a 28-pin socket to accommodate a 27(C)64 or 27(C)128 EPROM. This is addressed between $4000 and $7FFF and can be used to introduce software from an EPROM which has been programmed with code for de-bug. The monitor contains a routine which transfers the contents of this EPROM into RAM. Alternatively, code can be loaded serially via the RS232 interface.

The EBUG05 monitor EPROM (27C64) included in the external hardware is enabled from $C000 to $FFFF, although its actual start address is $E000. Figure 3 shows the memory map of the development system.

The monitor display is a 6-digit 4-backplane LCD (e.g., Hamlin type 4200 or the 8-digit GE type LXD69D3F09KG), which is driven by an MC145000P display driver. The driver is controlled by a 2-line serial link from the microprocessor. A single-backplane display can be used as an alternative, as shown in Figure 2. Three MC144115 driver chips are used along with a transistor inverter to supply the additional latch enable signal required by these drivers. This circuit requires more connections to the LCD but allows the use of a more commonly available display.

The optional RS232 interface can be implemented using the single-supply MC145407 driver-receiver chip. If outputting of S-records is not required, then a simple transistor inverter with a pull-up resistor and a reverse polarity protection diode can be used. This interface is shown in Figure 4.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

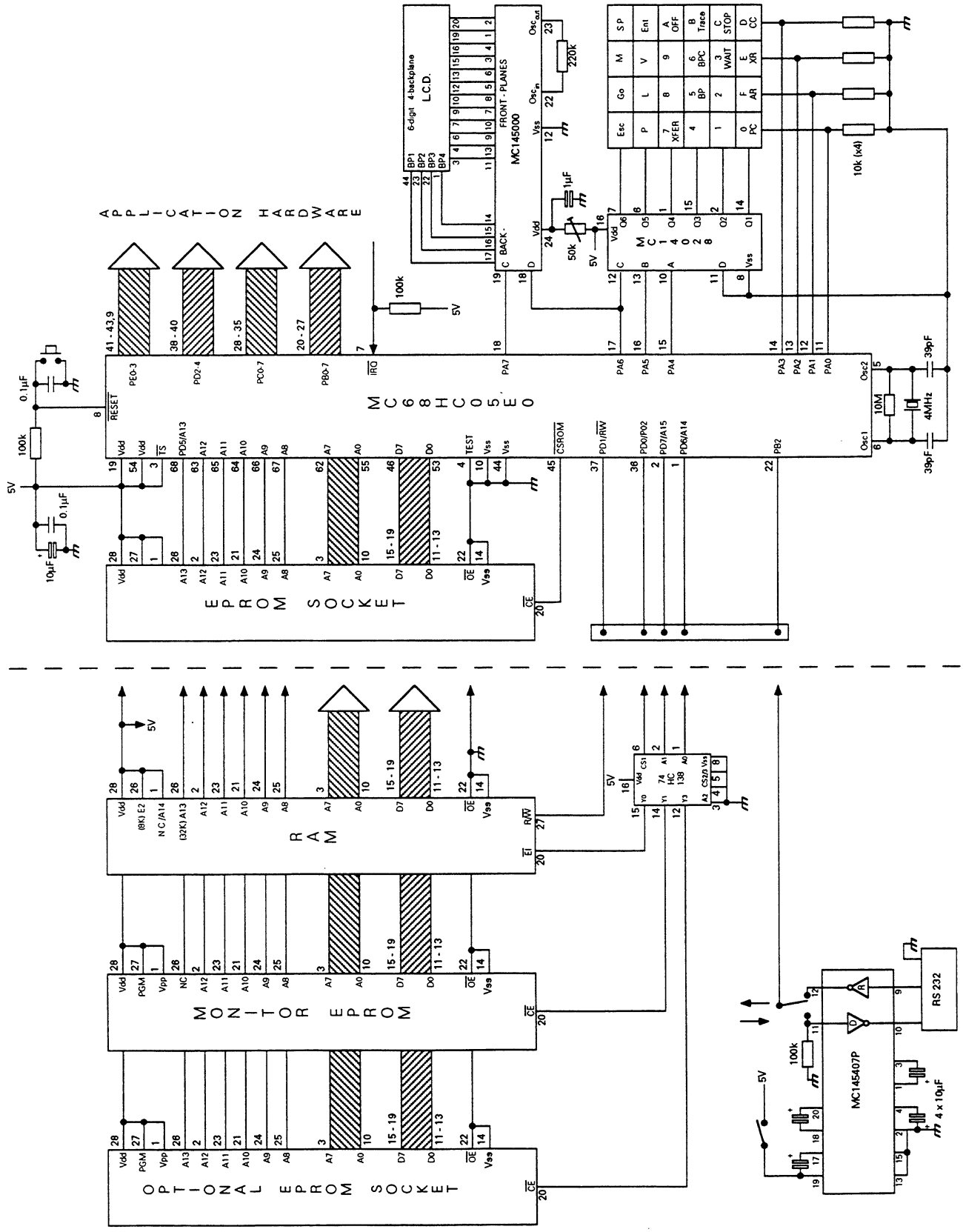

Figure 1. MC68HC05E0-based M6805 development system

Freescale Semiconductor, Inc.

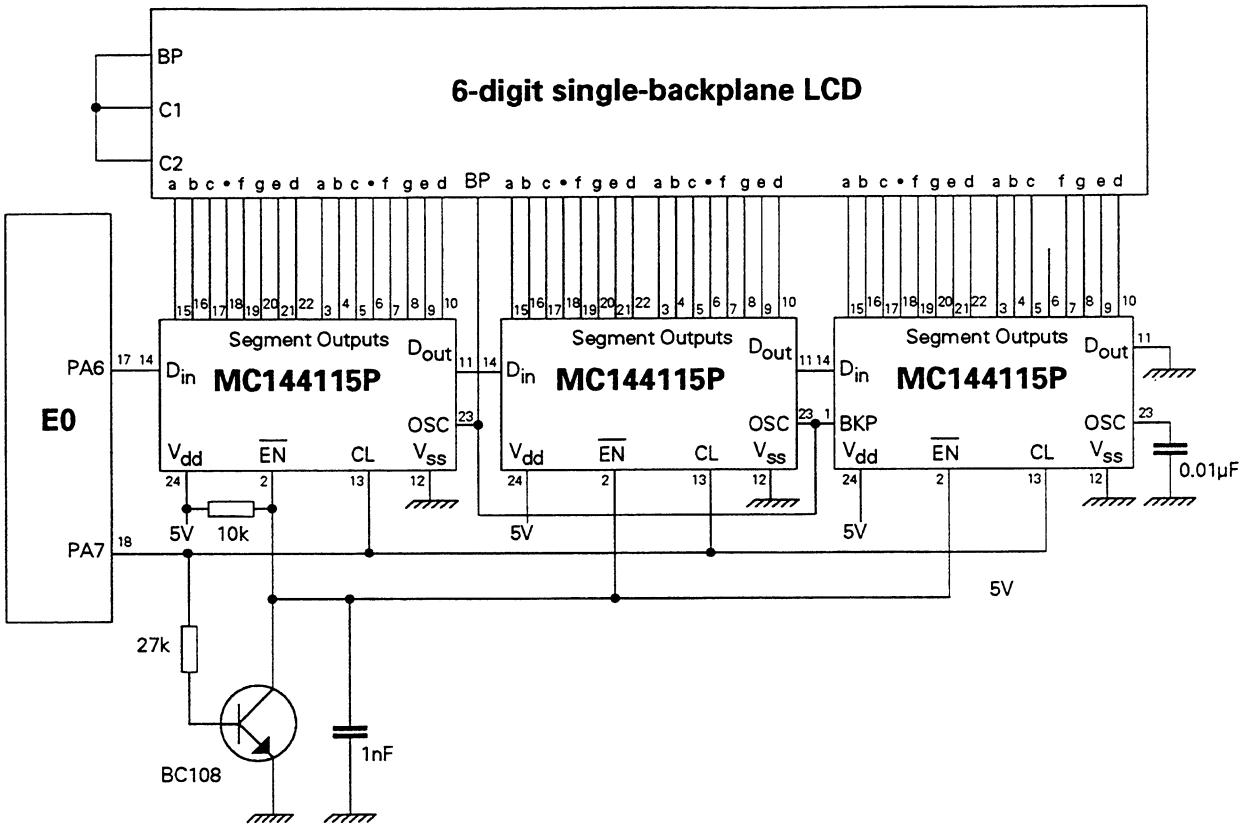

Figure 2. Alternative static LCD display

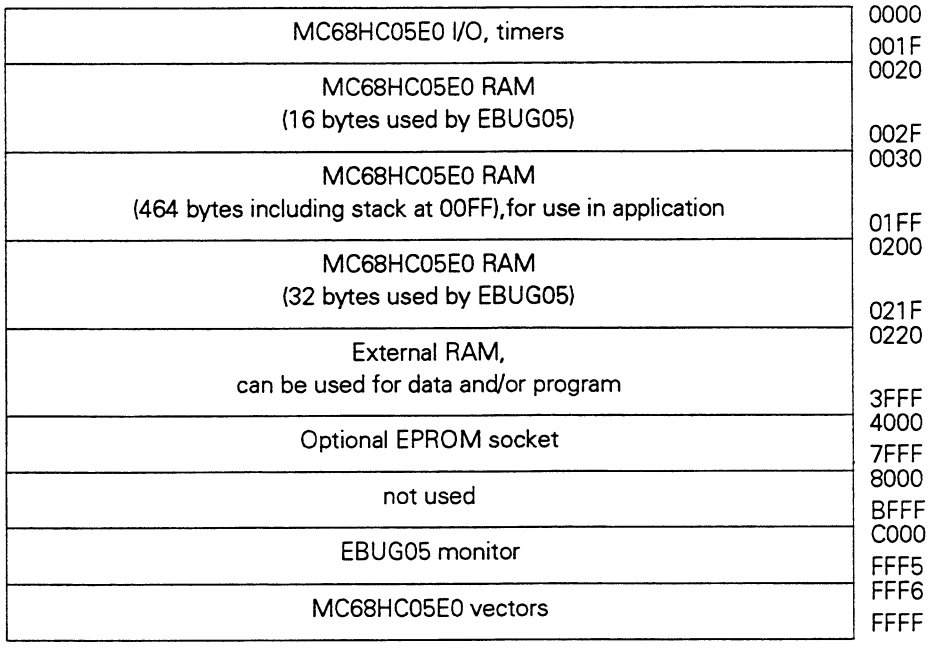

Figure 3. Memory Map

Freescale Semiconductor, Inc.

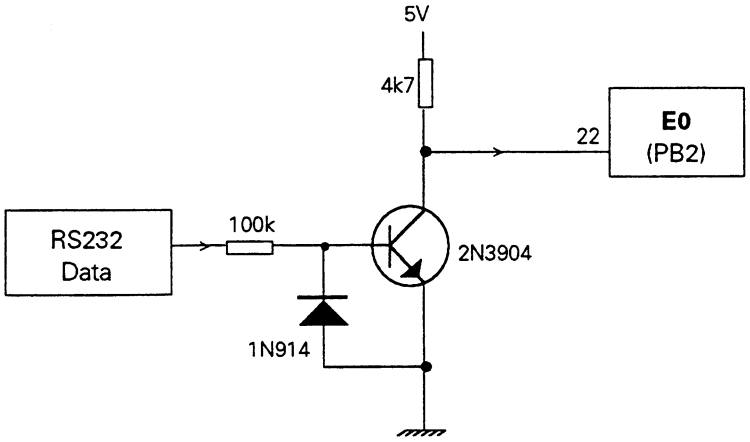

Figure 4. Simple input-only RS232 interface

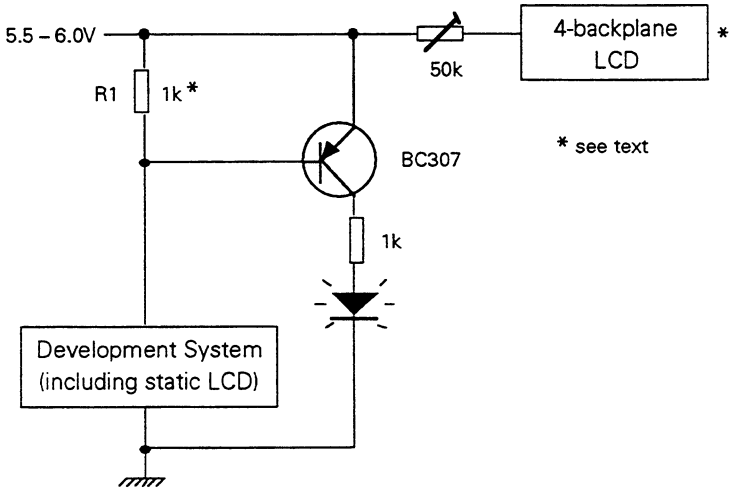

Figure 5. IDD Monitor Circuit

IDD INDICATOR

It is often useful with CMOS circuits to provide a simple IDD monitor which shows via an LED whether or not the IDD is above or below a specific value. In this application it shows whether or not the microprocessor is in the STOP mode. The required circuit is shown in Figure 5. The current threshold can be chosen by selecting the value of R1. A value of 1kohm sets the limit at about 500µA which means the LED should be off when the E0 is in STOP mode and on otherwise. The 500µA limit allows the LCD and perhaps a CMOS target application to be supplied without switching on the LED. When the microprocessor is not in STOP mode, its IDD is several milliamps and the LED should be lit. If a multiplexed LCD is used, it may be preferable not to supply it via this type of monitor circuit, as a significant change in contrast may occur as the microprocessor goes into its STOP mode (see Figure 5). The monitor drops about 600mV when the microprocessor is running, so the supply voltage should be chosen accordingly. While the microprocessor will operate down to 3V (at an appropriate frequency), the RAM and EPROM chips are specified at 5V ±0·5V. For battery operation 4 zinc-carbon or 5 Ni-Cad cells can be used.

Freescale Semiconductor, Inc.

EBUG05 KEY FUNCTIONS

The EBUG05 EPROM comprises a monitor, developed from the MC146805E2 CBUG05 program, specifically written to enter and debug 6805 code. The following functions are available using the 24-key keyboard:

Function	Key	Description of function
PC	0	Display program counter.
BP	5	Enable breakpoints. The first breakpoint is displayed if enabled or boff if it isn't. Enter new breakpoint address followed by ENTER to change, or just ENABLE to move to next one, three breakpoints are available. This number can be increased, the only cost being the additional RAM used (3 bytes per breakpoint).
BPC	6	Clear breakpoints. ENTER clears all breakpoints. Entering a number followed by ENTER clears that breakpoint only.
OFF	A	Calculate branch offset. The address of the branch instruction and that of the destination are requested. If a valid branch is calculated, it is written into memory and displayed. If not valid, then "or" for out of range is displayed. A branch of -128 through +127 relative to the start address of the next instruction is allowed.
TR	B	"Trace" one instruction. This is not a true trace as breakpoints are advanced sequentially through the code and do not follow branches or jumps (see below).
STOP	C	Put the E0 into STOP mode, clock stops.
WAIT	3	Put the E0 into WAIT mode.
CC	D	Display/change Condition Code register, new data is followed by ENTER. ESC returns to "□" prompt without changing contents.
XR	E	Display/change indeX Register, new data is followed by ENTER. ESC returns to "□" prompt without changing contents.
AR	F	Display/change Accumulator, new data is followed by ENTER. ESC returns to "□" prompt without changing contents.
P	P	Output S-records. When this key is pressed, "bA" for begin address is displayed; enter first address (and ENTER), then last address when "EA" is displayed. Another ENTER starts the RS232 S-record output.
L	L	Load S-records from the RS232 interface. Press L and "LOAd" is displayed. S-records sent to 2, PORTB, will be loaded until an S9 record is received. The prompt returns if the load was OK. If not, an error message is displayed (see next section).
V	V	Verify RAM against S-records. "UErIFy" is displayed, otherwise the same as LOAD including error checks.
ENTER	ENT	Enter keyed-in address or data (and move to next address in "M").
ESC	ESC	Escape from current function (except STOP, WAIT, V, L & P).

Freescale Semiconductor, Inc.

Function	Key	Description of function
GO	GO	Begin or continue program. When pressed current PC is displayed. To proceed from that location, press ENTER; to proceed from a different address, enter the address followed by ENTER.
M	M	Display/change a location in RAM. When pressed, last address is displayed. Press ENTER to display the contents of this address or enter a new address, followed by ENTER. New contents can be entered (followed by ENTER) if required. ENTER moves to next address; M moves to previous address.
SP	SP	Display stack pointer; cannot be modified.
XFER	7	Transfer code from an EPROM between $4220 and $7FFF into RAM ($0220 to $3FFF). When pressed, default start address (in RAM) is displayed. Press ENTER to start at this address ($0400), or enter new start address followed by ENTER. End address (in RAM) is displayed. Press ENTER to transfer code up to this end address ($1FFF), or enter new end address followed by ENTER. The lowest allowable start address is $0220 and the highest end address is $3FFF. During the transfer the code is read from the EPROM at the RAM addresses plus $4000. The code in EPROM should have been assembled with the start address where it will reside during execution (e.g., $0400).

USE OF THE MONITOR

The MC68HC05E0's vectors are within the address space of the EBUG05 EPROM. They operate via extended jumps in RAM. This gives the user access to the vectors if these jumps are written to, from within the user's program. The vectors' locations are shown in Table 1. The extended jump instruction ($CC) is written to RAM by the monitor; all the user has to do is to overwrite the destination addresses at the locations indicated in the Table. This must be done at the start of the user code, before interrupts are enabled. Vectoring the jumps through a table at a fixed location in the user code obviates the need to change this initialisation when re-assembling changes the addresses of the interrupt service routines. Lines of code for this purpose are included as comments in reference 2.

Clearly the RESET vector cannot be altered in this way, so during debug user software should be started using the GO facility in EBUG05. Software interrupts (SWIs) are used by the monitor to facilitate breakpoint and should therefore not be used in the application software.

Software for de-bug should be assembled to reside in RAM, starting at address $400 (or down to $220 if more space is required, see above). It can be introduced into the system in an EPROM at $4000 or loaded serially from a PC via the RS232 interface. The monitor includes a routine to move the code from EPROM to RAM, where it can be executed and debugged using EBUG05. This provides an alternative method of loading code for debug if a serial load is not appropriate. When debug is complete, the code should be re-assembled at $E000 (or $C000 if a 16K EPROM is used). When the code is in EPROM, the vectors should be included. The jumps required during debug are then no longer relevant.

CSROM will normally be suitable for selecting this EPROM, so that the MC74HC138 will not be required once the code has been finalised.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

Table 1.

Vector	Address	JMP location in RAM
SPI/IIC	$FFF4–5	$0209 (add: $20A/B)
Timer B	$FFF6–7	$0206 (add: $207/8)
Timer A	$FFF8–9	$0203 (add: $204/5)
IRQ	$FFFA–B	$0200 (add: $201/2)
SWI	$FFFC–D	used by monitor
RESET	$FFFE–F	use "GO" function

BREAKPOINTS

Up to three breakpoint addresses can be entered; this number can be easily increased if required by changing the EQU on line 25 of EBUG05 source code. To allow the continuation of execution from a breakpoint, a breakpoint at the start address specified by a GO is ignored. This means that a single breakpoint in a loop will only be encountered once if, after it has occurred, execution is re-commenced with a GO from the current address. If it is required to stop at the breakpoint again, then one of the two procedures described here should be used. Two breakpoints can be entered at different places in the loop. They will be encountered alternately; clearly, two GOs are required for each execution of the loop. A second method is to insert only one breakpoint but to trace one instruction before continuing with a GO. This trace takes the program counter past the breakpoint address and so allows the breakpoint to be re-inserted when GO is executed. This will only work reliably if the breakpoint in not on a branch, jump, or any other instruction that could cause the program to proceed to an address other than the next sequential address (see below).

If a single breakpoint has been entered and encountered and execution re-started with a GO from the same address, then pressing RESET is the only method of returning to the monitor. Normally pressing RESET when the program is running with breakpoints valid is not recommended as the application code will be left corrupted with SWIs replacing the instructions at the breakpoint addresses. If, however, control has been lost because execution was recommenced from the address of a single breakpoint, then corruption will not occur as the breakpoint will not have been re-inserted.

Breakpoints are inserted when TRace or GO are executed but removed when a breakpoint is encountered. The trace facility inserts a breakpoint (SWI) at the address of the instruction sequentially after the current instruction and then executes the current instruction. If the current instruction jumps or branches somewhere else then the SWI will not be encountered. This is thus not a true program-following trace but in many cases is more useful as often subroutines do not need to be traced, variables being required to be checked only after each subroutine has been executed. If it is required to trace the program flow to the other address, then another breakpoint should be inserted at that address.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

SERIAL LOAD

To load external Motorola S-records, the serial load key (L) should be pressed. The LCD will display "LOAd". S-records should then be supplied at 9600 baud (8-bit, no parity) on the RS232 interface. When an S9 termination record is received, the prompt returns. If an error is detected during a serial load, the load routine stops and displays the address at which the error occurred and the error type. The following error types are possible.

1: Checksum error, transmitted data or interface faulty.

2: RAM read-back error, RAM faulty or nonexistent.

3: ASCII character less than $30 (0) received.

4: ASCII character between $39 (9) and $41 (A) received.

5: ASCII character more than $46 (F) received.

7: Verify error when comparing S-records with RAM.

The verify function can be used to check that the RAM has not been corrupted. The VERIFY function is used exactly like LOAD except that RAM is compared with, rather than loaded by, the S-records.

The S-record format can be seen in the example below. The number following the S is the record type. The monitor only recognises S1 (8-bit data) and S9 (termination) records. The next byte (23 in most of the records) is the number (in hex) of bytes which follow. This is followed by a 2-byte address (e.g., E000 and FFF4), then the data. The last byte is a checksum byte. The sum of all bytes, including the byte count and the checksum byte, is $XXFF. This can be easily checked on the S9 record (03+00+00+FC=FF).

```
S123E000CDE05F24FB5FB728D6E09FB128270BC1E0BF273A5C5C5C5C20EEA601B70E5CDC50
S123E020E09FA6E3B7123F00A6F0B705A658B701A6FBB7063F02A6FFB7073F03A61CB708FF

S123E5E0305A26F8AD3020EAA1102705A11126E29981AE04BF28B72D4444444497D6E4CD2C
S123E600BE28E720B62DA40F97D6E4CDBE28E721CDE1ECB62D81B72EBF2CB6305FADD5B61C
S10DE6202BAE02ADCFB62EBE2C8146
S10FFFF4E022E022E022E04EE022E022C5
S9030000FC
```

SERIAL INTERFACE

Figure 6 shows a suggested method of wiring up the RS232 sockets in a system with both loading and dumping capabilities. This arrangement facilitates use of the serial LOAD and DUMP routines of the monitor either via a PC COM port or between a host and terminal connected by an RS232 link. When using a PC, the "host" socket should be used. As only one pin on the MC68HC05E0 is used, switching is required to make the required connections. S2 can be eliminated (or left at "L") if only loading is required, as will often be the case. To save power in battery applications, the RS232 interface chip can be switched off using S1. Table 2 shows possible methods of use.

Freescale Semiconductor, Inc.

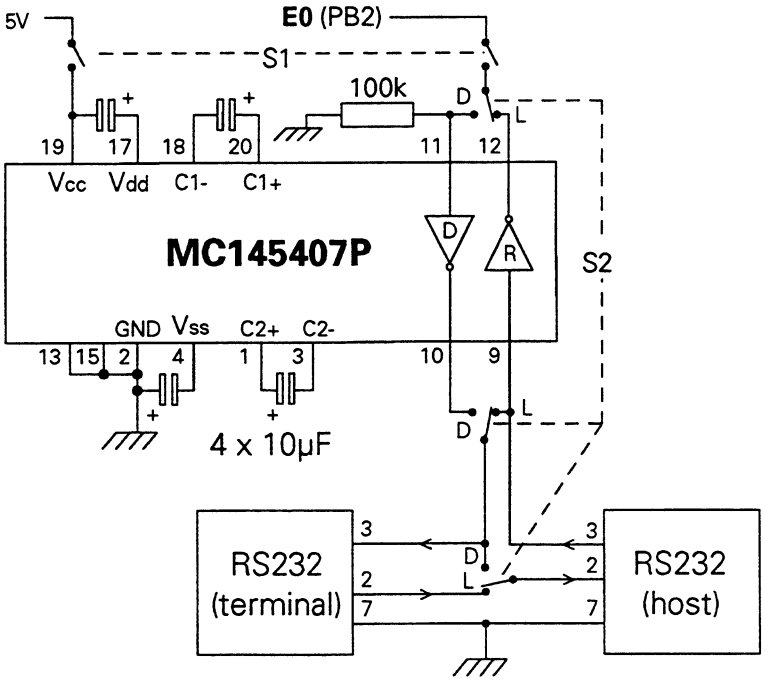

Figure 6. RS232 serial interface with Load/Dump switching

Table 2.

Set-up	Function	S1	S2	Comments
Host & terminal	Load	On	L	Terminal and host connected. Micro looks at data sent from host to terminal (pins 3).
	Dump	On	D	Connection between terminal and host broken, S-records sent to both host (2) and terminal (3).
PC "COM" port	Load	On	L	S-records loaded from pin 3.
	Dump	On	D	S-records sent to pin 2 on "host" socket (and pin 3 on "terminal" socket).

REFERENCES

1) AN441/D, An EPROM Emulator using the MC68HC05E0.

2) AN460/D, An RDS Decoder using the MC68HC05E0.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

APPENDIX – EBUG05 E0 monitor listing

```
0001
0002                    ***************************************************************
0003                    *                                                             *
0004                    *                     EBUG05 E0 monitor.                      *
0005                    *                                                             *
0006                    *                                                             *
0007                    *       P. Topping                      18th November '91     *
0008                    *                                                             *
0009                    ***************************************************************
0010
0011 0020                       ORG     $0020
0012
0013 0000               PORTA   EQU     0               PORT A ADDRESS
0014 0001               PORTB   EQU     1                *   B    *
0015 0002               PORTC   EQU     2                *   C    *
0016 0003               PORTD   EQU     3                *   D    *
0017 0004               PORTE   EQU     4                *   E    *
0018 0005               PORTAD  EQU     5               PORT A DATA DIRECTION REG.
0019 0006               PORTBD  EQU     6                *   B    *       *       *
0020 0007               PORTCD  EQU     7                *   C    *       *       *
0021 0008               PORTDD  EQU     8                *   D    *       *       *
0022 0009               PORTED  EQU     9                *   E    *       *       *
0023 000c               TCR     EQU     $0C             TIMER CONTROL REGISTER
0024 0012               PORTDSF EQU     $12             PORT D ALTERNATIVE FUNCTION REG.
0025 0003               NBKPT   EQU     3
0026
0027 0020               STAT    RMB     1               0: Not first SWI after RESET
0028                    *                               1: PROCEED (0: GO)
0029                    *                               4: No address entered (BLDRNG)
0030                    *                               5: Verifying (TLOAD)
0031                    *                               6: Register contents being changed
0032                    *                               7: Last S1 record (PUNCH)
0033 0021               WORK2   RMB     1               RAM SUB-ROUTINE LDA/STA
0034 0022               ADDRH   RMB     1                *       *       ADDRESS H
0035 0023               ADDRL   RMB     1                *       *       ADDRESS L
0036 0024               WORK3   RMB     1                *       *       RTS
0037 0025               WORK5   RMB     1
0038 0026               WORK6   RMB     1
0039 0027               TEMP    RMB     2               TEMP. ADDRESS
0040 0029               PNCNT   RMB     1               No. BREAKPOINTS
0041 002a               CHKSUM  RMB     1               CHECKSUM (SERIAL)
0042 002b               COUNT   RMB     1               BIT COUNTER
0043 002c               TMP1    RMB     1               TEMP (SERIAL)
0044 002d               TMP2    RMB     1                *       *
0045 002e               BCNT    RMB     1               BYTE COUNT
0046
0047 0200                       ORG     $0200
0048
0049 0200               IRQ     RMB     3               DE-BUG RAM VECTOR, IRQ
0050 0203               TIRQA   RMB     3                *       *       *       TIMER A
0051 0206               TIRQB   RMB     3                *       *       *       TIMER B
0052 0209               SIRQ    RMB     3                *       *       *       SERIAL
0053 020c               DTABL   RMB     6               DISPLAY TABLE
0054 0212               WORK1   RMB     1
0055 0213               WORK4   RMB     1
0056 0214               TMPTCR  RMB     1               TEMP. TCR (XROM USED BY MONITOR)
0057 0215               PROP    RMB     1               INSTRUCTION (PROCEED)
0058 0216               BKPTBL  RMB     3*NBKPT         B.P. TABLE
0059
0060
0061 e000                       ORG     $E000
0062
```

Freescale Semiconductor, Inc.

```
0063                    ***************************************************
0064                    *                                                 *
0065                    *       Reset.                                    *
0066                    *                                                 *
0067                    ***************************************************
0068
0069 e000 a6 f0        RESET   LDA     #$F0            SETUP PORT
0070 e002 b7 05                 STA     PORTAD          FOR KEYPAD
0071 e004 a6 ff                 LDA     #$FF            SETUP PORTD
0072 e006 b7 12                 STA     PORTDSF         FOR ADDRESSES ETC.
0073
0074 e008 a6 cc                 LDA     #$CC            VECTORS IN RAM
0075 e00a c7 02 00              STA     IRQ
0076 e00d c7 02 03              STA     TIRQA
0077 e010 c7 02 06              STA     TIRQB
0078 e013 c7 02 09              STA     SIRQ
0079 e016 c6 e0 ac              LDA     VECTOR
0080 e019 c7 02 01              STA     IRQ+1
0081 e01c c6 e0 ad              LDA     VECTOR+1
0082 e01f c7 02 02              STA     IRQ+2
0083 e022 c6 e0 ae              LDA     VECTOR+2
0084 e025 c7 02 07              STA     TIRQB+1
0085 e028 c6 e0 af              LDA     VECTOR+3
0086 e02b c7 02 08              STA     TIRQB+2
0087 e02e c6 e0 b0              LDA     VECTOR+4
0088 e031 c7 02 04              STA     TIRQA+1
0089 e034 c6 e0 b1              LDA     VECTOR+5
0090 e037 c7 02 05              STA     TIRQA+2
0091 e03a c6 e0 b2              LDA     VECTOR+6
0092 e03d c7 02 0a              STA     SIRQ+1
0093 e040 c6 e0 b3              LDA     VECTOR+7
0094 e043 c7 02 0b              STA     SIRQ+2
0095
0096 e046 ae 20                 LDX     #STAT
0097 e048 7f            INIT    CLR     0,X             CLEAR
0098 e049 5c                    INCX                    WORKING
0099 e04a a3 2e                 CPX     #BCNT           STORAGE
0100 e04c 23 fa                 BLS     INIT
0101
0102 e04e cd e6 1f              JSR     SCNBKP          CLEAR
0103 e051 a6 ff                 LDA     #$FF            ALL
0104 e053 d7 02 16      REBCLR  STA     BKPTBL,X        BREAKPOINTS
0105 e056 5c                    INCX
0106 e057 5c                    INCX
0107 e058 5c                    INCX
0108 e059 3a 29                 DEC     PNCNT
0109 e05b 26 f6                 BNE     REBCLR
0110 e05d 83                    SWI
0111
0112
```

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

```
0113                    ****************************************************
0114                    *                                                  *
0115                    *       SWI.                                       *
0116                    *                                                  *
0117                    ****************************************************
0118
0119 e05e 00 20 04      SWI     BRSET   0,STAT,SWICHK    FROM RESET?
0120 e061 10 20                 BSET    0,STAT          YES
0121 e063 20 4f                 BRA     GETCMD
0122 e065 b6 0c         SWICHK  LDA     TCR             GET CURRENT TCR
0123 e067 ae 0c                 LDX     #$0C            SET XROM SO THAT EXTERNAL RAM
0124 e069 bf 0c                 STX     TCR             CAN BE WRITTEN TO
0125 e06b c7 02 14              STA     TMPTCR          SAVE CURRENT TCR
0126 e06e cd e6 74      STAY    JSR     KEYSCN
0127 e071 cd e6 1f              JSR     SCNBKP          REMOVE
0128 e074 d6 02 16      SWIREP  LDA     BKPTBL,X        BREAKPOINTS
0129 e077 2b 0d                 BMI     SWINOB
0130 e079 b7 22                 STA     ADDRH
0131 e07b d6 02 17              LDA     BKPTBL+1,X
0132 e07e b7 23                 STA     ADDRL
0133 e080 d6 02 18              LDA     BKPTBL+2,X
0134 e083 cd e7 57              JSR     STORE
0135 e086 5c            SWINOB  INCX                    GET NEXT B.P.
0136 e087 5c                    INCX
0137 e088 5c                    INCX
0138 e089 3a 29                 DEC     PNCNT
0139 e08b 26 e7                 BNE     SWIREP
0140 e08d cd e1 23              JSR     LOCSTK          FIND STACK
0141 e090 e6 08                 LDA     8,X
0142 e092 a0 01                 SUB     #1              ADJUST PC
0143 e094 e7 08                 STA     8,X             (MINUS 1)
0144 e096 b7 23                 STA     ADDRL
0145 e098 e6 07                 LDA     7,X
0146 e09a a2 00                 SBC     #0
0147 e09c e7 07                 STA     7,X
0148 e09e b7 22                 STA     ADDRH
0149
0150 e0a0 03 20 06              BRCLR   1,STAT,NOPRO    PROCEED ?
0151 e0a3 c6 02 15              LDA     PROP
0152 e0a6 cd e7 57              JSR     STORE
0153
0154 e0a9 cc e1 35      NOPRO   JMP     PCOUNT          PRINT P.C.
0155
0156                    ****************************************************
0157                    *                                                  *
0158                    *       Default interrupt vectors.                 *
0159                    *                                                  *
0160                    ****************************************************
0161
0162 e0ac e8 9c         VECTOR  FDB     IRQV            IRQ
0163 e0ae e8 9c                 FDB     TIRQBV          TIMER B
0164 e0b0 e8 9c                 FDB     TIRQAV          TIMER A
0165 e0b2 e8 9c                 FDB     SIRQV           SERIAL
0166
0167
```

Freescale Semiconductor, Inc.

```
0168                    ***************************************************
0169                    *                                                 *
0170                    *       Print logo & scan for keypress.           *
0171                    *                                                 *
0172                    ***************************************************
0173
0174 e0b4 cd e6 42      GETCMD  JSR     CLRTAB
0175 e0b7 a6 e4                 LDA     #$E4            PRINT
0176 e0b9 c7 02 0c              STA     DTABL
0177 e0bc cd e6 4c      DSCN    JSR     DISTAB          PROMPT
0178 e0bf cd e6 74      CMDSCN  JSR     KEYSCN          CHECK KEYPAD
0179 e0c2 24 fb                 BCC     CMDSCN
0180 e0c4 5f                    CLRX
0181 e0c5 c7 02 12              STA     WORK1
0182 e0c8 d6 e0 df      RJUMP   LDA     PTABL,X         THIS COMMAND?
0183 e0cb c1 02 12              CMP     WORK1
0184 e0ce 27 0b                 BEQ     PJUMP           YES
0185 e0d0 c1 e1 1f              CMP     LAST
0186 e0d3 27 df                 BEQ     GETCMD
0187 e0d5 5c                    INCX                    NO
0188 e0d6 5c                    INCX                    GO TO
0189 e0d7 5c                    INCX                    NEXT
0190 e0d8 5c                    INCX                    POSSIBLE
0191 e0d9 20 ed                 BRA     RJUMP           TRY AGAIN
0192 e0db 5c            PJUMP   INCX                    GO TO
0193 e0dc dc e0 df              JMP     PTABL,X         COMMAND
0194
0195 e0df 11            PTABL   FCB     $11
0196 e0e0 cc e1 35              JMP     PCOUNT          0  PROGRAM COUNTER
0197 e0e3 12                    FCB     $12
0198 e0e4 cc e1 4f              JMP     AREG            F  ACCUMULATOR
0199 e0e7 14                    FCB     $14
0200 e0e8 cc e1 6d              JMP     XREG            S  INDEX REGISTER
0201 e0eb 18                    FCB     $18
0202 e0ec cc e1 8d              JMP     CCODE           D  CONDITION CODE
0203 e0ef 28                    FCB     $28
0204 e0f0 cc e8 9d              JMP     PWRDWN          C  STOP
0205 e0f3 24                    FCB     $24
0206 e0f4 cc e8 a3              JMP     WDWN            3  WAIT
0207 e0f7 32                    FCB     $32
0208 e0f8 cc e2 cf              JMP     BPDIS           8  DISPLAY/SET BP
0209 e0fb 34                    FCB     $34
0210 e0fc cc e3 46              JMP     BPCLR           9  CLEAR BP
0211 e0ff 38                    FCB     $38
0212 e100 cc e3 a7              JMP     PROC            B  PROCEED TO NEXT INSTRUCTION
0213 e103 41                    FCB     $41
0214 e104 cc e8 16              JMP     XFER            7  TRANSFER FROM EPROM TO RAM (OFFSET: $3C00)
0215 e107 48                    FCB     $48
0216 e108 cc e2 19              JMP     OFFSET          A  OFFSET CALCULATION
0217 e10b 51                    FCB     $51
0218 e10c cc e4 c5              JMP     PUNCH           P  OUTPUT S-RECORDS
0219 e10f 52                    FCB     $52
0220 e110 cc e3 f1              JMP     TLOAD           L  LOAD S-RECORDS
0221 e113 54                    FCB     $54
0222 e114 cc e3 eb              JMP     VERIFY          V  VERIFY S-RECORDS
0223 e117 62                    FCB     $62
0224 e118 cc e5 aa              JMP     NEWGO           G  NEW GO (SKIP CURRENT BP)
0225 e11b 64                    FCB     $64
0226 e11c cc e6 e3              JMP     MEMEX           M  MEMORY INSPECT/MODIFY
0227 e11f 68            LAST    FCB     $68
0228 e120 cc e6 25              JMP     STACK           S  STACK
0229
0230
0231                    ***************************************************
0232                    *                                                 *
0233                    *       Search for stack pointer, X <- SP-3       *
0234                    *                                                 *
0235                    ***************************************************
0236
0237 e123 ad 01         LOCSTK  BSR     LOCST2
0238 00e1               STKHI   EQU     -$8000+*/256+$80
0239 0025               STKLOW  EQU     -256*STKHI+*
0240 e125 81                    RTS
0241 e126 ae 7f         LOCST2  LDX     #$7F
0242 e128 a6 e1         LOCLOP  LDA     #STKHI
0243 e12a 5a            LOCDWN  DECX
0244 e12b f1                    CMP     0,X
0245 e12c 26 fc                 BNE     LOCDWN
0246 e12e a6 25                 LDA     #STKLOW
0247 e130 e1 01                 CMP     1,X
0248 e132 26 f4                 BNE     LOCLOP
0249 e134 81                    RTS
0250
```

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

```
0251                    ****************************************************
0252                    *                                                  *
0253                    *       Display program counter.                   *
0254                    *                                                  *
0255                    ****************************************************
0256
0257 e135 a6 73         PCOUNT  LDA     #$73            PRINT
0258 e137 c7 02 10              STA     DTABL+4         `PC`
0259 e13a a6 d1                 LDA     #$D1
0260 e13c c7 02 11              STA     DTABL+5
0261 e13f ad e2                 BSR     LOCSTK          FIND USER PC
0262 e141 e6 07                 LDA     7,X             HIGH BYTE
0263 e143 b7 22                 STA     ADDRH
0264 e145 e6 08                 LDA     8,X             LOW BYTE
0265 e147 b7 23                 STA     ADDRL           PRINT IT
0266 e149 cd e8 02              JSR     PRTADR
0267 e14c cc e0 bf              JMP     CMDSCN
0268
0269                    ****************************************************
0270                    *                                                  *
0271                    *       Accumulator examine/change.                *
0272                    *                                                  *
0273                    ****************************************************
0274
0275 e14f a6 77         AREG    LDA     #$77            PRINT `ACCA`
0276 e151 c7 02 0c              STA     DTABL
0277 e154 c7 02 0f              STA     DTABL+3
0278 e157 a6 d1                 LDA     #$D1
0279 e159 c7 02 0d              STA     DTABL+1
0280 e15c c7 02 0e              STA     DTABL+2
0281 e15f ad c2                 BSR     LOCSTK          FIND ACCUM. VALUE
0282 e161 9f                    TXA
0283 e162 ab 05                 ADD     #5
0284 e164 3f 22                 CLR     ADDRH           SETUP FOR
0285 e166 b7 23                 STA     ADDRL           EXAMINE/CHANGE
0286 e168 1c 20                 BSET    6,STAT
0287 e16a cc e6 ea              JMP     MEMEX3          USING MEMORY ROUTINE
0288
0289
0290                    ****************************************************
0291                    *                                                  *
0292                    *       Index reg. examine/change.                 *
0293                    *                                                  *
0294                    ****************************************************
0295
0296 e16d cd e6 42      XREG    JSR     CLRTAB
0297 e170 a6 06                 LDA     #6              PRINT `Idr`
0298 e172 c7 02 0d              STA     DTABL+1
0299 e175 a6 e6                 LDA     #$E6
0300 e177 c7 02 0e              STA     DTABL+2
0301 e17a a6 60                 LDA     #$60
0302 e17c c7 02 0f              STA     DTABL+3
0303 e17f ad a2                 BSR     LOCSTK          FIND INDEX
0304 e181 9f                    TXA                     REGISTER VALUE
0305 e182 ab 06                 ADD     #6
0306 e184 3f 22                 CLR     ADDRH           SETUP FOR
0307 e186 b7 23                 STA     ADDRL           EXAMINE/CHANGE
0308 e188 1c 20                 BSET    6,STAT
0309 e18a cc e6 ea              JMP     MEMEX3          USING MEMORY ROUTINE
0310
0311                    ****************************************************
0312                    *                                                  *
0313                    *       CC reg. examine/change.                    *
0314                    *                                                  *
0315                    ****************************************************
0316
0317 e18d cd e6 42      CCODE   JSR     CLRTAB
0318 e190 a6 d1                 LDA     #$D1
0319 e192 c7 02 0c              STA     DTABL
0320 e195 a6 d7                 LDA     #$D7
0321 e197 c7 02 0d              STA     DTABL+1
0322 e19a a6 e6                 LDA     #$E6
0323 e19c c7 02 0e              STA     DTABL+2
0324 e19f a6 f1                 LDA     #$F1
0325 e1a1 c7 02 0f              STA     DTABL+3
0326 e1a4 cd e1 23              JSR     LOCSTK          FIND CONDITION
0327 e1a7 9f                    TXA                     CODES
0328 e1a8 ab 04                 ADD     #4
0329 e1aa 3f 22                 CLR     ADDRH           SETUP FOR
0330 e1ac b7 23                 STA     ADDRL           EXAMINE/CHANGE
0331 e1ae 1c 20                 BSET    6,STAT
0332 e1b0 cc e6 ea              JMP     MEMEX3          USING MEMORY ROUTINE
0333
0334
```

Freescale Semiconductor, Inc.

```
0335                            ***************************************************
0336                            *                                                 *
0337                            *       Build a beginning and ending              *
0338                            *       address range.                            *
0339                            *       TEMP,TEMP+1 - ADDRH,ADDRL.                *
0340                            *                                                 *
0341                            ***************************************************
0342
0343 e1b3 19 20                 BLDRNG  BCLR    4,STAT
0344 e1b5 cd e6 42                      JSR     CLRTAB          PRINT
0345 e1b8 a6 f4                         LDA     #$F4            `BA`
0346 e1ba c7 02 10                      STA     DTABL+4
0347 e1bd a6 77                         LDA     #$77
0348 e1bf c7 02 11                      STA     DTABL+5
0349 e1c2 cd e6 4c                      JSR     DISTAB
0350 e1c5 cd e7 a0                      JSR     BLDADR          GET SOURCE ADDR.
0351 e1c8 24 25                         BCC     BLDRN1          VALID?
0352 e1ca b6 22                         LDA     ADDRH           YES
0353 e1cc b7 27                         STA     TEMP            NO SAVE IT
0354 e1ce b6 23                         LDA     ADDRL
0355 e1d0 b7 28                         STA     TEMP+1
0356 e1d2 cd e7 46                      JSR     LOAD            FETCH OPCODE OF INSTR.
0357 e1d5 b7 26                         STA     WORK6           SAVE IT
0358 e1d7 cd e6 42                      JSR     CLRTAB
0359 e1da a6 f1                         LDA     #$F1            PRINT `EA`
0360 e1dc c7 02 10                      STA     DTABL+4
0361 e1df a6 77                         LDA     #$77
0362 e1e1 c7 02 11                      STA     DTABL+5
0363 e1e4 cd e6 4c                      JSR     DISTAB
0364 e1e7 cd e7 a0                      JSR     BLDADR          GET DESTINATION ADDR
0365 e1ea 24 03                         BCC     BLDRN1          VALID?
0366 e1ec b6 22                         LDA     ADDRH           YES
0367 e1ee 81                            RTS
0368 e1ef 18 20                 BLDRN1  BSET    4,STAT          INVALID
0369 e1f1 81                            RTS
0370
0371                            ***************************************************
0372                            *                                                 *
0373                            *       Display message.                          *
0374                            *                                                 *
0375                            ***************************************************
0376
0377 e1f2 bf 26                 DISP    STX     WORK6
0378 e1f4 3f 2b                         CLR     COUNT
0379 e1f6 be 26                 DISLP   LDX     WORK6
0380 e1f8 d6 e2 0d                      LDA     DLOAD,X
0381 e1fb be 2b                         LDX     COUNT
0382 e1fd d7 02 0c                      STA     DTABL,X
0383 e200 3c 26                         INC     WORK6
0384 e202 3c 2b                         INC     COUNT
0385 e204 b6 2b                         LDA     COUNT
0386 e206 a1 06                         CMP     #6
0387 e208 25 ec                         BLO     DISLP
0388 e20a cc e6 4c                      JMP     DISTAB
0389
0390 e20d 00 00 d0 d7 77 e6     DLOAD   FCB     0,0,$D0,$D7,$77,$E6
0391 e213 d6 f1 60 06 71 b6     VERF    FCB     $D6,$F1,$60,$06,$71,$B6
0392
0393
```

Freescale Semiconductor, Inc.

```
0394                        ***************************************************
0395                        *                                                 *
0396                        *       Calculate branch offset.                  *
0397                        *                                                 *
0398                        ***************************************************
0399
0400 e219 ad 98             OFFSET  BSR     BLDRNG
0401 e21b 08 20 3e                 BRSET   4,STAT,ORET
0402 e21e b6 23                    LDA     ADDRL           NO FIND APPARENT
0403
0404 e220 a0 02                    SUB     #2
0405 e222 b7 23                    STA     ADDRL
0406 e224 b6 22                    LDA     ADDRH
0407 e226 a2 00                    SBC     #0
0408 e228 b7 22                    STA     ADDRH
0409 e22a b6 23                    LDA     ADDRL
0410 e22c b0 28                    SUB     TEMP+1          OFFSET
0411
0412 e22e b7 23                    STA     ADDRL
0413 e230 b6 22                    LDA     ADDRH
0414 e232 b2 27                    SBC     TEMP
0415 e234 b7 22                    STA     ADDRH
0416 e236 b6 26                    LDA     WORK6           CHECK OPCODE
0417 e238 a1 1f                    CMP     #$1F            FOR BIT BRANCH
0418 e23a 23 50                    BLS     OFFST1
0419 e23c b6 22                    LDA     ADDRH
0420 e23e a1 ff                    CMP     #$FF            + OR - OFFSET?
0421 e240 27 0b                    BEQ     OFFST2
0422 e242 4d                       TSTA                    CHECK OFFSET
0423 e243 26 7a                    BNE     OVRERR          FOR +/- 0
0424
0425 e245 b6 23                    LDA     ADDRL
0426 e247 a1 7f                    CMP     #$7F
0427 e249 22 74                    BHI     OVRERR
0428 e24b 20 0a                    BRA     OK1
0429
0430 e24d b6 23             OFFST2  LDA     ADDRL
0431 e24f a1 ff                    CMP     #$FF
0432 e251 27 6c                    BEQ     OVRERR
0433
0434 e253 a1 80                    CMP     #$80
0435 e255 25 68                    BLO     OVRERR
0436
0437 e257 ad 06             OK1    BSR     USE             PRINT IT IF VALID
0438 e259 cc e0 bf                 JMP     CMDSCN
0439 e25c cc e0 b4          ORET   JMP     GETCMD
0440
0441 e25f cd e6 42          USE    JSR     CLRTAB
0442 e262 a6 d6                    LDA     #$D6            PRINT `USED`
0443 e264 c7 02 0c                 STA     DTABL
0444 e267 a6 b5                    LDA     #$B5
0445
0446
```

Freescale Semiconductor, Inc.

```
0447                    ***************************************************
0448                    *                                                 *
0449                    *       Branch offset (continued).                *
0450                    *                                                 *
0451                    ***************************************************
0452
0453 e269 c7 02 0d              STA     DTABL+1
0454 e26c a6 f1                 LDA     #$F1
0455 e26e c7 02 0e              STA     DTABL+2
0456 e271 a6 e6                 LDA     #$E6
0457 e273 c7 02 0f              STA     DTABL+3
0458 e276 b6 23                 LDA     ADDRL           PRINT OFFSET
0459 e278 cd e7 d6              JSR     PRTDAT
0460 e27b 97                    TAX
0461 e27c b6 28                 LDA     TEMP+1
0462 e27e ab 01                 ADD     #1
0463 e280 b7 23                 STA     ADDRL
0464 e282 b6 27                 LDA     TEMP
0465 e284 a9 00                 ADC     #0              PUT INTO
0466 e286 b7 22                 STA     ADDRH           INSTRUCTION
0467 e288 9f                    TXA
0468 e289 cc e7 57              JMP     STORE
0469
0470 e28c b6 23         OFFST1  LDA     ADDRL           ADJUST FOR
0471 e28e a0 01                 SUB     #1              BIT BRANCH
0472 e290 b7 23                 STA     ADDRL
0473 e292 b6 22                 LDA     ADDRH
0474 e294 a2 00                 SBC     #0
0475 e296 b7 22                 STA     ADDRH
0476 e298 a1 ff                 CMP     #$FF            NEG OFFSET?
0477 e29a 27 0b                 BEQ     OFFST3          YES
0478 e29c 4d                    TSTA                    CHECK FOR
0479 e29d 26 20                 BNE     OVRERR          +/- 0 AND -1
0480 e29f b6 23                 LDA     ADDRL
0481 e2a1 a1 7f                 CMP     #$7F
0482 e2a3 22 1a                 BHI     OVRERR
0483 e2a5 20 0e                 BRA     OK2
0484
0485 e2a7 b6 23         OFFST3  LDA     ADDRL
0486 e2a9 a1 ff                 CMP     #$FF
0487 e2ab 27 12                 BEQ     OVRERR
0488 e2ad a1 fe                 CMP     #$FE
0489 e2af 27 0e                 BEQ     OVRERR          BHI ?
0490 e2b1 a1 80                 CMP     #$80
0491 e2b3 25 0a                 BLO     OVRERR
0492
0493 e2b5 3c 28         OK2     INC     TEMP+1
0494 e2b7 26 02                 BNE     OFFITS
0495 e2b9 3c 27                 INC     TEMP
0496 e2bb ad a2         OFFITS  BSR     USE             PRINT IF VALID
0497 e2bd 20 0d                 BRA     CMDJMP
0498
0499 e2bf a6 d7         OVRERR  LDA     #$D7            PRINT OR
0500 e2c1 c7 02 10              STA     DTABL+4
0501 e2c4 a6 60                 LDA     #$60
0502 e2c6 c7 02 11              STA     DTABL+5
0503 e2c9 cd e8 02              JSR     PRTADR
0504 e2cc cc e0 bf      CMDJMP  JMP     CMDSCN
0505
0506
```

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

```
0507                    ***************************************************
0508                    *                                                 *
0509                    *       Display/set breakpoints.                  *
0510                    *                                                 *
0511                    ***************************************************
0512
0513 e2cf 3f 26         BPDIS   CLR     WORK6
0514 e2d1 3a 26                 DEC     WORK6
0515 e2d3 cd e6 1f              JSR     SCNBKP          FIND B.P. TABLE
0516 e2d6 bf 21                 STX     WORK2
0517 e2d8 4f            BPDIS1  CLRA
0518 e2d9 c7 02 10              STA     DTABL+4
0519 e2dc d6 02 16              LDA     BKPTBL,X        GET B.P.
0520 e2df 2a 14                 BPL     BPDIS2          VALID?
0521 e2e1 a6 f4                 LDA     #$F4            NO
0522 e2e3 c7 02 0c              STA     DTABL           PRINT `BOFF`
0523 e2e6 a6 d7                 LDA     #$D7
0524 e2e8 c7 02 0d              STA     DTABL+1
0525 e2eb a6 71                 LDA     #$71
0526 e2ed c7 02 0e              STA     DTABL+2
0527 e2f0 c7 02 0f              STA     DTABL+3
0528 e2f3 20 0a                 BRA     BPDIS4
0529 e2f5 b7 22         BPDIS2  STA     ADDRH           PRINT B.P.
0530 e2f7 d6 02 17              LDA     BKPTBL+1,X
0531 e2fa b7 23                 STA     ADDRL
0532 e2fc cd e8 02              JSR     PRTADR
0533 e2ff 3c 26         BPDIS4  INC     WORK6           PRINT B.P. #
0534 e301 be 26                 LDX     WORK6
0535 e303 d6 e7 6e              LDA     CTABL,X
0536 e306 c7 02 11              STA     DTABL+5
0537 e309 cd e6 4c              JSR     DISTAB
0538 e30c cd e7 a0              JSR     BLDADR          NEW B.P.
0539 e30f be 21                 LDX     WORK2
0540 e311 25 08                 BCS     BPDIS7          YES
0541 e313 a1 10                 CMP     #$10            NO,ESC?
0542 e315 27 19                 BEQ     BPRET           GET OUT
0543 e317 a1 11                 CMP     #$11            ENTER?
0544 e319 27 0a                 BEQ     BPDIS5          GET NEXT B.P.
0545 e31b b6 22         BPDIS7  LDA     ADDRH
0546 e31d d7 02 16              STA     BKPTBL,X        STORE NEW B.P.
0547 e320 b6 23                 LDA     ADDRL
0548 e322 d7 02 17              STA     BKPTBL+1,X
0549 e325 5c            BPDIS5  INCX                    GET NEXT B.P.
0550 e326 5c                    INCX
0551 e327 5c                    INCX
0552 e328 bf 21                 STX     WORK2
0553 e32a 3a 29                 DEC     PNCNT
0554 e32c 26 aa                 BNE     BPDIS1          DONE?
0555 e32e 20 9f                 BRA     BPDIS           YES START OVER
0556 e330 cc e0 b4      BPRET   JMP     GETCMD
0557
0558 e333 cd e6 42      ERROR   JSR     CLRTAB
0559 e336 a6 f1                 LDA     #$F1
0560 e338 c7 02 0d              STA     DTABL+1
0561 e33b a6 60                 LDA     #$60
0562 e33d c7 02 0e              STA     DTABL+2
0563 e340 c7 02 0f              STA     DTABL+3
0564 e343 cc e0 bc              JMP     DSCN
0565
0566
```

Freescale Semiconductor, Inc.

```
0567                    ***************************************************
0568                    *                                                 *
0569                    *       Breakpoint clear.                         *
0570                    *                                                 *
0571                    ***************************************************
0572
0573 e346 cd e6 42      BPCLR   JSR     CLRTAB          PRINT `BCLR`
0574 e349 a6 f4                 LDA     #$F4
0575 e34b c7 02 0c              STA     DTABL
0576 e34e a6 d1                 LDA     #$D1
0577 e350 c7 02 0d              STA     DTABL+1
0578 e353 a6 d0                 LDA     #$D0
0579 e355 c7 02 0e              STA     DTABL+2
0580 e358 a6 60                 LDA     #$60
0581 e35a c7 02 0f              STA     DTABL+3
0582 e35d cd e6 4c              JSR     DISTAB
0583 e360 cd e6 1f              JSR     SCNBKP          FIND B.P. TABLE
0584 e363 bf 21                 STX     WORK2
0585 e365 cd e7 90              JSR     GETNYB
0586 e368 25 14                 BCS     BPCLR1          ENTER?
0587 e36a a1 11                 CMP     #$11
0588 e36c 26 36                 BNE     BPCRET          NO
0589 e36e a6 ff                 LDA     #$FF            YES,CLEAR ALL
0590 e370 be 21                 LDX     WORK2
0591 e372 d7 02 16      BPCLR2  STA     BKPTBL,X
0592 e375 5c                    INCX
0593 e376 5c                    INCX
0594 e377 5c                    INCX
0595 e378 3a 29                 DEC     PNCNT
0596 e37a 26 f6                 BNE     BPCLR2
0597 e37c 20 26                 BRA     BPCRET
0598 e37e a1 03         BPCLR1  CMP     #3              VALID B.P. #?
0599 e380 24 b1                 BHS     ERROR           NO
0600 e382 97                    TAX                     YES
0601 e383 d6 e7 6e              LDA     CTABL,X         PRINT B.P. #
0602 e386 c7 02 11              STA     DTABL+5
0603 e389 4f                    CLRA                    FIND IT
0604 e38a a0 03                 SUB     #3
0605 e38c ab 03         BPCLR3  ADD     #3
0606 e38e 5a                    DECX
0607 e38f 2a fb                 BPL     BPCLR3
0608 e391 b7 26                 STA     WORK6
0609 e393 cd e6 4c              JSR     DISTAB          PRINT B.P.
0610 e396 cd e6 bb              JSR     CHRIN
0611 e399 a1 11                 CMP     #$11            CLEAR IT?
0612 e39b 26 07                 BNE     BPCRET          NO
0613 e39d be 26                 LDX     WORK6           YES
0614 e39f a6 ff                 LDA     #$FF
0615 e3a1 d7 02 16              STA     BKPTBL,X
0616 e3a4 cc e0 b4      BPCRET  JMP     GETCMD
0617
0618
```

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

```
0619                        ***************************************************
0620                        *                                                 *
0621                        *       Proceed.                                  *
0622                        *                                                 *
0623                        ***************************************************
0624
0625 e3a7 12 20             PROC    BSET    1,STAT          SET PROCEED FLAG
0626 e3a9 cd e1 23                  JSR     LOCSTK          FIND S.P.
0627 e3ac e6 07                     LDA     7,X
0628 e3ae b7 22                     STA     ADDRH
0629 e3b0 b7 27                     STA     TEMP
0630 e3b2 e6 08                     LDA     8,X
0631 e3b4 b7 23                     STA     ADDRL
0632 e3b6 b7 28                     STA     TEMP+1
0633 e3b8 cd e7 46                  JSR     LOAD            GET OPCODE
0634 e3bb 44                        LSRA
0635 e3bc 44                        LSRA
0636 e3bd 44                        LSRA
0637 e3be 44                        LSRA
0638 e3bf 97                        TAX                     X <- MSB
0639 e3c0 d6 e3 db                  LDA     MAT,X           NUMBER OF BYTES THIS INSTRUCTION
0640 e3c3 bb 23                     ADD     ADDRL           CALCULATE NEXT ADDRESS AND MOVE
0641 e3c5 b7 23                     STA     ADDRL           CURRENT BREAKPOINT TO IT
0642 e3c7 b6 22                     LDA     ADDRH
0643 e3c9 a9 00                     ADC     #0
0644 e3cb b7 22                     STA     ADDRH
0645
0646 e3cd cd e7 46                  JSR     LOAD
0647 e3d0 c7 02 15                  STA     PROP
0648 e3d3 a6 83                     LDA     #$83
0649 e3d5 cd e7 57                  JSR     STORE
0650
0651 e3d8 cc e5 d3                  JMP     NCONT
0652
0653 e3db 03 02 02 02 01 01 MAT     FCB     3,2,2,2,1,1,2,1,1,1,2,2,3,3,2,1
     02 01 01 01 02 02
     03 03 02 01
0654
0655
```

Freescale Semiconductor, Inc.

```
0656                        ****************************************************
0657                        *                                                  *
0658                        *   RS232 (9600) S-Record receiver (E0 at 4MHz).   *
0659                        *                                                  *
0660                        ****************************************************
0661
0662 e3eb 1a 20             VERIFY  BSET    5,STAT          SERIAL VERIFY
0663 e3ed ae 06                     LDX     #6
0664 e3ef 20 03                     BRA     L4
0665 e3f1 1b 20             TLOAD   BCLR    5,STAT          SERIAL LOAD
0666 e3f3 5f                        CLRX
0667 e3f4 a6 81             L4      LDA     #$81
0668 e3f6 b7 24                     STA     WORK3
0669 e3f8 cd e1 f2                  JSR     DISP
0670
0671 e3fb ad 5c             INPUT   BSR     INCHD           7 BIT ASCII INTO A
0672 e3fd a1 53                     CMP     #'S'            S ?
0673 e3ff 26 fa                     BNE     INPUT           NO, TRY AGAIN
0674 e401 ad 56                     BSR     INCHD           YES, GET NEXT CHARACTER
0675 e403 a1 39                     CMP     #'9'            9 ?
0676 e405 27 6c                     BEQ     NINE            YES, FINISH
0677 e407 a1 31                     CMP     #'1'            NO, 1 ?
0678 e409 26 f0                     BNE     INPUT           NO, TRY AGAIN
0679
0680 e40b 3f 2a             LNGTH   CLR     CHKSUM          YES, CLEAR CHECKSUM
0681 e40d 3f 2d                     CLR     TMP2            AND TEMP. STORE
0682 e40f cd e4 8a                  JSR     BYTEI           AND GET BYTE COUNT
0683 e412 b7 2e                     STA     BCNT            AND SAVE IT
0684
0685 e414 cd e4 8a          ADDR    JSR     BYTEI           ADDRESS HIGH
0686 e417 b7 22                     STA     ADDRH
0687 e419 cd e4 8a                  JSR     BYTEI           ADDRESS LOW
0688 e41c b7 23                     STA     ADDRL
0689
0690 e41e cd e4 8a          DLOP    JSR     BYTEI               75   GET A BYTE
0691 e421 27 1e                     BEQ     CHCK            3   78   LAST BYTE ?
0692 e423 0b 20 0b                  BRCLR   5,STAT,L5       5   83   NO, VERIFYING ?
0693 e426 b7 26                     STA     WORK6           4   87   YES
0694 e428 cd e5 4b                  JSR     RAMACC         30  117   READ RAM
0695 e42b b1 26                     CMP     WORK6           3  120   SAME ?
0696 e42d 26 57                     BNE     ERR7            3  123
0697 e42f 20 08                     BRA     L6              3  126
0698 e431 cd e5 4b          L5      JSR     RAMACC         56  139   NO, WRITE TO RAM
0699 e434 c1 02 13                  CMP     WORK4           3  142
0700 e437 26 3d                     BNE     ERR2            3  145   READ-BACK OK ?
0701 e439 3c 23             L6      INC     ADDRL           5  150   5 131   INCREMENT LS ADDRESS
0702 e43b 26 02                     BNE     NOOVR           3  153   3 134   OVERFLOW ?
0703 e43d 3c 22                     INC     ADDRH           5  158   5 139   YES, INC. HIGH BYTE
0704 e43f 20 dd             NOOVR   BRA     DLOP            3  161   3 142
0705
0706
0707                        ****************************************************
0708                        *                                                  *
0709                        *       Checksum byte & error routine.             *
0710                        *                                                  *
0711                        ****************************************************
0712
0713 e441 bb 2a             CHCK    ADD     CHKSUM
0714 e443 b7 2a                     STA     CHKSUM          DEBUG
0715 e445 a1 ff                     CMP     #$FF            IS CHECKSUM BYTE OK ?
0716 e447 27 b2                     BEQ     INPUT           YES, AND AGAIN
0717
0718 e449 ae 01                     LDX     #1
0719 e44b 9f                ERR     TXA
0720 e44c cd e7 d6                  JSR     PRTDAT
0721 e44f 4f                        CLRA
0722 e450 c7 02 10                  STA     DTABL+4
0723 e453 cd e8 02                  JSR     PRTADR
0724 e456 cc e0 bf                  JMP     CMDSCN
0725
```

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

```
0726                    ****************************************************
0727                    *                                                  *
0728                    *       Input routine,  MC68HC05E0 : 0.5 uS.       *
0729                    *       Cycles per bit at 9600 baud : 208.         *
0730                    *                                                  *
0731                    ****************************************************
0732
0733 e459 ad 60         INCHD   BSR     DEL191          191         GET OUT OF BIT 7
0734 e45b 05 01 fd      INCH    BRCLR   2,PORTB,*         5         IS LINE HIGH ?
0735 e45e 04 01 fd              BRSET   2,PORTB,*         5         YES, WAIT FOR START
0736 e461 ae 07                 LDX     #7                2     6   7 DATA BITS TO READ
0737 e463 bf 2b                 STX     COUNT             4    10   WAIT TILL MIDDLE OF 1st BIT
0738 e465 ad 58                 BSR     DEL110          110   120   +/-3
0739
0740 e467 ad 52         INBT    BSR     DEL191          191         +120-208=103
0741 e469 05 01 00              BRCLR   2,PORTB,ZER       5   196   CYC 2 (105) READ
0742 e46c 46            ZER     RORA                      3   199   SAVE BIT
0743 e46d 3a 2b                 DEC     COUNT             5   204
0744 e46f 26 f6                 BNE     INBT              3   207
0745
0746 e471 44                    LSRA                      3    16   MSB A ZERO
0747 e472 81                    RTS                       6    22
0748
0749 e473 cc e0 b4      NINE    JMP     GETCMD
0750
0751 e476 ae 02         ERR2    LDX     #2                  RAM READBACK
0752 e478 20 d1                 BRA     ERR
0753 e47a ae 03         ERR3    LDX     #3                  LESS THAN ASCII 0
0754 e47c 20 cd                 BRA     ERR
0755 e47e ae 04         ERR4    LDX     #4                  BETWEEN ASCII 9 & A
0756 e480 20 c9                 BRA     ERR
0757 e482 ae 05         ERR5    LDX     #5                  MORE THAN ASCII F
0758 e484 20 c5                 BRA     ERR
0759 e486 ae 07         ERR7    LDX     #7                  VERIFY ERROR
0760 e488 20 c1                 BRA     ERR
0761
0762
```

Freescale Semiconductor, Inc.

```
0763                    ****************************************************
0764                    *                                                  *
0765                    *       Byte input sub-routines.                   *
0766                    *                                                  *
0767                    ****************************************************
0768
0769 e48a ad cd         BYTEI   BSR     INCHD           22          MS NIBBLE
0770 e48c ad 17                 BSR     ASCII           35   57     WHAT WAS IT
0771 e48e 48                    LSLA                     3          OK
0772 e48f 48                    LSLA                     3          SHIFT
0773 e490 48                    LSLA                     3          IT
0774 e491 48                    LSLA                     3   69     UP
0775 e492 b7 2c                 STA     TMP1             4   73     AND SAVE IT
0776 e494 b6 2d                 LDA     TMP2             3   76     RESTORE BYTE
0777 e496 bb 2a                 ADD     CHKSUM           3   79     ACCUMULATE
0778 e498 b7 2a                 STA     CHKSUM           4   83     IN CHECKSUM BYTE
0779 e49a ad bd                 BSR     INCHD                22     LS NIBBLE
0780 e49c ad 07                 BSR     ASCII           35   57     WHAT WAS IT
0781 e49e bb 2c                 ADD     TMP1             3   60     ADD TO MS NIBBLE
0782 e4a0 b7 2d                 STA     TMP2             4   64     SAVE BYTE
0783 e4a2 3a 2e                 DEC     BCNT             5   69     DECREMENT BYTE COUNT
0784 e4a4 81                    RTS                      6   75
0785
0786 e4a5 a1 30         ASCII   CMP     #$30             2          BEFORE ZERO ?
0787 e4a7 25 d1                 BLO     ERR3             3    5     YES, NOT LEGAL
0788 e4a9 a1 39                 CMP     #$39             2    7     AFTER NINE
0789 e4ab 22 03                 BHI     MT9              3   10     YES TRY A-F
0790 e4ad a0 30                 SUB     #$30             3   13     0-9, CONVERT TO HEX
0791 e4af 81                    RTS                      6   19
0792
0793 e4b0 a1 41         MT9     CMP     #$41             2   12     BEFORE A ?
0794 e4b2 25 ca                 BLO     ERR4             3   15     YES, NOT LEGAL
0795 e4b4 a1 46                 CMP     #$46             2   17     AFTER F ?
0796 e4b6 22 ca                 BHI     ERR5             3   20     YES, NOT LEGAL
0797 e4b8 a0 37                 SUB     #$37             3   23     A-F, CONVERT TO HEX
0798 e4ba 81            NFND    RTS                      6   29
0799
0800 e4bb ae 1d         DEL191  LDX     #29              2
0801 e4bd 20 02                 BRA     DELAY            3    5
0802 e4bf ae 10         DEL110  LDX     #16              2
0803 e4c1 5a            DELAY   DECX                     3
0804 e4c2 26 fd                 BNE     DELAY            3    6 x X
0805 e4c4 81                    RTS                      6    12+6X (INC BSR)
0806
0807
```

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

```
0808                        ****************************************************
0809                        *                                                  *
0810                        *       RS232 (9600 @ 4MHz) S-Record output.       *
0811                        *                                                  *
0812                        ****************************************************
0813
0814 e4c5 14 06             PUNCH   BSET    2,PORTBD        BIT 2 OUTPUT
0815 e4c7 cd e1 b3                  JSR     BLDRNG          BUILD RANGE
0816 e4ca 08 20 a6                  BRSET   4,STAT,NINE     NEW ADDRESS ENTERED ?
0817 e4cd be 27                     LDX     TEMP            NO, SWAP ADDRESSES
0818 e4cf b7 27                     STA     TEMP
0819 e4d1 bf 22                     STX     ADDRH
0820 e4d3 b6 23                     LDA     ADDRL
0821 e4d5 be 28                     LDX     TEMP+1
0822 e4d7 bf 23                     STX     ADDRL
0823 e4d9 b7 28                     STA     TEMP+1
0824 e4db 1f 20                     BCLR    7,STAT          CLEAR END FLAG
0825
0826 e4dd b6 28             LOOP1   LDA     TEMP+1          END LSB
0827 e4df b0 23                     SUB     ADDRL           CURRENT LSB
0828 e4e1 b7 26                     STA     WORK6           DIFFERENCE LSB
0829 e4e3 b6 27                     LDA     TEMP            END MSB
0830 e4e5 b2 22                     SBC     ADDRH           CURRENT MSB
0831 e4e7 26 0d                     BNE     LOTS            MSB ZERO ?
0832 e4e9 b6 26                     LDA     WORK6           YES, LOOK AT LSB
0833 e4eb 4c                        INCA                    ADJUST
0834 e4ec 27 08                     BEQ     LOTS            WAS $FF ?
0835 e4ee a1 20                     CMP     #$20            MORE THAN 23 ?
0836 e4f0 22 04                     BHI     LOTS            IF SO USE 23
0837 e4f2 1e 20                     BSET    7,STAT          NO, LAST S1 RECORD
0838 e4f4 20 02                     BRA     LTE20           LESS THAN OR EQUAL TO 20
0839
0840 e4f6 a6 20             LOTS    LDA     #$20
0841 e4f8 ab 03             LTE20   ADD     #$03            ADD BYTE COUNT & ADDRESS
0842 e4fa b7 2e                     STA     BCNT            No. BYTES THIS S1 RECORD
0843 e4fc a6 53                     LDA     #'S'            S
0844 e4fe cd e5 63                  JSR     OUCH
0845 e501 a6 31                     LDA     #'1'            1
0846 e503 ad 5e                     BSR     OUCH
0847 e505 3f 2a                     CLR     CHKSUM
0848 e507 b6 2e                     LDA     BCNT            BYTE COUNT
0849 e509 cd e5 8f                  JSR     BYTEO
0850 e50c b6 22                     LDA     ADDRH           ADDRESS HIGH
0851 e50e cd e5 8f                  JSR     BYTEO
0852 e511 b6 23                     LDA     ADDRL
0853 e513 ad 7a                     BSR     BYTEO           ADDRESS LOW
0854
0855 e515 cd e7 46          LOOP2   JSR     LOAD            GET BYTE
0856 e518 3c 23                     INC     ADDRL           INCREMENT ADDRESS
0857 e51a 26 02                     BNE     NOVR            OVERFLOW ?
0858 e51c 3c 22                     INC     ADDRH           YES, INC. HIGH BYTE
0859 e51e ad 6f             NOVR    BSR     BYTEO           SEND BYTE
0860 e520 26 f3                     BNE     LOOP2           LAST BYTE ?
0861
0862
0863                        ****************************************************
0864                        *                                                  *
0865                        *       Checksum byte.                             *
0866                        *                                                  *
0867                        ****************************************************
0868
0869 e522 b6 2a                     LDA     CHKSUM          CHECKSUM
0870 e524 43                        COMA                    REQUIRED CHECKSUM BYTE
0871 e525 ad 68                     BSR     BYTEO           SEND IT
0872 e527 ad 34                     BSR     CRLF            CRLF
0873 e529 0f 20 b1                  BRCLR   7,STAT,LOOP1    FINISHED ?
0874
```

Freescale Semiconductor, Inc.

```
0875                        ****************************************************
0876                        *                                                  *
0877                        *       S9 record.                                 *
0878                        *                                                  *
0879                        ****************************************************
0880
0881 e52c a6 53                     LDA     #'S'            S
0882 e52e ad 33                     BSR     OUCH
0883 e530 a6 39                     LDA     #'9'            9
0884 e532 ad 2f                     BSR     OUCH
0885 e534 a6 03                     LDA     #$03            3 BYTES
0886 e536 ad 57                     BSR     BYTEO
0887 e538 a6 00                     LDA     #$00
0888 e53a ad 53                     BSR     BYTEO           DUMMY (0)
0889 e53c a6 00                     LDA     #$00
0890 e53e ad 4f                     BSR     BYTEO           ADDRESS
0891 e540 a6 fc                     LDA     #$FC
0892 e542 ad 4b                     BSR     BYTEO           CHECKSUM
0893 e544 ad 17                     BSR     CRLF
0894 e546 15 06                     BCLR    2,PORTBD        BIT 2 INPUT
0895 e548 cc e0 b4                  JMP     GETCMD
0896
0897                        ****************************************************
0898                        *                                                  *
0899                        *       RAM R/W for S-record load.                 *
0900                        *                                                  *
0901                        ****************************************************
0902
0903 e54b 0a 20 09          RAMACC  BRSET   5,STAT,L3       5       READING ?
0904 e54e ae c7                     LDX     #$C7            2   7   NO, WRITING
0905 e550 bf 21                     STX     WORK2           4  11
0906 e552 bd 21                     JSR     WORK2          16  27
0907 e554 c7 02 13                  STA     WORK4           4  31
0908 e557 ae c6             L3      LDX     #$C6            2  33   READING
0909 e559 bf 21                     STX     WORK2           4  37
0910 e55b bc 21                     JMP     WORK2          13  50   (56 WITH JSR)
0911
0912
0913 e55d a6 0d             CRLF    LDA     #$0D            CR
0914 e55f ad 02                     BSR     OUCH
0915 e561 a6 0a                     LDA     #$0A            LF
0916
0917
0918                        ****************************************************
0919                        *                                                  *
0920                        *       Output routine, 208 cycles per bit.        *
0921                        *                                                  *
0922                        ****************************************************
0923
0924 e563 14 01             OUCH    BSET    2,PORTB         5       MAKE SURE IT'S HIGH
0925 e565 ae 0a                     LDX     #10             2   7   10 BITS TO SEND
0926 e567 bf 2b                     STX     COUNT           4  11   START, 8 DATA, STOP
0927 e569 cd e4 c1                  JSR     DELAY          72  83
0928 e56c 98                        CLC                     2  85   START A ZERO
0929 e56d 20 08                     BRA     STAR            3  88
0930
0931 e56f ae 1c             OUTBT   LDX     #28             2
0932 e571 cd e4 c1                  JSR     DELAY         180 182
0933 e574 9d                        NOP                     2 184
0934 e575 99                DEL3    SEC                     2 186   FILL WITH ONES FOR STOP
0935 e576 46                        RORA                    3 189   GET A BIT
0936 e577 25 04             STAR    BCS     OU1             3 192   1 ?
0937 e579 15 01                     BCLR    2,PORTB         5 197   NO
0938 e57b 20 04                     BRA     OBD             3 200
0939 e57d 14 01             OU1     BSET    2,PORTB         5       YES
0940 e57f 20 00                     BRA     OBD             3
0941 e581 3a 2b             OBD     DEC     COUNT           5 205
0942 e583 26 ea                     BNE     OUTBT           3 208   DONE ?
0943 e585 81                        RTS
0944
0945 e586 ab 30             ASCIO   ADD     #$30            3       CONVERT TO ASCII
0946 e588 a1 39                     CMP     #$39            2   5   0-9 ?
0947 e58a 23 02                     BLS     NMT9            3   8
0948 e58c ab 07                     ADD     #$07            3   8   NO, A-F
0949 e58e 81                NMT9    RTS                     6  14
0950
```

Freescale Semiconductor, Inc.

```
0951                        ****************************************************
0952                        *                                                  *
0953                        *       Byte output sub-routine.                   *
0954                        *                                                  *
0955                        ****************************************************
0956
0957 e58f b7 2c            BYTEO   STA     TMP1
0958 e591 44                        LSRA                    SHIFT
0959 e592 44                        LSRA                    DOWN TO
0960 e593 44                        LSRA                    GET MSB
0961 e594 44                        LSRA
0962 e595 ad ef                     BSR     ASCIO           & CONVERT
0963 e597 ad ca                     BSR     OUCH
0964 e599 b6 2c                     LDA     TMP1            RESTORE BYTE
0965 e59b bb 2a                     ADD     CHKSUM          ACCUMULATE
0966 e59d b7 2a                     STA     CHKSUM          IN CHECKSUM BYTE
0967 e59f b6 2c                     LDA     TMP1
0968 e5a1 a4 0f                     AND     #$0F            LSB
0969 e5a3 ad e1                     BSR     ASCIO           CONVERT IT
0970 e5a5 ad bc                     BSR     OUCH
0971 e5a7 3a 2e                     DEC     BCNT            DECREMENT BYTE COUNT
0972 e5a9 81                        RTS
0973
0974
```

Freescale Semiconductor, Inc.

```
0975                    ***************************************************
0976                    *                                                 *
0977                    *       Go.                                       *
0978                    *                                                 *
0979                    ***************************************************
0980
0981 e5aa 13 20         NEWGO   BCLR    1,STAT
0982 e5ac cd e1 23              JSR     LOCSTK
0983 e5af e6 08                 LDA     8,X
0984 e5b1 b7 23                 STA     ADDRL
0985 e5b3 e6 07                 LDA     7,X
0986 e5b5 b7 22                 STA     ADDRH
0987 e5b7 cd e7 9a              JSR     GETADR
0988 e5ba 25 08                 BCS     GOON            ADDR VALID?
0989 e5bc a1 10                 CMP     #$10
0990 e5be 27 5c                 BEQ     GOBACK
0991 e5c0 a1 11                 CMP     #$11
0992 e5c2 26 55                 BNE     GOERR
0993 e5c4 cd e1 23      GOON    JSR     LOCSTK          YES PUT IT
0994 e5c7 b6 22                 LDA     ADDRH           IN STACK
0995 e5c9 e7 07                 STA     7,X
0996 e5cb b7 27                 STA     TEMP
0997 e5cd b6 23                 LDA     ADDRL
0998 e5cf e7 08                 STA     8,X
0999 e5d1 b7 28                 STA     TEMP+1
1000
1001 e5d3 ad 4a         NCONT   BSR     SCNBKP          FIND B.P. TABLE
1002 e5d5 d6 02 16      GOINSB  LDA     BKPTBL,X        GET ADDRESS MSB
1003 e5d8 2b 0d                 BMI     GONOB           VALID ? (<32k ONLY)
1004 e5da b7 22                 STA     ADDRH           YES
1005 e5dc d6 02 17              LDA     BKPTBL+1,X      LSB
1006 e5df b7 23                 STA     ADDRL
1007 e5e1 cd e7 46              JSR     LOAD            READ INSTRUCTION
1008 e5e4 d7 02 18              STA     BKPTBL+2,X
1009 e5e7 5c            GONOB   INCX                    GET NEXT B.P.
1010 e5e8 5c                    INCX
1011 e5e9 5c                    INCX
1012 e5ea 3a 29                 DEC     PNCNT
1013 e5ec 26 e7                 BNE     GOINSB          DONE?
1014
1015 e5ee ad 2f                 BSR     SCNBKP          FIND B.P. TABLE
1016 e5f0 d6 02 16      GOINSB2 LDA     BKPTBL,X        INSERT B.P.'S
1017 e5f3 2b 17                 BMI     GONOB2          VALID MSB ?
1018 e5f5 b7 22                 STA     ADDRH           YES
1019 e5f7 b1 27                 CMP     TEMP            SAME AS MSB OF NEXT ADDRESS ?
1020 e5f9 26 07                 BNE     DOSW            IF NOT THEN INSERT SWI
1021 e5fb d6 02 17              LDA     BKPTBL+1,X      SAME, GET LSB
1022 e5fe b1 28                 CMP     TEMP+1          SAME AS THAT OF NEXT ADDRESS ?
1023 e600 27 0a                 BEQ     GONOB2          IF SO THEN NO SWI
1024 e602 d6 02 17      DOSW    LDA     BKPTBL+1,X      SAME, GET LSB
1025 e605 b7 23                 STA     ADDRL
1026 e607 a6 83                 LDA     #$83            INSERT SWI
1027 e609 cd e7 57              JSR     STORE
1028 e60c 5c            GONOB2  INCX                    GET NEXT B.P.
1029 e60d 5c                    INCX
1030 e60e 5c                    INCX
1031 e60f 3a 29                 DEC     PNCNT
1032 e611 26 dd                 BNE     GOINSB2         DONE?
1033
1034 e613 c6 02 14              LDA     TMPTCR          RESTORE ORIGINAL
1035 e616 b7 0c                 STA     TCR             TCR VALUE
1036 e618 80                    RTI                     YES
1037
1038
1039 e619 cc e3 33      GOERR   JMP     ERROR
1040 e61c cc e0 b4      GOBACK  JMP     GETCMD
1041
1042 e61f a6 03         SCNBKP  LDA     #NBKPT
1043 e621 b7 29                 STA     PNCNT
1044 e623 5f                    CLRX
1045 e624 81                    RTS
1046
```

Freescale Semiconductor, Inc.

```
1047                        ****************************************************
1048                        *                                                  *
1049                        *       Display stack pointer.                     *
1050                        *                                                  *
1051                        ****************************************************
1052
1053 e625 a6 b5             STACK   LDA     #$B5            PRINT
1054 e627 c7 02 10                  STA     DTABL+4         `SP`
1055 e62a a6 73                     LDA     #$73
1056 e62c c7 02 11                  STA     DTABL+5
1057 e62f 4f                        CLRA
1058 e630 5f                        CLRX
1059 e631 cd e7 d8                  JSR     PRTBYT
1060 e634 cd e1 23                  JSR     LOCSTK          FIND USER
1061 e637 9f                        TXA                     STACK POINTER
1062 e638 ab 03                     ADD     #3
1063 e63a ae 02                     LDX     #2
1064 e63c cd e7 d8                  JSR     PRTBYT          PRINT IT
1065 e63f cc e0 bf                  JMP     CMDSCN
1066
1067                        ****************************************************
1068                        *                                                  *
1069                        *       Clear display table.                       *
1070                        *                                                  *
1071                        ****************************************************
1072
1073 e642 ae 05             CLRTAB  LDX     #5
1074 e644 4f                CLRLOC  CLRA
1075 e645 d7 02 0c                  STA     DTABL,X         CLEAR SIX
1076 e648 5a                        DECX                    LOCATIONS IN
1077 e649 2a f9                     BPL     CLRLOC          DISPLAY TABLE
1078 e64b 81                        RTS
1079
1080                        ****************************************************
1081                        *                                                  *
1082                        *       Display table contenst.                    *
1083                        *                                                  *
1084                        ****************************************************
1085
1086 e64c ae 05             DISTAB  LDX     #5
1087 e64e d6 02 0c          DISCHR  LDA     DTABL,X         LOAD DISPLAY
1088 e651 ad 08                     BSR     DISPLY          TABLE INTO
1089 e653 5a                        DECX                    145000
1090 e654 2a f8                     BPL     DISCHR
1091 e656 81                        RTS
1092
1093
1094                        ****************************************************
1095                        *                                                  *
1096                        *       Blank display.                             *
1097                        *                                                  *
1098                        ****************************************************
1099
1100 e657 ad e9             CLRDIS  BSR     CLRTAB          BLANK
1101 e659 20 f1                     BRA     DISTAB          DISPLAY
1102
1103                        ****************************************************
1104                        *                                                  *
1105                        *       Shift one character into display.          *
1106                        *                                                  *
1107                        ****************************************************
1108
1109 e65b cf 02 12          DISPLY  STX     WORK1           SAVE INDEX
1110 e65e 1d 00                     BCLR    6,PORTA         CLEAR DATA
1111 e660 ae 08                     LDX     #8
1112 e662 48                DIS1    LSLA                    SET UP
1113 e663 24 02                     BCC     DIS2            BIT OF
1114 e665 1c 00                     BSET    6,PORTA         ACCUMULATOR
1115 e667 1e 00             DIS2    BSET    7,PORTA         CLOCK
1116 e669 1f 00                     BCLR    7,PORTA         IT
1117 e66b 1d 00                     BCLR    6,PORTA         CLEAR DATA
1118 e66d 5a                        DECX                    COMPLETE?
1119 e66e 26 f2                     BNE     DIS1            NO
1120 e670 ce 02 12                  LDX     WORK1           RESTORE INDEX
1121 e673 81                        RTS
1122
```

Freescale Semiconductor, Inc.

```
1123                    ****************************************************
1124                    *                                                  *
1125                    *       Keypad scan, carry set if valid output.    *
1126                    *                                                  *
1127                    ****************************************************
1128
1129 e674 98              KEYSCN  CLC
1130 e675 4f                      CLRA
1131 e676 ae 06                   LDX     #6              SETUP
1132 e678 ab 10            KEY1    ADD     #$10            ROW
1133 e67a b7 00                    STA     PORTA
1134 e67c ad 06                    BSR     COLUMN          CHECK COLUMNS
1135 e67e 25 03                    BCS     KEY2            IF VALID GET OUT
1136 e680 5a                       DECX                    ELSE TRY
1137 e681 26 f5                    BNE     KEY1            NEXT ROW
1138 e683 81                KEY2    RTS
1139
1140
1141                    ****************************************************
1142                    *                                                  *
1143                    *       Check for key closure.                     *
1144                    *       A contains value, C set if valid output.   *
1145                    *                                                  *
1146                    ****************************************************
1147
1148 e684 b6 00             COLUMN  LDA     PORTA           READ KEYPAD
1149 e686 c7 02 12                 STA     WORK1           STORE IT
1150 e689 a5 0f                    BIT     #$0F            KEY CLOSED?
1151 e68b 27 1b                    BEQ     COLRET          NO GET OUT
1152 e68d ad 1a                    BSR     DBOUNC          ELSE DEBOUNCE
1153 e68f b6 00                    LDA     PORTA           RE-READ KEYPAD
1154 e691 c1 02 12                 CMP     WORK1           SAME KEY CLOSED?
1155 e694 26 12                    BNE     COLRET          NO GET OUT
1156 e696 99                       SEC                     SET FLAG FOR VALID
1157 e697 b6 00             COL1    LDA     PORTA           KEY
1158 e699 a5 0f                    BIT     #$0F            RELEASED?
1159 e69b 26 fa                    BNE     COL1            NO TRY AGAIN
1160 e69d ad 0a                    BSR     DBOUNC          YES DEBOUNCE
1161 e69f b6 00                    LDA     PORTA           STILL
1162 e6a1 a5 0f                    BIT     #$0F            RELEASED?
1163 e6a3 26 f2                    BNE     COL1            NO TRY AGAIN
1164 e6a5 c6 02 12                 LDA     WORK1           RETURN CHAR IN A-REG
1165 e6a8 81                COLRET  RTS                     YES GO HOME
1166
1167                    ****************************************************
1168                    *                                                  *
1169                    *       Pause for 3075 cycles.                     *
1170                    *                                                  *
1171                    ****************************************************
1172
1173 e6a9 a6 0a             DBOUNC  LDA     #10             40mS
1174 e6ab b7 25                    STA     WORK5
1175 e6ad a6 ff             DLP     LDA     #$FF            PAUSE
1176 e6af 21 fe             DLOOP   BRN     *               256X12
1177 e6b1 21 fe                     BRN     *               CYCLES
1178 e6b3 4a                        DECA                    OR AT
1179 e6b4 26 f9                     BNE     DLOOP           LEAST 3.7mS
1180 e6b6 3a 25                     DEC     WORK5
1181 e6b8 26 f3                     BNE     DLP
1182 e6ba 81                        RTS
1183
1184
1185                    ****************************************************
1186                    *                                                  *
1187                    *       Input one character, A contains value.     *
1188                    *                                                  *
1189                    ****************************************************
1190
1191 e6bb cd e6 74          CHRIN   JSR     KEYSCN          GET KEY
1192 e6be 24 fb                     BCC     CHRIN           IF NOT VALID RETRY
1193 e6c0 5f                        CLRX
1194 e6c1 d1 e6 cb          CHRIN1  CMP     STABL,X         CONVERT
1195 e6c4 27 03                     BEQ     CHRIN2          TO HEX
1196 e6c6 5c                        INCX
1197 e6c7 20 f8                     BRA     CHRIN1
1198 e6c9 9f                CHRIN2  TXA                     IF CANCEL
1199 e6ca 81                        RTS
1200
```

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

```
1201                    ***************************************************
1202                    *                                                 *
1203                    *       Conversion table for keypad.              *
1204                    *                                                 *
1205                    ***************************************************
1206
1207 e6cb 11          STABL   FCB     $11     0
1208 e6cc 21                  FCB     $21     1
1209 e6cd 22                  FCB     $22     2
1210 e6ce 24                  FCB     $24     3
1211 e6cf 31                  FCB     $31     4
1212 e6d0 32                  FCB     $32     5
1213 e6d1 34                  FCB     $34     6
1214 e6d2 41                  FCB     $41     7
1215 e6d3 42                  FCB     $42     8
1216 e6d4 44                  FCB     $44     9
1217 e6d5 48                  FCB     $48     A
1218 e6d6 38                  FCB     $38     B
1219 e6d7 28                  FCB     $28     C
1220 e6d8 18                  FCB     $18     D
1221 e6d9 14                  FCB     $14     E
1222 e6da 12                  FCB     $12     F
1223 e6db 61                  FCB     $61     10  CANCEL COMMAND
1224 e6dc 58                  FCB     $58     11  ENTER COMMAND
1225 e6dd 68                  FCB     $68     12  STACK POINTER
1226 e6de 64                  FCB     $64     13  MEMORY
1227 e6df 62                  FCB     $62     14  GO
1228 e6e0 54                  FCB     $54     15  VERIFY TAPE
1229 e6e1 52                  FCB     $52     16  LOAD TAPE
1230 e6e2 51                  FCB     $51     17  PUNCH TAPE
1231
1232
1233                    ***************************************************
1234                    *                                                 *
1235                    *       Memory location examine/change.           *
1236                    *                                                 *
1237                    ***************************************************
1238
1239 e6e3 cd e7 9a     MEMEX   JSR     GETADR          BUILD ADDRESS
1240 e6e6 a1 10                 CMP     #$10
1241 e6e8 27 57                BEQ     MEMEX4
1242 e6ea c7 02 12     MEMEX3  STA     WORK1
1243 e6ed cd e7 46             JSR     LOAD            LOAD DATA
1244 e6f0 cd e7 d6             JSR     PRTDAT          PRINT IT
1245 e6f3 cd e7 90             JSR     GETNYB          GET NEW NIBBLE
1246 e6f6 a1 10                CMP     #$10
1247 e6f8 27 47                BEQ     MEMEX4
1248 e6fa a1 11                CMP     #$11
1249 e6fc 27 1b                BEQ     ADRINC
1250 e6fe a1 13                CMP     #$13
1251 e700 27 2f                BEQ     ADRDEC
1252 e702 a1 0f                CMP     #$0F
1253 e704 22 08                BHI     CMDMDL          IF VALID
1254 e706 cd e7 d6     MEMEX1  JSR     PRTDAT          PRINT IT
1255 e709 cd e7 7e             JSR     GETBY2          SHIFT IN NEXT
1256 e70c 25 f8                BCS     MEMEX1          IF VALID TRY AGAIN
1257
1258 e70e a1 11        CMDMDL  CMP     #$11            ENTER?
1259 e710 26 15                BNE     MEMEX2          NO
1260 e712 b6 21                LDA     WORK2           RESTORE ACCA
1261 e714 cd e7 57             JSR     STORE           YES STORE IT
1262 e717 25 d1                BCS     MEMEX3          STORE VALID?
1263 e719 0c 20 25     ADRINC  BRSET   6,STAT,MEMEX4
1264 e71c 3c 23                INC     ADDRL           YES GOTO
1265 e71e 26 02                BNE     MEMEX5          NEXT
1266 e720 3c 22                INC     ADDRH
1267 e722 cd e8 02     MEMEX5  JSR     PRTADR          PRINT IT
1268 e725 20 c3                BRA     MEMEX3          REPEAT
1269 e727 a1 13        MEMEX2  CMP     #$13
1270 e729 26 16                BNE     MEMEX4          NO
1271 e72b b6 21                LDA     WORK2
1272 e72d ad 28                BSR     STORE
1273 e72f 25 b9                BCS     MEMEX3
1274 e731 0c 20 0d     ADRDEC  BRSET   6,STAT,MEMEX4
1275 e734 3d 23                TST     ADDRL           YES THEN
1276 e736 26 02                BNE     CMDMB2          GET PREVIOUS
1277 e738 3a 22                DEC     ADDRH           ADDRESS
1278 e73a 3a 23        CMDMB2  DEC     ADDRL
1279 e73c cd e8 02             JSR     PRTADR          PRINT IT
1280 e73f 20 a9                BRA     MEMEX3          REPEAT
1281 e741 1d 20        MEMEX4  BCLR    6,STAT          INVALID CHAR
1282 e743 cc e0 b4             JMP     GETCMD
1283
1284
```

Freescale Semiconductor, Inc.

```
1285                        ****************************************************
1286                        *                                                  *
1287                        *       Load byte at ADDRH,ADDRL into A.           *
1288                        *                                                  *
1289                        ****************************************************
1290
1291 e746 cf 02 12        LOAD    STX     WORK1           SETUP
1292 e749 ae c6                    LDX     #$C6            ROUTINE
1293 e74b bf 21           LDSTCM  STX     WORK2           TO DO
1294 e74d ae 81                    LDX     #$81            TWO BYTE
1295 e74f bf 24                    STX     WORK3           LOAD
1296 e751 bd 21                    JSR     WORK2
1297 e753 ce 02 12                 LDX     WORK1
1298 e756 81                       RTS
1299
1300                        ****************************************************
1301                        *                                                  *
1302                        *       Store byte in A at ADDRH,ADDRL.            *
1303                        *                                                  *
1304                        ****************************************************
1305
1306 e757 cf 02 12        STORE   STX     WORK1
1307 e75a ae c7                    LDX     #$C7            SETUP
1308 e75c ad ed                    BSR     LDSTCM          ROUTINE
1309 e75e c7 02 13                 STA     WORK4           TO DO
1310 e761 cd e7 46                 JSR     LOAD            TWO BYTE
1311 e764 c1 02 13                 CMP     WORK4           STORE
1312 e767 27 01                    BEQ     STRTS
1313 e769 99                       SEC
1314 e76a ce 02 12        STRTS   LDX     WORK1
1315 e76d 81                       RTS
1316
1317                        ****************************************************
1318                        *                                                  *
1319                        *       Hex. to mux. display conversion.           *
1320                        *                                                  *
1321                        ****************************************************
1322
1323 e76e d7              CTABL   FCB     $D7     0
1324 e76f 06                      FCB     $06     1
1325 e770 e3                      FCB     $E3     2
1326 e771 a7                      FCB     $A7     3
1327 e772 36                      FCB     $36     4
1328 e773 b5                      FCB     $B5     5
1329 e774 f5                      FCB     $F5     6
1330 e775 07                      FCB     $07     7
1331 e776 f7                      FCB     $F7     8
1332 e777 b7                      FCB     $B7     9
1333 e778 77                      FCB     $77     A
1334 e779 f4                      FCB     $F4     B
1335 e77a d1                      FCB     $D1     C
1336 e77b e6                      FCB     $E6     D
1337 e77c f1                      FCB     $F1     E
1338 e77d 71                      FCB     $71     F
1339
1340
1341                        ****************************************************
1342                        *                                                  *
1343                        *       Build a byte in accumulator.               *
1344                        *                                                  *
1345                        ****************************************************
1346
1347 e77e b7 21             GETBY2  STA     WORK2
1348 e780 ad 0e                    BSR     GETNYB
1349 e782 24 0b                    BCC     GETBRT
1350 e784 38 21                    ASL     WORK2
1351 e786 38 21                    ASL     WORK2
1352 e788 38 21                    ASL     WORK2
1353 e78a 38 21                    ASL     WORK2
1354 e78c ba 21                    ORA     WORK2
1355 e78e 99                       SEC
1356 e78f 81              GETBRT  RTS
1357
```

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

```
1358                        ***************************************************
1359                        *                                                 *
1360                        *       Get one character and check for valid     *
1361                        *       hex, A contains output, carry set if      *
1362                        *       valid.                                    *
1363                        *                                                 *
1364                        ***************************************************
1365
1366 e790 cd e6 bb        GETNYB  JSR     CHRIN           GET CHARACTER
1367 e793 98                       CLC
1368 e794 a1 0f                    CMP     #$0F            VALID HEX?
1369 e796 22 01                    BHI     GETRET          NO
1370 e798 99                       SEC                     YES
1371 e799 81              GETRET  RTS
1372
1373
1374                        ***************************************************
1375                        *                                                 *
1376                        *       Build address in ADDRH,ADDRL, carry set   *
1377                        *       if new address.                           *
1378                        *                                                 *
1379                        ***************************************************
1380
1381 e79a cd e6 42        GETADR  JSR     CLRTAB          BLANK DISPLAY
1382 e79d cd e8 02        BLDA2   JSR     PRTADR
1383 e7a0 ad ee            BLDADR  BSR     GETNYB          GET CHARACTER
1384 e7a2 25 0a                    BCS     GETAD1          VALID HEX?
1385 e7a4 a1 11                    CMP     #$11            ENTER ?
1386 e7a6 27 2d                    BEQ     GETRTS
1387 e7a8 a1 10                    CMP     #$10            NO, CANCEL ?
1388 e7aa 27 29                    BEQ     GETRTS          NO, TRY AGAIN
1389 e7ac 20 ec                    BRA     GETADR
1390 e7ae 3f 22            GETAD1  CLR     ADDRH           INIT HIGH ADDRESS
1391 e7b0 b7 23                    STA     ADDRL           PUT CHAR AWAY
1392 e7b2 cd e8 02                 JSR     PRTADR          PRINT NEW ADDRESS
1393 e7b5 ad d9            GETALP  BSR     GETNYB          GET ANOTHER CHAR
1394 e7b7 24 13                    BCC     GETARG          VALID?
1395 e7b9 48                       ASLA                    YES
1396 e7ba 48                       ASLA                    SHIFT IT IN
1397 e7bb 48                       ASLA
1398 e7bc 48                       ASLA
1399 e7bd ae 04                    LDX     #4
1400 e7bf 48              GETASF  ASLA
1401 e7c0 39 23                    ROL     ADDRL
1402 e7c2 39 22                    ROL     ADDRH
1403 e7c4 5a                       DECX
1404 e7c5 26 f8                    BNE     GETASF
1405 e7c7 cd e8 02                 JSR     PRTADR          PRINT NEW ADDR
1406 e7ca 20 e9                    BRA     GETALP          GET ANOTHER CHAR
1407 e7cc a1 10            GETARG  CMP     #$10            NOT VALID HEX, CANCEL ?
1408 e7ce 27 05                    BEQ     GETRTS
1409 e7d0 a1 11                    CMP     #$11            ENTER ?
1410 e7d2 26 e1                    BNE     GETALP          NO TRY AGAIN
1411 e7d4 99                       SEC                     YES SET FLAG
1412 e7d5 81              GETRTS  RTS
1413
1414
1415                        ***************************************************
1416                        *                                                 *
1417                        *       Print one byte into pair of display       *
1418                        *       digits, A contains byte, X points to      *
1419                        *       first diget.                              *
1420                        *                                                 *
1421                        ***************************************************
1422
1423 e7d6 ae 04            PRTDAT  LDX     #4              PRINT IN LAST TWO DIGITS
1424 e7d8 cf 02 12         PRTBYT  STX     WORK1
1425 e7db c7 02 13                 STA     WORK4
1426 e7de 44                       LSRA
1427 e7df 44                       LSRA
1428 e7e0 44                       LSRA
1429 e7e1 44                       LSRA
1430 e7e2 97                       TAX
1431 e7e3 d6 e7 6e                 LDA     CTABL,X
1432 e7e6 ce 02 12                 LDX     WORK1
1433 e7e9 d7 02 0c                 STA     DTABL,X
1434 e7ec c6 02 13                 LDA     WORK4
1435 e7ef a4 0f                    AND     #$0F
1436 e7f1 97                       TAX
1437 e7f2 d6 e7 6e                 LDA     CTABL,X
1438
1439 e7f5 ce 02 12                 LDX     WORK1
1440 e7f8 d7 02 0d                 STA     DTABL+1,X
1441 e7fb cd e6 4c                 JSR     DISTAB
1442 e7fe c6 02 13                 LDA     WORK4
1443 e801 81                       RTS
```

Freescale Semiconductor, Inc.

```
1444
1445                    ****************************************************
1446                    *                                                  *
1447                    *       Print address ADDRH,ADDRL.                 *
1448                    *                                                  *
1449                    ****************************************************
1450
1451 e802 b7 25       PRTADR  STA     WORK5
1452 e804 bf 24               STX     WORK3
1453 e806 b6 22               LDA     ADDRH
1454 e808 5f                  CLRX
1455 e809 ad cd               BSR     PRTBYT
1456 e80b b6 23               LDA     ADDRL
1457 e80d ae 02               LDX     #2
1458 e80f ad c7               BSR     PRTBYT
1459 e811 b6 25               LDA     WORK5
1460 e813 be 24               LDX     WORK3
1461 e815 81                  RTS
1462
1463
1464                    ****************************************************
1465                    *                                                  *
1466                    *       Transfer code from EPROM to RAM, default   *
1467                    *       destination address:-  $0400-$1FFF, from   *
1468                    *       EPROM at $4400 - $5FFF.                    *
1469                    *       (TEMP,TEMP+1 -> ADDRH,ADDRL).              *
1470                    *                                                  *
1471                    ****************************************************
1472
1473 e816 a6 81       XFER    LDA     #$81            RTS
1474 e818 b7 24               STA     WORK3
1475 e81a a6 f4               LDA     #$F4            `BA`
1476 e81c c7 02 10            STA     DTABL+4
1477 e81f a6 77               LDA     #$77
1478 e821 c7 02 11            STA     DTABL+5
1479 e824 cd e6 4c            JSR     DISTAB
1480 e827 a6 04               LDA     #$04            DEFAULT TO $0400
1481 e829 b7 22               STA     ADDRH
1482 e82b a6 00               LDA     #$00
1483 e82d b7 23               STA     ADDRL
1484 e82f cd e7 9d            JSR     BLDA2           GET SOURCE ADDR.
1485 e832 25 04               BCS     SKPC1           VALID ?
1486 e834 a1 11               CMP     #$11            NO, ENTER ?
1487 e836 26 61               BNE     XAB             IF NOT THEN ABORT
1488 e838 b6 22       SKPC1   LDA     ADDRH           YES
1489 e83a b7 27               STA     TEMP            NO SAVE IT
1490 e83c b6 23               LDA     ADDRL
1491 e83e b7 28               STA     TEMP+1
1492 e840 a6 f1               LDA     #$F1            PRINT `EA`
1493 e842 c7 02 10            STA     DTABL+4
1494 e845 a6 77               LDA     #$77
1495 e847 c7 02 11            STA     DTABL+5
1496 e84a cd e6 4c            JSR     DISTAB
1497 e84d a6 1f               LDA     #$1F            DEFAULT TO $1FFF
1498 e84f b7 22               STA     ADDRH
1499 e851 a6 ff               LDA     #$FF
1500 e853 b7 23               STA     ADDRL
1501 e855 cd e7 9d            JSR     BLDA2           GET DESTINATION ADDR
1502 e858 25 04               BCS     SKPC2           VALID ?
1503 e85a a1 11               CMP     #$11            NO, ENTER ?
1504 e85c 26 3b               BNE     XAB             IF NOT THEN ABORT
1505 e85e b6 22       SKPC2   LDA     ADDRH
1506 e860 be 27               LDX     TEMP            SWAP ADDRESSES
1507 e862 b7 27               STA     TEMP
1508 e864 bf 22               STX     ADDRH
1509 e866 b6 23               LDA     ADDRL
1510 e868 be 28               LDX     TEMP+1
1511 e86a bf 23               STX     ADDRL
1512 e86c b7 28               STA     TEMP+1
1513
1514
```

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

```
1515                    ****************************************************
1516                    *                                                  *
1517                    *       Transfer code from EPROM to RAM.           *
1518                    *                                                  *
1519                    ****************************************************
1520
1521 e86e b6 22       XLOOP   LDA     ADDRH
1522 e870 ab 40               ADD     #$40
1523 e872 b7 22               STA     ADDRH
1524 e874 cd e7 46            JSR     LOAD
1525 e877 97                  TAX
1526 e878 b6 22               LDA     ADDRH
1527 e87a a0 40               SUB     #$40
1528 e87c b7 22               STA     ADDRH
1529 e87e 9f                  TXA
1530 e87f cd e7 57            JSR     STORE
1531 e882 24 03               BCC     RBOK
1532 e884 cc e4 76            JMP     ERR2
1533 e887 3c 23          RBOK INC     ADDRL
1534 e889 26 02               BNE     XSKP
1535 e88b 3c 22               INC     ADDRH
1536 e88d b6 22          XSKP LDA     ADDRH
1537 e88f b1 27               CMP     TEMP
1538 e891 26 db               BNE     XLOOP
1539 e893 b6 23               LDA     ADDRL
1540 e895 b1 28               CMP     TEMP+1
1541 e897 26 d5               BNE     XLOOP
1542
1543 e899 cc e0 b4       XAB  JMP     GETCMD
1544
1545                    ****************************************************
1546                    *                                                  *
1547                    *       Vectors etc.                               *
1548                    *                                                  *
1549                    ****************************************************
1550
1551                    SIRQV
1552                    TIRQAV
1553                    TIRQBV
1554 e89c 80            IRQV    RTI
1555
1556 e89d cd e6 57      PWRDWN  JSR     CLRDIS
1557 e8a0 8e            STP     STOP
1558 e8a1 20 fd               BRA     STP
1559
1560 e8a3 cd e6 57      WDWN    JSR     CLRDIS
1561 e8a6 8f            WIT     WAIT
1562 e8a7 20 fd               BRA     WIT
1563
1564 fff4                      ORG     $FFF4
1565
1566 fff4 e0 00               FDB     RESET           SERIAL
1567 fff6 02 06               FDB     TIRQB           TIMER B
1568 fff8 02 03               FDB     TIRQA           TIMER A
1569 fffa 02 00               FDB     IRQ             EXTERNAL INTERRUPT
1570 fffc e0 5e               FDB     SWI             SWI
1571 fffe e0 00               FDB     RESET           RESET
1572
1573                          END
```

Freescale Semiconductor, Inc.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.
Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters which may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals” must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale Semiconductor, Inc.