THE TARBEll CASSETTE INTERFACE
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Specifications</td>
</tr>
<tr>
<td>2. Interface History and Sales Pitch</td>
</tr>
<tr>
<td>3. Selecting a Cassette Unit for Digital Recording</td>
</tr>
<tr>
<td>4. The Kansas City (Byte/Lancaster/CUTS) Format</td>
</tr>
<tr>
<td>5. Parts List</td>
</tr>
<tr>
<td>6. Assembly Instructions</td>
</tr>
<tr>
<td>7. Assembly Drawings</td>
</tr>
<tr>
<td>8. Modifications</td>
</tr>
<tr>
<td>9. Soldering, Cleaning, and Installation Notes</td>
</tr>
<tr>
<td>10. Initial Adjustment Instructions</td>
</tr>
<tr>
<td>11. Operating Instructions</td>
</tr>
<tr>
<td>12. Bootstrap and Sync Code Generator Programs</td>
</tr>
<tr>
<td>13. Output Routine With Checksum</td>
</tr>
<tr>
<td>14. Input Routine With Checksum</td>
</tr>
<tr>
<td>15. How to Save and Load Data From a BASIC Program</td>
</tr>
<tr>
<td>16. Controlling the Start-Stop Function</td>
</tr>
<tr>
<td>17. Theory of Operation - Output Section</td>
</tr>
<tr>
<td>18. Theory of Operation - Input Section</td>
</tr>
<tr>
<td>19. Schematic - Output Section</td>
</tr>
<tr>
<td>20. Schematic - Input Section</td>
</tr>
<tr>
<td>21. Timing Diagram</td>
</tr>
<tr>
<td>22. Pin Function List</td>
</tr>
<tr>
<td>23. If You Have Problems</td>
</tr>
<tr>
<td>24. Ideas For Using the Cassette Interface</td>
</tr>
<tr>
<td>25. Modifications on Cassette BASIC CSAVE and CLOAD</td>
</tr>
<tr>
<td>26. Processor Technology Software Package #1 Modified</td>
</tr>
<tr>
<td>27. Writing Programs For the Cassette Interface</td>
</tr>
<tr>
<td>28. Interrupt Control</td>
</tr>
<tr>
<td>29. PHI-DECK Adapter Information</td>
</tr>
<tr>
<td>30. 74L00, 7403, 74L11, 7404, 74L04, 7406, 7425, 7408, 74L30</td>
</tr>
<tr>
<td>31. 74L75, 74L73, 74L74</td>
</tr>
<tr>
<td>32. 74L75, 74L86</td>
</tr>
<tr>
<td>33. 7496, DM131</td>
</tr>
<tr>
<td>34. 74L164, NE555</td>
</tr>
<tr>
<td>35. 8720</td>
</tr>
<tr>
<td>36. Customer Evaluation and Registration</td>
</tr>
</tbody>
</table>

January 14, 1977
THE TARBEI ELECTRONICS
20520 S. Leapwood Ave., Suite P
Carson, California 90746
(213) 538-4251

THE TARBEI CASSETTE INTERFACE (FOR ALTAIR 8800 USERS)

SPEED: UP TO 540 BYTES PER SECOND (2200 BITS PER INCH).
187 BYTES PER SECOND FOR TARBEI STANDARD 800 BITS/INCH.
30 BYTES PER SECOND FOR "BYTE/LANCaster" STANDARD.

ENCODING METHOD: PHASE-ENCODED (EXCLUSIVE-OR OF CLOCK AND DATA).
SELF-CLOCKING (CLOCK VARIES ALONG WITH TAPE SPEED).
USED ON MY OWN SYSTEM FOR THE LAST 4 YEARS.
CAN BE USED TO GENERATE "BYTE/LANCaster" TAPES.

CASSette: WILL WORK WITH MOST AUDIO CASSETTE UNITS. MAY BE ADAPTED
TO AUTOMATIC DIGITAL CASSETTE UNITS. WILL ALSO WORK
WITH REGULAR REEL-REEL TAPE RECORDERS. I HAVE BEEN USING
A REALISTIC (RADIO SHACK) CTR-19 AND A J.C. PENN 6536
($39.95). TAPE SHOULD BE OF LOW-NOISE TYPE.

8192 BYTE LOAD TIME: 15 SEC @ 540 BYTES PER SECOND.
43 SEC @ 187 BYTES PER SECOND.
4 MINUTES @ 30 BYTES PER SECOND.

DEVICE-CODE EASILY SELECTED WITH ON-BOARD DIP-SWITCH.

STATUS: 4 EXTRA STATUS LINES AVAILABLE FOR INPUT.

CONTROL: 4 EXTRA CONTROL LINES AVAILABLE FOR OUTPUT, WHICH MAY
BE USED TO DRIVE RELAYS FOR EXTRA CASSETTE UNITS.
2 SPARE IC SLOTS TO LET YOU DO YOUR OWN THING.

COMPATIBILITY: PLUGS DIRECTLY INTO ALTAIR 8800 OR IMSAI 8080.
HAS SERIAL-PARALLEL AND PARALLEL-SERIAL CONVERSION
ON BOARD. PATCHES PROVIDED FOR POPULAR SOFTWARE.

SOFTWARE: COMES WITH COMPLETE SET OF INPUT/OUTPUT SUBROUTINES,
BOOTSTRAP, AND "BYTE STANDARD" (LANCASTER) SOFTWARE.

COST: $120 FOR COMPLETE KIT, $175 ASSEMBLED AND CHECKED-OUT.

MANUAL:
ASSEMBLY INSTRUCTIONS AND DRAWING, PARTS AND PIN FUNCTION LISTS
SOLDERING, CLEANING, AND INSTALLATION NOTES; OPERATING INSTRUCTIONS
INITIAL ADJUSTMENT PROCEDURES, INPUT/OUTPUT ROUTINES WITH CHECKS
BOOTSTRAP PROGRAM AND TEST-STREAM GENERATOR PROGRAM

PARTS:
ALL RESISTORS, CAPACITORS, AND INTEGRATED CIRCUITS
CASSETTE CABLE, RIBBON CABLE, AND DIP CONNECTOR
LOW-NOISE CASSETTE WITH TEST STREAM
DOUBLE-SIDED BOARD WITH PLATED-THRU HOLES AND GOLD EDGE CONNECTOR

WARRANTY: IF NOT COMPLETELY SATISFIED, RETURN BOARD FOR REFUND
OR FREE REPAIR WITHIN 90 DAYS AFTER PURCHASE.

FIRST DELIVERIES WERE MADE IN SEPTEMBER, 1975. DELIVERY IS
1 TO 3 WEEKS AFTER RECEIVING ORDER. THE 25-PAGE MANUAL IS AVAILABLE
AT $4. CALIFORNIA RESIDENTS PLEASE ADD 6% SALES TAX. MAKE
CHECK OR MONEY ORDER PAYABLE TO TARBEI ELECTRONICS.
HISTORY AND SALES PITCH

I HAVE BEEN USING AN INEXPENSIVE AUDIO CASSETTE RECORDER IN MY HOME-DESIGNED COMPUTER SYSTEM SINCE 1972. I HAVE OVER 600 FILES ON CASSETTES, MOSTLY ABOUT 4 KBYTES EACH. MY ESTIMATE IS THAT THE ERROR RATE IS LESS THAN 1 ERROR IN 1,000,000 BITS. I SAY THIS BECAUSE I CAN USUALLY RECORD 30 4 KBYTE FILES ON ONE SIDE OF A C-60 CASSETTE WITHOUT ANY ERRORS. THIS INTERFACE GAVE ME VERY GOOD SERVICE WHILE I WAS WRITING THE DISK OPERATING SYSTEM FOR MY 500 KBYTE DISK. SINCE I STARTED USING MY DISK SYSTEM A FEW YEARS AGO, THE CASSETTE HAS SERVED AS BACKUP STORAGE - A RELIABLE PLACE TO STORE DATA AND PROGRAMS AFTER THEY ARE DEBUGGED.

THE ENCODING METHOD I USE IS VERY SIMPLE, AND HAS BEEN IN USE IN INDUSTRY FOR QUITE SOME TIME. PICTURE A SHIFT REGISTER WHICH IS LOADED WITH THE DATA TO BE RECORDED. THE REGISTER IS THEN CLOCKED WITH A SQUARE WAVE. THE OUTPUT OF THE SHIFT REGISTER IS EXCLUSIVE-ORED WITH THE CLOCK, PRODUCING THE BI-PHASE DATA. THIS DATA GOES DIRECTLY TO THE CASSETTE RECORDER'S INPUT. THE MOST DIFFICULT PART OF THE PROCESS IS RECOVERING THE DATA. MANY LONG HOURS WERE SPENT STUDYING THIS PROBLEM AND TRYING DIFFERENT METHODS. IN THE ORIGINAL INTERFACE THIS WAS ACCOMPLISHED WITH A 760 HIGH-SPEED COMPARATOR, A 74121 NON-RETRIGGERABLE ONE-SHOT, AND A DOUBLE-GLITCH GENERATOR MADE WITH AN EXCLUSIVE-OR GATE. SINCE THEN, THE 8720 HAS BEEN DEVELOPED, WHICH COMBINES THESE THREE FUNCTIONS ON A SINGLE CHIP, AND IS THE UNIT USED IN THE ALTAIR INTERFACE.

THE PRESENT DESIGN IS EVEN MORE RELIABLE THAN THE PREVIOUS ONE, AND IS CAPABLE OF RECORDING AND RECOVERING ERROR-FREE DATA AT A RATE OF 540 BYTES PER SECOND ON A STANDARD AUDIO CASSETTE RECORDER. (YES, THAT IS OVER 2200 BITS PER INCH!) I AM STILL, HOWEVER, ENCOURAGING USERS TO EXCHANGE DATA RECORDED AT 187 BYTES PER SECOND (1500 BITS PER SECOND, 800 BITS PER INCH). THE MAIN ADVANTAGE OF THIS METHOD OVER OTHERS IS IT'S ABILITY TO WITHSTAND A LARGE AMOUNT OF WOW AND FLUTTER, WHICH MAY BE INTRODUCED BY CHEAP RECORDERs, AND STILL RECOVER THE DATA RELIABLY. THIS FEATURE STEMS FROM THE SELF-CLOCKING NATURE OF THE RECORDED SIGNAL: THE RECOVERED CLOCK VARIES RIGHT ALONG WITH THE DATA, SO THAT THE SPEED VARIATIONS ARE ESSENTIALLY IGNORED. THE MAIN DISADVANTAGE OF THIS METHOD IS THAT IT REQUIRES GOOD LOW-NOISE TAPE, AND A DECENT FREQUENCY RESPONSE ON THE CASSETTE UNIT. THE CASSETTE UNIT I'VE BEEN USING LATELY (J.C. PENNY #6536) HAS A FREQUENCY RESPONSE 80-8,000 HZ. THE MOST IMPORTANT PART IS THE HIGH END. THESE REQUIREMENTS ARE DUE TO THE HIGH SPEED OF THE INTERFACE, AND WOULD BE THE SAME FOR ANY HIGH SPEED DEVICE.

THE SPEED MAY NOT SEEM VERY IMPORTANT TO YOU NOW, BUT A GOOD PORTION OF YOUR TIME IS GOING TO BE SPENT SAVING AND LOADING DATA, PROGRAMs, AND OTHER TEXT. THERE IS A WORLD OF DIFFERENCE BETWEEN LOADING BASIC AT SAY, 30 BYTES/SEC (4 MINUTES), AND AT 187 BYTES/SEC (40 SEC). IT DOESN'T SEEM LIKE MUCH, BUT WHEN YOU HAVE TO DO IT OVER, AND OVER, AND OVER... IT GETS TO BE A BIT MUCH. ESPECIALLY WHEN YOU'RE DEVELOPING YOUR OWN PROGRAMS, AND THEY TEND TO RUN AMUCK AND WIPE OUT CORE. THINK ABOUT IT, THEN BUY THE TARBELL CASSETTE INTERFACE. THE ONLY METHOD PROVEN WITH TIME. ASK YOUR FRIEND WHO HAS ONE.
SELECTING A CASSETTE UNIT FOR DIGITAL RECORDING

FIRST OF ALL, THE MOST EXPENSIVE CASSETTE RECORDERS ARE NOT NECESSARILY THE BEST FOR RECORDING DIGITAL DATA. THERE ARE SEVERAL FACTORS THAT COMBINE TO MAKE A GOOD UNIT FOR THE HOBBYIST:

1. IT SHOULD HAVE A GOOD HIGH-FREQUENCY RESPONSE, PREFERABLY UP TO AT LEAST 8,000 HZ.

2. IT SHOULD HAVE A TONE CONTROL, SO THAT THE INHERENT FREQUENCY RESPONSE MAY BE REALIZED.

3. ALTHOUGH AUTOMATIC VOLUME CONTROL IS MORE CONVENIENT FROM AN OPERATIONAL POINT OF VIEW, IT ALSO REQUIRES A FEW SECONDS OF SETTLING TIME BEFORE STARTING TO RECORD.

4. IF IT DOES NOT HAVE AUTOMATIC VOLUME CONTROL, IT IS GOOD TO HAVE A RECORDING LEVEL METER. THIS ALLOWS EASIER ADJUSTMENT FOR THE CORRECT RECORDING LEVEL.

5. IT IS VERY IMPORTANT TO HAVE A DIGITAL COUNTER. THIS MAKES IT POSSIBLE TO QUICKLY LOCATE THE DESIRED PROGRAM AMONG SEVERAL.

6. IT SHOULD BE CAPABLE OF RUNNING DIRECTLY ON THE AC LINE. BATTERIES TEND TO MAKE THE MOTOR GET SLOWER AS THEY WEAR.

7. IT IS HANDY TO HAVE AN AUXILIARY INPUT, SO THAT A FAIRLY HIGH LEVEL MAY BE FED TO THE RECORDER, AND NOISE KEPT TO A MINIMUM.

8. A REMOTE INPUT JACK IS VALUABLE TO CONTROL START-STOP DURING ASSEMBLER AND COMPILER OPERATIONS.

9. JACKS FOR MIC, AUX, REMOTE, AND EARPHONE ARE USUALLY INCLUDED, BUT YOU SHOULD CHECK TO MAKE SURE THEY ARE THERE ANYWAY.

10. LOW WOW AND FLUTTER CHARACTERISTICS ARE IMPORTANT, BUT ARE MUCH MORE IMPORTANT WHEN USING AN ASYNCHRONOUS INTERFACE, WHICH IS NOT SELF-CLOCKING.

11. IF IT IS DESIRED TO DO AUTOMATIC REWIND, FAST-FORWARD, AND RECORD/PLAYBACK SWITCHING UNDER PROGRAM CONTROL, YOU MUST PURCHASE A RECORDER THAT HAS THESE FACILITIES. ANOTHER FEATURE TO LOOK FOR ON THIS TYPE OF UNIT IS A WAY TO KEEP TRACK OF WHERE YOU ARE ON THE CASSETTE TAPE.

12. PEOPLE HAVE EXPERIENCED PROBLEMS WITH PANASONIC RECORDERS. I AM PRESENTLY RECOMMENDING THE J.C. PENNEY MODEL 6536 AT $39.95 AS THE BEST BUY I KNOW, AND SEVERAL PEOPLE ARE USING THEM WITH MY INTERFACE WITH GOOD RESULTS. OTHER BRANDS THAT I KNOW HAVE BEEN SUCCESSFUL ARE SONY AND REALISTIC.

OF COURSE, NONE OF THE ABOVE ITEMS IS ABSOLUTELY NECESSARY FOR RECORDING DIGITAL DATA ON AN AUDIO CASSETTE. BUT THE MORE OF THESE REQUIREMENTS THAT ARE FILLED, THE MORE CAPABLE YOUR UNIT WILL BE, AND THE EASIER IT WILL BE TO USE.
USING THE TARBELL CASSETTE INTERFACE FOR THE KANSAS CITY FORMAT

Some time ago, there was a meeting of various cassette interface manufacturers to determine a standard for exchange of programs and data on cassettes among computer hobbyists. The format that was proposed as a result of the meeting is a modified version of the coding technique described by Don Lancaster in the first issue of "BYTE" Magazine. In this format, each 8-bit byte is written on tape in an asynchronous format, with one start bit (zero), 8 data bits (zero or one), and two stop bits (ones). A one is defined as 8 cycles at 2400 bits per second, and a zero is defined as 4 cycles at 1200 bits per second. This provides a data transfer speed of 300 baud, or a little less than 30 bytes per second, and may be generated and decoded using a variety of techniques.

Since the standard is fairly slow, it suggests that many people may want to have two methods available. One that provides for the Kansas City (BYTE/Lancaster) format, and another that is much faster, to speed program loading and development. The Tarbell Cassette Interface may easily be modified for both methods.

First, the output oscillator frequency will have to be raised from 3000 hz to 4800 hz. This is because a higher bit density is required of the tape, although the actual data transfer rate is much slower. A one may be generated by writing a word of all zeroes (00000000), and a zero may be generated by writing a word of alternating ones and zeroes (01010101). An output subroutine converts each byte to be written in this format from parallel to serial form (required only for this format).

On the input side, the adjustment of the potentiometer (R8) will have to be changed for the higher frequency. The sync detector circuit (IC's 9 and 10) will have to be changed so that it recognizes the alternating bit pattern as a sync byte in addition to the normal sync byte of E6 (hex). An input subroutine converts each byte from its serial form to its parallel form (required only for this format).

Using the method outlined above, the Tarbell Cassette Interface can be modified so that a double-pole, single-throw switch will determine which frequency will be used. The software determines the format. Another alternative is to change to the higher frequency permanently, so that no switch is necessary. The disadvantage of this is that you would have to readjust the potentiometer to read tapes made with the standard 3000 hz oscillator (187 bytes per second), and that a slightly higher frequency response is required on the part of your recorder.

A description of the hardware modifications, and listings of the subroutines for the operations described above are available upon request. Please include a self-addressed stamped envelope.
# Cassette Interface Parts List

## Integrated Circuits

<table>
<thead>
<tr>
<th>REF NO.</th>
<th>Description</th>
<th>QTY</th>
<th>PT NO.</th>
<th>+5</th>
<th>GND</th>
<th>-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Quad 2-input AND</td>
<td>1</td>
<td>7406</td>
<td>4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Dual J-K Flip-Flop</td>
<td>1</td>
<td>7473</td>
<td>4</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8-bit shift register</td>
<td>1</td>
<td>74L164</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>4, 23</td>
<td>Dual type-D flip-flop</td>
<td>2</td>
<td>74L74</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>5-volt regulator</td>
<td>1</td>
<td>LN308K</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7, 13, 19, 25</td>
<td>Quad 2-input NAND 0/C</td>
<td>4</td>
<td>7403</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6, 29</td>
<td>Dual J-K Flip-Flop</td>
<td>2</td>
<td>74L73</td>
<td>4</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Bi-directional one-shot</td>
<td>1</td>
<td>CT20</td>
<td>16</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>9, 31</td>
<td>Hex inverter</td>
<td>2</td>
<td>74L04</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>6-input NAND</td>
<td>1</td>
<td>74L30</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Triple 3-input AND</td>
<td>1</td>
<td>74L11</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Hex inverter</td>
<td>1</td>
<td>7404</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>21, 27</td>
<td>5-bit shift register</td>
<td>2</td>
<td>7496</td>
<td>5</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Timer</td>
<td>1</td>
<td>NE555</td>
<td>8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Quad 2-input NAND</td>
<td>1</td>
<td>74L00</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Dual 4-input NOR</td>
<td>1</td>
<td>7425</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Quad exclusive-OR</td>
<td>1</td>
<td>74L66</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>6-bit comparators</td>
<td>1</td>
<td>74L131</td>
<td>16</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Quad latch</td>
<td>1</td>
<td>74L75</td>
<td>5</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Hex inverter buffer 0/C</td>
<td>1</td>
<td>7406</td>
<td>14</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

**C1-C5, C10-C15**: 1 mfd capacitor

**C6**: 0.02 mfd capacitor

**C7**: 0.033 or 0.039 mfd capacitor

**C8**: 0.01 mfd capacitor

**C9**: 22 or 25 mfd capacitor

**C16**: 2200 pf capacitor

**C17**: 0.01 mfd capacitor

**R1**: 24 kohm resistor (red, yellow, red)

**R2**: 4.7 kohm resistor (yellow, violet, red)

**R3**: 1.5 kohm resistor (brown, green, red)

**R4**: 330 ohm resistor (orange, orange, brown)

**R5**: 220 ohm resistor 1W (red, red, brown)

**R6**: 27 kohm resistor (red, violet, orange)

**R7**: Approx 10 kohm resistor (brown, black, orange)

**R8**: 50 kohm potentiometer

**R9**: 100 ohm resistor (brown, black, brown)

**R10-R17**: 1 kohm resistor (brown, black, red)

**CR1**: 1N914 signal diode

**CR2**: 1N750 4.7 volt zener diode

**CR3**: Light-emitting diode

**S1**: DIP-switch

**J1**: DIP-socket

**P1**: DIP-plug

2 Cassette Coax Cables

Flat Ribbon Cable

**PC1**: Printed circuit board

**CS1**: Scotch low-noise tape

TO-3 Insulating Wafer

2 Sets of 4-40 nuts, screws, and washers

*The items marked with an asterisk are matched and bagged separately. 74LS may be substituted for 74L series IC's in most cases.*
ASSEMBLY DRAWING

NOTE: IC5 HAS PIN 1 IN UPPER RIGHT, REST IN LOWER LEFT.

COMPONENT SIDE
ASSEMBLY INSTRUCTIONS

1. TAKE OUT ALL THE PARTS AND CHECK THEM AGAINST THE PARTS LIST. IF THERE ARE ANY PARTS MISSING, DROP US A NOTE, AND WE WILL SEND THEM TO YOU. NOTE THAT THE NE555, 10K (APPROX.) RESISTOR, 27K RESISTOR, AND .01 MFD CAPACITOR WHICH ARE PACKAGED IN THE SEPARATE BAG ARE MATCHED FOR 1500 HZ AND ARE MARKED WITH ASTERISKS ON THE PARTS LIST. OTHER COMPONENTS USED FOR THESE WILL CAUSE THE OUTPUT TO RUN AT A DIFFERENT FREQUENCY.

2. USING THE ASSEMBLY DRAWING ON PAGE 7, INSERT THE DISCRETE COMPONENTS (BAG WITH CAPACITORS, RESISTORS, LED, DIP-SWITCH) INTO THEIR PROPER LOCATIONS. ON THE LIGHT-EMITTING DIODE (LED), THE RED LEAD GOES TO THE RESISTOR. IF THE RED HAS BEEN REMOVED OFF THE LEAD, USE AN OHMMETER TO DETERMINE WHICH LEAD GOES TO THE RESISTOR. THE IN750 ZENER DIODE (SMALL GLASS DEVICE) SHOULD BE MOUNTED WITH THE LINE SIDE TOWARD THE LEFT. THE IN914 DIODE (OTHER SMALL GLASS DEVICE) SHOULD BE MOUNTED WITH THE LINE SIDE TOWARD THE BOTTOM. THE ONLY CAPACITOR WHICH SHOULD BE ORIENTED IN A PARTICULAR DIRECTION IS THE FILTER CAPACITOR C9, WHICH SHOULD BE MOUNTED WITH THE PLUS SIDE TOWARD THE LEFT, AS MARKED ON THE BOARD. THE DIP-SOCKET SHOULD BE MOUNTED IN THE J1 POSITION, AS IS SHOWN ON THE ASSEMBLY DRAWING.

3. INSERT ALL THE INTEGRATED CIRCUITS (IC'S) (SMALL BLACK DEVICES AND ONE LARGE METAL DEVICE) INTO THEIR PROPER LOCATIONS AS INDICATED BY THE ASSEMBLY DRAWING AND THE PARTS LIST. NOTE THAT IC #5 (6T20) IS ORIENTED WITH PIN 1 AT UPPER RIGHT, WHEREAS ALL THE OTHERS ARE ORIENTED WITH PIN 1 AT LOWER LEFT. THERE ARE 2 WAYS TO TELL WHICH IS PIN 1 ON AN IC. ON SOME, THERE IS A SMALL DOT BY PIN 1. ON OTHERS, THERE IS A NOTCH AT ONE END OF THE IC, AND PIN 1 IS ON THE LOWER LEFT WHEN THE NOTCH IS AT THE LEFT. BE SURE TO PUT THE MICA INSULATOR UNDER THE REGULATOR (IC 6) BEFORE MOUNTING IT. BE SURE TO PUT IC 6 IN SO THAT THE SCREW HOLES LINE UP, AND THEN INSTALL THE SCREWS WITH THE WASHERS UNDER THE NUTS. BE SURE THAT THE WASHERS DON'T TOUCH THE TRACE ON SIDE B OF THE PC BOARD, AND THAT CAPACITOR C9 IS NOT TOUCHING ANY PART OF IC 6.

4. INSTALL THE SEVEN JUMPER WIRES, PREFERABLY USING SMALL INSULATED SINGLE-STRAND COPPER WIRE. THE JUMPER WIRES ARE MARKED ON THE ASSEMBLY DRAWING IN HEAVY BLACK LINES. FIVE OF THEM ARE ALSO MARKED ON THE BOARD WITH MATCHING LETTERS (A,B,C,D,E) BESIDE THEIR HOLES. NOTICE THAT THE BOTTOM OF JUMPER D GOES TO THE HOLE ABOVE THE "D". THERE ARE TWO MORE VERTICAL JUMPERS DOWN NEXT TO THE CONNECTOR. THE LAST IS A SHORT ONE (E), BELOW IC 30, TO SELECT THE OUTPUT PHASE. MINE IS CONNECTED BETWEEN THE CENTER AND LEFT HOLES, AS SHOWN IN THE SCHEMATIC. THIS MAY HAVE TO BE CHANGED LATER, AS NOTED IN THE ADJUSTMENT PROCEDURE.

5. CONSULT THE INSTALLATION NOTES FOR INSTRUCTIONS ABOUT THE CASSETTE CABLES.

6. SEE THE MODIFICATION SHEET (PAGE 8) FOR ANY LATE MODIFICATIONS. REVISION C HAS ALL MODIFICATIONS THROUGH MOD. #6. BE SURE THAT ALL MODIFICATIONS WHICH ARE NOT ON YOUR REVISION ARE INSTALLED.
MODIFICATIONS ON THE TARBELL CASSETTE INTERFACE AS OF SEPT 1, 1976.

INCLUDED ON REVISION B:

1. RESISTOR R1 HAS BEEN CHANGED TO A 2.4 KOHM RESISTOR.
2. ON THE OUTPUT VOLTAGE DIVIDER, R10 IS RECOMMENDED TO BE 1 KOHM FOR AUXILIARY CASSETTE INPUTS, INSTEAD OF 10 KOHM.

INCLUDED ON REVISION C:

3. A .1 MFD CAPACITOR HAS BEEN ADDED IN PARALLEL WITH R11. THIS IMPROVES RELIABILITY WITH SOME TYPES OF RECORDERS.
4. A 1 KOHM RESISTOR HAS BEEN ADDED BETWEEN PINS 10 AND 14 OF IC35. THIS PROVIDES PULL-UP TO DRIVE THE 7475 LATCH.
5. THE TRACE TO PIN 8 OF IC5 (GROUND) HAS BEEN CUT AND A LINE RUN DIRECTLY TO THE GROUND BUS ON THE BOTTOM. THIS ELIMINATES CROSS-TALK FROM THE NE555 OSCILLATOR.
6. A 2200 PF CAPCITOR SHOULD BE ADDED BETWEEN PINS 6 AND 8 ON IC5. THIS REDUCES THE EFFECTS OF HIGH-FREQUENCY NOISE GENERATED IN SOME COMPUTERS.

INCLUDED ON REVISION D:

7. ADD A .01 MFD CAPACITOR BETWEEN PINS 15 AND 8 OF IC5.
9. CONNECT ONE SIDE OF A 1K RESISTOR TO 5 VOLTS, THE OTHER TO IC29-7&10, IC2-14&3&7&10, IC6-14&3&7.
11. CONNECT ONE SIDE OF A 1K RESISTOR TO 5 VOLTS, THE OTHER TO IC4-13&10, IC3-9.
12. CONNECT PIN 1 TO PIN 2 ON IC3.

THERE HAVE BEEN SEVERAL QUESTIONS REGARDING THE USE OF RATES HIGHER THAN THE STANDARD 167 BYTES PER SECOND. CHANGES ARE REQUIRED ON BOTH THE INPUT AND OUTPUT SECTIONS. ON THE INPUT SECTION, THE POTENTIOMETER THAT IS PROVIDED ON THE BOARD MAY BE ADJUSTED TO CHANGE THE FREQUENCY. ON THE OUTPUT SECTION, ANY OF THREE COMPONENTS MAY BE CHANGED TO CHANGE THE FREQUENCY: R6, R7, OR C8. IF YOU WANT TO OPERATE AT TWO FREQUENCIES, FOR EXAMPLE A HIGHER ONE, AND THE STANDARD, IT IS FEASIBLE TO INSTALL A SWITCH FOR THE ABOVE MENTIONED COMPONENTS. FOLLOWING WOULD BE A REASONABLE PROCEDURE FOR EXPERIMENTING WITH THE HIGHER RATES:

1. REDUCE THE VALUE OF R6, R7, OR C8 BY ABOUT THE AMOUNT YOU WANT TO INCREASE THE FREQUENCY.
2. USE THE CASSETTE OUTPUT ROUTINE AND A STOP WATCH TO VERIFY THAT THE INCREASE IN OUTPUT SPEED HAS BEEN ATTAINED.
3. USE THE SYNC GENERATOR PROGRAM TO MAKE A TAPE WITH A LONG STREAM OF SYNC BYTES AT THE NEW FREQUENCY.
4. PLAY IT BACK, ADJUSTING THE POT FOR THE LED TO COME ON. CLOSER ADJUSTMENT OF THE CASSETTE VOLUME CONTROL MAY ALSO BE NECESSARY.
5. THE .1 CAPACITOR MODIFICATION MENTIONED IN ITEM 3 ABOVE WILL ALSO HAVE TO BE REDUCED ACCORDINGLY.
SOLDERING, CLEANING, AND INSTALLATION NOTES

SOLDERING:

Be sure to use good resin-core solder. Acid-core solder will corrode. Use a small soldering element, preferably about 27 watts. Keep your tip clean by wiping on a sponge. Apply heat to the joint first, then solder, then remove solder, then remove the heat (soldering iron). Don't leave the heat applied to the connection more than a few seconds at a time. Some of the components can be destroyed by too much heat, especially the integrated circuits (IC's). Be sure there is a smooth flow of solder over the complete connection, and that the joint looks shiny.

CLEANING:

After you finish soldering, there will be many small conductive particles on the board which you cannot always see. Take a small pointed instrument of some sort, such as a jeweler's screwdriver, and scrape between all printed wiring which is close together, such as those leading to the IC pins. This may take some time, but it is well worth it. Then scrub the bottom (side B) of the board with alcohol. Then visually inspect the board under a strong light, and again remove any dangerous looking particles.

INSTALLATION:

First set the dip-switches to the following positions:

1-off, 2-off, 3-on, 4-off, 5-off, 6-on, 7-off (input phase inversion)

Switches 1 through 6 correspond to address bits 2 through 7 respectively, and off is a one, on is a zero. Address bit 1 can be either way, as it is ignored by the present interface. Address bit 0 is zero for status/control, and one for data. Therefore, the switch settings above correspond to device address 011011XX (most significant bit first), where X indicates bits that can be either way. This is the device select code that is used in all software for the cassette interface that is supplied by Tarbell Electronics.

Then insert the board into the 100-pin socket, being sure that the component side of the board is to the right (Altair) or front (IMSAI) as viewed from the front of the computer. Then install the ribbon cable between the dipconnector on the interface board and the 25-pin connector slot in the rear. Then install the coax cables between the 25-pin connector and the cassette recorder. The coax cables may also be connected directly between the interface board and the recorder. See the pin function list page for the proper connections.
INITIAL ADJUSTMENT INSTRUCTIONS

THIS INTERFACE WAS DESIGNED TO BE AS EASY AS POSSIBLE TO GET UP AND GOING. THERE ARE, HOWEVER, SOME INITIAL ADJUSTMENTS TO BE MADE, AFTER WHICH THERE NEED NEVER BE ANY MORE.

1. PUT THE TEST CASSETTE INTO YOUR CASSETTE RECORDER.

2. IF YOUR RECORDER HAS A TONE CONTROL, TURN IT TO THE MAXIMUM (BEST HIGH-FREQUENCY RESPONSE) POSITION.

3. TURN YOUR VOLUME CONTROL TO A MIDDLE POSITION.

4. TURN THE POTENTIOMETER ON THE INTERFACE TO A MIDDLE POSITION.

5. PRESS THE "PLAY" BUTTON ON YOUR RECORDER.

6. IF THE LED (RED LIGHT) ON THE INTERFACE DOES NOT COME ON AFTER A FEW SECONDS, ADJUST YOUR VOLUME AND THE INTERFACE POTENTIOMETER UNTIL THE LIGHT COMES ON.

7. IF THE LED STILL DOESN'T COME ON, CHANGE SWITCH #7 (INPUT PHASE REVERSAL) ON THE DIP-SWITCH TO THE OPPOSITE POSITION; THEN REPEAT STEP 6.

8. IF THE LED STILL DOESN'T COME ON AFTER ADJUSTING YOUR VOLUME AND THE INTERFACE POT, SOMETHING IS WRONG WITH YOUR RECEIVER SECTION.

9. WHEN THE LED COMES ON, THIS INDICATES THAT THE RECEIVER IS OPERATING PROPERLY, AND IS DETECTING THE CONTINUOUS STREAM OF SYNC BYTES WHICH IS ON THE TEST TAPE. FURTHER ADJUST BOTH THE VOLUME CONTROL AND THE INTERFACE POT SO THAT YOU CAN TURN EACH OF THEM FROM SIDE TO SIDE A LITTLE WITHOUT THE LIGHT GOING OUT. THE LIGHT SHOULD BE VERY STABLE, WITH NO FLICKER.

NOTE: SINCE THE LED ONLY STAYS ON WHEN IT IS DETECTING CONTINUOUS SYNC BYTES, IN NORMAL OPERATION (WITH REAL DATA) IT WILL JUST FLICKER ONCE IN A WHILE.

1. RUN THE SYNC CODE GENERATOR PROGRAM WITH YOUR RECORDER IN RECORD MODE, ONTO A BLANK TAPE.

2. TRY THE PROCEDURE ABOVE. IF THE LIGHT DOES NOT COME ON CONTINUOUSLY, YOU MAY BE RECORDING AT TOO HIGH OR TOO LOW A LEVEL. TRY DIFFERENT LEVELS UNTIL YOU FIND THE BEST PLACE. YOU ALSO MAY BE RECORDING IN THE OPPOSITE PHASE. IF SO, CHANGE THE JUMPER FROM PIN 9 TO PIN 8 ON IC 23.

NOTE: IF YOU HAVE AN OSCILLOSCOPE, IC 4 PIN 11 SHOULD SHOW A NICE CLEAN WAVE FORM, WITH ABOUT 25% DUTY CYCLE. ALTHOUGH THERE MAY BE LONG-TERM JITTER, BECAUSE OF THE FLUTTER AND WOW ON THE CASSETTE RECORDER, FAST JITTER ON THE EDGES OF THE WAVE-FORM SHOULD BE FAIRLY SMALL. THE LESS THIS HIGH-SPEED JITTER IS, THE MORE TOLERANCE YOUR INTERFACE WILL HAVE TO TAPE SPEED VARIATIONS.
CASSETTE INTERFACE OPERATING INSTRUCTIONS

These instructions pertain to operating the interface with an ordinary audio cassette recorder, and assume that the proper software (programs, subroutines) is in the computer to communicate with the interface.

To perform an output (save, write) operation:

1. If your volume control has an effect on the recording function, first turn it all the way down.

2. Get to the point in your program where all it takes is a push of a button to start into the cassette output routine.

3. Use fast-forward or rewind to move to the desired location on the cassette tape.

4. Start your cassette recording.

5. If your volume control has an effect during record, slowly increase the volume until your indicator shows a correct recording level.

6. Wait for about 5 seconds to record leader.

7. Push the button that starts the output routine on the recorder. (This might be the carriage-return after "Csave" in Basic, or the front-panel "run" button for stand-alone programs.)

8. When the program indicates that the data transfer is complete, stop your cassette recorder.

To perform an input (load, read) operation:

1. Be sure your volume control is at the position that you left it in the adjustment procedure.

2. Get to the point in your program where all it takes is a push of a button to start into the cassette input routine.

3. Use fast-forward or rewind to move to the desired location on the cassette tape. This should be a few seconds into the leader of a previous recording.

4. Start your cassette in the playback mode.

5. Press the button which causes the input routine to start running. (In Basic, this might be the carriage-return after a CLOAD, or the front-panel "run" button for stand-alone programs, such as bootstrap.)

6. When the program indicates that the data transfer is complete, stop your cassette recorder.

Note: Always be sure that all memory into which programs or data is to be read, is unprotected first.
CASSETTE BOOTSTRAP PROGRAM

This program loads data starting at zero and keeps on going. There is no count of bytes, and no checksum. It is assembled to run at 2F00 (HEX), but may be assembled to run anywhere, provided that it does not load data over itself. Use the second hand on your watch to determine how long to wait until stopping this program. Allow about 45 seconds to load an 8 KBYTE BLOCK.

2F00 3E 10 MVI A,10H SET BIT 4 OF A = 1
2F02 D3 6E OUT CASC RESET INTERFACE.
2F04 21 00 00 LXI H,O PUT STARTING ADDRESS IN H,L.
2F07 DB 6E LOOP IN CASC READ STATUS.
2F09 E6 10 ANI 10H CLEAR ALL BUT BIT 4.
2F0B C2 07 2F JNZ LOOP WAIT IN LOOP UNTIL READY.
2F0E DB 6F IN CASD READ A DATA BYTE.
2F10 FB EI SIGNAL OPERATOR.
2F11 77 MOV M,A PUT DATA INTO MEMORY.
2F12 23 INX H INCREMENT MEMORY POINTER.
2F13 C3 07 2F JMP LOOP REPEAT THE ABOVE OPERATION.

CASC EQU 6EH CASSETTE'STATUS PORT.
CASD EQU 6FH CASSETTE DATA PORT.

END

NOTE: IF YOU HAVE AN IMSAI OR ALT AIR WITH AN OUTPUT PORT ON THE FRONT PANEL (8 LED'S), YOU CAN USE THE BOOTSTRAP PROGRAM FOR TROUBLESHOOTING THE INPUT SECTION WITH THE FOLLOWING MODIFICATION: AT INSTEAD OF SUBSTITUTE
2F10 EI (FB) CMA (2F)
2F11 MOV M,A (77) OUT (D3)
2F12 INX H (23) LEDS (FF)

SYNC CODE GENERATOR PROGRAM

This program may be used to generate a continuous stream of E6 (HEX), the same as is on the supplied cassette. If you find that the received stream is inverted from the one supplied, you may change the circuit so that IC23-8 is hooked to the 74L86 instead of IC23-9. This will make your recordings the same phase as mine.

0000 DB 6E LOOP IN CASC READ STATUS.
0002 E6 20 ANI 20H LOOK AT BIT 5.
0004 C2 00 00 JNZ LOOP WAIT UNTIL READY.
0007 3E E6 MVI A,0E6H GET SYNC BYTE.
0009 D3 6F OUT CASD WRITE IT INTO CASSETTE.
000B C3 00 00 JMP LOOP REPEAT.

CASC EQU 6EH STATUS PORT.
CASD EQU 6FH DATA PORT.

END
CASSETTE INTERFACE OUTPUT ROUTINE

This program writes a block of memory out onto cassette tape.
The program is assembled to start at 3100 (hex), but may be reassembled to start anywhere. The block starting address is located at address 3104 (hex). The block length (2 bytes) is located at address 3107 (hex). The program will write a "W" on the comment device when it is through with its data transfer.

```
3100 31 43 31 LXI SP,STAK SET STACK POINTER.
3103 21 00 00 LXI H,0 GET BLOCK ADDRESS.
3106 01 00 20 LXI B,2000H SET BLOCK LENGTH = 8192.
3109 1E 00 MVI E,0 SET E=0.
310B 3E 3C MVI A,3CH GET START BYTE.
310D CD 32 31 CALL COUT OUTPUT START BYTE TO CASSETTE.
3110 3E E6 MVI A,OE6H GET SYNC BYTE.
3112 CD 32 31 CALL COUT OUTPUT SYNC BYTE TO CASSETTE.
3115 7E LOOP MOV A,E GET A DATA BYTE FROM MEMORY.
3116 CD 32 31 CALL COUT OUTPUT DATA BYTE TO CASSETTE.
3119 83 ADD E ADD E (CHECKSUM) TO A.
311A 5F MOV E,A PUT NEW CHECKSUM INTO E.
311B 23 INX H INCREMENT MEMORY POINTER.
311C 0B DCX B DECREMENT COUNTER.
311D 3E 00 MVI A,0 MAKE A=0.
311F B6 CMP B IF B NOT = 0,
3120 C2 15 31 JNZ LOOP REPEAT LOOP.
3123 B9 CMP C IF C NOT = 0,
3124 C2 15 31 JNZ LOOP REPEAT LOOP.
3127 7B MOV A,E OTHERWISE, GET CHECKSUM
3128 CD 32 31 CALL COUT AND OUTPUT IT.
312B 3E 57 MVI A,\"W\" WRITE \"W\" (END OF WRITE).
312D D3 01 OUT I PRINT ON CONSOLE.
312F C3 2F 31 WAIT JK PUSH WAIT HERE WHEN DONE.
3132 F5 COUT PUSH PSW SAVE A AND FLAGS.
3133 DB 6E CLOP IN CASC READ CASSETTE STATUS.
3135 E6 20 ANI 20H CLEAR ALL BUT BIT 5.
3137 C2 33 31 JNZ CLOP TRY AGAIN IF NOT READY.
313A F1 POP PSW RESTORE A AND FLAGS.
313B D3 6F OUT CASSD OUTPUT DATA TO CASSETTE.
313D C9 RET RETURN FROM COUT.
313E 00 0
313F 00 0
3140 00 0
3141 00 0
3142 00 0
3143 00 STAK 0
3146 3F 6B CASD EQU 6FH
```
CASSETTE INTERFACE INPUT ROUTINE

This program reads a block of bytes from cassette into memory. The program is assembled to start at 3100 (hex), but may be reassembled to start anywhere, although care should be taken to insure that the data it is reading does not write over the program itself. This may be accomplished by locating the program immediately below or a block length above the data to be read in. The starting address for the block is located in address 3185 (hex). The block length is located in address 3188 (hex) (two bytes).

```
3180 3E 10 MVI A,10H set bit 4 of A=1.
3182 D3 6E OUT CASC reset interface.
3184 21 00 00 LXI H,0 get starting address.
3187 11 00 20 LXI D,2000H get block length.
3189 06 00 MVI B,0 set checksum = 0.
318C D3 6E LOOP IN CASC read cassette status.
318E E6 10 ANI 10H look at bit 4.
3190 C2 EC 31 JNZ LOOP wait if not ready.
3193 D3 6F IN CASD read data from cassette.
3195 77 MOV M,A put data into memory.
3196 80 ADD B add checksum to A.
3197 47 MOV B,A put it back in B.
3198 23 INX H increment memory pointer.
3199 13 DCX D decrement counter.
319A 3E 00 MVI A,0 clear A.
319C BA CMP D if D not = 0.
319D C2 EC 31 JNZ LOOP read more.
31A0 BB CMP E if E not = 0.
31A1 C2 EC 31 JNZ LOOP read more.
31A4 DB 6E CHEK IN CASC read status.
31A6 E6 10 ANI 10H look at bit 4.
31A8 C2 AA 31 JNZ CHEK wait if not ready.
31AB DB 6F IN CASD read checksum.
31AD 36 CMP B compare to B.
31AE 3E 45 MVI A,45 put code for "E" in A.
31B0 C2 B5 31 JNZ ERR if checksums not equal, error.
31B3 C6 02 ADI 2 add a 2 to make "G" if equal.
31B5 D3 01 ERR OUT CRTD print "E" for "G".
31B7 C3 B7 31 END JMP END wait here when done.
CASC EQU 6EH cassette status/control port.
CASD EQU 6FH cassette data port.
CRTD EQU 01H console data port.
```
1 REM THIS PROGRAM SHOWS HOW TO SAVE DATA ONTO A CASSETTE
2 REM AND LOAD IT BACK INTO MEMORY FROM A PROGRAM RUNNING
3 REM UNDER 8K BASIC 3.1. IT ALLOWS YOU TO ENTER LINES OF
4 REM TEXT FROM THE CONSOLE KEYBOARD, SAVE THEM ON CASSETTE,
5 REM LOAD THEM BACK INTO MEMORY, AND PRINT THEM.
10 CLEAR 3000: L=50: DIM A$(50): REM RESERVE FOR UP TO 50 LINES.
20 CC=110: CD=111: REM CONTROL AND DATA PORT NUMBERS.
30 T$="I": REM END-OF-FILE CHARACTER.
40 D=100
100 INPUT "COMMAND": CS
110 IF CS="ENTER" THEN 1000
120 IF CS="PRINT" THEN 2000
130 IF CS="SAVE" THEN 3000
140 IF CS="GET" THEN 4000
900 PRINT "INVALID COMMAND.:" : GOTO 100
1000 REM ENTER TEXT FROM THE CONSOLE KEYBOARD
1010 FOR N=1 TO L: REM ENTER A MAXIMUM OF L LINES.
1020 INPUT BS: REM READ A LINE FROM KEYBOARD.
1030 IF BS="X" THEN 1070: REM A % TERMINATES THE INPUT.
1040 A$(N)=BS: NEXT N: REM PUT LINE INTO BUFFER.
1070 N=N+1: GOTO 1000: REM N=THE NUMBER OF LINES ENTERED.
2000 REM PRINT THE BUFFER AREA ON THE CONSOLE.
2010 FOR I=1 TO N: PRINT A$(I): NEXT I: GOTO 100
3000 REM SAVE THE BUFFER ONTO CASSETTE TAPE.
3010 SS=CHRS(195)+CHRS(230): REM SS=START & SYNC BYTES.
3019 REM CHANGE THE CONSOLE OUTPUT ROUTINE FOR CASSETTE.
3020 POKE 1230, CC: POKE 1232, 32: POKE 1236, CD
3030 FOR I=1 TO N
3040 FOR K=1 TO D: NEXT K: REM DELAY FOR COUNT OF D.
3050 BS=SS+A$(I): REM HOOK START & SYNC BYTES TO LINE.
3060 PRINT BS: REM WRITE LINE ONTO CASSETTE.
3070 NEXT I
3080 BS=SS+TS: REM HOOK START & SYNC BYTES TO TERMINATOR.
3090 FOR K=1 TO D: NEXT K: REM DELAY FOR COUNT OF D.
3095 PRINT BS: REM WRITE THE END-OF-FILE MARK.
3100 REM CHANGE CONSOLE ROUTINE BACK TO NORMAL.
3101 POKE 1230, 0: POKE 1232, 128: POKE 1236, 1
3110 GOTO 100
4000 REM GET TEXT FROM CASSETTE AND PUT INTO BUFFER.
4010 REM CHANGE CONSOLE INPUT ROUTINE FOR CASSETTE.
4011 POKE 1241, CC: POKE 1243, 16: POKE 1245, CD: POKE 1232, 0
4012 POKE 1236, 255
4020 FOR I=1 TO L
4030 OUT CC, 16: REM RESET CASSETTE INPUT SECTION.
4040 INPUT BS: REM READ A LINE OF TEXT FROM CASSETTE.
4050 IF BS=TS THEN 4080
4060 A$(I)=BS
4070 NEXT I
4080 N=N+1
4090 REM CHANGE CONSOLE ROUTINE BACK TO NORMAL.
4091 POKE 1241, 0: POKE 1243, 1: POKE 1245, 1: POKE 1232, 128
4092 POKE 1236, 1
4100 GOTO 100
8K
SAVING AND LOADING DATA TO AND FROM CASSETTE FROM BASIC PROGRAMS

It is quite often desireable to save and load data to and from cassette, while running a BASIC program. For example, you might have a nice inventory program running in BASIC, but it's of limited utility if there is no way to save the inventory on cassette for overnight storage. You can leave the computer running, but if there is an interruption in power, or a computer failure, your inventory is lost. Other applications include accounts receivable, mailing lists, and payroll.

There are several ways of handling this function, depending on the version of BASIC you use. Some versions of BASIC have a command which can save and load a numerical array to and from cassette. ALTAIR* DISK BASIC is one of these. The only problem with this way, is that DISK BASIC is fairly large, and that strings have to be converted to numerical arrays. Another way is described explicitly by a program on page 15 of the Tarbell Cassette Interface Manual. In this method, the console (TTY, CRT) routines are modified by POKE commands, so that they are temporarily cassette I/O routines. Then the PRINT and INPUT statements may be used to transfer the data. After the data is transferred, POKE statements restore the console routines to their original form. The disadvantage here is that different versions of BASIC have their console I/O routines in different places, so the program has to be adapted when changing from one version to another.

Another possibility is to POKE the data into an unused area of memory. The USR function then is used to transfer to your own output routine (possibly an adaptation from one in the manual). This routine only needs to transfer a block of memory onto tape. The USR function can be used with a different argument to run an input routine, then the BASIC program can retrieve the data with the PEEK function. The main problem with this method is that the I/O routines generally have to be loaded separately, and are not yet written and tested.

Still another way is to use a general purpose monitor program. This program would also be loaded separately, but might reside in read-only-memory (ROM). The monitor would handle all input/output functions including console and cassette I/O. One of the commands in the monitor is to assign different functions to different I/O devices than normally handle these functions. For example, the console function could be assigned to the cassette interface instead of the TTY. Then PRINT and INPUT statements could be used for I/O to and from cassette.

The monitor described above is presently under development by Tarbell Electronics. In addition to the Assign command, it has commands for dumping and loading memory, checking records for errors, and moving data from one area in memory to another.
CONTROLLING THE START-STOP (REMOTE) FUNCTION ON YOUR CASSETTE

Most cassette recorders have a remote control input, meant for control from a microphone switch. This facility can be operated by a computer program to start and stop the tape automatically, according to the needs of the program. This is particularly important if the amount of data on tape is more than will fit into main memory all at once. In this case, the data may be "blocked"; that is, gaps may be inserted between blocks of data, which allow time for the tape to start and stop. The program may then start the tape, read some data, stop the tape, work on the data, start another tape, write some data, and stop the tape. This process may be repeated until all the data is processed. Some examples where this operation might be necessary are as follows:

1) An assembler, where the source is larger than memory, may read source from one tape and write machine code to another.
2) A compiler, in the same situation.
3) A merging program, where an old file is updated with changes to form a new file.

The circuit for controlling one tape unit is shown below:

These routines may be used for starting and stopping the cassette before and after input and output operations:

START
LDA CTLS GET CONTROL STATUS BYTE.
ORI 01 SET BIT 0 = ONE.
STA CTLS UPDATE CONTROL STATUS BYTE.
OUT CASC START THE TAPE.
CALL DELAY WAIT FOR TAPE TO GET UP TO SPEED.
RET RETURN (NEXT DO YOUR I/O).

STOP
LDA CTLS GET CONTROL STATUS BYTE.
AND OFEH SET BIT 0 TO ZERO.
STA CTLS UPDATE CONTROL STATUS BYTE.
OUT CASC STOP THE TAPE.
RET RETURN FROM I/O ROUTINE.

CTLS DB 0 CONTROL STATUS BYTE.
CASC EQU 6EH CASSETTE CONTROL PORT.

The start-up delay is determined by your recorder, and should be longer before a write than before a read operation.

Note: A module which allows the control of up to four cassette recorders with a Tarbell cassette interface is available from: RO-CHE systems, 7101 Mammoth Ave, Van Nuys, CA 91405.
THEORY OF CASSETTE INTERFACE OPERATION

OUTPUT SECTION

THE PURPOSE OF THIS SECTION IS TO CONVERT 8-BIT PARALLEL BYTES FROM THE COMPUTER TO A SERIAL BI-PHASE ENCODED DATA STREAM FOR THE RECORDER. THE NE555 (IC 22) IS CONNECTED AS AN OSCILLATOR TO OSCILLATE AT TWICE THE FREQUENCY OF THE REQUIRED CLOCK RATE. FOR 800 BITS PER INCH, THE CLOCK RATE NEEDS TO BE 1500 HZ FOR A RECORDER RUNNING AT 1 7/8 INCHES PER SECOND. SO IC 22 RUNS AT 3000 HZ, DETERMINED BY THE MATCHED SET OF COMPONENTS (IC 22, C6, R6, AND R7). THE OUTPUT ON PIN 3 IS FED TO A J-K FLIP-FLOP, WHICH DIVIDES THE FREQUENCY BY TWO. THE MAIN FUNCTION OF THIS FLIP-FLOP IS TO MAKE THE WAVEFORM PERFECTLY SYMMETRICAL. THIS SQUARE-WAVE IS THEN FED TO AN INPUT OF THE EXCLUSIVE-OR GATE AT IC 30 PIN 6.

IC 32, THE DM8131, IS A 6-BIT DIGITAL COMPARATOR. IT'S PURPOSE IS TO COMPARE THE ADDRESS ON BITS 2 THROUGH 7 OF THE ADDRESS BUS WITH THE SETTING ON THE DIP SWITCH. WHEN THEY MATCH, THE OUTPUT AT PIN 9 GOES LOW, INDICATING THAT THIS DEVICE IS BEING SELECTED. THIS SIGNAL IS INVERTED AND ANDED WITH THE WRITE SIGNAL AND THE STATUS OUTPUT SIGNAL ON THE BUS. THIS PRODUCES A HIGH OUTPUT AT IC 14 PIN 6 WHEN THIS DEVICE IS BEING WRITTEN TO.

THIS SIGNAL IS IN TURN ANDED WITH ADDRESS BIT 0 TO STROBE DATA INTO THE SHIFT REGISTER (IC 26 PIN 6 AND IC 20 PINS 4 AND 2). THE DATA IS SHIFTED OUT OF THE SHIFT REGISTER BY THE CLOCK PREVIOUSLY MENTIONED. THE SHIFT REGISTER IS MADE UP OF IC'S 21, 27, AND 23. DATA ENTERS AT THE BOTTOM AND IS SHIFTED OUT TOWARD THE RIGHT, WHERE IT IS COMBINED WITH THE CLOCK TO FORM THE BI-PHASE SIGNAL FOR THE RECORDER.

THE FIRST EIGHT OUTPUTS OF THE SHIFT REGISTER ARE FED INTO IC 26. SINCE A ZERO IS BEING FED INTO THE LEFT END OF THE SHIFT REGISTER FOR EACH SHIFT (IC 21 PIN 9), THESE LEFT EIGHT BITS WILL ALL BECOME ZERO AFTER 8 SHIFTS. AT THIS TIME THE OUTPUTS AT IC 26 PINS 6 AND 6 WILL BOTH BE HIGH, CAUSING THE OUTPUT AT IC 14 PIN 6 TO GO HIGH. THIS SIGNAL WILL BE USED BY THE COMPUTER TO DETERMINE WHEN THE INTERFACE NEEDS MORE OUTPUT DATA.

WHEN ADDRESS BIT 0 IS LOW, THE OUTPUT AT IC 14 PIN 6 ANDED WITH IC 31 PIN 2 CAUSES A HIGH OUTPUT AT IC 35 PIN 10. THIS STROBES BITS 0, 1, 2, AND 3 ON THE DATA BUS INTO THE LATCH OF IC 34. THE OUTPUTS OF THIS LATCH ARE BUFFERED WITH IC 35, AND APPEAR AT PINS 2, 4, 6, AND 8. THESE ARE THEN FED TO THE DIP-SOCKET. IF DATA BUS BIT 4 IS HIGH, A PULSE APPEARS AT PIN 6 OF IC 26. THIS IS USED TO RESET THE COUNTER IN THE INPUT SECTION.
THEORY OF OPERATION

INPUT SECTION

The purpose of this section is to convert the bi-phase audio signal coming from the cassette recorder to 6-bit parallel bytes for the computer. The audio signal coming directly from the cassette earphone or speaker output is terminated by the 100 ohm resistor and fed through the .02 capacitor to the input of the 820 (IC 5 pins 6 and 7). This input has a built-in voltage-divider, which biases the DC level to a good midway TTL reference voltage between one and zero. This is one input of a high-speed comparator. The other input (IC 5 pin 5) is connected through a resistor divider and a 1N914 diode to one of the comparator outputs (IC 5 pin 9). This provides a small amount of hysteresis to combat noise problems. The other output of the comparator (IC 5 pin 1) is exclusive-ored with switch 7 on the dip-switch to provide a way to invert the input data stream. This is then fed to a D-type flip-flop (IC 4 pin 12).

Meanwhile, inside IC 5, the outputs of the comparator section are fed to an edge-detector, which detects both positive and negative-going transitions. The output of this detector is then used to trigger a stable non-retriggerable one-shot. The capacitor for this one-shot is between pins 12 and 14 of IC 5, and the resistor is the 50 kohm potentiometer. The output of the one-shot is the recovered clock (IC 5 pins 10 and 11). This is fed to three different places: 1) IC 4 pin 11, where it triggers the flip-flop to recover the serial data stream; 2) IC 3 pin 8 where it is used to shift the serial-parallel shift-register; 3) IC 29 pin 1 and IC 26 pin 13, where it is used to step the 8 counter.

In a start-up position, IC 29 pin 12 is zero, having been reset either by the reset switch or by a reset command from the program. This stops the clock from triggering the 8 counter, which has also been reset. As the serial stream flows through the shift-register (IC 3), it is continually inspected by the sync decoder made up of IC's 9 and 10. Whenever a sync code appears in the shift register, IC 10 pin 8 goes low. This lights the sync LED and also allows the flip-flop at IC 9 pin 12 to go high. This allows the clock to appear at IC 26 pin 11, and trigger the 8 counter. When the counter has counted to 8, IC 6 pin 12 goes high, triggering the ready flip-flop at IC 6 pin 9. This ready condition indicates to the computer that there is a byte in the shift register ready to read. The computer may then read this byte through gates of IC's 13 and 19.

In order to read data the signal at point B, which comes from the output section is anded with the two input gate signals PDBIN and SINP (IC 14 pins 13, 1 and 2). When address bit 0 is high, this signal is anded with it to gate data from the shift register onto the input data bus (IC 1 pin 3). When address bit 0 is low, this signal is anded with it to gate various status bits onto the input data bus. Four of these inputs are general-purpose, and come from the dip-socket. The others are picked up at IC 25 pin 3 (input status), and IC 25 pin 6 (output status).
CASSETTE INTERFACE TIMING DIAGRAM

SERIAL BINARY DATA 1 1 1 0 0 1 1 1 0 1 1 1 0 0 1 1

CLOCK (PIN 1-9)

NRZ DATA (PIN 23-9)

BI-PHASE TO TAPE (30-4)

AUDIO FROM TAPE

RECOVERED BI-PHASE (30-11)

EDGE DETECTOR OUTPUT * *

RECOVERED CLOCK (5-11)

RECOVERED NRZ DATA (4-9) [INVALID HERE]

* CLOCK IS IN PROPER PHASE AFTER FIRST DATA CHANGE.
### CASSETTE INTERFACE PIN FUNCTIONS

<table>
<thead>
<tr>
<th>J1 (DIP SOCKET)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 GP STATUS IN BIT 0</td>
<td>16 GROUND</td>
</tr>
<tr>
<td>2 GP STATUS IN BIT 1</td>
<td>15 DATA FROM CASSETTE</td>
</tr>
<tr>
<td>3 GP STATUS IN BIT 2</td>
<td>14 SPARE</td>
</tr>
<tr>
<td>4 GP STATUS IN BIT 3</td>
<td>13 SPARE</td>
</tr>
<tr>
<td>5 GP CONTROL OUT BIT 3</td>
<td>12 +5 VOLTS</td>
</tr>
<tr>
<td>6 GP CONTROL OUT BIT 2</td>
<td>11 SPARE</td>
</tr>
<tr>
<td>7 GP CONTROL OUT BIT 1</td>
<td>10 DATA TO CASSETTE</td>
</tr>
<tr>
<td>8 GP CONTROL OUT BIT 0</td>
<td>9 GROUND</td>
</tr>
</tbody>
</table>

**NOTES:**
- GP STANDS FOR GENERAL-PURPOSE
- DATA FROM CASSETTE SHOULD BE CONNECTED TO EARPHONE JACK
- DATA TO CASSETTE SHOULD BE CONNECTED TO AUXILIARY JACK.
- IF MIKE JACK IS USED, OR IF THE RECORDER ONLY OPERATES WITH AUTOMATIC VOLUME CONTROL ON RECORD, THEN DIVIDER RESISTORS MAY HAVE TO BE CHANGED FOR THE PROPER RECORDING LEVEL.
- THE DIVIDER RESISTORS ARE R10 AND R11.

### P2 (6800 BUS PINS)

|  |
|------------------|--|
| 1 +8V |
| 29 ADDRESS LINE #5 |
| 30 ADDRESS LINE #4 |
| 31 ADDRESS LINE #3 |
| 35 DATA OUT LINE #1 |
| 36 DATA OUT LINE #0 |
| 38 DATA OUT LINE #4 |
| 39 DATA OUT LINE #5 |
| 40 DATA OUT LINE #6 |
| 41 DATA IN LINE #2 |
| 42 DATA IN LINE #3 |
| 43 DATA IN LINE #7 |
| 45 OUT |
| 46 INP |
| 50 GROUND |
| 51 +8V |
| 52 -16V |
| 75 RESET-NOT |
| 77 WRITE-NOT |
| 78 DATA BUS-IN |
| 79 ADDRESS LINE #0 |
| 81 ADDRESS LINE #2 |
| 82 ADDRESS LINE #6 |
| 83 ADDRESS LINE #7 |
| 86 DATA OUT LINE #2 |
| 89 DATA OUT LINE #3 |
| 90 DATA OUT LINE #7 |
| 91 DATA IN LINE #4 |
| 92 DATA IN LINE #5 |
| 93 DATA IN LINE #6 |
| 94 DATA IN LINE #1 |
| 95 DATA IN LINE #0 |
| 100 GROUND |
IF YOU CANNOT MAKE AT LEAST TEN 8K-BYTE TRANSFERS WITH NO ERRORS, YOU HAVE A PROBLEM, AND THE ITEMS BELOW MAY BE OF SOME HELP:

1. CHECK TO MAKE SURE THAT ALL THE COMPONENTS AND JUMPERS ARE IN THEIR PROPER LOCATIONS, AND THAT THEY ARE ORIENTED AS SHOWN IN THE ASSEMBLY DRAWING.

2. MAKE SURE THAT THE BOARD IS CLEAN, ESPECIALLY THAT THERE IS NO FLUX RESIDUE BETWEEN IC PINS OR OTHER CLOSE LINES.

3. DEMAGNETIZE AND CLEAN THE RECORD/PLAYBACK HEAD ON YOUR RECORDER.

4. WHEN YOU PLUG IN THE BOARD, BE SURE THAT THE PINS ON THE BOARD EDGE CONNECTOR LINE UP WITH THE PINS IN THE MOTHERBOARD CONNECTOR.

5. HAVE YOU TRIED BOTH PHASES WITH THE PHASE SWITCH, AND ARE THE OTHER SETTINGS ON THE DIP-SWITCH CORRECT?

6. YOU SHOULD BE ABLE TO ADJUST THE VOLUME ON YOUR RECORDER BY ABOUT 50% DURING PLAYBACK, AND STILL HAVE THE SYNC LIGHT LIT WHEN READING THE SYNC STREAM. IF IT DOESN'T LIGHT AT ALL, THERE IS PROBABLY SOME GROSS PROBLEM ON THE BOARD, SUCH AS A BAD PLATED THROUGH HOLE, A SOLDER BRIDGE, OR A BAD INTEGRATED CIRCUIT. IF YOU ARE USING A TAPE DECK THAT HAS ONLY A PREAMP, YOU MAY NEED TO ADD AN EXTRA STAGE OF AMPLIFICATION IN ONE OF THE EXTRA IC SLOTS.

7. IF YOU HAVE AN OSCILLOSCOPE, THE BEST PLACE TO LOOK TO SEE HOW THE RECEIVER INPUT SECTION IS OPERATING IS AT IC 4, PIN 11. THIS SIGNAL SHOULD BE FAIRLY CLEAN, WITH SOME OVERALL JITTER, DUE TO THE TAPE WOW AND FLUTTER, AND SOME HIGH-SPEED JITTER ON THE EDGE OF THE WAVEFORM. IT IS THIS HIGH-SPEED JITTER THAT YOU SHOULD TRY TO MAKE A MINIMUM.

8. IF YOU HAVE A VIDEO INTERFACE, OR OTHER SOURCE OF HIGH-FREQUENCY NOISE, TRY LOCATING IT FURTHER AWAY FROM THE CASSETTE INTERFACE.

9. ARE YOU SURE THAT YOUR RECORDER HAS A FREQUENCY RESPONSE TO 8KHZ?

10. HAVE YOU USED THE PROPER VOLTAGE DIVIDER (R10, R11) FOR YOUR PARTICULAR RECORDER? IF YOU ARE ABLE TO RECOVER THE SYNC STREAM I WROTE SATISFACTORILY, BUT ARE HAVING TROUBLE WITH RECORDINGS YOU MAKE YOURSELF, THE LEVEL GOING FROM THE INTERFACE TO THE RECORDER MAY BE TOO HIGH OR TOO LOW, ESPECIALLY IF YOU HAVE AUTOMATIC LEVEL CONTROL. YOU MAY ALSO WANT TO TRY OPERATING WITHOUT C15.

11. HAVE YOU CHECKED YOUR +5 VOLT POWER? TOO MANY BOARDS IN YOUR COMPUTER COULD INTRODUCE RIPPLE ON THIS SUPPLY.

12. DON'T USE DIGITALLY CERTIFIED TAPE, ONLY AUDIO LOW-NOISE TAPE.

13. IF YOUR RECORDER HAS AN INTERNAL MICROPHONE, BE SURE IT IS NOT ACTIVE WHILE YOU ARE MAKING A RECORDING (THE J.C. PENNY HAS A SWITCH ON THE TONE CONTROL WHICH CUTS OFF THE INTERNAL MIC).

14. IF YOU STILL HAVE PROBLEMS, PLEASE RETURN THE UNIT, PREFERABLY WITH YOUR CASSETTE RECORDER, AND I WILL GET IT OPERATING PERFECTLY WITHOUT CHARGE. THE REPAIR TURNAROUND TIME IS 1 TO 3 WEEKS.

15. IF YOU ARE COMPLETELY DISSATISFIED, YOU MAY RETURN THE INTERFACE FOR A COMPLETE REFUND WITHIN 90 DAYS AFTER YOU ACCEPTED DELIVERY.
IDEAS FOR USING THE CASSETTE INTERFACE

SAVING AND LOADING PROGRAMS

Programs may be toggled into memory, loaded from paper tape, or loaded by some other means. They may then be dumped onto cassette using the cassette output routine. The output routine itself may be dumped along with the other program, so that it will be available later for further dumps. Routines may then be loaded from cassette, modified, and dumped back out to cassette in a continuous process of development.

USING A BACKUP

A backup is a method of making sure that valuable programs or data is not lost. One simple way of providing a backup is to record a particular program in two different places on the same tape, or on two different tapes. You may want to go back and forth between the two copies, each time the program is changed. In this way, you always have a copy of the last program and only the most recent changes are lost if a power failure or other equipment trouble develops during the process of saving.

STARTING AND STOPPING AUTOMATICALLY

Most audio cassette recorders have an input labeled "remote". This is normally operated from a switch on the microphone, so that the recorder can be started and stopped while dictating. This input can be used to start and stop the recorder under program control from the computer. One of the 4 extra control lines coming from the cassette interface (DIP-socket pins 5, 6, 7, or 8) can be used to drive a relay which would have its contacts connected to the recorder remote jack. The main requirements are that the 40 mA available from the control line be able to drive the relay, and that the relay contacts are able to handle the current into the remote jack, which can be as high as one ampere.

OPERATING WITH MORE THAN ONE CASSETTE READER

Sometimes it is desirable to operate with two or more cassette units. With the Tarbell cassette interface, it is already possible to read from one cassette recorder, while writing onto another. This is because the input and output sections are entirely independent, and may be programmed separately. If it is desired to read from one of two units, and write onto another, such as during a merging operation, relays may be used to switch back and forth between the two input units, under control of one of the 4 control lines. If, however, it is necessary to read simultaneously from two different units, or write different information onto two units at the same time, it will be necessary to have two cassette interfaces.
MODIFICATIONS ON CASSETTE BASIC FOR CSAVE AND CLOAD USING TARBEll CASSETTE INTERFACE

### 8K BASIC 3.2

<table>
<thead>
<tr>
<th>ADDR OLD</th>
<th>NEW</th>
<th>ADDR OLD</th>
<th>NEW</th>
<th>ADDR OLD</th>
<th>NEW</th>
<th>ADDR OLD</th>
<th>NEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1008 06 6E</td>
<td>155A 06 6E</td>
<td>0E5E 06 FA</td>
<td>1DFE DB DB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1009 E6 E6</td>
<td>155B E6 E6</td>
<td>0E59 D2 67</td>
<td>1DFF 06 6E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100A 01 10</td>
<td>155C 01 10</td>
<td>0E5A FA 0E</td>
<td>1E00 E6 E6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100B C2 C2</td>
<td>155D C2 C2</td>
<td>0E5B 67 3E</td>
<td>1E01 60 20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100C 07 07</td>
<td>155E 59 59</td>
<td>0E5C 0E 19</td>
<td>1E02 C2 C2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100D 10 10</td>
<td>155F 15 15</td>
<td>0E5D 78 CD</td>
<td>1E03 FE FE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100E DB DB</td>
<td>1560 DB DB</td>
<td>0E5E CD FD</td>
<td>1E04 1D 1D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100F 07 6F</td>
<td>1561 07 6F</td>
<td>0E5F FA 1D</td>
<td>1E05 F1 F1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1016 06 6E</td>
<td>1568 06 6E</td>
<td>0E61 CD CD</td>
<td>1E07 07 6F</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1017 E6 E6</td>
<td>1569 E6 E6</td>
<td>0E62 FA FD</td>
<td>1E08 C9 C9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1018 80 20</td>
<td>156A 80 20</td>
<td>0E63 1D 1D</td>
<td>1E09 06 06</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1019 C2 C2</td>
<td>156B C2 C2</td>
<td>0E64 C3 C3</td>
<td>1E0A 01 01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101A 15 15</td>
<td>156C 67 67</td>
<td>0E65 74 74</td>
<td>1E0B FE FE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101B 10 10</td>
<td>156D 10 10</td>
<td>0E66 0E 0E</td>
<td>1E0C 2E 2E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101C F1 F1</td>
<td>156E F1 F1</td>
<td>0E67 0E 3E</td>
<td>1E0D CA CA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101D D3 D3</td>
<td>156F D3 D3</td>
<td>0E68 04 10</td>
<td>1E0E 12 12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101E 07 6F</td>
<td>1570 07 6F</td>
<td>0E69 CD D3</td>
<td>1E0F 0E 0E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101F C9 C9</td>
<td>1571 C9 C9</td>
<td>0E6A F0 6E</td>
<td>1E10 E5 E5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1020 E5 E5</td>
<td>1572 E5 E5</td>
<td>0E6B 1D 00</td>
<td>1E11 3E CD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1021 3E CD</td>
<td>1573 3E CD</td>
<td>0E6C B6 00</td>
<td>1E12 D3 FD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1022 D3 14</td>
<td>1574 D3 66</td>
<td>0E6D C2 00</td>
<td>1E13 CD 1D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1023 CD 10</td>
<td>1575 CD 15</td>
<td>0E6E 67 00</td>
<td>1E14 FD 3E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1024 14 3E</td>
<td>1576 66 3E</td>
<td>0E6F 0E 00</td>
<td>1E15 1D 6E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1025 10 6E</td>
<td>1577 15 E6</td>
<td>0E70 0D 00</td>
<td>1E16 CD CD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1026 06 3E</td>
<td>1578 06 3E</td>
<td>0E71 C2 00</td>
<td>1E17 FA FD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1027 03 10</td>
<td>1579 03 10</td>
<td>0E72 69 00</td>
<td>1E18 1D 1D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1028 CD D3</td>
<td>157A CD D3</td>
<td>0E73 0E 00</td>
<td>1E19 D3 D3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1029 07 6E</td>
<td>157B 07 6E</td>
<td>0E74 F7 F7</td>
<td>1E48 06 3E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102A 10 00</td>
<td>157C 10 00</td>
<td>0E75 F7 F7</td>
<td>1E49 03 10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102B D6 00</td>
<td>157D D6 00</td>
<td>0E76 F7 F7</td>
<td>1E4A CD D3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102C 03 00</td>
<td>157E 03 00</td>
<td>0E77 F7 F7</td>
<td>1E4B D3 D3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102D A3 00</td>
<td>157F A3 00</td>
<td>0E78 F7 F7</td>
<td>1E4C D3 D3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102E 15 00</td>
<td>1580 15 00</td>
<td>0E79 F7 F7</td>
<td>1E4D D3 D3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102F 05 00</td>
<td>1581 05 00</td>
<td>0E7A F7 F7</td>
<td>1E4E D3 D3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1030 19 00</td>
<td>1582 19 00</td>
<td>0E7B F7 F7</td>
<td>1E4F D3 D3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 12K BASIC 3.2

### DISK BASIC
The main purpose of this software is to allow development of assembly-language programs. Programs may be entered from the keyboard in standard assembly-language format, edited, assembled, and saved on cassette in either source or machine format. These programs may be then loaded back into memory at some future time for listing, editing or running. Following is a summary of the commands. All address and byte information is in hex.

**FILE /NAME/ AAAA**
- Creates a file of name NAME at address AAAA (hex).
- Up to six text files can be in memory at the same time. The file just created is the current file.

**FILE /NAME/ 0**
- This command deletes the file named NAME.

**FILE /NAME/**
- Makes the named file the current file.

**FILE**
- Prints the starting and ending addresses of the current file, and it's name.

**FILES**
- Lists the name, starting & ending address for each of the files in the system. Current file at top.

**DUMP SSSS EEEE**
- Dumps the contents of memory from SSSS to EEEE.

**EXEC AAAA**
- Executes the machine-language program at address AAAA.

**EXEC /NAME/**
- Searches the cassette for named file, loads and exec.

**ENTR AAAA**
- Enter data into memory: B1 B2 B3 B4/

**LIST NNNN**
- Lists the current file starting at line NNNN.

**DELT NNNN MMTT**
- Deletes lines NNNN through MMTT from current file.

**NNNN (text entered)**
- Enter a line of text into the current file. Always use 4 decimal digits. It works similar to BASIC, in that the numbered lines are ordered automatically.

**ASSM AAAA**
- Assembles current file into address AAAA.

**ASSM AAAA BBBB**
- Assembles at address AAAA, but puts code at BBBB.

**CUST**
- Execute a customer (user) routine at address EOOO.

**SFIL**
- Save the current file on cassette.

**LFIL**
- Load the current file from cassette. The name of the file on tape must match the name of the current file. Be sure there is enough room in memory.

**CFIL**
- Check a cassette file written with the SFIL command above, for errors, without overwriting current file.

**AFIL**
- Append a file from cassette to the current file.

**SAVE SSSS EEEE**
- Save a block of memory from SSSS to EEEE on cassette.

**LOAD SSSS**
- Load a block of memory from cassette starting at SSSS.

**CHEK SSSS**
- Check a cassette file written with SAVE command.

**NAME /NAME/**
- Rename the current file to NAME.

**RNUM N**
- Renumber the current file by increment N.

**NLIS NNNN**
- Lists the current file at NNNN without line numbers.

This software is useful, but may not be completely free of bugs. It loads into the first 4096 bytes of memory, and uses about 2048 bytes following, for tables and scratch pad. Ctrl-C escapes from any printing. Ctrl-W freezes printing until another character is typed. The package is available from Tarbell Electronics for the prices listed below:

Cassette Tape with software and instructions: $5.00
A reassembled and patched (updated) listing: $5.00
Writing Programs for the Cassette Interface

Sometimes it is necessary to write assembly or machine language programs for a particular interface. The class of programs we are talking about here are called "drivers". These would be required for linkage to a piece of software for which patches are not provided in the manual.

This interface is a synchronous device. One of the implications of this is that data or programs are most efficiently written as a contiguous block, rather than as separate bytes. There are a few rules that must be followed when writing software for this device:

1. The first byte must be a "start byte" which may be any byte except 00, FF, or E6 (hexadecimal).
2. The second byte must be a "sync byte" which must be E6 (hex).
3. The software must be able to deliver bytes to the interface as fast as it can accept them, which at the standard speed is 167 bytes per second. This means that any loop that the program goes through which is between bytes must last less than 5.3 milliseconds. An average instruction time on the 8080 with no wait states is 2 cycles, or 1 microsecond. Thus, there should be no more than about 5300 instruction executions between bytes being sent out. This is normally not a problem.
4. A similar constraint must be observed with respect to the input software, which should be able to accept data as fast as it is being made ready by the interface.

There are situations in which the data cannot be provided or accepted fast enough by the software. One example of this is data which is being generated by a program running in BASIC. This problem is solved by sending the data out and reading it in a line at a time, with nulls in between. The page entitled "How to Save and Load Data from a BASIC Program" is a sample program that shows how to do this. There are at least two other ways to handle this problem: 1) Send each byte as a separate block with its own start and sync bytes. 2) Accumulate bytes in a buffer area of memory, and start and stop the cassette recorder under control of the computer when it is time to dump and refill the buffer.

Other items to be considered when writing software:

1. Since tape is an imperfect medium, it is generally useful to incorporate an error-checking scheme, such as the checksum system that is demonstrated on the pages entitled "Output Routine with Checksum" and "Input Routine with Checksum". There are many systems possible, each providing different kinds and levels of protection, and books have been written about these.
2. It is sometimes handy to have an identifier, such as a name, written along with the file onto cassette. An example is the one-letter name given to ALTAIR BASIC programs when writing to cassette.
3. Another item that is useful is a way for the program to tell how long the file is. A one or two-byte header to indicate length of a file or block is sometimes used. In the modified processor technology software package I support, blocks are preceded by one length byte that may be from 1 to 255. A length of zero indicates the end of the file.
4. A "type byte" is a unique byte for a particular format of file, so that a sophisticated loader may distinguish between different types, and load them appropriately. I use a 90 (hex) type byte immediately following the sync byte on the Proc. Tech. software.
USING THE TARBELL CASSETTE INTERFACE UNDER INTERRUPT CONTROL

There is no built-in provision for interrupts in the design. However, it is not too difficult to make a modification to the board to provide for interrupts. Two bits in the control line register are used as interrupt-enable flip-flops. It is necessary to install another 7403 quad 2-input open-collector NAND gate in one of the spare IC positions at the top of the board. The circuit diagram below shows how to connect the 7403:

If you do not have a vectored interrupt card, connect the output pins 3 and 6 to the edge connector pin 75. An interrupt will then be caused when the following conditions are true: 1) Interrupts are enabled with an EI instruction. 2) Control bit 0 is high and the receiver is ready with an input byte; or control bit 7 is high and the transmitter is ready for an output byte. Both conditions 1 and 2 must be true for an interrupt to be caused.

If you do have a vectored interrupt card, connect the output pins 3 and 6 to one of the edge connector pins 4, 5, 6, 7, 8, 9, 10, or 11, depending on what level you want the cassette interface. Be sure that you have the correct polarity to cause an interrupt. The output at pins 3 and 6 goes low to cause an interrupt. If you have a vectored interrupt card that requires a high-going signal to cause an interrupt, you may use the extra gate shown above to invert the line. Read your manual on your interrupt board to make sure.

There is not enough room here to explain how to use the interrupt system, but the following lines show how to enable and disable the two interrupt bits on this board.

<table>
<thead>
<tr>
<th>Enable Receiver Interrupt:</th>
<th>MVI A,1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OUT 6EH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enable Transmitter Interrupt:</th>
<th>MVI A,2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OUT 6EH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enable Both Interrupts:</th>
<th>MVI A,3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OUT 6EH</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Disable Both Interrupts:</th>
<th>MVI A,0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OUT 6EH</td>
</tr>
</tbody>
</table>
We have purchased the introductory variable-speed PHI-DECK package, which includes the deck, control electronics, power supply, and control box. We have been experimenting with an adapter which connects between the Tarbell Cassette Interface, the read-write heads, and the control electronics. This adapter provides start-stop, forward, rewind, and fast-forward control for the PHI-DECK from the Tarbell Cassette Interface. It also includes read-write electronics for one channel. It is still in the experimental stage, but speeds of 1000 bytes per second have been attained. The information is being provided here for those who would like to experiment further along these lines. The circuit is not guaranteed to work for your application, and will probably not be exactly what we end up with for our use. As progress is made, further refinements will be published, including more supporting software.

The simple program below is handy for experimenting. Flip sense switch 2 up momentarily to pulse the control line.

```
LOOP IN FFH DB FF Read Sense Switches.
OUT 6EH D3 6E Write to Control Port.
IN 6EH DB 6E Read Status Lines.
OUT FFH D3 FF Write To Display Lights.
JMP LOOP C3 00 00 Do it all over again.
```

Control Table

<table>
<thead>
<tr>
<th>s1</th>
<th>s0</th>
<th>function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>RUN</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>STOP</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>FF</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>REW</td>
</tr>
</tbody>
</table>

Sense switch 3 should be up for read, down for write.
**74L73/7473 T-K Flip-Flop with Clear**

Inputs
- CLR
- CK
- J
- K

Outputs
- Q
- \( \overline{Q} \)

<table>
<thead>
<tr>
<th>CLR</th>
<th>CK</th>
<th>J</th>
<th>K</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>0, 1</td>
<td>Q, ( \overline{Q} )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>NO CHANGE</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1, 0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0, 1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( \overline{Q} ), Q</td>
<td></td>
</tr>
</tbody>
</table>

No change in output until falling edge of clock

X = Don't Care

**74L74 D-Type Flip-Flop with Preset and Clear**

Inputs
- CLR
- CK
- PR

Outputs
- Q
- \( \overline{Q} \)

<table>
<thead>
<tr>
<th>CLR</th>
<th>PR</th>
<th>CK</th>
<th>D</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0, 1</td>
<td>Q, ( \overline{Q} )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>0, 1</td>
<td>Q, ( \overline{Q} )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>0, 1</td>
<td>ILLEGAL</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1, 0</td>
<td></td>
</tr>
</tbody>
</table>

No change in output until rising edge of clock

X = Don't Care
74LTS 4-BIT BISTABLE LATCH

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>OUTPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENB</td>
<td>D</td>
</tr>
<tr>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

X: DON'T CARE
Q0: THE STATE OF Q BEFORE ENB IS SET LOW

74L86 EXCLUSIVE-OR

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Y = \overline{A \oplus B} = A \bar{B} + \bar{A}B

+5 VDC
SER* SERIAL DATA INPUT. DATA IS MOVED FROM SERIAL INPUT TO QA AFTER THE FIRST CLOCK.

A-B* PARALLEL DATA INPUT. DATA IS PRESENT ON ASSOCIATED OUTPUT (A→QA) AFTER CLOCK. IF MODE IS SHIFTED FROM PARALLEL, DATA WILL MOVE TOWARDS QE ON EACH FOLLOWING CLOCK.

ENB P* ENABLE PARALLEL MODE. WHEN SET, REGISTER IS IN PARALLEL Mode. WHEN RESET, REGISTER IS IN SERIAL Mode.

CK* CLOCK. NO CHANGE IN OUTPUT UNTIL RISING EDGE OF CLOCK.

CLR* CLEAR. WHEN RESET ALL OUTPUTS ARE FORCED TO RESET WITHOUT REGARDS TO ANY OTHER INPUT.

QA-QE* OUTPUTS. ALL OUTPUTS USED FOR PARALLEL OUTPUT. FOR SERIAL OUTPUT, ONE OF THE OUTPUTS IS USED DEPENDING ON THE LENGTH DESIRED.

7476 6-BIT SHIFT REGISTER

IF B1 THROUGH B6 ARE BIT-FOR-BIT EQUAL TO THE BITS RESET ON T1 THROUGH T6, Q IS RESET. FOR ALL OTHER CONDITIONS Q IS SET

5* STROBE. WHEN SET Q IS DISABLED

DMB131 6-BIT COMPARATOR
A, B = SERIAL DATA INPUTS. ONE OF THE TWO MUST BE SET TO ENABLE THE OTHER. DATA IS PRESENT AT QA AFTER ONE CLOCK AND MOVES TOWARD QH ON EACH FOLLOWING CLOCK.

CK = CLOCK. NO CHANGE IN OUTPUT UNTIL RISING EDGE OF CLOCK.

CLR = CLEAR. WHEN RESET, ALL OUTPUTS ARE RESET WITHOUT REGARDS TO OTHER INPUTS.

QA - QH = OUTPUTS. DATA IS MOVED FROM A AND B TO QA ON EACH CLOCK AND THE DATA AT QA IS MOVED TOWARDS QH.

74L164 8-BIT PARALLEL-OUT SHIFT REGISTER

NE555 TIMER
ST20 BIDIRECTIONAL ONE-SHOT

INPUT SIGNAL

REFERENCE

LIMITER OUTPUT A

LIMITER OUTPUT A

ONE-SHOT OUTPUT Q

ONE-SHOT OUTPUT Q

ONE-SHOT OUTPUT Q

ONE-SHOT OUTPUT Q

PIN 2 SET
PIN 13 RESET

PIN 2 RESET
PIN 15 SET

PIN 2 SET
PIN 13 SET

ONE-SHOT ON TIME IS DETERMINED
BY TIMING COMPONENTS CONNECTED
TO PINS 12, 14, AND 15.

ST20 BIDIRECTIONAL ONE-SHOT INPUT
AND OUTPUT WAVEFORMS
DEAR CUSTOMER,

THANK YOU VERY MUCH FOR PURCHASING A CASSETTE INTERFACE.
I AM INTERESTED TO KNOW WHAT KIND OF PROGRESS YOU HAVE MADE
WITH YOUR TARBEll CASSETTE INTERFACE. I WOULD REALLY APPRECIATE
IT IF YOU WOULD TAKE TIME TO FILL THIS QUESTIONNAIRE OUT. THIS
WILL HELP ME TO PROVIDE YOU WITH BETTER SERVICE IN THE FUTURE.

DID YOUR INTERFACE ARRIVE IN A REASONABLE LENGTH OF TIME? YES NO
WERE ANY OF THE ITEMS DAMAGED IN SHIPMENT? YES NO
WERE ANY OF THE ITEMS MISSING? IF SO, WHAT? YES NO
WAS THE QUALITY OF WORKMANSHIP ON THE BOARD REASONABLE? YES NO
HAVE YOU STARTED CONSTRUCTION YET? YES NO
HAVE YOU HAD ANY PROBLEM UNDERSTANDING THE MANUAL? WHERE? YES NO
WERE THE COMPONENTS OF REASONABLE QUALITY? WHAT WASN'T? YES NO
HAVE YOU COMPLETED THE CONSTRUCTION YET? YES NO
HAVE YOU TESTED THE INTERFACE YET? YES NO
WHAT IS THE MAKE AND MODEL OF YOUR RECORDER?
HAVE YOU ENCOUNTERED ANY PROBLEMS? WHAT? YES NO
IF THERE WERE PROBLEMS, ARE THEY FIXED? IN WHAT WAY? YES NO
HAVE YOU FOUND ANY OF THE ADJUSTMENTS CRITICAL? WHICH? YES NO
ARE YOU DISSATISFIED IN ANY WAY WITH THE UNIT? HOW? YES NO
DO YOU HAVE ANY SUGGESTIONS FOR IMPROVEMENT OF THE DESIGN YES NO
WHAT DO YOU FEEL IS THE MAJOR DISADVANTAGE OF THIS UNIT?
HAVE YOU DEVELOPED ANY SOFTWARE OR HARDWARE RELATIVE TO THE
INTERFACE THAT MIGHT BE OF USE TO OTHER PEOPLE? WHAT?
HAVE YOU TRIED PACKING DENSITIES HIGHER THAN THE STANDARD
800 BITS/INCH? WHAT DENSITY? WITH WHAT RESULTS?
WHAT IS YOUR NAME, ADDRESS, AND PHONE NUMBER?
DO YOU MIND IF I GIVE OUT YOUR NAME TO OTHER CASSETTE USERS?
PLEASE USE THIS SPACE, AND THE REVERSE SIDE, IF NECESSARY, TO
MAKE ANY COMMENTS ON THE INTERFACE THAT MAY BE HELPFUL.

SINCERELY,

Donald E. Tarbell
DONALD E. TARBEll
144 MIRALESTE DRIVE #106
MIRALESTE, CALIF. 90732
TARBELL MODE TAPES

P.T. SOFTWARE#1 : BLOCK LENGTH 1000H
STARTING ADDRESS 0000H
NEEDS A 1024 BYTE BLOCK OF MEMORY @ 1000H
WHEN USED AS ASSEMBLER, BE SURE TO INCLUDE IN THE PROGRAM THE LINES
PSW EQU 6
SP EQU 6

IMSAI 8K BASIC : BLOCK LENGTH 2000H
STARTING ADDRESS 0000H
I/O SETUP
STATUS PORT 3 0
DATA PORT 2 1
THE BIT 7 BIT 7
RDA BIT 1 BIT 0

IMSAI: BITS ACTIVE HIGH
THESE TAPES: BITS ACTIVE LOW
THE TAPES SUPPLIED ARE ON THE OLD MITS STANDARD.
TO CONVERT TO THE IMSAI SUBSTITUTE AS FOLLOWS:

ADDRESS   IMSAI   THESE
004A   03   00
004E   03   00
0047   03   00
0049   02   01
004A   02   01
18F6   02   01
18F8   02   01
18F9   CA   C2
1929   03   00
192A   1F   07
192B   02   DA
1930   02   01
1A0F   03   00
1A11   02   01
1A12   02   01
1A14   02   01

IMSAI ASSEMBLER, EDITOR AND MONITOR
IMSAI SOFTWARE DEVELOPMENT PACKAGE
AS IMSAI CONFIGURED SOFTWARE
AND
ARE PRESENT ON THE TAPES

BASIC PROGRAMS:
12K PROGRAMS RUN ON MITS 12K VERSION 3.2
8K CHESS RUNS ON MITS 8K V 3.2
CHECKERS RUNS ON MITS 8K V 3.1.

ALL ARE LOADED IN VIA THE LOAD COMMAND. TO MODIFY YOUR MITS BASIC,
CONSULT YOUR TARBELL MANUAL FOR THE PATCHES.
INSTRUCTIONS FOR CASSETTE READ-ONLY MEMORY PROGRAM 1 (ROMP!)
AUG 12, 1976

THIS PROGRAM, WHICH WILL RUN IN EITHER READ-ONLY OR READ-WRITE MEMORY, ALLOWS THE USER TO PERFORM THE 6 FUNCTIONS LISTED BELOW:

<table>
<thead>
<tr>
<th>LETTER EXAMPLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>E  E3000</td>
<td>EXECUTE PROGRAM AT 3000 (HEX).</td>
</tr>
<tr>
<td>C  C100020E7</td>
<td>CHECK CHECKSUM OF CASSETTE RECORD.</td>
</tr>
<tr>
<td>S  S</td>
<td>GENERATE SYNC STREAM.</td>
</tr>
<tr>
<td>O  0200000000</td>
<td>OUTPUT RECORD OF LENGTH 2000, STARTING AT 0000.</td>
</tr>
<tr>
<td>I  I10003000</td>
<td>INPUT RECORD OF LENGTH 1000, STARTING AT 3000.</td>
</tr>
<tr>
<td>L  L47DF1257</td>
<td>LOAD AND EXECUTE, LENGTH 47DF, STARTING AT 1257.</td>
</tr>
</tbody>
</table>


THE SECOND AND THIRD BYTES OF THIS PROGRAM SPECIFY WHERE THE STACK IS LOCATED, AND SHOULD BE THE TOP OF SOME RANDOM-ACCESS MEMORY (NOT READ-ONLY MEMORY) WHICH IS NOT USED FOR OTHER THINGS. THE STACK WILL THEN EXPAND DOWN FOR ABOUT 8 OR 10 BYTES FROM THIS ADDRESS.
FF00 31 00 EC
FF03 3E 0D
FF06 3E 0A
FF0A CD 58 FF
FF0D 3E 3F
FF0F CD 5C FF
FF12 CD 64 FF
FF15 FF 45
FF17 CC 36 FF
FF1A FE 43
FF1C CC 80 FF
FF1F FE 53
FF21 CC 73 FF
FF24 FE 4F
FF25 CA 7B FF
FF29 FE 49
FF2B CC 80 FF
FF2E FE 4C
FF30 CC 30 FF
FF33 CC 00 FF
FF36
FF36 CD 3A FF
FF39 C9
FF3A
FF3A 21 00 00
FF3D 0E 04
FF3F CD 64 FF
FF42 2F
FF43 2F
FF44 29
FF45 29
FF46 CD 50 FF
FF49 C5
FF4A 6F
FF4B 0D
FF4C C2 3F FF
FF4F C9
0005 *
0010 *** READ-ONLY MEMORY PROGRAM ***
0015 *
0020 START LXI SP,0E00H SET STACK PTR.
0025 LXI A,DOH PRINT CR, LF.
002A CALL PTCN
0030 CALL PTCN
003B HN CONSOLE.
003C HN CONSOLE.
0040 CALL PTCN
0045 HN CONSOLE.
0049 CALL RDcn READ KEYBOARD.
0054 CPI 'E' IF E,
0059 CPI 'C' IF C,
0064 CPI 'S' IF S,
0069 CPI 'G' IF G,
006E CPI 'T' IF T,
0073 CPI 'L' IF L,
0078 CPI 'D' IF D,
007D CPI 'K' IF K,
0082 CPI 'Z' IF Z,
0087 CPI 'X' IF X,
008C CPI 'C' IF C,
0091 CPI 'I' IF I,
0096 CPI 'T' IF T,
009B CPI 'P' IF P,
009C CPI 'A' IF A,
00A1 CPI 'O' IF O,
00A6 CPI 'U' IF U,
00B1 CPI 'E' IF E,
00B6 CPI 'J' IF J,
00C1 CPI 'X' IF X,
00C6 CPI 'S' IF S,
00D1 CPI 'L' IF L,
00D6 CPI 'D' IF D,
00E1 CPI 'K' IF K,
00E6 CPI 'Z' IF Z,
00F1 CPI 'T' IF T,
00F6 CPI 'C' IF C,
00FA CALL AHEX READ ADDRESS FROM KB.
0104 PCHL JUMP TO IT.
0109
0110 *** EXECUTE THE PROGRAM AT THE ADDRESS ***
0115 *
0120 EXEC CALL AHEX READ ADDRESS FROM KB.
0125 PCHL JUMP TO IT.
0130 CALL AHEX READ ADDRESS FROM KB.
0135 LXI H,4 GET A 16-BIT ZERO.
0140 LXI C,0 COUNT OF 4 DIGITS.
0145 CALL RDCN READ A BYTE.
0150 DAD H SHIFT 4 LEFT.
0155 DAD H
0160 DAD H
0165 DAD H
0170 DAD H
0175 DAD H
0180 CALL AHSI CONVERT TO BINARY.
0185 ADD L
0190 ADD H
0195 ADD L
01A0 ADD H
01A5 ADD L
01B0 ADD H
01B5 ADD L
01C0 CALL AHSI CONVERT TO BINARY.
01C5 ADD L
01D0 ADD H
01D5 ADD L
01E0 ADD H
01E5 ADD L
01F0 CALL AHSI CONVERT TO BINARY.
01F5 ADD L
0200 ADD H
0205 ADD L
0210 ADD H
0215 ADD L
0220 CALL AHSI CONVERT TO BINARY.
0225 ADD L
0230 ADD H
0235 ADD L
0240 CALL AHSI CONVERT TO BINARY.
0245 ADD L
0250 ADD H
0255 ADD L
0260 ADD H
0265 ADD L
0270 CALL AHSI CONVERT TO BINARY.
0275 ADD L
0280 ADD H
0285 ADD L
0290 CALL AHSI CONVERT TO BINARY.
0295 ADD L
02A0 ADD H
02A5 ADD L
02B0 ADD H
02B5 ADD L
02C0 CALL AHSI CONVERT TO BINARY.
02C5 ADD L
02D0 CALL AHSI CONVERT TO BINARY.
02D5 ADD L
02E0 CALL AHSI CONVERT TO BINARY.
02E5 ADD L
02F0 CALL AHSI CONVERT TO BINARY.
02F5 ADD L
0300 *** PRINT REGISTER A ON CONSOLE ***
0305
0310
0315
0320
0325
0330
0335
0340
0345
0350
0355
0360
0365
0370
0375
0380
0385
0390
0395
03A0
03A5
03B0
03B5
03C0
03C5
03D0
03D5
03E0
03E5
03F0
03F5
FFBC F1  9365  P6P  PSW   GET CONTROL CHAR.
FFBD E5  9370  PUSH H   SAVE STARTING ADDRESS
FFBE F5  9375  PUSH PSW  UNDER CONTROL CHAR.
FFBF 06 00  9380  MVI B,0   SET CHECKSUM = 0.
FFC1 CD F2 FF  9390  CALL PSW   READ A BYTE FROM CASS.
FFC4 4F    9400  MVI C,A   SAVE IT IN REG C.
FFC5 F1    9410  P6P PSW  GET CONTROL CHAR.
FFC6 F5    9420  PUSH PSW  SAVE IT BACK.
FFC7 FE 43  9430  CPI 'C' IS IT A C?
FFCA CA 00 FF  9440  JZ  CIN0 IF C .. DON'T STORE IT.
FFCD 77    9450  JNZ CIN0 IF NOT .. STORE IT.
FFCE 60    9460  ADD B   ADD TO CHECKSUM.
FFCF 47    9470  MVI B,A   INCREMENT P0INTER.
FFD0 23    9480  INX H   INCREMENT POINTER.
FFD1 13    9490  DEX D   DECREMENT COUNTER.
FFD2 97    9500  SUB A   CLEAR A.
FFD3 3A    9510  CMP D   IF D NOT = 0,..READ MORE.
FFD4 C2 01 FF  9520  CMP E   IF E NOT = 0,..READ MORE.
FFD6 C2 01 FF  9530  CALL PSW   READ LAST BYTE.
FFD8 CD F2 FF  9540  CMP B   COMPARE TO CHECKSUM.
FFDF E5 05  9550  MVI A,'E' PRINT E FOR ERROR.
FFE1 C2 0C FF  9560  JNZ CERR PRINT NOW IF ERROR.
FFE4 F1    9570  P6P PSW  RECOVER CTL CHAR.
FFE5 FE 40  9580  CPI 'L' IF IT'S NOT L,..DON'T EXECUTE.
FFE7 C2 0C FF  9590  JNZ CERR ELSE EXECUTE.
FFEA E1    9600  P6P H   ELSEWISE, EXECUTE.
FFEB E9    9610  PHL   AT STARTING ADDRESS.
FFEC CD 5E FF  9620  CERR CALL PTCN PRINT C,E, OR I.
FFEF C3 00 FF  9630  JMP START.
FFF2 D3 6E  9640  CIN IN CASC READ STATUS.
FFF4 E5 10  9650  ANI 10H LOOK AT BIT 4.
FFF6 C2 F2 FF  9660  JNZ CIN WAIT UNTIL LOW.
FFF9 DB 6F  9670  IN CASS READ DATA FROM CASS.
FFFB C9    9680  RET RETURN FROM CIN.
FFFC 9996 CASD EQU 6FH CASSETTE DATA PORT.
FFFC 9997 CASC EQU 6EH CASSETTE STATUS PORT.
FFFC 9998 PSW EQU 6   
FFFC 9999 SP EQU 6   
FFFC