System Monitor Board II

User's Manual

Manual Revision 0
Sept. 26, 1979

Computer Design Labs
342 Columbus Ave.
Trenton, N. J. 08629

Copyright (c) 1979 by Computer Design Labs
Table of Contents

I. Introduction .............................................. 1

II. Features .................................................. 2

III. Overlay Drawing ......................................... 3

IV. User Guide ................................................ 4
   A. Switch Settings ....................................... 4
      1. I/O Default Switch (SW1) ......................... 4
      2. Jump Address Switch (SW2) ...................... 4
      3. Cassette Options & Protect Switches (SW3) .... 5
   B. VDB Set Up ............................................ 6
   C. Baud Rate Selection .................................. 6
   D. Memory Addressing ................................... 7
   E. ROM & ROM Options ................................... 10
   F. RAM .................................................... 11
   G. Extended Memory Addressing Options .............. 11
   H. I/O Port Addressing ................................... 11
   I. I/O Byte (Second Parallel Port) J2 ............... 12
   J. TTY 20ma/EIA Option ................................. 12
   K. Vectored Interrupt Options ......................... 12
# Table of Contents

L. Wait State Options & MWRITE Generation......13

M. SMEMR Clamp...........................................13

N. Audio Cassette Connector.........................13

O. Ribbon Cable Connector (J1)......................14
   1. Connection of Ribbon Cable......................15-18

V. Functional Description of SMB2..................19

   A. Introduction...........................................19
   B. Bus Interface...........................................19
   C. ROM/EPROM Memories.................................20
   D. Static RAM..............................................21
   E. Parallel Port...........................................21
   F. Serial Ports...........................................21
   G. Cassette Interface.....................................22
   H. Jump On Reset and Sense Switches................23

VI. Software Documentation & Listings..............25

   A. 1K Extension ROM........................................25
      1. TDL VDB Driver Software...........................25
      2. Buffered Cassette Routine........................25-28
      3. 2708 Programmer Software
      4. Extended User-defined Commands
      5. Current Assignment Display

   B. Cassette Interface

   C. Extension Routines for SMB2
VII. Parts List

VIII. Schematics
I. Introduction

The System Monitor Board, SMB2, is designed to be used with the Technical Design Labs Z-80 CPU board (the ZPU) or any other compatible Z-80 CPU which operates in the S-100 bus.

The SMB2 is a ROM/RAM/I-O cassette board. It has space for 3 ROM's and 2K of RAM. There are 2 serial ports with EIA Interface (1 optional 20ma), and CDL's audio cassette interface. The SMB2 also contains the circuitry necessary to cause the processor to Jump to any 256 byte boundary on RESET, to generate the MWRITE signal on bus pin 68 in those mainframes that lack this circuitry, and a wait state circuit that will allow TDL's ZPU to access it's memory at 4Mhz. There are jumper options which will allow the user to change (within reason) the memory and I/O addresses that the SMB2 will decode. One ROM socket is set up for a 2K masked ROM (Zapple) and can be changed to work with a 2716. The other 2 ROM sockets are set up for 2708's and may be changed to work with 2716's. To put more than 5K of ROM on the board is possible but not recommended.
II. Features

Supports 2-2708 EPROMs, 1 included
Regulators on all supply voltages
More reliable cassette
1200/2400 baud cassette operation
Jump to any page in memory on Reset
Expanded Zapple Monitor, with VDB driver
IV. User Guide

In the following sections please refer to the overlay drawing on the preceding page, which shows the areas being discussed. Before plugging the board into your system, the following considerations must be observed to achieve any degree of success.

A. Switch Settings

1. I/O Default Switch (SW1)

This switch is accessible to the computer by inputting from either ports 7A or 7B hex. The remainder of this text will discuss the switch as used by the Zapple Monitor program.

The I/O switch on the SMB2 is shipped set to the TTY. This setting configuration is all toggle switches OFF (bottom part of switch depressed). The 4 logical devices (i.e. LPR-lineprinter, PUN-punch, RDR-reader, CON-console) are assigned as the TTY.

If you want to configure your system for Video <CRT>, the switch configuration would be 01000001. Toggle switch positions, going from left to right, 2 and 8 should be toggled ON (top part of switch depressed). (Note this switch is installed right side up with S1 on the LEFT and S8 on the RIGHT).

If you want to configure your system for the VDB, see Section "B. on VDB Set Up.

Please note that assignment of the I/O device to one of the logical devices as is done by the 8-bit dip switch can also be accomplished under software control by using the ASSIGN command of the monitor's operating system. See the Zapple Monitor manual.

2. Jump Address Switch (SW2)

Upon a RESET of the system a jump to any 256 byte boundary in memory will occur depending on the switch settings of SW2. The SMB2 is set to jump to F000, which is the location of the Zapple Monitor. The setting configuration for SW2 is 11110000, which means the four left-most toggle switches are toggled ON (top part of switch pushed in).

You may set this switch to any 256 byte location. For example, if you wanted to have your system RESET to jump to the location at which BASIC resides, say at 0C0000H, SW2 should be toggled in as follows: 11000000.
If you wish to disable the jump, just set the switch for 0000:00000000 (All bottom buttons pushed in).

3. Cassette Options & Protect Switches (SW3)

The left-most position (S1) labeled "LV" stands for level and should be toggled ON. S1 controls the level of the signal sent to the tape recorder. If you are using a microphone (MIC) input, then the signal should be low, and the bottom button of S1 should be pushed in (OFF). If you are using the auxiliary (AUX) input, then the signal should be high, and the top button of S1 should be pushed in (ON).

The next position (S2) labeled "BR" for baud rate should be OFF. The next two positions, S3 and S4, labeled "IP" and "OP" stand for Input Polarity and Output Polarity respectively. S3 labeled "IP" causes the data coming into the interface to be inverted. S4 causes an inversion of the data sent to the cassette, and should be adjusted so as to allow the writing of a standard cassette. Once you have adjusted S3 to allow the playback of TDL pre-recorded tapes, then adjust S4 so that the tapes that you record can be played back without changing S3. If you change tape recorders, you may have to go through the procedure again. If a particular cassette recorder has an odd number of inverting audio stages in its input circuit, the data recorded on the tape will be inverted. If the number of inverting audio stages in its output is odd, the data will appear inverted from what was recorded on the tape. Thus two conditions exist. One in which the data is either inverted or not on RECORD. And the other in which the data is either inverted or not on PLAYBACK. When recording a tape and playing it back on the same unit, it is a simple matter to determine the position of the invert switch. It will be the same for all such recordings. However, when playing back tapes recorded on another unit, the switch should be tried in the opposite position if the recording unit did not have the same inversion as yours. Trial and error will find the correct setting quickly.

Switches labeled "R1" and "R2" protect RAM memory. "R1" protects RAM1 and "R2" protects RAM2. Depressing the top half of the toggle switches positions S5 and S6, unprotects the RAM memory while depressing the bottom half causes it to be protected. Both RAM segments should be unprotected so it can be written into as well as read from.

The remaining toggle switches labeled "24" and "12" are for the baud rate the cassette is running under. Depress the top half of the switch designating the correct baud rate. The default setting is 1200 baud.
TDL tapes are recorded at 1200 baud. Remember that turning on both switches at the same time will upset the baud rate generator and may prevent the proper operation of the 2 serial ports.

B. VDB Set Up

The VDB driver may be used as both the "LIST" device (AL-U) or as the console output. When used as console output, the parallel port on the VDB becomes the console input. This assumes that the user has a parallel keyboard hooked up to the VDB (as per the VDB manual), and a CRT hooked to the Video output of the VDB. You are able to initialize the system using the aforementioned configuration by setting the "LIST" device to "USER" on the I/O initialization switches (SW1), and setting the console switches to "BATCH" mode. Providing that both conditions have been set-up, the system will sign-on to the VDB. Note, you must have the R2 Ram protect switch in the unprotect mode (i.e. top button pushed in.)

Normal Batch operation has been preserved, providing that the "LIST" device is not set to "USER" on the I/O initialization switches. Once Zapple has signed on, you may re-define the List device to something other than USER with no ill effects. It is the setting of the I/O initialization switches (port 7AH), not the I/O Byte (port 76H) that determines the VDB as the console. Note that the I/O Byte is modified after sign-on to reflect that the console is the USER, and not the Batch mode. (See table below)

For a better understanding, see the software listings of the 1K Extension ROM.

<table>
<thead>
<tr>
<th>I/O Init. Switch</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>11xxxx10</td>
<td>VDB=CONSOLE</td>
</tr>
<tr>
<td>00xxxx10</td>
<td>Batch Mode (READER=CONSOLE)</td>
</tr>
<tr>
<td>01xxxx10</td>
<td></td>
</tr>
<tr>
<td>10xxxx10</td>
<td></td>
</tr>
<tr>
<td>xxxxxxx00</td>
<td></td>
</tr>
<tr>
<td>xxxxxxx01</td>
<td>TTY=CONSOLE</td>
</tr>
<tr>
<td>xxxxxxx11</td>
<td>USER DEFINED CONSOLE</td>
</tr>
</tbody>
</table>

C. Baud Rate Selection

Depending on the particular device you are using, a baud rate must be set. For a TTY (Teletype), (designated CLKT on the silk screen), running at 110 baud, connect a jumper between CLKT terminal and 110
baud rate terminal. See the overlay drawing labeled BAUD RATE SELECTION to determine location of jumpers. For Video <CRT> use at 9600 baud connect a jumper between CLKV terminal and 9600 baud rate pin designation. The TTY at 110 and Video at 9600 baud are the default settings. The available baud rates are labeled from left to right 9600, 4800, 2400, 1200, 600, 300 and 110. See the diagram below.

BAUD RATE JUMPERS

9 4 2 1 6 3 C  C  I
6 8 4 2 0 0 L  L  I
0 0 0 0 0 0 K  K  0
0 0 0 0

STANDARD CONFIGURATION

D. Memory Addressing

The SMB2 standard memory addressing is configured for a 1k RAM, 2K Zapple ROM and 1K Extension EPROM system. In the standard factory configuration the jumpers are connected so ROM 1 is Zapple at 0F000H - 0F7FFH, ROM 2 is 1K Extension at 0F800H - 0FBFFH and RAM 2 at 0FC00H - 0FFFFH.

1. I/O Set Up

The board is supplied at the TDL standard I/O block 7OH. This may be altered if desired to allow more than one SMB2 in the system at the same time. Remember, however, that the 2K Masked ROM program was designed to use ONLY the 70H I/O block and execute at F000H.
The 4K Memory Address Decoder, 74LS138's, designated U22 and U23 together form a 4-bit decoder which decodes 1 of the 16 4-K blocks in the machine by looking at address bits A12 through A15. This circuit also has a possible input from the Extended Memory Circuitry. The 74LS139 chip, a dual 2-bit decoder, designated by U24, performs a 2K select and a 1K select. The 1K Decoder looks at Bits 10 and 11 and selects 1K block addresses which yield 4 possible combinations. The 2K Decoder looks at Bit 11 and selects 2K block addresses which yield 2 possible combinations. Please study the diagram on the next page.
I/O SET UP

FACTORY JUMPERED for I/O BLOCK at 70H

Cut to change
Then jump from SEL- to desired pin.

U15 74LS138

Cut and jumper (as shown by dotted line) to make I/O Block 8→F
The following pages show the available jumpers, the Standard Jumper Configuration and a Memory Map describing the system's boundaries, plus two examples of other jumper configurations.
DECODING

4K MEMORY ADDRESS BLOCK DECODER

1K DECODER

C00 - FFF
800 - FFF
400 - 7FF
000 - 3FF

3 → IK RAM
2 → IK EXT EPR

2K DECODER

800 - FFF
000 - 7FF

1 → ZAPPLE ROM

0 →
Available Jumper Configurations

Longest jumper length possible to 7
Longest jumper length possible to A

RAM1 is 1 K RAM U7 + U9
RAM2 is 1 K RAM U6 + U8
ROM1 is 2 K ZAPPEL Rom U1
ROM2 is 1 K Extension EPROM U2
ROM3 is 1 K Extension EPROM U3
Standard Jumper Configurations

RAM 2 is RAM at 0FC00H – 0FFFFH.
ROM 1 is ZAPPLE at 0F000H – 0F7FFH.
ROM 2 is 1K Extension at 0F800H – 0FBFFH.
### System Monitor Board II Memory Map

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFFFFh</td>
<td>TOP OF MEMORY</td>
</tr>
<tr>
<td>OFD80H</td>
<td>USER WORKSPACE</td>
</tr>
<tr>
<td>OFC74H</td>
<td>BUFFERED CASSETTE BUFFERS</td>
</tr>
<tr>
<td>OFC24H</td>
<td>VDB SCROLL BUFFER</td>
</tr>
<tr>
<td>OFC00H</td>
<td>EXTENDED USER JUMP VECTORS</td>
</tr>
<tr>
<td>OF800H</td>
<td>ONE-K EXTENSION ROUTINES</td>
</tr>
<tr>
<td>OF000H</td>
<td>ZAPPLE TWO-K MASKED ROM</td>
</tr>
<tr>
<td>OE000H</td>
<td>TOP OF SYSTEM MEMORY IN A MAXIMUM CONFIGURATION</td>
</tr>
<tr>
<td></td>
<td>(60-K CONTIGUOUS)</td>
</tr>
<tr>
<td>0100H</td>
<td>TRANSIENT PROGRAM AREA</td>
</tr>
<tr>
<td>0038H</td>
<td>RST 7 (USED ONLY DURING DEBUGGING)</td>
</tr>
<tr>
<td>0000H</td>
<td>MEMORY BOTTOM</td>
</tr>
</tbody>
</table>
Example A Jumper Configurations

RAM 1 is RAM at 0F800H-0FBFFH
RAM 2 is RAM at 0FC00H-FFFFFH
ROM 1 is ZAPPLE at 0F000H-0F7FFH
Example B Jumper Configurations

RAM 1 is RAM at 0400H-07FFH
RAM 2 is RAM at 0800H-0BFFH
ROM 1 is ZAPPLE at 0F000H-0F7FFH
ROM 2 is ROM at 0000H-03FFH
E. ROM and ROM Options

The SMB2 is factory shipped with 2K Zapple ROM and 1K Extension EPROM. The chips used are a MOSTEK MK34038N or equivalent Mask ROM in U1 socket position on the board (ROM1) and an INTEL 2708 UV erasable 1024x8 ROM in U2 socket position on the board (ROM2). The MK34038 is a 16K bit, 2K byte ROM which contains the Zapple Monitor. ROM sockets 2 and 3 are configured to accept either a 2708, (1Kx8) or a 2716 (2Kx8) ROM.

The ROM option jumpers allow a user to configure any or all 3 sockets for 2716 EPROMs. A 2716 is a 16K bit, 2K byte EPROM. The ROM options involved when inserting 2716 chips deal with jumpering pins 18, 19, 20 and 21 to agree with their specific requirements. Please note that there are marked differences when inserting an INTEL 2716 and a TI2716. The voltages are different and if not jumpered correctly may cause the destruction of the chip.

The figures diagrammed for you on the following pages show the standard jumper configuration, and the jumpers needed to convert your board to use with two INTEL 2716 chips and TI2716 chips respectively. Note that traces have to be cut from the standard configuration to allow 2716 usage.

Jumpers as shown are standard configuration
Jumpers as shown are configured for use with two INTEL 2716 EPROMS.

(require 2K address select CS-)

Jumpers as shown are configured for use with two TI 2716 EPROMS

(require 2K address select CS-)
F. RAM

RAM is provided in socket positions U6 and U8 by 2 INTEL 2114s. RAM occupies 1K of memory from OFCOOH to OFFFH. The first section includes Extended User Jump Vectors, the next section includes the VDB Scroll buffer, followed by the Buffered Cassette Buffers. The area from OFD80H to OFFFH is reserved for the user as workspace. (See Standard Memory Map diagram in section D.) Memory Addressing shows RAM and its reference to the system. (See software listings contained elsewhere in this manual).

G. Extended Memory Addressing Options

The SMB2, when enabled decodes extended memory addresses, lines A16-A19. If these lines are not used in the system, the BANK select jumper is connected to ground (GND), as it is per the factory. The Extended Memory Addressing Option will be further developed with the introduction of forthcoming TDL products.

H. I/O Port Addressing

U18 on the SMB2 is a Motorola or equivalent 6820 PIA (Peripheral Interface Adapter) that contains two parallel I/O ports. One of these is used by the operating system to specify the I/O device currently being used.

Port Assignments

The devices are assigned to ports on the System Monitor Card in the following manner:

<table>
<thead>
<tr>
<th>DEVICE</th>
<th>STATUS/CONTROL</th>
<th>DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Teletype (serial)</td>
<td>70</td>
<td>71</td>
</tr>
<tr>
<td>Video/CRT(serial)</td>
<td>72</td>
<td>73</td>
</tr>
<tr>
<td>Cassette</td>
<td>74</td>
<td>75</td>
</tr>
<tr>
<td>Parallel Port (User)</td>
<td>79</td>
<td>78</td>
</tr>
<tr>
<td>I/O Byte</td>
<td>77*</td>
<td>76*</td>
</tr>
<tr>
<td>Sense Switch</td>
<td>7A</td>
<td>7B</td>
</tr>
<tr>
<td>Unused</td>
<td>7C,7D,7E,7F</td>
<td></td>
</tr>
</tbody>
</table>

* Note: Ports 76 and 77 are used internally for operation of the monitor.

The Motorola Data Sheets are included in APPENDIX
the MC6820 chip. Also see I/O Set Up in section D.

I. I/O Byte (Second Parallel Port)

The I/O Byte (second parallel port) is used by Zapple for the storage of the I/O Byte and thus is not available to the user as a second parallel port. If the user were to remove the Zapple ROM chip and create his own monitor, the second parallel port would, of course, be available to him for whatever he desires. As an input port, a keyboard, high speed paper tape reader, or other device can be used. As an output port, a high speed paper tape punch, line printer, etc. can be used. In order to use it, however, it must be set up with the proper software.

J. TTY 20ma/EIA Option

Your TTY runs at either 20ma Current Loop or at RS232. A jumper is provided on the SMB2 and must be set before your board will operate. Observe the drawing below:

K. Vectored Interrupt Options

The SMB2 generates interrupts from the PIA and 3 ACIAs. These interrupts are buffered and may be connected to any of the 8 interrupt lines. However, note that the masked ROM Zapple does not use interrupts for its operation.
L. Wait State Options and MWRITE Generation

The SMB2 can cause single wait states to any cycle if the PRDY jumper is installed. The Wait State is necessary if the system is running at 4MHz.

The board generates the MWRITE signal equal to PWR.SOUT- and drives it with a buffered driver. This connection should be included if MWRITE is not generated elsewhere in the system. (Usually a front-panel generated signal.)

M. SMEMR Clamp

The SMEMR clamp must be included for the Jump On Reset. It is factory jumpered as standard.

Note: If used with the original TDL ZPU board, pin 15 of U33 (74LS175) must be bent out of its socket pin. This is required because PSYNC is high before a RESET is released.

N. Audio Cassette Connector

The SMB2 provides a cassette connector, plus a plug adaptor. See the diagram below:

R stands for Record (write).
0. Ribbon Cable Connector

If you have already purchased the Interface One, the ribbon cable is properly terminated on the back of the mainframe. (see the diagram of connector.) If not you must attach the twenty six (26) wires in the ribbon cable to their proper destinations. The pin assignments for the SMB2 connector are as follows:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TTY CTS (Clear to Send)</td>
</tr>
<tr>
<td>2</td>
<td>Video CTS (Clear to Send)</td>
</tr>
<tr>
<td>3</td>
<td>CAS RTS (Request to Send)</td>
</tr>
<tr>
<td>4</td>
<td>B Output Pulse</td>
</tr>
<tr>
<td>5</td>
<td>TTY RTS (Request to Send)</td>
</tr>
<tr>
<td>6</td>
<td>Not Used</td>
</tr>
<tr>
<td>7</td>
<td>Video RTS (Request to Send)</td>
</tr>
<tr>
<td>8</td>
<td>TTY 20 MA Out</td>
</tr>
<tr>
<td>9</td>
<td>Video RS232 Out</td>
</tr>
<tr>
<td>10</td>
<td>TTY 20 MA In</td>
</tr>
<tr>
<td>11</td>
<td>TTY RS232 Out</td>
</tr>
<tr>
<td>12</td>
<td>Not Used</td>
</tr>
<tr>
<td>13</td>
<td>TTY RS232 In</td>
</tr>
<tr>
<td>14</td>
<td>-12 Volts</td>
</tr>
<tr>
<td>15</td>
<td>Video RS232 In</td>
</tr>
<tr>
<td>16</td>
<td>GND (Ground)</td>
</tr>
<tr>
<td>17</td>
<td>PIA Handshake CB2</td>
</tr>
<tr>
<td>18</td>
<td>PIA Handshake CB1</td>
</tr>
<tr>
<td>19</td>
<td>PIA Data PB7</td>
</tr>
<tr>
<td>20</td>
<td>PIA Data PB6</td>
</tr>
<tr>
<td>21</td>
<td>PIA Data PB5</td>
</tr>
<tr>
<td>22</td>
<td>PIA Data PB4</td>
</tr>
<tr>
<td>23</td>
<td>PIA Data PB3</td>
</tr>
<tr>
<td>24</td>
<td>PIA Data PB2</td>
</tr>
<tr>
<td>25</td>
<td>PIA Data PB1</td>
</tr>
<tr>
<td>26</td>
<td>PIA Data PB0</td>
</tr>
</tbody>
</table>
1. Connection of Ribbon Cable

TTY RS232:

Using the standard EIA 25 pin connector which should be wired to the proper places on the TTY, the following connections should be made:

TTY (DB25)  |  SMB (J1)*
---|---
1 Frame Ground | 16 Ground
2 Transmit | 13 TTY Input RS232
3 Receive | 11 TTY Output RS232
4 Request to Send | 10 TTY 20ma IN
5 Clear to Send | 14 minus (-) 12 volts
6 Data Set Ready | 8 TTY 20ma OUT
7 Signal Ground | J1 on the SMB is the blue Ansley ribbon cable connector at the top middle of the board.
8 Rec'd Line Detect | 20 Data Terminal Ready
20 Data Terminal Ready

TTY RS232 Checklist:

Make the following connections on the 25 pin RS232 connector (DB25) going to your TTY.

(✓) Connect a jumper between pin 4, Request to Send, and pin 5, Clear to Send.
( ) Connect a jumper between pin 6, Data Set Ready, pin 8, Received Line Signal Detect, and pin 20, Data Terminal Ready.
( ) Connect a jumper between pin 1, Frame Ground, and pin 7, Signal Ground.

Make the following connections on the SMB's J1-- the blue Ansley ribbon cable connector at the top right of the board.

( ) Connect 20ma/EIA option jumper to EIA.

Make the following connections between the DB25 connector and J1 on the SMB.

(✓) Connect pin 1 of the DB25 to pin 16 of J1.
(✓) Connect pin 2, Transmit, of the DB25 to pin 13, TTY RS232 IN, of J1.
(✓) Connect pin 3, Receive, of the DB25 to pin 11, TTY RS232 OUT, of J1.
TTY 20ma Current Loop:

Connection to the TTY's current loop can be made at either the Terminal Strip (TS) or J2. Note: In this reference, J1 refers to the jack on the SMB and J2 refers to the jack on the TTY. Follow this procedure:

( ) Connect pin 8 of J1 to either pin 7 of TS or pin 8 of J2.
( ) Connect pin 10 of J1 to either pin 4 of TS or pin 6 of J2.
( ) Connect pin 14 of J1 to either pins 3 and 6 of TS or pins 5 and 7 of J2.
( ) Connect 20ma/EIA option jumper to 20ma.

CONVERTING A TELETYPE TERMINAL FROM HALF- TO FULL-DUPEX OPERATION

To convert a Teletype terminal connected for half-duplex operation to full-duplex operation, the following modifications should be made.

1. Locate the black terminal strip in the back of the data terminal. See Fig. 5a.
2. Move the brown/yellow and white/blue wires from pins 3 and 4 to pin 5.

CONVERTING A TELETYPE TERMINAL FROM 60-ma to 20-ma OPERATION

To convert a Teletype terminal connected for 60-ma operation to 20-ma operation, the following modifications should be made.

1. Locate the black terminal strip in the back of the data terminal. See Fig. 5
2. Move the violet wire from pin 8 to pin 9.
3. Move the blue wire connected to the current source resistor (a flat green resistor having four tabs located to the right of the keyboard) from the 750-ohm tab to the 1450-ohm tab.
Video:

You may connect either a TTY or a CRT terminal to the "Video" port, however, a CRT terminal is usually connected. The connections are very similar to the TTY connections. Refer to the TTY section for any abbreviations used here without explanation. The explanations are not repeated.

(1) Make the same 3 jumper connections on the DB 25 as described for the TTY (i.e., 4 to 5, 6 to 8 and 20, and 1 to 7).

(2) Connect pin 1 of DB 25 to pin 16 of J1.

(3) Connect pin 2, Transmit, of DB 25 to pin 15, VIDEO RS232 IN, of J1.

(4) Connect pin 3, Receive, of DB 25 to pin 9, VIDEO RS232 OUT, of J1.

Note: The Video port does NOT have a 20ma option.
Cassette:

At the top of the SMB, left-hand side of board, there are three connection points. The far left of these is "R" for Record (write). The center one is "GND" for ground. And the right one is "P" for Play (read). The following is the connection procedure:

( ) Connect the shields of two shielded audio cables together and then connect them to the center ("GND") terminal.
( ) Connect the center wire of the one going to the recorder's input to the left terminal ("R").
( ) Connect the center wire of the cable coming from the recorder's output to the right terminal ("P").

Parallel Port:

See the schematic for the pin designations on J1 which pertain to the parallel port. They are labelled "PB" for the port's data bits and "CB" for the port's control bits.
V. Functional Description of SMB2

A. Introduction

The System Monitor Board, SMB2, is a flexible multi-function board which provides a variety of features commonly required in microcomputer systems. The SMB2 includes:

1. Zapple monitor in 2K masked ROM
2. 2 EPROM sockets (one 2708 1K EPROM is included)
3. 2K static RAM (only 1K included)
4. Parallel 8-bit port (PIA)
5. 2 serial RS-232 ports (ACIA)
   (20ma current loop interface is available on one port.)
6. Cassette Interface
7. Jump-On-Reset to any 256 byte boundary
8. 8-bit switch register

B. Bus Interface

An internal tristate 8-bit data bus (DO-D7) interconnects the functional units on the SMB2. This bus is connected to the S-100 data-out (DOO-D07) and data-in (DIO-D17) buses through 74LS244 octal tristate buffers (U13 and U14). The internal data bus drives the DI bus when the processor inputs data from the SMB2. The enable signal (ENOUT- at U34 pin 3) is generated when the board is selected (SMB) and the processor does a memory read operation (SMEMR) or an input operation (SINP). The Jump-On-Reset circuit also enables the board outputs. The internal data bus is driven by the DO bus whenever the processor transfers data to the SMB2. This enable signal (ENIN- at U30 pin 11) is generated for all memory write operations (MWRITE) and output operations (SOUT), regardless of the board select. However, this will have no effect on the SMB2 unless one of its functions is selected.

The SMB2 contains both memory and input/output interfaces. These are addressed independently. The extended memory address lines (A16-A19) are decoded by the 74LS138 at U32. A19 is decoded by a jumper connection to either an inverting (A19=0) or a non-inverting (A19=1) enable input. A16-A18 select one of eight outputs which can be jumpered to the active-low bank enable signal (BANK-). BANK- must be jumpered to ground if extended addressing is not used.

Memory is addressed only when no input/output (IOP=SINP+SOUT) operation is in progress. A12-A15 are decoded by 74LS138's at U22 and U23. Each output corresponds to one 4K address block. These blocks are
subdivided into 1K and 2K address blocks by the 74LS139, U24. These signals are attached to the chip enable inputs of the appropriate memory units. (See also the detailed description in the ROM and RAM sections.) Address lines A0-A11 are buffered by 74LS244's at U19 and U20. The buffered address lines address bytes within each semiconductor memory.

The SMB2 responds to a block of 12 input/output addresses. During input/output operations, the 74LS138 at U15 decodes A4-A7 to select a block of 16 addresses. The decoder is disabled when A2-A3=1 to limit each block to 12 addresses. The SEL- jumper selects which block enables the board. However, the software in the Zapple ROM requires that addresses 70-7B be used. Input/output address assignments are: (Addresses in hexadecimal)

70,71 Serial port (TTY- "teletype")
72,73 Serial prot (VID- "video terminal")
74,75 Cassette serial port (CAS-)
76,77,78,79 Parallel interface port (PIA-)
7A or 7B Switch register input (SWIT-)

When either memory or an input/output port is selected, the board select signal (SMB at U21 pin 8) is true.

C. ROM/EPROM Memories

The SMB2 layout includes sockets for three ROMs (Read-Only-Memory) or EPROMs (Erasable-Programmable-Read-Only-Memory). A wide variety of standard memory chips have similar pinouts in a 24 pin package. Pins 18, 19, 20 and 21 vary, as shown below:

<table>
<thead>
<tr>
<th>Pin 18</th>
<th>Pin 19</th>
<th>Pin 20</th>
<th>Pin 21</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTEL 2708 1Kx8 EPROM</td>
<td>gnd(PRG)</td>
<td>+12v</td>
<td>CS-</td>
</tr>
<tr>
<td>TI 2716 2Kx8 EPROM</td>
<td>CS-(PRG)</td>
<td>+12v</td>
<td>A10</td>
</tr>
<tr>
<td>INTEL 2716 2Kx8 EPROM</td>
<td>CS-(PRG)</td>
<td>A10</td>
<td>OE-</td>
</tr>
<tr>
<td>TI 2532 4Kx8 EPROM</td>
<td>A11</td>
<td>A10</td>
<td>CS-</td>
</tr>
<tr>
<td>MOS 34000 2Kx8 ROM</td>
<td>CS-(gnd)</td>
<td>A10</td>
<td>CS-</td>
</tr>
<tr>
<td>MOS 32000 4Kx8 ROM</td>
<td>A11</td>
<td>A10</td>
<td>CS-</td>
</tr>
<tr>
<td>MOS 36000 8Kx8 ROM</td>
<td>A11</td>
<td>A10</td>
<td>CS-</td>
</tr>
</tbody>
</table>

CAUTION: INTEL 2716 and TI 2716 are very different parts, electrically. CS- indicates the chip select pin which enables the memory when at logic 0. When this pin is high (logic 1), the memory chip is disabled and its
outputs are in the high-impedance tri-state condition. When several CS- lines are indicated, all must be low to enable the memory. (Some ROM chips may have high true enables.) Additional CS- lines should be grounded or tied to an address line to aid in decoding.

A10, A11, and A12 indicate high order address lines. +12v, -5v, and +5v indicate power supply connections. PRGM and Vpp indicate pins on EPROMs which are used when the device is programmed. The SMB2 cannot program EPROMs.

Standard configuration includes the Zapple Monitor program in a 2Kx8 mask-programmed ROM and a 2708 EPROM programmed with the device driver for the VDB (Video Display Board). The power supply and address line connections required for these devices are part of the printed circuit etching. These traces must be cut if other devices are used. Plated-thru holes are provided to allow neat reconnection for the desired devices.

D. Static RAM

Sockets for four 2114 static RAMs (Random-Access-Memories, read-write) provide two 1Kx8 memory blocks. Select RAM1- enables memories at U7 and U9; select RAM2- enables memories at U6 and U8. (Standard configuration includes memories at U6 and U8 only). The two blocks are driven by separate 1K enables from the address decoders. The contents of the memory may be protected from accidental alteration by opening the write-protect switches on the WR- write pulse line. Memory is lost anytime power is OFF; these switches must be closed in order to store data into the RAM.

E. Parallel Port

A Motorola 6821 Parallel Interface Adapter circuit (PIA) provides two 8-bit parallel input/output ports. The Zapple monitor uses port A to store the I/O byte, precluding its use for input/output. If Zapple is not used, a connector may be installed in the J2 connector holes. Port B is connected to J1.

F. Serial Ports

Two serial ports are implemented using Motorola 6850 Asynchronous Communication Interface Adapter circuits (ACIA). These ports are labeled teletype (TTY- at U10) and video (VID- at U11). The serial communication lines and modem control lines are buffered using RS232 standard line drivers (SN75188 or MC4188)
and line receivers (SN75189 or MC4189). The buffered lines are on connector J1. The teletype port includes circuitry to connect to a 20mA current loop device (jumper selects which input is used; current loop and RS232 signal are assigned separate pins on J1).

A Motorola 14411 Baud Rate Generator (U17) provides clock timing signals for each ACIA. The clocks are 16 times the frequency of the desired serial baud rate. Jumper wires select the desired baud rate for each serial port separately.

G. Cassette Interface

The third serial port adapts an external low-cost audio cassette deck for recording digital data and programs. The cassette ACIA (U12) is switch selected to operate at either 1200 or 2400 baud. The higher baud rate does require higher fidelity in the signal played back by the cassette deck. This interface only processes the serial signal; it does not control the motion of the tape. Motion control may be manual or controlled by another port. Data is recorded in standard 8-level asynchronous code, including a start and two stop bits per character. Transmission is not required to be continuous. However, whenever the recorder is started, a resynchronization sequence must precede data transfer. TDL software accomplishes this by preceding most punch operations with leader (NULLS) followed by 8 rubouts (OFFH), followed by the data. In addition, upon loading tapes, the software waits until the appearance of 4 or more rubouts, and then begins loading when reaching the first non-rubout that follows. This prevents the 1-2 characters of garbage, that result from initial cassette start-up, from being treated as valid data. (See the "L" and "U" commands in the Zapple Manual).

The cassette transmit clock (TCC), is switch selected to either the 75 baud (1200 hz) or the 150 baud (2400 hz) output of the baud rate generator. The two baud rate select switches should not both be closed at the same time. Note that the generator outputs are set 16 times the baud rate. The cassette ACIA runs at 1x the baud rate. The ACIA transmit data output (TDC) is a NRZ signal clocked by the falling edge of TCC. TDC and TCC are exclusive-or'ed to generate a double-frequency self-clocked signal for the recorder. There is always a clock transition in the center of a bit cell (rising edge of TCC). Data is the level of the signal following this transition (i.e. while TCC is high). This recording scheme is polarity sensitive. A transmit polarity select switch (SW3-4) allows the signal to generate the polarity required by the cassette. The flux polarity written on the tape should match that of TDL's distributed software. The microphone inputs of
cassette recorders are intended for small amplitude signals. Thus, the output is attenuated to 0.4 volts (SW3-1 closed) or 0.04 volts (SW3-1 open).

The playback input is configured to connect to the auxiliary speaker output of the cassette deck. The input resistors provide a low impedance 32 ohm input resistance compatible with the cassette output. This output is usually heavily distorted. The LM339 analog comparator restores this signal to a clean square wave. Two 1k ohm resistors provide a 2.5 volt reference. The comparator negative input is the a.c. component of the cassette signal referenced to 2.5 volts. The positive input is the same reference with 50 millivolts of hysteresis (positive feedback). The hysteresis allows the circuit to time between the sharp leading edges of the input pulses. The trailing portion of the pulse lacks definition and is sensitive to noise. Note that both inputs are referenced to the same voltage; the offset error of the circuit is only that of the comparator itself.

The receiver clock is generated from the input square wave. The second comparator is used to delay the square wave by 3 microseconds. The exclusive-or of these signals is a 3 usec. pulse following each transition. This pulse triggers a non-retriggerable one-shot whose period is 3/4 of the bit period. Clock transitions trigger the one-shot; data transitions do not trigger the one-shot because the output is still active. The rising edge of the complementary output strobes the data into the ACIA. SW3-3 sets the polarity of the input data.

1200 baud corresponds to a bit period of 850 usec. which requires that the one-shot period be 630 usec. At 2400 baud, the times are 425 usec. and 315 usec, respectively. The one-shot timing is halved by closing SW3-2 which parallels a second equal resistor with timing resistor.

The output of the cassette's read pre-amplifier is usually less distorted than the speaker output. The 22 ohm (R7) input resistor should be replaced with a much larger value if it is driven by the pre-amplifier. Very low amplitude input signals may require less hysteresis, i.e. larger value resistor in the feedback (R11).

D. Jump-On-Reset and Switch Register

Dip-switch SW1 is eight switches which can be read as a byte from port x'nA'. Dip-switch SW2 sets the high-order address byte for Jump-On-Reset. The low-order byte is zero. These switches are gated onto the data bus with two quad 2-line to 1-line tri-state multiplexors (74LS258 at U4 and U5).
requires that the processor execute the sequence as shown in the diagram on the following page.

The 74LS258s have inverted outputs. Thus when ENSW=1 and the SELECT=0 inputs are all high, the multiplexors force the data bus low. The jump instruction code conveniently contains four "1" bits (0,1,6,7) and four "0" bits (2,3,4,5). Thus x'C3' is put on the data lines by enabling only the multiplexor (U5) which drives bits 2,3,4,5 low. Pull up resistors cause bits 0,1,6,7 to be high. The x'00' is caused by enabling both multiplexors. The SW2 switch settings are read by bringing the common side of the switches low.

The 74LS175 (U33) is connected as a shift register. It is reset by the bus RESET- signal. This activates the Jump-On-Reset circuit. The shift register is clocked by the leading edge of PSYNC. This precedes the PDBIN signal that the processor uses to input the data. Thus an extra flip-flop is set before JMPA is set. (However, the original ZPU from TDL has PSYNC high when RESET- is released. The extra flip-flop must be removed from the circuit. This is easily done by bending pin 15 of the 74LS175 at U33 out of the socket.)
SMB2 Jump-on-Reset Timing Diagram

- M1 Instruction Fetch
- Low order address
- High order address
- next

<table>
<thead>
<tr>
<th>JMPA</th>
<th>JMPB</th>
<th>JMPC</th>
<th>EN0167-</th>
<th>EN2345-</th>
<th>SELECT (JMPC)</th>
<th>ENSW (JMPB)</th>
<th>DO-D7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>x'00' (A0-A7)</td>
<td>SW2 (A8-A15)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Jump instruction code
- Low-order address byte
- High-order address byte
- tristate
- Circuit disabled
- SW1 Read Switch Register
SMB2 Cassette Record Circuit

Cassette Play-Back Circuit

Timing Components

2400 BAUD 315\mu\text{s} cycle = 425\mu\text{s}
1200 BAUD 630\mu\text{s} cycle = 850\mu\text{s}
SMB 2  Cassette Timing Diagram

TCC
TDC

A
B

R (same as B, but only 40 mV or 400 mV peak)  (Polarity selectable: shown with SW3-4 open.)

P (amplitude, polarity and waveform depend on recorder used)

SW3-4 open.

2.55V
2.5V
2.45V

hysteresis

245V

D

E

F (exaggerated)

RC - delay - 3μsec

2.5V

G

H

3μsec pulse

3125μsec at 2400 baud
630μsec at 1200 baud
685 μs.

RCC

RDC

( Polarity selectable, shown with SW3-3 open.)

425 μsec at 2400 baud
850 μsec at 1200 baud
VI. Software Documentation & Listings

A. 1K Extension ROM

Included in the Standard SMB2 is one 2708 ROM chip, factory programmed with the following:

1. TDL VDB Driver Software
2. Buffered Cassette Routines
3. 2708 Programmer Software
4. Extended User-defined Commands
5. Current Assignment Display

1. VDB Driver

The information for the VDB driver is included in Section IV. User Guide under B. VDB Set Up.

2. Buffered Cassette Routine

The Buffered Cassette Routine provides utility commands to open and close the input/output buffers and to control the cassette motors.

Please note! When using the Buffered Cassette Routine you must be careful not to do the right thing at the wrong time. I.E. you must open, close and rewind cassettes at various times during the running of your programs, and it is very easy to get mixed up. Cassette player 0 is the playback unit and cassette recorder 1 is the record unit.

The Buffered Cassette Routine is provided by Technical Design Labs for use with the ROM Zapple and the System Monitor Board. This routine provides the user with the ability to simulate a controlled paper tape reader and punch with two cassette tape recorders of adequate quality.

You must implement two motor control relays which are driven from the REQUEST TO SEND leads from two of the ACIA's on the System Monitor board.
This circuit was used at TDL.

The RTS leads come out on the 26 wire ribbon cable. The RTS leads from the TTY and cassette ACIA's are used for motor control, and the RTS lead from the video (crt) ACIA is not used.

RTS TTY = Cassette 1 = Rec

RTS RCSS = Cassette 0 = Play.
Study this example of text editing and assembly:

1) Sign on Zapple Monitor (turns on both motors).
2) Place a cassette containing the cassette routines on cassette 0 (playback). Type AR=C (cr) which assigns the reader to cassette.
3) Place a cassette containing the Text Editor on cassette 0 (playback). Type R,100 (cr) and turn on cassette 0 which loads the Text Editor into memory at 0100 Hex. When the prompt character ">" is printed, stop and rewind the Editor tape.
4) Type K.C,O,F to stop the cassette motors. Place the cassette containing the text material to be edited on cassette 0 (playback). Place a blank cassette on cassette 1 (record). Press the play button on cassette 0 and the record button on cassette 1. Note the motors should be off at this time.
5) Type AR=U (cr) and AP=U (cr) to assign the reader and punch to the Buffered Cassette Routine.
6) Type K.C,O,O to open the output file (reset the pointer). Type K.C,O,I to open the input file (starts the cassette 0 and preloads the buffer, then turns the motor off).
7) Type G100 (cr) to execute the Text Editor (follow the Text Editor manual for procedures). Every time you type A (escape) (escape) the Text Editor will read 50 lines of text. The cassette routine will start up the cassette 0 motor every time the input buffer gets empty. When you type W (escape) (escape) or E (escape) (escape) the Text Editor writes out text to the punch buffer and when the buffer is full, the Buffered Cassette Routine will start cassette 1 (record) and write the buffer to the cassette and then stop the cassette.
8) When ending the text edit with the E command, the last buffer load may not have been written to the cassette. Type X (escape) (escape) to get back to Zapple. Then type K.C,C,O to close the output file and write the last buffer to the cassette.
9) Turn off the cassette, type K.C,O,N to start the motors. Then rewind the cassette in the record machine.
10) Type AR=C (cr), place the assembler tape in the playback machine, type R,100 (cr), and start the cassette to load in the assembler. When the prompt character ">" is printed, stop and rewind the assembler tape.
11) Remove the source tape from cassette 1 (record), see step 9, and place it in cassette 0 (playback). Type AR=U (cr) and AP=U (cr) to assign the reader and punch to the Buffered Cassette Routine. Type K.C,O,I and K.C,O,O to open both output and input files. Cassette 0 (playback) will start up and preload the first buffer. Place a blank tape in the output machine.
12) Type G100 (cr), and when the assembler types "PASS=" respond with "1". Cassette 0 (playback) should start up, and the assembler should process pass 1.
13) When the assembler next types "PASS=" respond with "0", the assembler should trap to the monitor. Type K.C,O,I - the cassette 0 (playback) motor should start, rewind and play the tape again. The machine should stop after
loading the first buffer.

14) Type G (cr) to return to the assembler. The assembler types "PASS=" and you respond with "4". The assembler should process the second pass and type the listing on the lineprinter device and output the Hex Object tape to cassette 1 (record).

15) At the end of this pass, when the assembler types "PASS=" answer "0" and you type K,C,0 to cause the writing of the last block to cassette 1 (record).

16) Load the object tape using the Buffered Cassette Routines and the R command. I.E. AR=U instead of AR=C as with the TDL supplied tapes, because the tapes that you create using these routines are in a blocked format.

3. 2708 Programmer Software

The Standard SMB2 is provided with the Zapple (tm) Monitor in 2K Masked ROM. In addition, one 2708 has been provided as an "Extension" to the 2K Zapple. Additional commands have been included and are accessed through the user-defined "K" command.

4. Extended User-defined Commands

The syntax of the extended user-defined "K" command is:

> K.[a] where [a] may be A thru Z.

The commands included in the 2708 are:

K.A = Current I/O assignment map
K.C = Buffered Cassette Routines
K.K = Branches to a user-defined "K" command
K.P = Programs a 2708 (in conjunction with a bytesaver)
K.T = Sends a form-feed to the current List Device
K.X = Resets and clears the VDB hardware
K.Z = Absolute JMP to Zapple

Any other commands have been turned off. They may be set to another address by programming the .WORD address into the proper location (using a bytesaver).

The Commands

K.A Simply typing "K.A" will respond with the current I/O assignment map printed on the console. Example:
which means:

- C=V the current console is the Video device
- R=T the current reader device is the TTY
- P=T the current punch device is the TTY
- L=L the current list device the (user routine) is the lineprinter.

See section VI.-A.-2. on the Buffered Cassette.

This vectors to a "JMP" located at OFCIEH. Note that this JMP and the routine must have been placed in memory by the user prior to its use.

This program allows direct manipulation of the Bytesaver through the Zapple Monitor. It is self-prompting and easy to use. Prior to typing "K.P", you should put the code you want to program somewhere easy to remember, (i.e. 1000H) and put the bytesaver in the machine at some convenient address (i.e. 8000H).

Type "K.P" and the message:

BASE ADDR:

will be printed. This refers to the base address of the bytesaver. In the above example, you would type 8000. Study the following:

BASE ADDR:8000(return)

The computer will then type:

NMBR & DATA ADDR:

The "NMBR" refers to which of the 8 ROM sockets you want to program. "0" is the one on the extreme right of the bytesaver board and "7" would be on the extreme left.

The "DATA ADDR:" refers to the address of the DATA (byte pattern) you wish to place in the "NMBR" ROM.
In the above example, you might type

NMBR & DATA ADDR:0,1000(return)

Note the comma between 0 and 1000.
The computer then prompts with:

SW ON-

Waiting for you to turn the "Program Power" switch to the UP (on) position.

When you have turned it on, type a return and wait. After a short time you should get another prompt that reads:

SW OFF-

NOTE:  If before the SW OFF- message you get *BAD* printed, this means a bad ROM was detected and should be changed.

If no *BAD* message was encountered, turn the switch to the DOWN (off) position and type return.

The computer will then type:

NMBR & DATA ADDR:

This allows programming up to 8 ROMs in one session. You would then type:

NMBR & DATA ADDR:1,1400(return)

and the process will be repeated for the next ROM. Note that the address must also reflect the next ROM as well (+400H).

When you are done, simply answer the "NMBR & DATA ADDR:" question with a carriage return only, and you will return to the Zapple command mode.

K.T  This command will send a form-feed (0CH) to the currently assigned list device.

K.X  This command will perform a software reset of TDL's VDB board and initialize the screen and send the cursor "Home", to the upper left corner.
NOTE: This command assumes the VDB is set up for the TDL standard of 9CH for the control port. Do not implement this command unless you have a VDB in the system. (It will "hang" the system, and you would have to use RESET to get back to Zapple.)

K.Z This is an unconditional branch to Zapple. It is to be used after clearing the memory area where the monitor's stack is located.

5. Current Assignment Display

This routine will print on the console the current I/O assignments. It may be evoked by the user-defined expanded "K.A" command.

B. Cassette Interface

A good quality cassette recorder should be used with this interface. Most tape recorders selling for $69.95 and above should be excellent choices. Due to the wide variety of recorders available and variation from recorder to recorder it is difficult to recommend a particular brand. However, the above price range should be of some help. Although less expensive recorders such as the General Electric (NO. 3-5105) at $39.95 and others have been used successfully at TDL, it is observed that they are harder to adjust and operate and some of them are extremely difficult, if not impossible, to get running reliably.

The volume control is the most critical adjustment to make. If the unit has a tone control, it should be set at the extreme "treble" setting. It is best to begin by making a recording of the Zapple Monitor contained in ROM and then playing it back and verifying it against the ROM. Begin by plugging the audio cable coming from "R" on the SMB to the "RECORD" jack of the recorder and the cable from "P" to the "SPEAKER" or "EARPHONE" jack of the recorder. The "RECORD" jack may be labelled either "MIC" or "AUX". Some recorders have both. See User Guide section A.3. Cassette Options on setting of S3, the MIC/AUX switch, and make sure it is in the correct position.

To copy the monitor, the following command should be used:
The tape recorder should be started well in advance of hitting return and a few nulls (N<return>) may be inserted before actually recording. When the monitor comes back with the ">" prompt, enter the following:

E (return)

This latter step is very important as it signals the end of the file. What has just been recorded is a checksummed hex file of the Zapple Monitor.

Rewind the tape to the starting position and note the volume control setting. The following command should be used on playback:

R1000 (return)

Start the recorder before hitting return so the speed has a chance to become stable. When the playback is finished, the monitor will return with the ">" prompt. At this time, if the volume setting was adequate, there will be a copy of the monitor at 0000H to 07FFH. Make sure there is RAM at that location. Test for accuracy of the copy by using the verify command:

V0,07FF,F000 (return)

If everything is OK, the ">" prompt will return. If not, the addresses that do not match will be printed followed by the hex representation of the copy and then the hex of the monitor’s byte.

Before changing the volume setting and re-recording, switch S4 to its other state (invert). If that isn’t successful, re-record at different volume settings until the proper one is found. Don’t forget to try playback in both the inverted and non-inverted states of S4.

Some units, especially the less expensive ones are very poorly isolated internally and will pick up the continuous tone put out by the interface’s record circuitry and feed it back on playback. If trouble is experienced, try unplugging the jack to the recorder while playing back.
Extension routines for TDL "SYSTEM MONITOR BOARD".
VERSION 1.0 - MAR 21, 1978 - Gathered by Roger Amidon

; THIS SOFTWARE IS PROVIDED IN A ONE-K ROM
; FOR USE IN TDL "SMB2".

; COPYRIGHT 1978 By Technical Design Labs, Inc.
; 1101 State Rd. Research Park, Bldg. H
; Princeton, New Jersey 08540

+++ CONSTANTS +++

FFFF
OFF == -1 ; UNIMPLEMENTED BRANCH COMMAND

+++ ASCII CONTROL CHARACTERS +++

0008 BS == 08H ; BACK SPACE
000D CR == 0DH ; CARRIAGE RETURN
000A LF == 0AH ; LINE FEED
000C FF == 0CH ; FORM FEED
0007 BEL == 07H ; BELL
0011 DC1 == 11H ; DC CONTROLS
0012 DC2 == 12H
0013 DC3 == 13H
0014 DC4 == 14H
007F DEL == 7FH ; DELETE

; THIS SOFTWARE MAY BE MODIFIED BY THE USER
; WITH FULL PERMISSION HEREBY GRANTED BY TDL.
; HOWEVER, TECHNICAL DESIGN LABS ASSUMES NO
; LIABILITY WITH REGARD TO EITHER THE USAGE
; OR PERFORMANCE OF THE FOLLOWING PROGRAMS.

; THIS SOFTWARE WAS DESIGNED TO BE USED ONLY
; IN CONJUNCTION WITH HARDWARE SUPPLIED BY
; TECHNICAL DESIGN LABS.
Extension routines for TDL "SYSTEM MONITOR BOARD".
ZAPPLE MONITOR VARIABLE EQUATES

+++ ZAPPLE MONITOR EQUATES +++

F000 START == ZAPPLE ; START OF MONITOR

F003 CI == ZAPPLE+3
F006 RI == ZAPPLE+6
F009 CO == ZAPPLE+9
F00C PO == ZAPPLE+12
F00F LO == ZAPPLE+15
F01E TRAP == ZAPPLE+30

0070 TTS == IO ; TTY STATUS
0074 RCSS == IO+4 ; CASSETTE STATUS
0076 IOBYT == IO+6 ; I/O BYTE LATCH

BUFFERED CASSETTE ROUTINE EQUATES

0080 BSIZE == 128 ; BUFFER SIZE
0016 SYN == 16H ; SYNC CHARACTER

F464 ERROR == OF464H ; ERROR RETURN
F574 HILO == OF574H ; INX H & CMP HL TO DE
F59E MARK == OF59EH ; PUNCH 8 RUB OUTS
F5A3 LEAD == OF5A3H ; PUNCH NULLS
F794 LTBL == OF794H ; DEVICE ASSIGNMENT TABLE
F488 BLK == OF488H ; PRINT A SPACE
F452 TOM == OF452H
F736 TI == OF736H ; GET KEYBOARD & ECHO
F730 KI == OF730H ; GET KEYBOARD, NO ECHO
F60A PCHK == OF60AH ; TEST FOR DELIMITER
F540 EXPR == OF540H ; GET 16 BIT ADDRESS
F512 CRLF == OF512H ; IN SMB1

F800 ROM == ZAPPLE+800H
Extension routines for TDL "SYSTEM MONITOR BOARD".

BRANCH TABLE AT ZAPPLE VECTOR LOCATIONS

<table>
<thead>
<tr>
<th>Address</th>
<th>Opcode 1</th>
<th>Opcode 2</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F800</td>
<td>C3</td>
<td>F824</td>
<td>JMP CIU ; USER CONSOLE INPUT</td>
</tr>
<tr>
<td>F803</td>
<td>C3</td>
<td>F82D</td>
<td>JMP COU ; USER CONSOLE OUTPUT</td>
</tr>
<tr>
<td>F806</td>
<td>C3</td>
<td>FC06</td>
<td>JMP RIPLOC ; AR=P (VECTOR TO RAM)</td>
</tr>
<tr>
<td>F809</td>
<td>C3</td>
<td>FA9D</td>
<td>JMP READ ; BUFFERED CASSETTE READ ROUTINE</td>
</tr>
<tr>
<td>F80C</td>
<td>C3</td>
<td>FC0C</td>
<td>JMP POPLoc ; AP=P (VECTOR TO RAM)</td>
</tr>
<tr>
<td>F80F</td>
<td>C3</td>
<td>FA41</td>
<td>JMP PUNCH ; BUFFERED CASSETTE ROUTINE</td>
</tr>
<tr>
<td>F812</td>
<td>C3</td>
<td>FC12</td>
<td>JMP LLOLC ; AL=L (VECTOR TO RAM)</td>
</tr>
<tr>
<td>F815</td>
<td>C3</td>
<td>F846</td>
<td>JMP LOU ; USER LIST DEVICE (VDB)</td>
</tr>
<tr>
<td>F818</td>
<td>C3</td>
<td>F836</td>
<td>JMP CSU ; USER CONSOLE STATUS</td>
</tr>
<tr>
<td>F81B</td>
<td>C3</td>
<td>FC1B</td>
<td>JMP IUCMND ; USER DEFINED &quot;I&quot; COMMAND</td>
</tr>
<tr>
<td>F81E</td>
<td>C3</td>
<td>F85E</td>
<td>JMP KUSER ; K COMMAND</td>
</tr>
<tr>
<td>F821</td>
<td>C3</td>
<td>FC21</td>
<td>JMP OUICMND ; USER DEFINED &quot;O&quot; COMMAND</td>
</tr>
</tbody>
</table>

SUPPORT ROUTINES TO ENABLE USAGE OF THE TOTAL VDB CAPABILITIES, AND STILL ALLOW USER-DEFINABLE CONSOLE CONFIGURATIONS.

<table>
<thead>
<tr>
<th>Address</th>
<th>Opcode 1</th>
<th>Opcode 2</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F824</td>
<td>CD</td>
<td>F83F</td>
<td>CALL TEST ; DETERMINE CONFIGURATION</td>
</tr>
<tr>
<td>F827</td>
<td>C2</td>
<td>FC00</td>
<td>JNZ CIULOC ; NOT BATCH MODE, VECTOR OUT</td>
</tr>
<tr>
<td>F82A</td>
<td>C3</td>
<td>F9E4</td>
<td>JMP KBIN ; ELSE USE VDB KEYBOARD</td>
</tr>
<tr>
<td>F82D</td>
<td>CD</td>
<td>F83F</td>
<td>CALL TEST ; DETERMINE CONFIGURATION</td>
</tr>
<tr>
<td>F830</td>
<td>C2</td>
<td>FC03</td>
<td>JNZ COULOC ; NOT BATCH MODE, VECTOR OUT</td>
</tr>
<tr>
<td>F833</td>
<td>C3</td>
<td>F900</td>
<td>JMP VDB ; ELSE USE VDB DRIVER</td>
</tr>
<tr>
<td>F836</td>
<td>CD</td>
<td>F83F</td>
<td>CALL TEST ; DETERMINE CONFIGURATION</td>
</tr>
<tr>
<td>F839</td>
<td>C2</td>
<td>FC18</td>
<td>JNZ CSULOC ; NOT BATCH MODE, VECTOR OUT</td>
</tr>
<tr>
<td>F83C</td>
<td>C3</td>
<td>F9ED</td>
<td>JMP KBSTS ; ELSE USE VDB KEYBOARD</td>
</tr>
<tr>
<td>F83F</td>
<td>DB7A</td>
<td></td>
<td>TEST IN IO+10 ; READ CONFIGURATION SW.</td>
</tr>
<tr>
<td>F841</td>
<td>E603</td>
<td></td>
<td>ANI 3 ; LOOK AT CONSOLE SECTION</td>
</tr>
<tr>
<td>F843</td>
<td>FEO2</td>
<td></td>
<td>CPI 2 ; BATCH MODE? (VDB)</td>
</tr>
<tr>
<td>F845</td>
<td>C9</td>
<td></td>
<td>RET</td>
</tr>
<tr>
<td>F846</td>
<td>DB77</td>
<td></td>
<td>LOU IN IO+7 ; LOOK AT IOBYTE CONTROL</td>
</tr>
<tr>
<td>F848</td>
<td>CB47</td>
<td></td>
<td>BIT 0,A ; HAS VDB BEEN INITIALIZED?</td>
</tr>
<tr>
<td>F84A</td>
<td>C2</td>
<td>F900</td>
<td>JNZ VDB ; YES, USE DRIVER</td>
</tr>
<tr>
<td>F84D</td>
<td>CBC7</td>
<td></td>
<td>SET 0,A ; ELSE SET-UP SMB HARDWARE</td>
</tr>
<tr>
<td>F84F</td>
<td>D377</td>
<td></td>
<td>OUT IO+7 ; STORE THE FACT</td>
</tr>
<tr>
<td>F851</td>
<td>CD</td>
<td>F83F</td>
<td>CALL TEST ; BATCH MODE?</td>
</tr>
<tr>
<td>F854</td>
<td>2006</td>
<td></td>
<td>JRNZ .NO ; JUST USER LIST</td>
</tr>
<tr>
<td>F856</td>
<td>DB76</td>
<td></td>
<td>IN IO+6 ; MODIFY CONSOLE TO USER</td>
</tr>
<tr>
<td>F858</td>
<td>F603</td>
<td></td>
<td>ORI 3 ; FORCE TO USER</td>
</tr>
<tr>
<td>F85A</td>
<td>D376</td>
<td></td>
<td>OUT IO+6</td>
</tr>
<tr>
<td>F85C</td>
<td>1856</td>
<td></td>
<td>.NO: JMPRI INIT ; INITIALIZE THE VDB</td>
</tr>
</tbody>
</table>
Extension routines for TDL "SYSTEM MONITOR BOARD".

**EXPANDED COMMAND BRANCH TABLE**

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Instruction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F85E</td>
<td>LXI H,UTAB</td>
<td>Point to command table</td>
</tr>
<tr>
<td>F861</td>
<td>LXI H,UTAB</td>
<td>Get next character</td>
</tr>
<tr>
<td>F864</td>
<td>LXI H,UTAB</td>
<td>Must be a period</td>
</tr>
<tr>
<td>F866</td>
<td>LXI H,UTAB</td>
<td>ERROR</td>
</tr>
<tr>
<td>F869</td>
<td>LXI H,UTAB</td>
<td>Get next character</td>
</tr>
<tr>
<td>F86C</td>
<td>LXI H,UTAB</td>
<td>Qualify it</td>
</tr>
<tr>
<td>F86E</td>
<td>LXI H,UTAB</td>
<td>&lt;A</td>
</tr>
<tr>
<td>F86F</td>
<td>LXI H,UTAB</td>
<td>CPI 'Z'-'A'+1</td>
</tr>
<tr>
<td>F871</td>
<td>LXI H,UTAB</td>
<td>D0</td>
</tr>
<tr>
<td>F872</td>
<td>LXI H,UTAB</td>
<td>ADD A ;A*2</td>
</tr>
<tr>
<td>F873</td>
<td>LXI H,UTAB</td>
<td>ADD L ;+UTAB</td>
</tr>
<tr>
<td>F874</td>
<td>LXI H,UTAB</td>
<td>MOV L,A ;INDEX INTO TABLE</td>
</tr>
<tr>
<td>F875</td>
<td>LXI H,UTAB</td>
<td>MOV A,M ;GET LOW BYTE</td>
</tr>
<tr>
<td>F876</td>
<td>LXI H,UTAB</td>
<td>INX H</td>
</tr>
<tr>
<td>F877</td>
<td>LXI H,UTAB</td>
<td>MOV H,M ;GET HIGH BYTE</td>
</tr>
<tr>
<td>F878</td>
<td>LXI H,UTAB</td>
<td>MOV L,A ;HL=ROUTINE ADDRESS</td>
</tr>
<tr>
<td>F879</td>
<td>LXI H,UTAB</td>
<td>A4</td>
</tr>
<tr>
<td>F87A</td>
<td>LXI H,UTAB</td>
<td>INR A</td>
</tr>
<tr>
<td>F87B</td>
<td>LXI H,UTAB</td>
<td>CA F464</td>
</tr>
<tr>
<td>F87E</td>
<td>LXI H,UTAB</td>
<td>E9</td>
</tr>
<tr>
<td>F880</td>
<td>LXI H,UTAB</td>
<td>PCHL</td>
</tr>
</tbody>
</table>

**UTAB: .LOC ROM+128 ;BRANCH TABLE LOCATION**

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Instruction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F880</td>
<td>LXI H,UTAB</td>
<td>Assign display</td>
</tr>
<tr>
<td>F882</td>
<td>LXI H,UTAB</td>
<td>UTILITY</td>
</tr>
<tr>
<td>F884</td>
<td>LXI H,UTAB</td>
<td>CASSETTE ROUTINES</td>
</tr>
<tr>
<td>F886</td>
<td>LXI H,UTAB</td>
<td>LOAD</td>
</tr>
<tr>
<td>F888</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F88A</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F88C</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F88E</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F890</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F892</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F894</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F896</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F898</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F89A</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F89C</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F89E</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A0</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A2</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A4</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A6</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A8</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8AA</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8AC</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8AE</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8B0</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8B2</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
</tbody>
</table>

**F880**

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Instruction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F8C5</td>
<td>LXI H,UTAB</td>
<td>ADISP ;A - ASSIGNMENT DISPLAY</td>
</tr>
<tr>
<td>F882</td>
<td>LXI H,UTAB</td>
<td>UTILITY</td>
</tr>
<tr>
<td>F884</td>
<td>LXI H,UTAB</td>
<td>PGM ;P - PROGRAM A 2708</td>
</tr>
<tr>
<td>F886</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F888</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F88A</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F88C</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F88E</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F890</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F892</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F894</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F896</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F898</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F89A</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F89C</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F89E</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A0</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A2</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A4</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A6</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A8</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8AA</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8AC</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8AE</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8B0</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8B2</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
</tbody>
</table>

**F8B2**

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Instruction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F000</td>
<td>LXI H,UTAB</td>
<td>ZAPPLE ;Z - RESTART ZAPPLE</td>
</tr>
</tbody>
</table>

**F880**

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Instruction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F8C5</td>
<td>LXI H,UTAB</td>
<td>ADISP ;A - ASSIGNMENT DISPLAY</td>
</tr>
<tr>
<td>F882</td>
<td>LXI H,UTAB</td>
<td>UTILITY</td>
</tr>
<tr>
<td>F884</td>
<td>LXI H,UTAB</td>
<td>PGM ;P - PROGRAM A 2708</td>
</tr>
<tr>
<td>F886</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F888</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F88A</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F88C</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F88E</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F890</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F892</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F894</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F896</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F898</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F89A</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F89C</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F89E</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A0</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A2</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A4</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A6</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8A8</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8AA</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8AC</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8AE</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8B0</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
<tr>
<td>F8B2</td>
<td>LXI H,UTAB</td>
<td>OFF</td>
</tr>
</tbody>
</table>

**F8B2**

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Instruction</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F000</td>
<td>LXI H,UTAB</td>
<td>ZAPPLE ;Z - RESTART ZAPPLE</td>
</tr>
</tbody>
</table>
Extension routines for TDL "SYSTEM MONITOR BOARD".

VDB INITIALIZATION SECTION

+++ VDB INITIALIZATION SECTION +++

0080

MDBIT = 80H

; THIS CAN BE CHANGED TO ALLOW
; A USER DEFINED INITIALIZATION.

; 80H = NORMAL MODE, BLINKING CURSOR
; 90H = NORMAL MODE, NO CURSOR
; 88H = REVERSE VIDEO, BLINKING CUR.
; 98H = REVERSE VIDEO, NO CURSOR

F8B4 3EE3 INIT: MVI A, VDBRES ; RESET VDB
F8B6 D39C OUT VDBCTL
F8B8 3E7F MVI A, #MDBIT ; WRITE TO MODE REG.
F8BB 2F CMA
F8BB D39C OUT VDBCTL
F8BD AF XRA A ; CLEAR PARALLEL PORT
F8BE D39F OUT VDBKD
F8C0 3EOC MVI A, FF ; FORM FEED
F8C2 C3 F968 JMP CNTL ; FAKE IT
Extension routines for TDL "SYSTEM MONITOR BOARD".
CURRENT ASSIGNMENT DISPLAY PROGRAM

; THIS ROUTINE WILL PRINT ON THE CONSOLE THE
; CURRENT I/O ASSIGNMENTS.
;
; IT MAY BE EVOKED BY EITHER THE USER-DEFINED
; EXPANDED "K" COMMAND [K.A] IN ZAPPLE V1.X,
; OR THE "BRANCH" COMMAND IN ZAPPLE V2.X
; [B.A]

F8C5    CD F512    ADISP: CALL  CRLF
F8C8    21 F793
F8CB    1E04
F8CD    DB76
F8CF    57    ..AD1: MOV    D.A  ;SAVE IT IN DE
F8D0    08
F8D1    0604
F8D3    CD F488
F8D6    23
F8D7    4E
F8D8    CD F009
F8DB    0E3D
F8DD    CD F009
F8E0    7A
F8E1    E603
F8E3    57
F8E4    14
F8E5    23    ..AD2: INX    H  ;POINT TO ASSIGNMENT
F8E6    4E
F8E7    15
F8E8    CC F009
F8EB    10F8
F8ED    08
F8EE    1F
F8EF    1F
F8F0    1D
F8F1    20DC
F8F3    C9
F8F4    CD F730
F8F7    FE0D
F8F9    20F9
F8FB    C9

; SUBROUTINE FOR ROM PROGRAMMER
; WAITS FOR A CR FROM THE CONSOLE
; (HERE TO FILL OUT PAGE...)

; WAIT: CALL  KI
;       CPI  CR
;       JRNZ  WAIT
;       RET
Extension routines for TDL "SYSTEM MONITOR BOARD".
VDB DRIVER PROGRAM, WITH RAM BUFFER

MARCH 1978 - by Roger Amidon

+++ VDB PORT AND MASK DEFINITIONS +++

009D VDBDAT == VDBCTL+1 ;DATA PORT ADDRESS
009E VDBK.S == VDBCTL+2 ;VDB KEYBOARD STATUS
009F VDBK.D == VDBCTL+3 ;KEYBOARD DATA
0080 VDBMRF == 10000000B ;MODE REGISTER FLAG
00C0 VDBYCF == 11000000B ;Y CURSOR FLAG
00E0 VDBXCR == 11100000B ;X CURSOR READ
00E1 VDBYCR == 11100001B ;Y CURSOR READ
00E2 VDBMRR == 11100010B ;MODE REGISTER READ
00E3 VDBRES == 11100011B ;RESET VDB
0060 VDBGMK == 01100000B ;GRAPHIC MASK

+++ MODE BIT DEFINITIONS +++

0000 VDBZAP == 0 ;ALTERNATE PAGE
0001 VDBZBE == 1 ;BLINK ENABLE
0002 VDBZS == 2 ;INVERT SYMBOL
0003 VDBZID == 3 ;INVERT DISPLAY
0004 VDBZDC == 4 ;DISABLE CURSOR
0005 VDBZDD == 5 ;DISABLE DISPLAY

+++ VDB TELETYPewriter SIMULATOR +++

F900 .LOC ROM+100H ;PUT THIS ON A PAGE BORDER

F900 79 VDB: MOV A,C ;GET OUTPUT CHARACTER
F901 E67F ANI 7FH ;GET RID OF PARITY BIT
F903 C8 RZ ;IGNORE NULLS
F904 FE7F CPI DEL ;RUBOUT?
F906 C8 RZ ;IGNORE IT
F907 C5 PUSH B ;SAVE REGISTERS
F908 D5 PUSH D ;
F909 E5 PUSH H ;
F90A CD F912 CALL MAIN ;CALL DRIVER
F90D E1 POP H ;RESTORE REGISTERS
F90E D1 POP D ;
F90F C1 POP B ;
F910 79 MOV A,C ;OUTPUT CHARACTER INTO A
F911 C9 RET ;DONE

+++ VDB DRIVER +++

F912 0E9C MAIN: MVI C,VDBCTL ;SET C UP
F914 FE20 CPI " " ;CONTROL CHARACTER?
F916 3850 JRC CNTL ;YES
F918 47 MOV B,A ;SAVE PRINTING CHAR.
F919 3EE2 MVI A,VDBMRR ;GET THE MODE
F91B D39C OUT VDBCTL
F91D DB9C IN VDBCTL
Extension routines for TDL "SYSTEM MONITOR BOARD".
VDB DRIVER PROGRAM, WITH RAM BUFFER

```
F91F E606  ANI 1< VDBXBE ! 1< VDBXIS
F921 2802  JRZ  .NO7  NO BLINK OR DISPLAY INVERT
F923 CBF8  SET  7,B
F925 78  .NO7:  MOV A,B  WRITE THE CHARACTER
F926 D39D  OUT VDBDAT
F928 3EE0  MVI A,VDBXCR
F92A D39C  OUT VDBCTL ; TEST X POS
F92C ED40  INP B  SEE IF WE JUST WENT ZERO
F92E C0  RNZ  NOPE
F92F 3C  INR A  ; A = VDBYCR
F930 D39C  OUT VDBCTL ; TEST Y POS
F932 ED40  INP B  DID WE GO ZERO?
F934 C0  RNZ  NOPE
F935 21 FC24 SCROL: LXI H,BUFF  POINT TO BUFFER
F938 5D  MOV E,L  SAVE POINTER LSB
F939 0C  INR C  ; C = VDBDAT
F93A CD F9CE CALL GTMD  GET CURRENT MODE
F93D F5  PUSH PSW  FOR LATER
F93E F610  ORI 1< VDBZDC  KILL CURSOR
F940 D39C  OUT VDBCTL  INHIBIT
F942 AF  XRA A  X = 0
F943 D39C  OUT VDBCTL
F945 3EC1  MVI A,VDBYCF+1  SET UP Y
F947 1618  MVI D,24  DO THIS 24 TIMES
F949 D39C  .SC1:  OUT VDBCTL  SET Y POS
F94B 0650  MVI B,80  GET THE 80 CHARACTERS AT Y
F94D EDB2  INIR
F94F 6B  MOV L,E  RESET BUFF
F950 3D  DCR A  Y = Y - 1
F951 D39C  OUT VDBCTL
F953 0650  MVI B,80
F955 EDB3  OUTIR
F957 6B  MOV L,E  RESET BUFF
F958 C602  ADI 2  DO NEXT LINE
F95A 15  DCR D
F95B 20EC  JRNZ .SC1
F95D CD F9D7 CALL CLIN  CLEAR LAST LINE
F960 3ED8  MVI A,24! VDBYCF
F962 D39C  OUT VDBCTL
F964 F1  POP PSW
F965 D39C  OUT VDBCTL ; ENABLE DISPLAY
F967 C9  RET
```

```
F968 FE0A  CNTL: CPI LF  ; IS IT LF?
F96A 282F  JRZ .LF  ; GO DO IT
F96C FE0D  CPI CR  ; IS IT CR?
F96E 284F  JRZ .CR  ; GO DO IT
F970 FE08  CPI BS  ; IS IT BS?
F972 284F  JRZ .BS  ; GO DO IT
F974 FE0C  CPI FF  ; IS IT FF?
F976 2833  JRZ .FF  ; GO DO IT
F978 FE07  CPI BEL  ; IS IT BELL?
F97A 2810  JRZ .BEL  ; GO DO IT
F97C D611  SUI DC1  ; IS IT DC1?
```
Extension routines for TDL "SYSTEM MONITOR BOARD"

VDB DRIVER PROGRAM. WITH RAM BUFFER

<table>
<thead>
<tr>
<th>Address</th>
<th>Opcode</th>
<th>Instruction</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>F97E</td>
<td>D8</td>
<td>RC</td>
<td></td>
</tr>
<tr>
<td>F97F</td>
<td>FE04</td>
<td>CPI 4</td>
<td></td>
</tr>
<tr>
<td>F981</td>
<td>D0</td>
<td>RNC</td>
<td></td>
</tr>
<tr>
<td>F982</td>
<td>0601</td>
<td>MVI B,1</td>
<td>THE SHIFT BIT</td>
</tr>
<tr>
<td>F984</td>
<td>3C</td>
<td>INR A</td>
<td></td>
</tr>
<tr>
<td>F985</td>
<td>3D</td>
<td>DCR A</td>
<td></td>
</tr>
<tr>
<td>F986</td>
<td>280C</td>
<td>JRZ .OK</td>
<td></td>
</tr>
<tr>
<td>F988</td>
<td>CB20</td>
<td>SLAR B</td>
<td></td>
</tr>
<tr>
<td>F98A</td>
<td>18F9</td>
<td>JMPR .SHFT</td>
<td>SET B UP</td>
</tr>
<tr>
<td>F98C</td>
<td>3E80</td>
<td>MVI A,80H</td>
<td>PULSE BIT 7 TO</td>
</tr>
<tr>
<td>F98E</td>
<td>D39F</td>
<td>OUT VDBK.D</td>
<td>RING BELL</td>
</tr>
<tr>
<td>F990</td>
<td>AF</td>
<td>XRA A</td>
<td></td>
</tr>
<tr>
<td>F991</td>
<td>D39F</td>
<td>OUT VDBK.D</td>
<td></td>
</tr>
<tr>
<td>F993</td>
<td>C9</td>
<td>RET</td>
<td></td>
</tr>
<tr>
<td>F994</td>
<td>CD F9CE</td>
<td>CALL GTMD</td>
<td>GET CURRENT MODE</td>
</tr>
<tr>
<td>F997</td>
<td>A8</td>
<td>XRA B</td>
<td>COMPLIMENT SELECTED BIT</td>
</tr>
<tr>
<td>F998</td>
<td>D39C</td>
<td>OUT VDBCTL</td>
<td>SET NEW MODE</td>
</tr>
<tr>
<td>F99A</td>
<td>C9</td>
<td>RET</td>
<td></td>
</tr>
<tr>
<td>F99B</td>
<td>3EE1</td>
<td>MVI A,VDBYCR</td>
<td>ENABLE TO READ Y POS</td>
</tr>
<tr>
<td>F99D</td>
<td>D39C</td>
<td>OUT VDBCTL</td>
<td></td>
</tr>
<tr>
<td>F99F</td>
<td>D99C</td>
<td>IN VDBCTL</td>
<td>READ Y</td>
</tr>
<tr>
<td>F9A1</td>
<td>3C</td>
<td>INR A</td>
<td></td>
</tr>
<tr>
<td>F9A2</td>
<td>FE19</td>
<td>CPI 25</td>
<td>TIME TO SCROLL?</td>
</tr>
<tr>
<td>F9A4</td>
<td>308F</td>
<td>JRNC SCROL</td>
<td></td>
</tr>
<tr>
<td>F9A6</td>
<td>F6C0</td>
<td>ORI VDBYCF</td>
<td>SET Y</td>
</tr>
<tr>
<td>F9A8</td>
<td>D39C</td>
<td>OUT VDBCTL</td>
<td></td>
</tr>
<tr>
<td>F9AA</td>
<td>C9</td>
<td>RET</td>
<td></td>
</tr>
<tr>
<td>F9AB</td>
<td>3EC0</td>
<td>MVI A,VDBYCF</td>
<td></td>
</tr>
<tr>
<td>F9AD</td>
<td>D39C</td>
<td>OUT VDBCTL</td>
<td>RESET X&amp;Y</td>
</tr>
<tr>
<td>F9AF</td>
<td>AF</td>
<td>XRA A</td>
<td></td>
</tr>
<tr>
<td>F9B0</td>
<td>D39C</td>
<td>OUT VDBCTL</td>
<td></td>
</tr>
<tr>
<td>F9B2</td>
<td>CD F9D7</td>
<td>CALL CLIN</td>
<td>CLEAR THE LINE</td>
</tr>
<tr>
<td>F9B5</td>
<td>3EE1</td>
<td>MVI A,VDBYCR</td>
<td>SET TO READ Y</td>
</tr>
<tr>
<td>F9B7</td>
<td>D39C</td>
<td>OUT VDBCTL</td>
<td></td>
</tr>
<tr>
<td>F9B9</td>
<td>DB9C</td>
<td>IN VDBCTL</td>
<td>INPUT Y CURSOR</td>
</tr>
<tr>
<td>F9BB</td>
<td>B7</td>
<td>ORA A</td>
<td></td>
</tr>
<tr>
<td>F9BC</td>
<td>20F4</td>
<td>JRNZ .FF1</td>
<td>NOT DONE, TRY AGAIN</td>
</tr>
<tr>
<td>F9BE</td>
<td>C9</td>
<td>RET</td>
<td>DONE</td>
</tr>
<tr>
<td>F9BF</td>
<td>AF</td>
<td>XRA A</td>
<td>SET X=0</td>
</tr>
<tr>
<td>F9C0</td>
<td>D39C</td>
<td>OUT VDBCTL</td>
<td></td>
</tr>
<tr>
<td>F9C2</td>
<td>C9</td>
<td>RET</td>
<td></td>
</tr>
<tr>
<td>F9C3</td>
<td>3EE0</td>
<td>MVI A,VDBXCR</td>
<td></td>
</tr>
<tr>
<td>F9C5</td>
<td>D39C</td>
<td>OUT VDBCTL</td>
<td></td>
</tr>
<tr>
<td>F9C7</td>
<td>ED58</td>
<td>INP E</td>
<td></td>
</tr>
<tr>
<td>F9C9</td>
<td>C8</td>
<td>RZ DCR E</td>
<td>AT LEFT MARGIN</td>
</tr>
<tr>
<td>F9CA</td>
<td>1D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F9CB</td>
<td>ED59</td>
<td>OUTP E</td>
<td>X=X-1</td>
</tr>
<tr>
<td>F9CD</td>
<td>C9</td>
<td>RET</td>
<td></td>
</tr>
</tbody>
</table>
Extension routines for TDL "SYSTEM MONITOR BOARD".
VDB DRIVER PROGRAM, WITH RAM BUFFER

; GTMD: MVI A, VDBMRR ; SET TO READ MODE
F9CE 3EE2
F9D0 D39C OUT VDBCTL ; SET IT
F9D2 DB9C IN VDBCTL ; GET CURRENT MODE
F9D4 F680 ORI VDBMRF ; SET TO WRITE MODE
F9D6 C9 RET

; CLIN: MVI A, VDBXCR
F9D7 3EE0 OUT VDBCTL
F9D9 D39C XRA A
F9DB AF ..CL1: OUT VDBDAT
F9DC D39D IN VDBCTL
F9DE DB9C ORA A
F9E0 B7 JRNZ ..CL1
F9E1 20F8 RET
F9E3 C9
Extension routines for TDL "SYSTEM MONITOR BOARD".
ADDITIONAL SUPPORT ROUTINES

+++ VDB PARALLEL KEYBOARD INTERFACE +++

NOTE: Assumes jumpers from 'A' to 'B'
on VDB board. (See MANUAL)

F9E4 DB9E KBIN: IN VDBK.S ;READ STATUS
F9E6 E680 ANI 80H ;TEST FOR KEYPRESS
F9E8 28FA JRZ KBIN ;NO, KEEP LOOKING
F9EA DB9F IN VDBK.D ;OK, READ IT
F9EC C9 RET ;DONE

F9ED DB9E KBSTS: IN VDBK.S ;READ STATUS
F9EF E680 ANI 80H ;TEST BIT 7
F9F1 C8 RZ ;RETURN FALSE
F9F2 3EFF MVI A,-1 ;SET-UP A, NOT FLAGS
F9F4 C9 RET ;RETURN TRUE

+++ TOP OF FORM +++

F9F5 0E0C TForm: MVI C,FF ;SEND OUT A FORM FEED
F9F7 C3 F00F JMP LO ;TO THE CURRENT LIST DEVICE
Extension routines for TDL "SYSTEM MONITOR BOARD".  
TDL/SMB ROM BUFFERED CASSETTE ROUTINES

; BY TOM KIRK 10/5/77

; THIS BUFFERED CASSETTE ROUTINE IS CALLED
; USING THE "USER" ASSIGNMENT FOR THE READER/
; PUNCH LOGICAL DEVICES.

; COMMANDS ARE AS FOLLOWS:
; *NOTE* - COMPUTER TYPES THE COMMAS

; K.C,O,N                TURN ON THE MOTORS OF THE
; CASSETTE RECORDERS. TO REWIND
; TAPES, ETC.

; K.C,O,F                TURN OFF THE MOTORS OF THE
; CASSETTE RECORDERS.

; K.C,O,O                OPEN THE OUTPUT FILE. RESET
; THE POINTER TO THE BEGINNING
; OF THE BUFFER. USED BEFORE
; CREATING ANY OUTPUT.

; K.C,O,I                OPEN THE INPUT FILE. START UP
; THE PLAY CASSETTE, LOAD THE
; FIRST BLOCK, STOP THE MOTOR,
; RESET THE POINTER TO THE
; BEGINNING OF THE BUFFER.
; USED PRIOR TO THE START OF
; READING ANY INPUT.

; K.C,C,O                CLOSE THE OUTPUT FILE. FILL
; THE REST OF THE BUFFER WITH
; NULLS ( OOH ), START THE RECORD
; CASSETTE, AND WRITE OUT THE
; LAST BLOCK. MUST BE USED TO
; WRITE THE LAST BLOCK TO THE
; CASSETTE AT THE END OF A PROGRAM.

; K.C,C,I                CLOSE THE INPUT FILE. FILL THE
; INPUT BUFFER WITH CONTROL Z
; ( 1AH ) CHARACTERS AND RESET THE
; POINTER. MUST BE USED WITH THE
; TEXT EDITOR TO TERMINATE THE
; COPY INPUT TO OUTPUT PHASE OF
; THE "E" COMMAND WHEN ENDING A
; TEXT EDITING SESSION THAT DOES
; NOT HAVE AN INPUT TAPE.
Extension routines for TDL "SYSTEM MONITOR BOARD".

TDL/SMB ROM BUFFERED CASSETTE ROUTINES

; UTILITY: CALL GCHAR ; GET SECOND LETTER

F9FA CD FA1A UTILITY: CALL GCHAR ; GET SECOND LETTER
F9FD 41 MOV B,C
F9FE CD FA1A CALL GCHAR ; GET THIRD LETTER
FA01 21 FA25 LXI H,TABLE-3 ; LOOK UP TABLE
FA04 23 ...3: INX H
FA05 23 ...2: INX H
FA06 23 INX H
FA07 7E MOV A,M
FA08 B7 ORA A ; END OF
FA09 CA F464 JZ ERROR ; TABLE
FA0C B8 CMP B ; 1ST?
FA0D 23 INX H ; NEXT ENTRY
FA0E 20F4 JRNZ ..3 ; NO
FA10 7E MOV A,M
FA11 B9 CMP C ; 2ED?
FA12 23 INX H
FA13 20F0 JRNZ ..2 ; NO
FA15 5E MOV E,M ; YES
FA16 23 INX H
FA17 56 MOV D,M
FA18 EB XCHG
FA19 E9 PCHL ; GO DOIT!

FA1A 0E2C GCHAR: MVI C,',' ; PROVIDE OWN
FA1C CD F009 CALL CO ; DELIMITER
FA1F CD F730 CALL KI ; GET COMMAND
FA22 E65F ANI 5FH ; MAKE UPPER CASE
FA24 4F MOV C,A ; ECHO
FA25 C3 F009 JMP CO

.DEFIN E JTBL[L12,ADDR] = [
 .WORD 'L12'>'8!''L12'<'8,ADDR]

TABLE: JTBL CO,CLOUT[
 .WORD 'CO'>81,'CO'<8,CLOUT]
JTBL CI,CLEN[
 .WORD 'CI'>81,'CI'<8,CLEN]
JTBL 00,OPOUT[
 .WORD '00'>81,'00'<8,OPOUT]
JTBL 0I,OPIN[
 .WORD '0I'>81,'0I'<8,OPIN]
JTBL ON,MOT.ON[
 .WORD 'ON'>81,'ON'<8,MOT.ON]
JTBL OF,MOT.OF[
 .WORD 'OF'>81,'OF'<8,MOT.OF]
 .BYTE 0
Extension routines for TDL "SYSTEM MONITOR BOARD".

FIXED BLOCK WRITE ROUTINE

FA41 E5 \_PUNCH: \_PUSH H \_SAVE REG'S
FA42 D5 \_PUSH D
FA43 C5 \_PUSH B
FA44 11 FCF8 \_LXI END \_END OF BUFFER
FA47 2A FC75 \_LHLD R \_POINTER
FA4A 71 \_MOV \_SAVE IT
FA4B CD F574 \_CALL \_STEP POINTER
FA4E 22 FC75 \_SHLD B R \_SAVE POINTER
FA51 DC FA59 \_F \_IF BUFFER FULL
FA54 C1 \_POP B \_RESTORE
FA55 D1 \_POP D \_REG'S
FA56 E1 \_POP B
FA57 79 \_MOV A,C
FA58 C9 \_RET

FA59 CD FB37 \_WRTBF: \_CALL SW10 \_START
FA5C 3E51 \_MVI A 51H \_UP
FA5E D370 \_OUT \_DRIVE 1
FA60 21 FC79 \_LXI \_H PBUFF \_START OF
FA63 22 FC75 \_SHLD P PBUFF \_BUFFER
FA66 CD F5A3 \_CALL \_LEAD
FA69 CD F5A3 \_CALL \_LEAD
FA6C CD F59E \_CALL \_MARK
FA6F 0E16 \_MVI C I \_SYNC
FA71 CD F00C \_CALL \_PO
FA74 4E \_MOV C M \_GET CHARACTER
FA75 CD F00C \_CALL \_PO \_TO CASSET
FA78 CD F574 \_CALL \_HILO \_TEST FOR END
FA7B 30F7 \_JRN C \_NOPE
FA7D 0E16 \_MVI C \_WRITE
FA7F CD F00C \_CALL \_PO \_END-MARK
FA82 CD F59E \_CALL \_MARK \_DON'T STOMP ON SYNC
FA85 C3 FB2A \_JMP \_STOP \_STOP TAPES

FA88 F5 \_NOSYN: \_CALL STOP \_SHUT DOWN
FA89 CD FB2A \_POP PSW
FA8C F1 \_JMP TRAP
FA8D C3 F01E \_ABORT: \_CALL STOP \_SHUT DOWN
FA90 CD FB2A \_JMP ERROR
FA93 C3 F464 \_RIFF: \_CALL RET \_GET READER CHARACTER
FA96 CD F006 \_JRC ABORT \_ABORT ON CARRY
FA99 38F5 \_C \_TEST D
FA9B BA \_RET
Extension routines for TDL "SYSTEM MONITOR BOARD".

FIXED BLOCK READ ROUTINE

`FA9D` `E5` READ: `PUSH H` ;SAVE REG'S
`FA9E` `D5` `PUSH D`
`FA9F` `C5` `PUSH B`
`FAA0` `11 FD79` ..RD1: `LXI D,RBEND+1 ;END OF BUFFER`
`FAA3` `2A FC77` LHLR DBPTR ;POINTER
`FAA6` `7E` MOV A,M ;GET CHARACTER
`FAA7` `F5` `PUSH PSW ;SAVE IT`
`FAA8` `CD F574` CALL HILO ;STEP POINTER
`FAAB` `22 FC77` SHLD RBPTR ;SAVE POINTER
`FAAE` `3006` JRNZ ..RD2 ;BUFFER OK
`FABA` `CD FABC` CALL RDBUF ;REFILL BUFFER
`FABB` `F1` POP PSW ;CLEAR GARbage
`FABB` `C9` RET

`FABC` `CD FB37` RDBUF: CALL SWIO ;START
`FABF` `3E50` MVI A,50H ;UP
`FAC1` `D374` OUT RCSS ;DRIVE 0
`FAC3` `21 FCF9` LXI H,RBUFF ;BUFFER START
`FAC6` `16FF` MVI D,OFFH ;START-OF-FILE
`FAC8` `0604` ..RD0: MVI B,4 ;FIND 4
`FACA` `CD FA96` ..RD1: CALL RIFF
`FACD` `20F9` JRNZ ..RD0
`FACF` `10F9` DJNZ ..RD1
`FAD1` `1616` MVI D,SYN ;FILE SYNC
`FAD3` `CD FA96` ..RD2: CALL RIFF ;WAIT FOR FILE
`FAD6` `20FB` JRNZ ..RD2
`FAD8` `11 FD78` LXI D,RBEND ;END OF BUFFER
`FADB` `CD F006` ..RD3: CALL RI ;GET CHARACTER
`FADE` `38B0` JRC ABORT ;MANUAL ABORT
`FAE0` `77` MOV M,A ;SAVE IT
`FAE1` `CD F574` CALL HILO ;TEST FOR END
`FAE4` `30F5` JRNZ ..RD3 ;NOPE
`FAE6` `CD F006` CALL RI ;GET END MARK
`FAE9` `FE16` CPI SYN ;CHECK FOR END
`FAEB` `209B` JRNZ NOSYN ;BLOCK ERROR
`FAED` `21 FCF9` LXI H,RBUFF ;RESET
`FAF0` `22 FC77` SHLD RBPTR ;POINTER
`FAF3` `1835` JMPR STOP ;STOP TAPES
### Extension routines for TDL "SYSTEM MONITOR BOARD".

**UTILITY AND SUPPORT ROUTINES**

<table>
<thead>
<tr>
<th>Address</th>
<th>Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAF5</td>
<td>21 FC79</td>
<td>OPUT: LXI H, PBUFF; RESET</td>
</tr>
<tr>
<td>FAF8</td>
<td>22 FC75</td>
<td>SHLD PBPTR; BUFFER</td>
</tr>
<tr>
<td>FAFB</td>
<td>C9</td>
<td>RET</td>
</tr>
<tr>
<td>FAFC</td>
<td>2A FC75</td>
<td>CLOUT: LHLD PBPTR; POINTER</td>
</tr>
<tr>
<td>FAFF</td>
<td>11 FCF8</td>
<td>LXI D, PBEND; END OF BUFFER</td>
</tr>
<tr>
<td>FB02</td>
<td>3600</td>
<td>..CL: MVI M, 0; CLEAR</td>
</tr>
<tr>
<td>FB04</td>
<td>CD F574</td>
<td>CALL HILO; REST OF</td>
</tr>
<tr>
<td>FB07</td>
<td>30F9</td>
<td>JRNC ..CL; BUFFER</td>
</tr>
<tr>
<td>FB09</td>
<td>C3 FA59</td>
<td>JMP WRTBF; AND OUTPUT</td>
</tr>
<tr>
<td>FB0C</td>
<td>CD FB11</td>
<td>OPIN: CALL CLEN; SET EMPTY</td>
</tr>
<tr>
<td>FB0F</td>
<td>18AB</td>
<td>JMPR RDBUF; FILL BUFFER</td>
</tr>
<tr>
<td>FB11</td>
<td>21 FCF9</td>
<td>CLEN: LXI H, RBUFF; RESET</td>
</tr>
<tr>
<td>FB14</td>
<td>22 FC77</td>
<td>SHLD RBPTR; TO FULL</td>
</tr>
<tr>
<td>FB17</td>
<td>E5</td>
<td>PUSH H; SAVE IT</td>
</tr>
<tr>
<td>FB18</td>
<td>361A</td>
<td>MVI M, 1AH; OZ (EOF)</td>
</tr>
<tr>
<td>FB1A</td>
<td>D1</td>
<td>POP D; OVERLAP</td>
</tr>
<tr>
<td>FB1B</td>
<td>13</td>
<td>INX D; MOVE</td>
</tr>
<tr>
<td>FB1C</td>
<td>01 007F</td>
<td>LXI B, BSIZE-1</td>
</tr>
<tr>
<td>FB1F</td>
<td>EDB0</td>
<td>LDIR; FILL BUFFER</td>
</tr>
<tr>
<td>FB21</td>
<td>C9</td>
<td>RET</td>
</tr>
<tr>
<td>FB22</td>
<td>3E50</td>
<td>MOT.ON: MVI A, 50H; START</td>
</tr>
<tr>
<td>FB24</td>
<td>D374</td>
<td>OUT RCSS; DRIVE 0 (PLAY)</td>
</tr>
<tr>
<td>FB26</td>
<td>3C</td>
<td>INR A; START</td>
</tr>
<tr>
<td>FB27</td>
<td>D370</td>
<td>OUT TTS; DRIVE 1 (RECORD)</td>
</tr>
<tr>
<td>FB29</td>
<td>C9</td>
<td>RET</td>
</tr>
<tr>
<td>FB2A</td>
<td>3A FC74</td>
<td>STOP: LDA IOSAV; RESTORE</td>
</tr>
<tr>
<td>FB2D</td>
<td>D376</td>
<td>OUT IOBYT; I/O BYTE</td>
</tr>
<tr>
<td>FB2F</td>
<td>3E10</td>
<td>MOT.OF: MVI A, 10H; STOP</td>
</tr>
<tr>
<td>FB31</td>
<td>D374</td>
<td>OUT RCSS; DRIVE 0 (PLAY)</td>
</tr>
<tr>
<td>FB33</td>
<td>3C</td>
<td>INR A; STOP</td>
</tr>
<tr>
<td>FB34</td>
<td>D370</td>
<td>OUT TTS; DRIVE 1 (RECORD)</td>
</tr>
<tr>
<td>FB36</td>
<td>C9</td>
<td>RET</td>
</tr>
<tr>
<td>FB37</td>
<td>DB76</td>
<td>SWIO: IN IOBYT; GET I/O BYTE</td>
</tr>
<tr>
<td>FB39</td>
<td>32 FC74</td>
<td>STA IOSAV; SAVE IT</td>
</tr>
<tr>
<td>FB3C</td>
<td>E6C3</td>
<td>ANI 0C3H; MASK READER/PUNCH</td>
</tr>
<tr>
<td>FB3E</td>
<td>F628</td>
<td>ORI 028H; SET TO</td>
</tr>
<tr>
<td>FB40</td>
<td>D376</td>
<td>OUT IOBYT; CASSETTE</td>
</tr>
<tr>
<td>FB42</td>
<td>C9</td>
<td>RET</td>
</tr>
</tbody>
</table>

---
; WRITTEN BY JOHN MONTAGNA
;
; THIS ROUTINE WILL PROGRAM A BYTE SAVER
; ADDRESSED AT LOCATION 'BASE'. A PROM
; NUMBER (0-7) IS REQUIRED ALONG WITH A DATA
; PICK UP ADDRESS. THIS ROUTINE ASSUMES 1K ROMS.
;
; *NOTE: ANY UNUSED PORTION OF THE 1K BLOCK SHOULD
; BE FILLED WITH 'FF' USING THE FILL COMMAND.

; EXTENDED ROUTINES FOR TDL "SYSTEM MONITOR BOARD"
;
BYTE MOVER ROUTINE FOR ZAPPLE

; PGM: LXI H, MSG1
;      MVI B, MSG1L
;      CALL TOM
;      CALL Expr ; GET BASE OF BYTESAVER
;      POP H ; SAVE IN HL'
; MORE: LXI H, MSG2
;      MVI B, MSG2L
;      CALL TOM
;      CALL PCHK ; GET KEYBOARD
;      RC ; CR ENTERED, QUIT
;      SUI '0' ; LOOKING FOR A NUMBER
;      JRC .. ER ; TOO SMALL
;      MOV D, A ; SAVE IN D
;      ANI #7H ; BETWEEN 0 & 7
;      PCHK ; GET DELIMITER
;      JNZ ERROR ; NOT DELIMITER, ABORT
;      CALL Expr ; GET THE ADDRESS
;      EXX ; GET PROM BOARD START
;      PUSH H
;      EXX
;      POP H
;      LXI B, 400H ; ONE THOUSAND
;      INR D
;      DCR D ; TEST FOR ZERO
;      JRZ SKIP ; TEST FOR FINISHED
;      JNZ ERROR
;      DAD B ; ADD # OF K'S
;      DCR D ; DONE ONE MORE
;      JRNZ .. LP
;      JPUSH H ; SAVE PUT DOWN ADDR.
;      LXI H, MSG0 ; POINT TO MESSAGE
;      MVI B, MSG0L ; # OF CHARS
;      CALL TOM ; SEND MESSAGE
;      CALL WAIT
;      CALL CRLF
;      POP H ; PUT DOWN ADDRESS
;      POP D ; PICK UP ADDRESS
;      MVI A, 50 ; DO THIS 50 TIMES
;      CALL PSH PSW ; SAVE COUNT
;      PUSH D ; SAVE PICK UP
;      PUSH H ; SAVE PUT DOWN
;      LD X Y, 400H ; 1K MUST MATCH
;      LXi B, 400H ; HOW MANY LOCATIONS
Extension routines for TDL "SYSTEM MONITOR BOARD".

BYTE MOVER ROUTINE FOR ZAPPLE

FB98 1A  LOOP3:  LDAX  D  ;PICKUP
FB99 77  MOV  M,A  ;PUT DOWN
FB9A BE  CMP  M  ;VERIFY PUT DOWN
FB9B 2002  JRNZ  SKIP2  ;NO GOOD?
FB9D FD2B  DCX  Y  ;YES GOOD
FB9F 13  SKIP2:  INX  D
FBAB 23  INX  H
FBAD OB  DCX  B
FBAA 78  MOV  A,B  ;TEST FOR FINISHED
FBAC B1  ORA  C  ;WITH THIS PASS
FBAD 20F2  JRNZ  LOOP3
FBA6 E1  POP  H
FBA7 D1  POP  D
FBA8 F1  POP  PSW
FBA9 3D  DCR  A  ;DONE ONE MORE PASS
FBAE FD2B  DCX  Y  ;TEST FOR 100%
FBAF FD29  DADY  Y  ;MATCH ON THIS PASS
FBBD 20DE  JRNZ  LOOP2  ;DO ANOTHER PASS
FBBO 21 FBF7  LXI  H,ERMSG
FBB3 0609  MVI  B,ERMSGL
FBB5 D4 F452  CNC  TOM  ;CALL IF ERROR
FBB8 21 FBEA  LXI  H,MSG3
FBBB 060A  MVI  B,MSG3L
FBBE CD F452  CALL  TOM
FBC0 CD F8F4  CALL  WAIT  ;WAIT FOR CR
FBC3 188B  JMPR  MORE

FBC5 0DOA  MSG0:  .BYTE  CR,LF
FBC7 5357204F4E07  .ASCII  #SW ON-
0009 SW - ON  MSG0L = .MSG0
FBCE 0DOA  MSG1:  .BYTE  CR,LF
FBD0 424153452041  .ASCII  #BASE ADDR:
000C BASE - A  MSG1L = .MSG1
FBDA 0DOA  MSG2:  .BYTE  CR,LF
FBD8 4E4D42522026  .ASCII  "NMBR & DATA ADDR:"
0013 NMBR - &  MSG2L = .MSG2
FBED 0DOA  MSG3:  .BYTE  CR,LF
FBEF 5357204F4646  .ASCII  #SW OFF-
000A SW - OFF  MSG3L = .MSG3
FBB7 0DOA  ERMSG:  .BYTE  CR,LF
FBB9 2A42414422A  .ASCII  #*BAD*
FBBE 0DOA  .BYTE  CR,LF
0009 ERMSGL = .ERMSG
Extension routines for TDL "SYSTEM MONITOR BOARD".
READ/ WRITE (RAM) STORAGE DEFINITIONS

+++ DATA STORAGE (RAM) SECTION +++

FC00
.LOC ROM+400H

FC00
CIULOC: .BLKB 3 ;*
FC03
COULOC: .BLKB 3 ;*
FC06
RIULOC: .BLKB 3 ;*
FC09
RIULOC: .BLKB 3 ;USED BY BUFFERED CASES
FC0C
POULOC: .BLKB 3 ;*
FC0F
POULOC: .BLKB 3 ;USED BY BUFFERED CASES
FC12
LULOC: .BLKB 3 ;*
FC15
LULOC: .BLKB 3 ;PERMANENTLY ASSIGNED TO VDB
FC18
CSULOC: .BLKB 3 ;*
FC1B
IUCMND: .BLKB 3 ;USER DEFINED "I" COMMAND
FC1E
KUCMND: .BLKB 3 ;USER DEFINED "K" COMMAND
FC21
OUCMND: .BLKB 3 ;USER DEFINED "O" COMMAND

FC24
BUFF: .BLKB 80 ;VDB BUFFER STORAGE

+++ BUFFERED CASES STORAGE AREAS +++

FC74
IOSAV: .BLKB 1 ;I/O BYTE STORAGE
FC75
PBPT: .BLKB 2 ;PUNCH BUFFER POINTER
FC77
RBPT: .BLKB 2 ;READ BUFFER POINTER

FC79
PBPT: .BLKB BSIZE ;PUNCH BUFFER
FCF8
PBPT: == PBPT+BSIZE-1

FCF9
RBPT: .BLKB BSIZE ;READ BUFFER
FD78
RBPT: == RBPT+BSIZE-1

.END
Extension routines for TDL "SYSTEM MONITOR BOARD".

<table>
<thead>
<tr>
<th>ABORT</th>
<th>ABRT</th>
<th>ADISP</th>
<th>ADISP</th>
<th>ADISP</th>
</tr>
</thead>
<tbody>
<tr>
<td>B5</td>
<td>0008</td>
<td>BSIZE</td>
<td>0080</td>
<td>BSIZE</td>
</tr>
<tr>
<td>CIE</td>
<td>F824</td>
<td>CIULOC</td>
<td>F000</td>
<td>CIULOC</td>
</tr>
<tr>
<td>CLOUT</td>
<td>FAFC</td>
<td>CNTL</td>
<td>F968</td>
<td>CNTL</td>
</tr>
<tr>
<td>COULOC</td>
<td>FC03</td>
<td>CR</td>
<td>000D</td>
<td>CR</td>
</tr>
<tr>
<td>CSULOC</td>
<td>FC18</td>
<td>DC1</td>
<td>0011</td>
<td>DC1</td>
</tr>
<tr>
<td>DC4</td>
<td>0014</td>
<td>DEL</td>
<td>007F</td>
<td>DEL</td>
</tr>
<tr>
<td>ERROR</td>
<td>F464</td>
<td>EXPR</td>
<td>F540</td>
<td>EXPR</td>
</tr>
<tr>
<td>GTMD</td>
<td>F90E</td>
<td>HILO</td>
<td>F574</td>
<td>HILO</td>
</tr>
<tr>
<td>IOBYT</td>
<td>0076</td>
<td>IOSAV</td>
<td>FC74</td>
<td>IOSAV</td>
</tr>
<tr>
<td>KBSTS</td>
<td>F9ED</td>
<td>KI</td>
<td>F730</td>
<td>KI</td>
</tr>
<tr>
<td>LEAD</td>
<td>F5A3</td>
<td>LF</td>
<td>000A</td>
<td>LF</td>
</tr>
<tr>
<td>LOOP2</td>
<td>F88E</td>
<td>LOOP3</td>
<td>F988</td>
<td>LOOP3</td>
</tr>
<tr>
<td>LTBL</td>
<td>F794</td>
<td>MAIN</td>
<td>F912</td>
<td>MAIN</td>
</tr>
<tr>
<td>MORE</td>
<td>F850</td>
<td>MOT.0F</td>
<td>B2F2</td>
<td>MOT.0F</td>
</tr>
<tr>
<td>MSGOL</td>
<td>0009</td>
<td>MSG1</td>
<td>FBCE</td>
<td>MSG1</td>
</tr>
<tr>
<td>MSG2L</td>
<td>0013</td>
<td>MSG3</td>
<td>FBED</td>
<td>MSG3</td>
</tr>
<tr>
<td>OFF</td>
<td>FFFF</td>
<td>OPIN</td>
<td>FB0C</td>
<td>OPIN</td>
</tr>
<tr>
<td>PBEND</td>
<td>FCG8</td>
<td>PBPTR</td>
<td>FC75</td>
<td>PBPTR</td>
</tr>
<tr>
<td>PGM</td>
<td>F843</td>
<td>PO</td>
<td>F00C</td>
<td>PO</td>
</tr>
<tr>
<td>PUNCH</td>
<td>FA41</td>
<td>RBEND</td>
<td>FD78</td>
<td>RBEND</td>
</tr>
<tr>
<td>RCSS</td>
<td>0074</td>
<td>RDBUF</td>
<td>FABC</td>
<td>RDBUF</td>
</tr>
<tr>
<td>RIFF</td>
<td>FA96</td>
<td>RIPLOC</td>
<td>FC06</td>
<td>RIPLOC</td>
</tr>
<tr>
<td>SCROL</td>
<td>F935</td>
<td>SKIP</td>
<td>FB7B</td>
<td>SKIP</td>
</tr>
<tr>
<td>START</td>
<td>F000</td>
<td>STOP</td>
<td>FB2A</td>
<td>STOP</td>
</tr>
<tr>
<td>TABLE</td>
<td>FA28</td>
<td>TEST</td>
<td>F83F</td>
<td>TEST</td>
</tr>
<tr>
<td>UTILITY</td>
<td>F9FA</td>
<td>VDB</td>
<td>F900</td>
<td>VDB</td>
</tr>
<tr>
<td>VDBGK</td>
<td>0060</td>
<td>VDBK. D</td>
<td>009F</td>
<td>VDBK. D</td>
</tr>
<tr>
<td>VDBMRR</td>
<td>00E2</td>
<td>VDBRES</td>
<td>00E3</td>
<td>VDBRES</td>
</tr>
<tr>
<td>VDBYCR</td>
<td>00E1</td>
<td>VDBXAP</td>
<td>0000</td>
<td>VDBXAP</td>
</tr>
<tr>
<td>VDB2DD</td>
<td>0005</td>
<td>VDBZID</td>
<td>0003</td>
<td>VDBZID</td>
</tr>
<tr>
<td>WRTBF</td>
<td>FA59</td>
<td>ZAPPLE</td>
<td>F000</td>
<td>ZAPPLE</td>
</tr>
</tbody>
</table>

```
<table>
<thead>
<tr>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>SN74LS00 QUAD 2-NAND U34,U36</td>
</tr>
<tr>
<td>1</td>
<td>SN74LS04 HEX inverter U28</td>
</tr>
<tr>
<td>1</td>
<td>SN74LS08 QUAD 2-AND U30</td>
</tr>
<tr>
<td>1</td>
<td>SN74LS32 QUAD 2-OR U29</td>
</tr>
<tr>
<td>1</td>
<td>SN74LS33 QUAD 2-NOR OC buffer U27</td>
</tr>
<tr>
<td>1</td>
<td>SN74LS74 DUAL D-flip flop U35</td>
</tr>
<tr>
<td>1</td>
<td>SN74LS86 QUAD 2-XOR U38</td>
</tr>
<tr>
<td>1</td>
<td>SN74LS175 QUAD D-flip flop U33</td>
</tr>
<tr>
<td>4</td>
<td>SN74LS244 OCTAL bus driver U13,U14,U19,U20</td>
</tr>
<tr>
<td>2</td>
<td>SN74LS258 QUAD 2-MPX 3-5 INV U4, U5</td>
</tr>
<tr>
<td>1</td>
<td>TIL 113 Opto-coupler U31</td>
</tr>
<tr>
<td>1</td>
<td>LM 339 QUAD Analog comparator U37</td>
</tr>
<tr>
<td>1</td>
<td>SN75188 (1488) QUAD RS-232 driver U25</td>
</tr>
<tr>
<td>1</td>
<td>SN75189 (1489) QUAD RS-232 Receiver U26</td>
</tr>
<tr>
<td>1</td>
<td>Motorola 14411 Baud rate generator U17</td>
</tr>
<tr>
<td>1</td>
<td>Motorola MC6820 PIA (parallel) U18</td>
</tr>
<tr>
<td>3</td>
<td>Motorola MC6850 ACIA (serial) U10-U12</td>
</tr>
<tr>
<td>1</td>
<td>Mostek MK34038N Mask ROM U1</td>
</tr>
<tr>
<td></td>
<td>(or c28050M)</td>
</tr>
<tr>
<td>1</td>
<td>INTEL 2708 UV erasable 1024x8 ROM U2</td>
</tr>
<tr>
<td>2</td>
<td>INTEL 2114 Static 1024x4 RAM U6,U8</td>
</tr>
<tr>
<td></td>
<td>(or AMD/INTERSIL/MOTOROLA 6614)</td>
</tr>
<tr>
<td>1</td>
<td>4047 CMOS Multivibrator U39</td>
</tr>
<tr>
<td>1</td>
<td>SN74LS30 8-NAND U21</td>
</tr>
<tr>
<td>5</td>
<td>SN74LS138 Decoder 3 line-8 line U15,U16,U22,U23,U32</td>
</tr>
<tr>
<td>1</td>
<td>SN74LS139 Dual decoder 2 line-4 line U24</td>
</tr>
<tr>
<td></td>
<td>Resistor SIP 10 pin 9 resistor 4.7K ohm RN1-RN4</td>
</tr>
<tr>
<td>2</td>
<td>(Signal) diodes CR1,CR2</td>
</tr>
<tr>
<td>2</td>
<td>Resistor 24.9K 1% R4,R5</td>
</tr>
<tr>
<td>1</td>
<td>Resistor 100 ohm R1</td>
</tr>
<tr>
<td>1</td>
<td>Resistor 820 ohm R17</td>
</tr>
<tr>
<td>5</td>
<td>Resistor 1K R2,R9,R12,R15,R20</td>
</tr>
<tr>
<td>1</td>
<td>Resistor 1.2K R16</td>
</tr>
<tr>
<td>1</td>
<td>Resistor 2.2K R18</td>
</tr>
<tr>
<td>2</td>
<td>Resistor 33K R10,R19</td>
</tr>
<tr>
<td>1</td>
<td>Resistor 10K R3</td>
</tr>
<tr>
<td>1</td>
<td>Resistor 22K R8</td>
</tr>
<tr>
<td>1</td>
<td>Resistor 100K R11</td>
</tr>
<tr>
<td>1</td>
<td>Resistor 15M ohm R21</td>
</tr>
<tr>
<td>2</td>
<td>Resistor 4.7K ohm R13,R14</td>
</tr>
<tr>
<td>1</td>
<td>Resistor 10 ohm R6</td>
</tr>
<tr>
<td>1</td>
<td>Resistor 22 ohm R7</td>
</tr>
<tr>
<td>1</td>
<td>Capacitor 0.01uf 2% C1</td>
</tr>
<tr>
<td>1</td>
<td>Capacitor 33uf C5</td>
</tr>
<tr>
<td>9</td>
<td>Capacitor 4.7uf C3,C4,C6-C10,C24,C26</td>
</tr>
<tr>
<td>17</td>
<td>Capacitor 0.1uf C2,C11-C23,C25,C27,C28</td>
</tr>
<tr>
<td>Quantity</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>uA7805C +5v Regulator 1.5 amp</td>
</tr>
<tr>
<td>1</td>
<td>uA7812C +12v Regulator</td>
</tr>
<tr>
<td>1</td>
<td>uA7905C -5v Regulator</td>
</tr>
<tr>
<td>1</td>
<td>uA7912C -12v Regulator</td>
</tr>
<tr>
<td>3</td>
<td>Dip switches 8-position SW1, SW2, SW3</td>
</tr>
<tr>
<td>108</td>
<td>Augat pins</td>
</tr>
<tr>
<td>1</td>
<td>Heatsink Wakefield 680-.5-220</td>
</tr>
</tbody>
</table>
When using ZOMA connect the circuit between -12 volts and the specified pin on the ribbon cable.

When not using ZOMA connect the keyboard (ZOMA In) circuit to -12 volts.

Select Monitor Drive
20 MA OUT

20 MA IN

Rear View

SMB connector

Copyright 1977 by CDC
Computer Design Labs