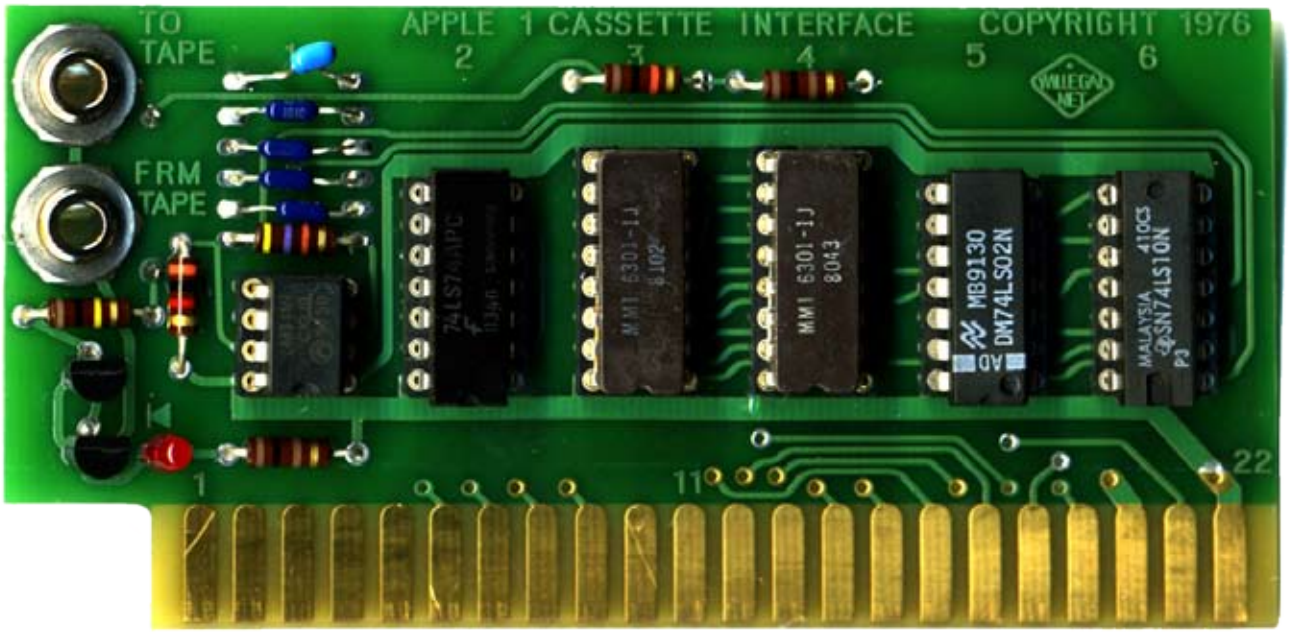



# Mimeo Cassette Interface Assembly and Operations Guide

**By Mike Willegal**

**www.willegal.net**

**version 0.15**

## CAUTION!!!

Incorrect assembly or connecting of the Mimeo Cassette Interface can cause fatal damage to the interface and/or the motherboard. Double and triple check your connections before powering on. Pay special attention to orientation of the card when you plug it into the motherboard's expansion slot. Plugging it in backwards will result in damage to the card and/or motherboard.

## Forward

In the mid 1970's, around the time the Apple 1 was developed, the only reasonably affordable interface for home computer hobbiyist was repurposing an ordinary cassette recorder as a data storage device. Soon after the introduction of the Apple 1, Apple Computer released the Apple Cassette Interface (ACI) for the Apple 1. This small card had a list price of $75 and turned out to be the only peripheral card that Apple ever released for the Apple 1. The Mimeo Cassette Interface is a clone of Apple's original ACI, duplicating the form, fit and function of the original ACI in exacting detail. This manual refers to the board as the ACI, since that is the name Apple used and for all intents and purposes the implementations and operation are identical.

## Reliability

Reliability of the ACI card, in it's stock form, is not very good by 21st centrury standards. Apple made improvements to the cassette interface circuit when it came out with the Apple II. I have spent considerable effort looking for improvements in reliablity, without altering the design of the original ACI printed circuit board. In the end, I have found three items that can help with reliability,

Two of these involve changing component values. I am providing parts that can used to build the board with either original component values or with components that will improve reliabilty. Because of the reliability problems with the original design, I recommend building this kit with the components that improve reliability. Look and feel of the board is not affected but reliability is improved quite a bit. Even with these changes, reliablity is not perfect, but the system will be more reliable.

The third reliability improvement I recommend, is using an Apple recommended cassette recorder. I have had great difficulty with a different vintage cassette recorder that works quite well with an Apple II. The good news is that the original Apple recommended recorder happens to remain on the market. The is a Panasonic RQ-2102. There may be other cassette recorders that perform as well or better than the RQ2102, but I don't have the time or resources to investigate the possibilities.

# *Chapter 1 – Assemble Components, Tools, and Equipment*

### 1. Recommended Tools and Equipment

- Quality soldering station - I use a Weller WES51. Whatever you use, I recommend that it has some kind of temperature controlled tip. This will help prevent damage to the PCB when soldering. Soldering irons that do not have a tempurature controlled tip can overheat and damage the PCB or component being soldered
- Solder - use quality solder - thinner solder is vastly easier to work with than fat solder. The fat stuff sold at hardware stores is not suitable for these sort of electronics projects
- Wire cutters – for trimming component leads and cutting wire to length
- Wire strippers - for stripping ends of jumper wire
- Your favorite PCB cleaning agent - Isopropyl Alcohol will dissolve many kinds of soldering resin. Windex will also help with cleaning PCBs
- Ohm meter - to check for good connections and shorts
- Logic probe or oscilloscope – handy if you are having trouble with bring up
- Your host computer schematics or hardware interfacing guide – Direction for connecting to Mimeo 1 computers are provided in this manual

### 2. Additional Components (not included)

- Cassette Recorder - I strongly recommend the Panasonic RQ-2102
- Cassette Tapes - ordinary 30 or 60 minute tapes work well
- Two mono to mono 1/8" audio cables. One end plugs into the ACI, the other into jacks on the cassette recorder

## 3. Compare Received Components With Parts List

Examine and identify all parts provided with the kit.

| PART | DESCRIPTION | QUANTITY | PRESENT |
|---|---|---|---|
| 16 pin socket | For PROMs | 2 | |
| 14 pin socket | For 74LS parts | 3 | |
| 8 pin socket | For LM311 | 1 | |
| LM311 | Voltage comparator | 1 | |
| 74LS02 | Quad 2 input nor gate | 1 | |
| 74LS10 | Triple 3 input nand gate | 1 | |
| 74LS74 | Dual D type flip flop | 1 | |
| 7474 | Reliability improvment replacement for 74LS74 | 1 | |
| 6301 - APPLE-A3 | 256x4 PROM - location A3 | 1 | |
| 6301 - APPLE-A4 | 256x4 PROM - location A4 | 1 | |
| .01uF capacitor | Input coupling capacitor | 1 | |
| .1uF capacitor | Reliability improvement replacement for .01uF | 1 | |
| 100 ohm | brown-black-brown Low part of voltage divider for tape output & current limiter for LED input monitor | 2 | |
| 3K resistor | orange-black-red Voltage comparator feedback | 1 | |
| 10K resitor | brown-black-orange-gold High part of voltage divider for tape output | 1 | |
| 10K 1% resistor | brown-black-orange-black-brown Voltage dividers for inputs to voltage comparator | 4 | |
| 47K resistor | yellow-violet-orange Voltage comparator feedback | 1 | |
| 100K resistor | brown-black-yellow Sense resistor for input monitor LED | 1 | |
| PCB | Printed circuit board | 1 | |
| MPS3704 | Sense transistors for input monitor LED | 2 | |
| RED LED | Read level indicator | 1 | |
| Audio Jacks | Switchcraft #41 | 2 | |
| 1’ jumper wire | For jumpering from audio jack to PCB | 1 | |
| **PARTS COUNT** | | **32** | |
| **COUNT OF TYPES** | | **23** | |

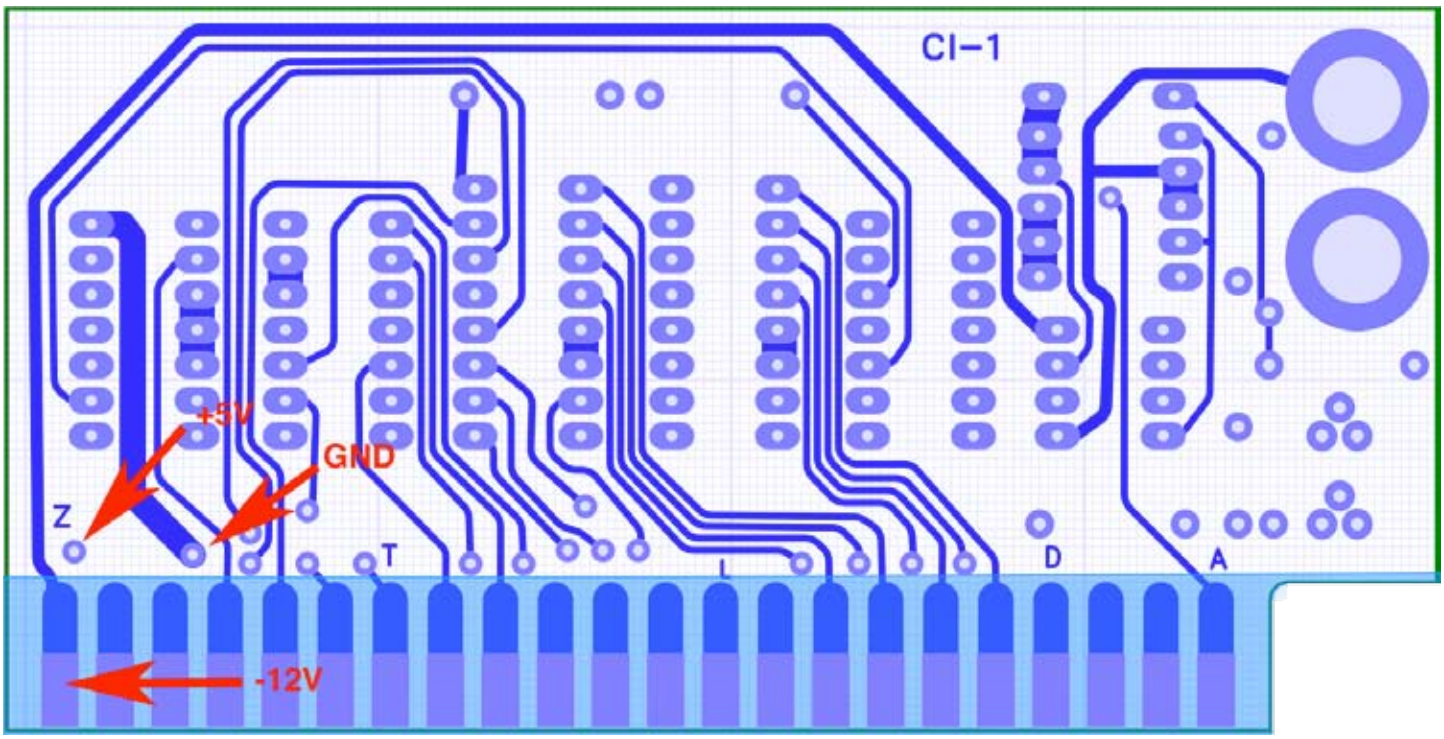
## *Chapter 2 – Solder In Components*

### 1. Overview

The key thing here is to check orientation and make sure that you don't put the sockets or transistors in wrong. For the IC sockets, make sure that the parts are oriented correctly with pin 1 of the socket or chip near the edge of the PCB that contains the gold fingers. All components go on the front of the board (the side with the words "Apple Cassette Interface 1" etched in copper.

Make sure the socket or chip is fully seated. I accomplish this by resting the socket upside down on a small object with the board on top. The weight of the board should keep the socket or chip completely seated. Then tack down a couple of corner pins and recheck orientation and seating. Then finish soldering the rest of the pins.

Take your time and enjoy the process, double checking orientation of devices as you go. The red or blue arrows indicate places to pay special attention when placing components.

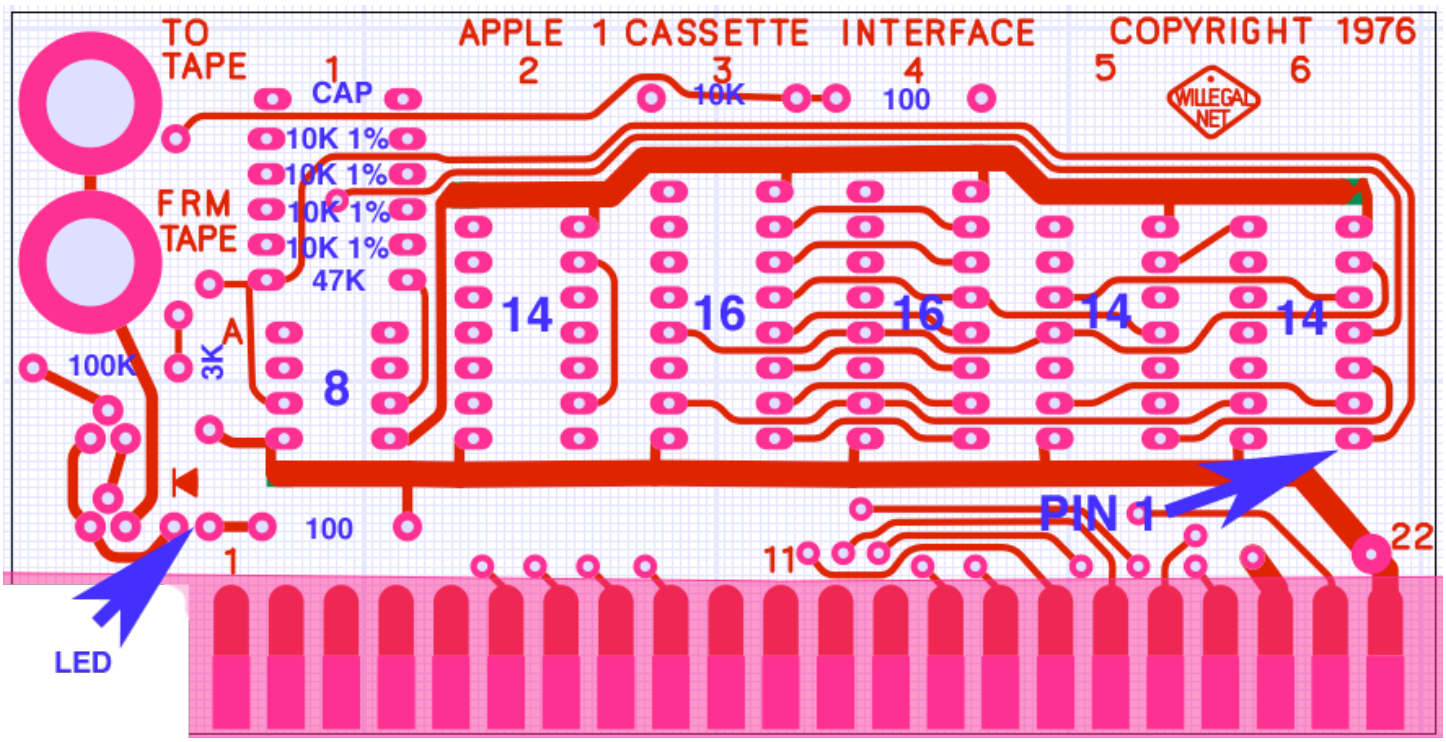



back side of board

### 2. Check for Power and Ground Shorts on PCB

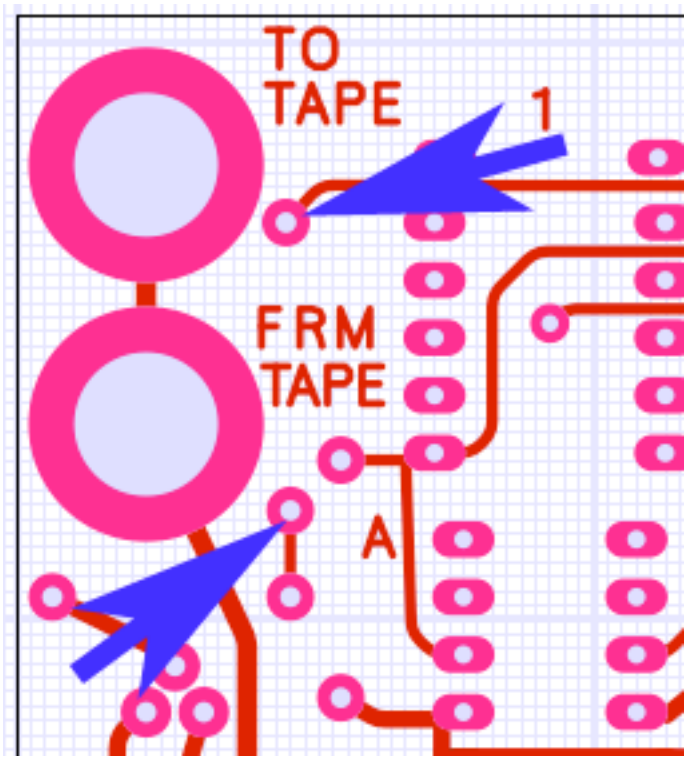
Easiest way to do this is to use an ohm-meter to make sure that there is no connection between +5 volts, -12 volts and ground. The Ohm meter should show no connections between any of these nets. A convenient place to use to check for shorts, is this area on the back of the board (red arrows above).

## 3. Solder in All Components Except 1/8” Phono Jacks




front view of board (components are mounted on front side of board)

| PART | DESCRIPTION | QUANTITY | COMPLETE |
|---|---|---|---|
| 16 pin socket | A-3 and A-4 - pin 1 toward gold finger edge | 2 | |
| 14 pin socket | A-2, A-5, A-6 - pin 1 toward gold finger edge | 3 | |
| 8 pin socket | A-1 - pin 1 toward gold finger edge | 1 | |
| capacitor | Input coupling capacitor - topmost device in row of components at A-1. Use .1uF (104) capacitor for better read reliability. Use .01uF (103) capacitor to exactly replicate original design. | 1 | |
| 100 ohm | brown-black-brown Top of row at A4 | 2 | |
| 100 ohm | brown-black-brown Next to gold fingers in row A1 | | |
| 3K ohm | orange-black-red Vertically mounted - left of 8 pin dip in row A-1 | 1 | |
| 10K resitor | brown-black-orange-gold Top of row at A-3 | 1 | |
| 10K 1% resistor | brown-black-orange-black-brown Four in a row below cap in row A-1 | 4 | |
| 47K resistor | yellow-violet-orange Just above 8 pin dip in row A-1. | 1 | |
| 100K resistor | brown-black-yellow just below two 1/8” jacks | 1 | |
| MPS3704 | Below two 1/8” jacks - flat side toward top of board (middle pin goes in hole closer to top of board) | 2 | |
| RED LED | Long lead (anode) on right | 1 | |




### 4. Install 1/8” Phono Jacks

After mounting the jacks, a short wire must be connected from tab on jack to PCB hole to connect read and write circuits to the jacks. See the illustrations above for locations.

| PART | DESCRIPTION | QUANTITY | COMPLETE |
|---|---|---|---|
| Read and Write Jacks | The jack is mounted with the recepticle facing the front of the board (the same side as the components). Firmly tighten the nut, but not so tight that you risk damaging the PCB. Ground is through this connection. Cut two short lengths of wire that are long enough to reach from the tab on jack to the hole in the PCB. Strip ends of short wire. If you prefer, you can strip insulation from entire length of wire. Original ACIs had no isulation on these short lengths of wire. From the back of the board, solder one end to tab on jack and the other to the appropriate hole in the PCB. There are two tabs. Be sure to connect the wire to the tab that connects to the tip of the plug. | 2 | |

### 5. Recheck for Power and Ground Shorts on PCB

Easiest way to do this is to use an ohm-meter to make sure that there is no direct connection between +5 volts, -12 volts and ground. With the resistors now soldered in, you should note about 9.6K ohms resistance between +5 volts and ground. -12 volts should have no connectivity with either +5 volts or ground.

### 6. Install ICs

| PART | DESCRIPTION | QUANTITY | COMPLETE |
|---|---|---|---|
| LM311 | 8 Pin Socket at A-1. Pin 1 toward gold fingers | 2 | |
| 74LS74 | 14 Pin Socket at A-2. Pin 1 toward gold fingers. Use 74LS74 for original performance. Because of the floating inputs, a replacement 7474 is provided as an optional substitute part and should provide for slightly more reliable operation. A 7474 with floating inputs has more predictable behavior than a 74LS74. | | |
| PROM A-3 | 16 Pin Socket at A-3. Pin 1 toward gold fingers. Prom is printed with "APPLE A-3" on top of the package and has an A3 label on the bottom. | | |
| PROM A-4 | 16 Pin Socket at A-4. Pin 1 toward gold fingers. Prom is printed with "APPLE A-4" on top of the package and has an A4 label on the bottom. | | |
| 74LS02 | 14 Pin Socket at A-5. Pin 1 toward gold fingers. | | |
| 74LS10 | 14 Pin Socket aat A-6. Pin 1 toward gold fingers. | | |

### 7. Clean PCB of Rosin and By-products of Soldering

Once soldering is complete, clean the back of PCB of excess flux and rosin. 90% or higher isopropyl alcohol. IPA will dissolve soldering resin. Note that the IPA will also remove the APPLE-AX printing on the PROMs so keep it away from these parts. Spray it on the back of the board and lightly scrub with a very soft brush that will not scratch the surface of the PCB. Soak up the IPA and contaminates with a clean soft cloth before the IPA evaporates in order to remove the by products of soldering. I have also discovered that "Windex" window cleaner can help remove the by-products from the soldering job. Removing contaminates is important as many kinds of rosins are corrosive. Let dry overnight. Position a fan to blow over the board to make sure that all remaining moisture evaporates.

### 8. Check Board for Solder Bridges and Cold Solder Joints

While the board is drying, you should carefully check your work for bad solder joints and solder bridges.

## *Chapter 3 – Installation, Operation and Help*

### 1. Installation and Operation

Completely read and understand the original Apple Cassette Interface Manual reproduced in appendix C for installation and operation instructions.

### *2. Troubleshooting and Help*

A good job of soldering the components into place should eliminate most if not all trouble. First step, in case of trouble, should be to check for bad solder joints or bridges.

Refer to my Apple II repair page at www.willegal.net for some general troubleshooting hints.

Feel free to send email to: mike@willegal.net if you run into difficulties.

•

## *Appendix A - Using an iPod With the ACI*

An iPod may be used in place of a cassette player with the ACI. Almost any iPod can be used for loading programs with the same cable that is used for reading from a cassette player. Programs must be put into AIFF format prior to loading. I have listed several programs already in that format on this web page:
http://www.willegal.net/appleii/apple1-software.htm
This same page has the source code for a UNIX shell program that will convert programs in Apple monitor format into AIFF files, so that you can convert your own programs to be loaded from a iPod.

Writing to the iPod requries an iPod that supports microphone input, a special cable and an iPad application that uses a lossless recording format. A detailed writeup on the process can be found here.
http://www.apple1notes.com/Home/Notes.html

## *Appendix B Replica 1 Notes*

At the time of this printing, the ACI has not been tested with a Briel Computer Systems Replica 1. Watch my bog at www.willegal.net/blog for updates on the results of this planned testing. I do not expect issues with this testing. If you do try it before I get the chance, remember that because the ACI requires -12volts, the Replica 1 must be powered by an ATX power supply.

The Achatz replica does not have a provision for -12 volts, so the ACI will not work with that system.

## *Appendix C - Apple's Original ACI Manual*

Appendix C is a digitized reproduction of the original eight page Apple Cassette Interface manual. Fonts and layout are similar to the original, but not exact reproductions. Disregard waranty, address, phone number information - this data is left in place, so the complete manual is preserved.

The manual contains accurate and useful information for operation of the ACI. It is my opinion is that the reliability of the ACI is not as good as the manual suggests, especially with the stock .01uF capacitor in place.

Unlike what the manual indicates, performance with various cassette recorders can vary from not functional to works pretty well. I use and recommend a Panasonic RQ-2102. The best volume setting for read operations on my recent production Panasonic RQ 2102 is around a 4.

## *Appendix D – ACI Source Code Listing*

Appendix D is the source code listing for the 256 byte PROM bank that exists on the ACI card.


# APPLE-1 CASSETTE INTERFACE

# INTRODUCTION

The Apple Cassette Interface [ACI] is a peripheral device for the Apple Computer which enables the user to store and retrieve information (data and programs) using a standard audio grade cassette recorder. The ACI attaches directly to the Apple Computer and jacks are provided on the ACI board to connect to the cassette recorder. The ACI reads and writes data at the rate of approximately 1500 baud (depending upon the data), All the ACI timing is done in software. resulting in extreme accuracy, no adjustments, and consistency between units.

# TAPE RECORDERS & TAPE

Almost any cassette recorder will work well with the ACI. As a recommendation, we have found the least expensive (under $40) Panasonic to be very reliable and of good quality, although it is not equipped with a tape counter. which is useful (though not essential) for locating files within a tape. (An alternative method of discerning files is to record a voice identification between files.)

Among the 'under $25' cassette recorders there may be variations in head alignment and internal electronics, resulting in the inability to accurately read a tape that has been recorded on a different machine. However, if the same unit is used for both recording and reading, even the cheapest of cassette recorders will work reliably.

Most tapes available in the $2-$4 category work well for data storage. You may experience an occasional tape which 'loses bits' which is caused by severe oxide thickness variations on the tape and cannot be corrected. Special leaderless tapes need not be used as the ACI automatically transmits a ten second 'header' of all ones before transmitting the data, which insures that the leader will have passed.

# JUMPERS

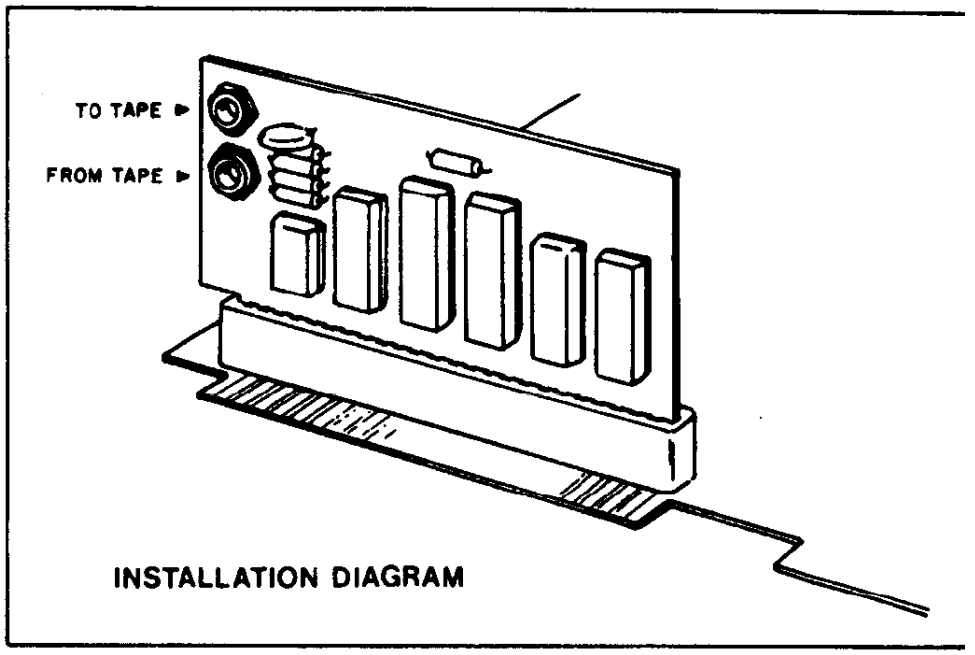
For operation of the ACI, a permanent jumpcr must be placed between 'R' and 'C' in the block select area of the main board (B9,10). This jumper from 'R' which is connectcd to enable the ACI, to 'C' se1ects the ACI when the 12th 4K block ('C') is addressed.

Also, for running Apple Basic in the 2nd 4K band of on-board memory, it is necessary to break the solder jumper between 'W' and '1' and then jumper 'W' to 'E.' This moves the 2nd 4K bank from the '1' block to the 'E' block, which is where Apple Basic resides.

# INSTALLATION

Install the ACI board into the connector on the main board **with the components on the ACI board facing away from the main board (the jacks to the edge of the main board).** SEE FIGURE 1. The system power should be OFF whenever installing or removing the ACI board.

Install cables from the ACI jack marked "TO TAPE" to the recorder microphone input and from the ACI jack marked "FROM TAPE" to the recorder earphone output. One cable can be appropriately switched between the two paths if necessary.




INSTALLATION DIAGRAM

# USING THE ACI

The Cassette program is contained in two PROMS on the ACI board and runs at C100. When entered, the program should echo an "*." The format for specifying the memory address ranges to be either stored (write) or deposited into (read) is identical to the standard monitor format: **Beginning . End ,** suffixed with either a 'W' (write) or an 'R' (read). Execution will start following a carriage return (CR). The cassette program will return control to the system monitor upon completion of a read or write. Illegal characters (or the absence of characters) in the address line will return control to the system monitor without execution, following a carriage return.

# MULTIPLE RANGES

The ACI is capable of reading and writing multiple address ranges. The format is: **A.BW C.DW** (R for read). Again, spaces are ignored. The ACI will write a ten second header, the first range, another header and the second range. **100.200W 300.500W** will write a header, 100 in 200, a second header, and 300 to 500. When reading a multiple range tape. YOU MUST USE THE SAME ADDRESS INCREMENTS AS WERE USED IN WRITING THE TAPE. This does not mean the same absolute addresses, but rather the same increments.

The procedure for reading from a tape into memory is:

**C100R (RET)** This enters the cassette program and should echo an '*'

**E000 . EFFFR** This will load the tape data into memory locations E000.EFFF . 'R' denotes a read, and spaces are ignored. [Don't hit 'return' yet.)

**Start the Tape**

**Hit 'RETURN'** A carriage return will start execution of a read. The 'return' can be hit immediately, however, it must be hit within 5 seconds after the start of tape motion.

When the last location (EFFF in this example) has been loaded, the program will print a '/' and return control to the monitor.

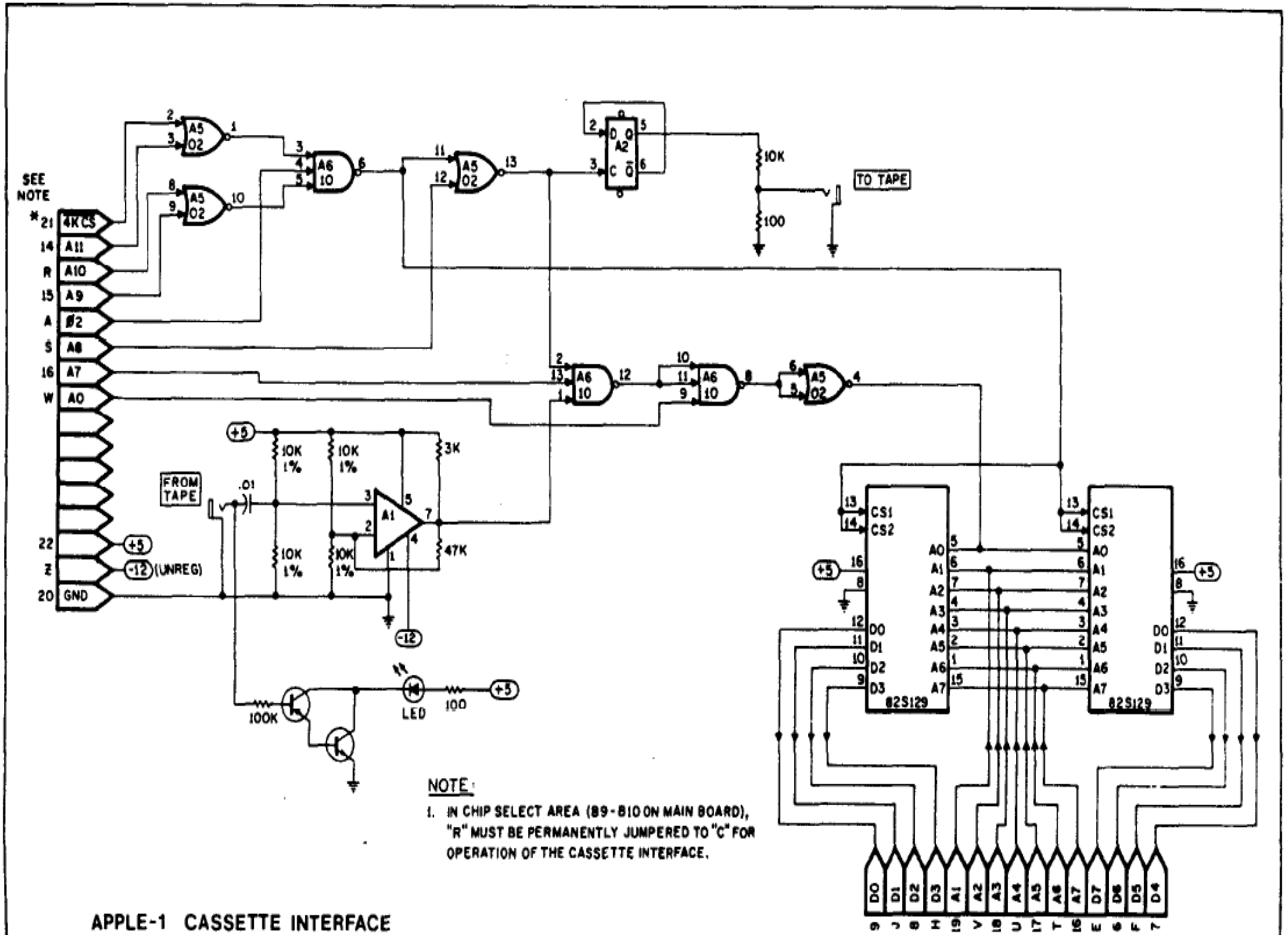
The procedure for a write is identical except the suffix 'W' is substituted for 'R' in the address line. For both read and write, the tape should always be moving before hitting the RETURN.

# LEVEL

The Cassette recorder output level should be set to where the LED on the ACI is just fully lit. Increase the level from zero until the LED glows fully. If you experience a bad read, try it a LITTLE higher. The LED indicator is operational even when the cassette program is not executing. and the level should be set prior to reading a tape, NOT during the reading of a tape.

# SPEED

The ACI uses the technique of recording a whole cycle of either a 1kHz cycle (representing a 'one' data bit) or a 2 kHz cycle (representing a 'zero' data bit). Therefore, with an average data mix of one's and zero`s, data will be recorded at 1500 baud. A ten second header of all ones will automatically be recorded on the tape prior to memory data. This is to insure that the clear leader portion of the tape will have passed. See schematic for further details.




NOTE:

1. IN CHIP SELECT AREA (B9-B10 ON MAIN BOARD), "R" MUST BE PERMANENTLY JUMPERED TO "C" FOR OPERATION OF THE CASSETTE INTERFACE.

APPLE-1 CASSETTE INTERFACE

# WARRANTY

Apple Computer Company hereby warrants each of its products, and all components therein contained, to be free from defects in materials and/or workmanship for a period of thirty (30) days from date of purchase. In the event of the occurrence of malfunction, or other indication of failure attributable directly to faulty workmanship and/or material, then, upon return of the product to the Apple Computer Company at 770 Welch Road, Palo Alto, California 94304 (postage prepaid), the Apple Computer Company will, at its option, repair or replace said products or components thereof, to whatever extent Apple Computer Company shall deem necessary, to restore said product to proper operating condition. All such repairs or replacements shall he rendered by Apple Computer Company without charge to the customer.

The responsibilty for the failure of any Apple Computer product, or component thereof, which, at the discretion of the Apple Computer Company , shall have resulted either directly or indirectly from accident, abuse, or misapplication of the product, shall be assumed by the cutomer and the Apple Computer Company shall assume no liability as a consequence of such events under the terms of this warranty.

While every effort, on the part of Apple Computer Company, is made to provide clear and accurate technical instruction on the use, implementation, and application of its products, the Apple Computer Company shall assume no liability in events which arise from the application of such technical instruction, nor shall the Apple Computer Company be held liable for the quality, interconnection, or application of periferal products, which may have been recommended by Apple Computer Company, but which have not been supplied as part of the product.

This warranty contains and embodies the limits of responsibility of the Apple Computer Company with regard to its products, and no other liability is expressed, implied, or should be assumed by the purchaser, and in no event shall the Apple Computer Company be held liable for the loss of time, effort , or transportation costs, nor for loss of potential profits or other consequential losses which might arise from the purchase, assembly, use, application, or subsequent sale of the products of Apple Computer Company, nor from any instructions and/or mechanical information thereto related.

**APPLE COMPUTER COMPANY**

**770 Welch Road, Suite 154**

**Palo Alto, California  94304**

**Phone:(415) 326-4248**

```
------- FILE wozaci.asm LEVEL 1 PASS 2
   1 c200                                       PROCESSOR 6502
   2 c200 ????                                  LIST      ON
   3 c200 ????                  ;-------------------------------------------------------------------
   4 c200 ????                  ;
   5 c200 ????                  ; The WOZ Apple Cassette Interface for the Apple 1
   6 c200 ????                  ; Written by Steve Wozniak somewhere around 1976
   7 c200 ????                  ;
   8 c200 ????                  ;-------------------------------------------------------------------
   9 c200 ????
  10 c100                                       ORG       $C100
  11 c100
  12 c100                       ;-------------------------------------------------------------------
  13 c100                       ; Memory declaration
  14 c100                       ;-------------------------------------------------------------------
  15 c100
  16 c100               00 24   HEX1L           EQU       $24         ;End address of dump block
  17 c100               00 25   HEX1H           EQU       $25
  18 c100               00 26   HEX2L           EQU       $26         ;Begin address of dump block
  19 c100               00 27   HEX2H           EQU       $27
  20 c100               00 28   SAVEINDEX       EQU       $28         ;Save index in input buffer
  21 c100               00 29   LASTSTATE       EQU       $29         ;Last input state
  22 c100
  23 c100               02 00   IN              EQU       $0200       ;Input buffer
  24 c100               c0 00   FLIP            EQU       $C000       ;Output flip-flop
  25 c100               c0 81   TAPEIN          EQU       $C081       ;Tape input
  26 c100               d0 10   KBD             EQU       $D010       ;PIA.A keyboard input
  27 c100               d0 11   KBDCR           EQU       $D011       ;PIA.A keyboard control register
  28 c100               ff 1a   ESCAPE          EQU       $FF1A       ;Escape back to monitor
  29 c100               ff ef   ECHO            EQU       $FFEF       ;Echo character to terminal
  30 c100
  31 c100                       ;-------------------------------------------------------------------
  32 c100                       ; Constants
  33 c100                       ;-------------------------------------------------------------------
  34 c100
  35 c100               00 8d   CR              EQU       $8D         ;Carriage Return
  36 c100               00 9b   ESC             EQU       $9B         ;ASCII ESC
  37 c100
  38 c100                       ;-------------------------------------------------------------------
  39 c100                       ; Let's get started
  40 c100                       ;-------------------------------------------------------------------
  41 c100
  42 c100                a9 aa  WOZACI          LDA       #$AA        ;Print the Tape prompt "*"
  43 c102               20 ef ff                JSR       ECHO
  44 c105               a9 8d                   LDA       #CR         ;And drop the cursor one line
  45 c107               20 ef ff                JSR       ECHO
  46 c10a
  47 c10a                a0 ff                  LDY       #-1         ;Reset the input buffer index
  48 c10c               c8      NEXTCHAR        INY
  49 c10d               ad 11 d0 KBDWAIT        LDA       KBDCR       ;Wait for a key
  50 c110               10 fb                   BPL       KBDWAIT     ;Still no key!
  51 c112
  52 c112               ad 10 d0                LDA       KBD         ;Read key from keyboard
  53 c115               99 00 02                STA       IN,Y        ;Save it into buffer
  54 c118               20 ef ff                JSR       ECHO        ;And type it on the screen
  55 c11b               c9 9b                   CMP       #ESC
  56 c11d               f0 e1                   BEQ       WOZACI      ;Start from scratch if ESC!
  57 c11f               c9 8d                   CMP       #CR
  58 c121               d0 e9                   BNE       NEXTCHAR    ;Read keys until CR
  59 c123
  60 c123                a2 ff                  LDX       #-1         ;Initialize parse buffer pointer
  61 c125
```

```
 62 c125                    ;--------------------------------------------------------------------
 63 c125                    ; Start parsing first or a new tape command
 64 c125                    ;--------------------------------------------------------------------
 65 c125
 66 c125  a9 00     NEXTCMD LDA  #0          ;Clear begin and end values
 67 c127  85 24             STA  HEX1L
 68 c129  85 25             STA  HEX1H
 69 c12b  85 26             STA  HEX2L
 70 c12d  85 27             STA  HEX2H
 71 c12f
 72 c12f  e8        NEXTCHR INX              ;Increment input pointer
 73 c130  bd 00 02          LDA  IN,X        ;Get next char from input line
 74 c133  c9 d2             CMP  #$D2        ;Read command? "R"
 75 c135  f0 56             BEQ  READ        ;Yes!
 76 c137  c9 d7             CMP  #$D7        ;Write command? "W"
 77 c139  f0 35             BEQ  WRITE       ;Yes! (note: CY=1)
 78 c13b  c9 ae             CMP  #$AE        ;Separator?"."
 79 c13d  f0 27             BEQ  SEP         ;Yes!
 80 c13f  c9 8d             CMP  #CR         ;End of line?
 81 c141  f0 20             BEQ  GOESC       ;Escape to monitor! We're done
 82 c143  c9 a0             CMP  #$A0        ;Ignore spaces: " "
 83 c145  f0 e8             BEQ  NEXTCHR
 84 c147  49 b0             EOR  #$B0        ;Map digits to 0-9 "0"
 85 c149  c9 0a             CMP  #9+1        ;Is it a decimal digit?
 86 c14b  90 06             BCC  DIG         ;Yes!
 87 c14d  69 88             ADC  #$88        ;Map letter "A"-"F" to $FA-$FF
 88 c14f  c9 fa             CMP  #$FA        ;Hex letter?
 89 c151  90 ad             BCC  WOZACI      ;No! Character not hex!
 90 c153
 91 c153  0a        DIG     ASL              ;Hex digit to MSD of A
 92 c154  0a                ASL
 93 c155  0a                ASL
 94 c156  0a                ASL
 95 c157
 96 c157  a0 04             LDY  #4          ;Shift count
 97 c159  0a        HEXSHIFT ASL             ;Hex digit left, MSB to carry
 98 c15a  26 24             ROL  HEX1L       ;Rotate into LSD
 99 c15c  26 25             ROL  HEX1H       ;Rotate into MSD
100 c15e  88                DEY              ;Done 4 shifts?
101 c15f  d0 f8             BNE  HEXSHIFT    ;No! Loop
102 c161  f0 cc             BEQ  NEXTCHR     ;Handle next character
103 c163
104 c163                    ;--------------------------------------------------------------------
105 c163                    ; Return to monitor, prints \ first
106 c163                    ;--------------------------------------------------------------------
107 c163
108 c163  4c 1a ff GOESC    JMP  ESCAPE      ;Escape back to monitor
109 c166
110 c166                    ;--------------------------------------------------------------------
111 c166                    ; Separating . found. Copy HEX1 to Hex2. Doesn't clear HEX1!!!
112 c166                    ;--------------------------------------------------------------------
113 c166
114 c166  a5 24    SEP     LDA  HEX1L       ;Copy hex value 1 to hex value 2
115 c168  85 26             STA  HEX2L
116 c16a  a5 25             LDA  HEX1H
117 c16c  85 27             STA  HEX2H
118 c16e  b0 bf             BCS  NEXTCHR     ;Always taken!
119 c170
```

```
120 c170                ;-------------------------------------------------------------------------
121 c170                ; Write a block of memory to tape
122 c170                ;-------------------------------------------------------------------------
123 c170
124 c170  a9 40   WRITE       LDA   #64          ;Write 10 second header
125 c172  20 cc c1            JSR   WHEADER
126 c175
127 c175  88      WRNEXT      DEY                ;Compensate timing for extra work
128 c176  a2 00               LDX   #0           ;Get next byte to write
129 c178  a1 26               LDA   (HEX2L,X)
130 c17a
131 c17a  a2 10               LDX   #8*2         ;Shift 8 bits (decremented twice)
132 c17c  0a      WBITLOOP    ASL                ;Shift MSB to carry
133 c17d  20 db c1            JSR   WRITEBIT     ;Write this bit
134 c180  d0 fa               BNE   WBITLOOP     ;Do all 8 bits!
135 c182
136 c182  20 f1 c1            JSR   INCADDR      ;Increment address
137 c185  a0 1e               LDY   #30          ;Compensate timer for extra work
138 c187  90 ec               BCC   WRNEXT       ;Not done yet! Write next byte
139 c189
140 c189  a6 28     RESTIDX   LDX   SAVEINDEX    ;Restore index in input line
141 c18b  b0 98               BCS   NEXTCMD      ;Always taken!
142 c18d
143 c18d                ;-------------------------------------------------------------------------
144 c18d                ; Read from tape
145 c18d                ;-------------------------------------------------------------------------
146 c18d
147 c18d  20 bc c1 READ       JSR   FULLCYCLE    ;Wait until full cycle is detected
148 c190  a9 16               LDA   #22          ;Introduce some delay to allow
149 c192  20 cc c1            JSR   WHEADER      ; the tape speed to stabilize
150 c195  20 bc c1            JSR   FULLCYCLE    ;Synchronize with full cycle
151 c198
152 c198  a0 1f     NOTSTART  LDY   #31          ;Try to detect the much shorter
153 c19a  20 bf c1            JSR   CMPLEVEL     ;  start bit
154 c19d  b0 f9               BCS   NOTSTART     ;Start bit not detected yet!
155 c19f
156 c19f  20 bf c1            JSR   CMPLEVEL     ;Wait for 2nd phase of start bit
157 c1a2
158 c1a2  a0 3a               LDY   #58          ;Set threshold value in middle
159 c1a4  a2 08     RDBYTE    LDX   #8           ;Receiver 8 bits
160 c1a6  48        RDBIT     PHA
161 c1a7  20 bc c1            JSR   FULLCYCLE    ;Detect a full cycle
162 c1aa  68                  PLA
163 c1ab  2a                  ROL                ;Roll new bit into result
164 c1ac  a0 39               LDY   #57          ;Set threshold value in middle
165 c1ae  ca                  DEX                ;Decrement bit counter
166 c1af  d0 f5               BNE   RDBIT        ;Read next bit!
167 c1b1  81 26               STA   (HEX2L,X)    ;Save new byte
168 c1b3
169 c1b3  20 f1 c1            JSR   INCADDR      ;Increment address
170 c1b6  a0 35               LDY   #53          ;Compensate threshold with workload
171 c1b8  90 ea               BCC   RDBYTE       ;Do next byte if not done yet!
172 c1ba  b0 cd               BCS   RESTIDX      ;Always taken! Restore parse index
173 c1bc
174 c1bc  20 bf c1  FULLCYCLE JSR   CMPLEVEL     ;Wait for two level changes
175 c1bf  88        CMPLEVEL  DEY                ;Decrement time counter
176 c1c0  ad 81 c0            LDA   TAPEIN       ;Get Tape In data
177 c1c3  c5 29               CMP   LASTSTATE    ;Same as before?
178 c1c5  f0 f8               BEQ   CMPLEVEL     ;Yes!
179 c1c7  85 29               STA   LASTSTATE    ;Save new data
180 c1c9
181 c1c9  c0 80               CPY   #128         ;Compare threshold
182 c1cb  60                  RTS
183 c1cc
```

```
184 c1cc                    ;-------------------------------------------------------------------------
185 c1cc                    ; Write header to tape
186 c1cc                    ;
187 c1cc                    ; The header consists of an asymmetric cycle, starting with one phase of
188 c1cc                    ; approximately (66+47)x5=565us, followed by a second phase of
189 c1cc                    ; approximately (44+47)x5=455us.
190 c1cc                    ; Total cycle duration is approximately 1020us ~ 1kHz. The actual
191 c1cc                    ; frequencywill be a bit lower because of the additional workload between
192 c1cc                    ; the twoloops.
193 c1cc                    ; The header ends with a short phase of (30+47)x5=385us and a normal
194 c1cc                    ; phase of (44+47)x5=455us. This start bit must be detected by the read
195 c1cc                    ; routine to trigger the reading of the actual data.
196 c1cc                    ;-------------------------------------------------------------------------
197 c1cc
198 c1cc  86 28     WHEADER     STX   SAVEINDEX   ;Save index in input line
199 c1ce  a0 42     HCOUNT      LDY   #66         ;Extra long delay
200 c1d0  20 e0 c1              JSR   WDELAY      ;CY is constantly 1, writing a 1
201 c1d3  d0 f9                 BNE   HCOUNT      ;Do this 64 * 256 time!
202 c1d5  69 fe                 ADC   #-2         ;Decrement A (CY=1 all the time)
203 c1d7  b0 f5                 BCS   HCOUNT      ;Not all done!
204 c1d9  a0 1e                 LDY   #30         ;Write a final short bit (start)
205 c1db                    ;
206 c1db                    ;-------------------------------------------------------------------------
207 c1db                    ; Write a full bit cycle
208 c1db                    ;
209 c1db                    ; Upon entry Y contains a compensated value for the first phase of 0
210 c1db                    ; bit length. All subsequent loops don't have to be time compensated.
211 c1db                    ;-------------------------------------------------------------------------
212 c1db
213 c1db  20 e0 c1  WRITEBIT    JSR   WDELAY      ;Do two equal phases
214 c1de  a0 2c                 LDY   #44         ;Load 250us counter - compensation
215 c1e0
216 c1e0  88        WDELAY      DEY               ;Delay 250us (one phase of 2kHz)
217 c1e1  d0 fd                 BNE   WDELAY
218 c1e3  90 05                 BCC   WRITE1      ;Write a '1' (2kHz)
219 c1e5
220 c1e5  a0 2f                 LDY   #47         ;Additional delay for '0' (1kHz)
221 c1e7  88        WDELAY0     DEY               ; (delay 250us)
222 c1e8  d0 fd                 BNE   WDELAY0
223 c1ea
224 c1ea  bc 00 c0  WRITE1      LDY   FLIP,X      ;Flip the output bit
225 c1ed  a0 29                 LDY   #41         ;Reload 250us cntr (compensation)
226 c1ef  ca                    DEX               ;Decrement bit counter
227 c1f0  60                    RTS
228 c1f1
229 c1f1                    ;-------------------------------------------------------------------------
230 c1f1                    ; Increment current address and compare with last address
231 c1f1                    ;-------------------------------------------------------------------------
232 c1f1
233 c1f1  a5 26     INCADDR     LDA   HEX2L       ;Compare current address with
234 c1f3  c5 24                 CMP   HEX1L       ; end address
235 c1f5  a5 27                 LDA   HEX2H
236 c1f7  e5 25                 SBC   HEX1H
237 c1f9  e6 26                 INC   HEX2L       ;And increment current address
238 c1fb  d0 02                 BNE   NOCARRY     ;No carry to MSB!
239 c1fd  e6 27                 INC   HEX2H
240 c1ff  60        NOCARRY     RTS
241 c200
242 c200                    ;-------------------------------------------------------------------------
243 c200
```