NETRONICS

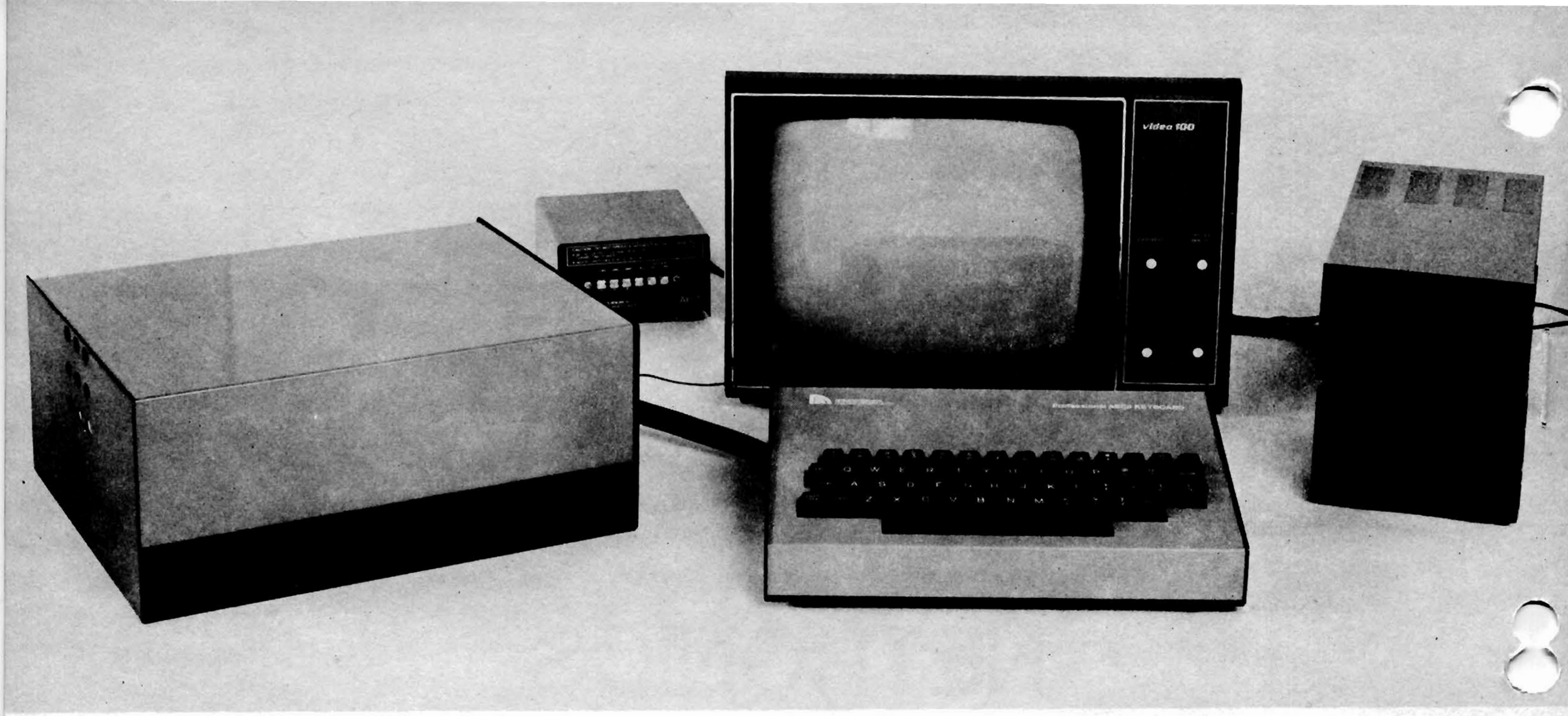
ASSEMBLY & INSTRUCTION MANUAL

EXPLORER 85

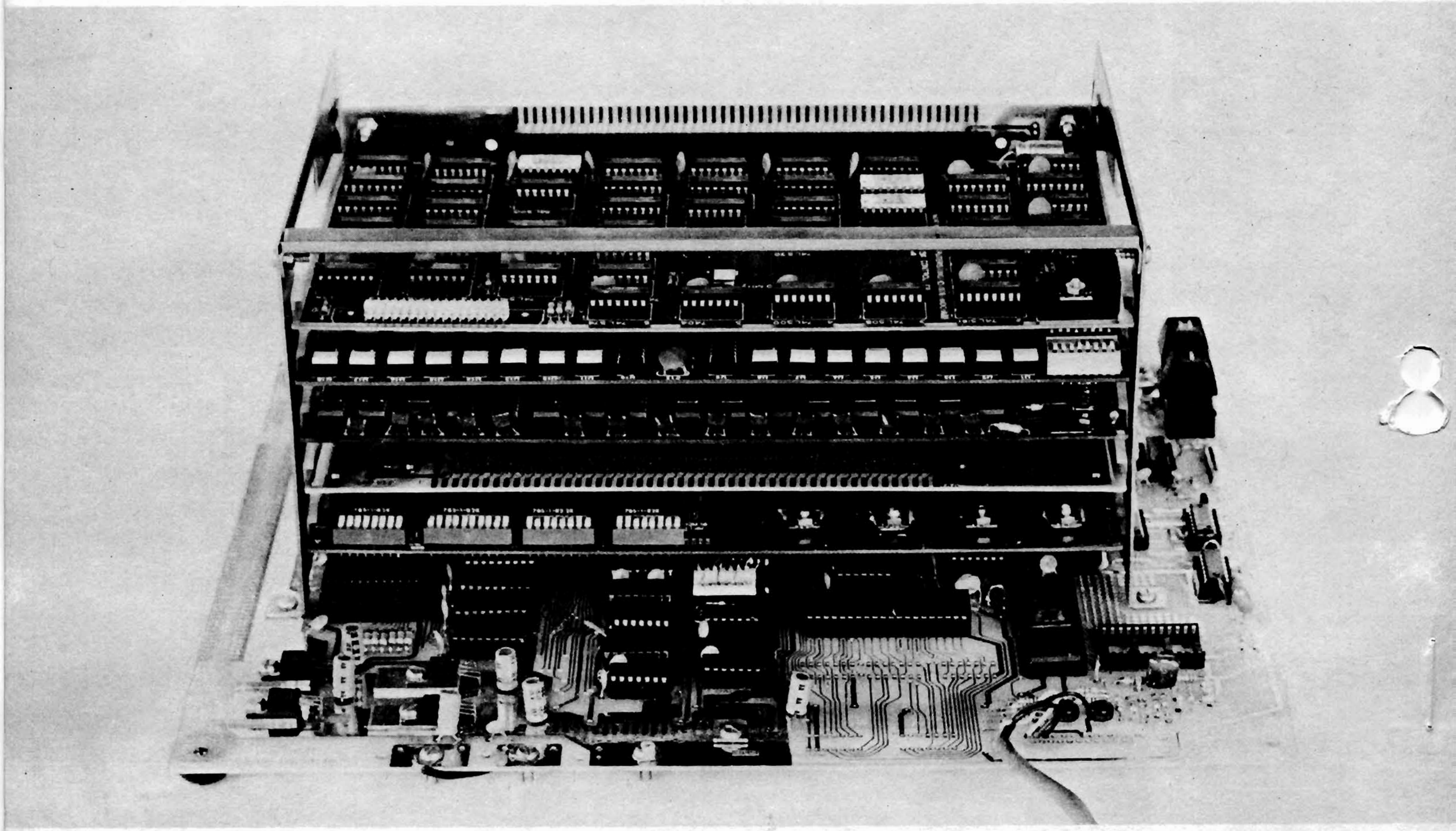
LEVEL A

RT.202 NEW MILFORD, CT 06776

REVISION C-1



EXPLORER SYSTEM SHOWN IN ENCLOSURES AVAILABLE WITH FLOPPY DISC DRIVES AND MONITOR



FULLY EXPANDED EXPLORER

INTRODUCTION

The Explorer 85 microcomputer system is designed to satisfy the needs of all levels of microcomputer system design,as well as an extremely cost effective expandable learning tool. The Explorer can be expanded to a versitile single board computer for industrial controllers or to a full blown data processing system utilizing floppy disc drives, printers, terminals, etc. The S100 bus addition (Level B) allows unlimited expansion using the hundreds of boards available ranging from custom controllers and memory to exotic voice and music generators. The mother board has provisions for all levels of expansion. Level A (minimum system), level B (S100 expansion), level D (4K on board RAM), level E (8K EPROM expansion) all on the same mother board. Level C utilizes a main frame concept which mounts right on the mother board expanding the number of S100 sockets to six. The mother board also has a generous amount of prototyping area for custom designs.

The minimum system can be supplied to communicate with either a standard terminal or teletype with a EIA (RS/232-C) or 20MA current loop as well as a version to communicate with a hex keypad (with an output display). The EIA, and hex keypad systems are identical except for the monitor ROM which can be replaced any time you wish to change inputs.

Each level is separately regulated, there is even a regulator for the hex keypad for the ultimate in interference and noise free operation. The cabinet is designed to house the minimum as well as the fully expanded system in a neat and professional manner. Adequate ventilation ensure low operating temperatures, long and reliable life. The level A system requires +8V DC at 650MA. (-8V also required if using RS232 terminal). Adding level B, D and E requires approximately 2.5 amps total. The Netronics ±8V,5 amp supply should be adequate for most systems. If additional power is required, the board is designed to accomodate more than one supply. The S100 boards can be powered separately for an extremely low noise S100 backplane.

The Explorer 85 utilizing the 8085 microprocessor is 100% software compatible with 8080 programs. This includes the 8 RST vector interrupts which are designed to run or to be executed in the users program. The four additional hardware interrupts are automatically channeled to the monitor with a register save routine and RAM area addresses which redirect the processor to the desired interrupt program.

SPECIFICATIONS
LEVEL A

P.C. Board 13¼" x 10 3/4" glass epoxy, plated through holes with solder mask.

I/O .Provisions for 25 pin (DB25) connector for terminal serial I/O, which can also support a paper tape reader.
- .Provision for 24 pin DIP socket for hex key board I/O.
- .Cassette tape recorder input.
- .Cassette tape recorder output.
- .Cassette tape control output.
- .Speaker output.
- .LED output indicator on SOD line.
- .Printer interface (less drivers).
- .Total of four 8 bit plus one 6 bit I/O ports.

Crystal Frequency 6.144 MHz

Control Switches Reset and user (RST 7.5) interrupt. Additional provisions for RST 5.5, 6.5 and TRAP interrupts on board.

Programmable 14 Bit Binary Counter/Timer

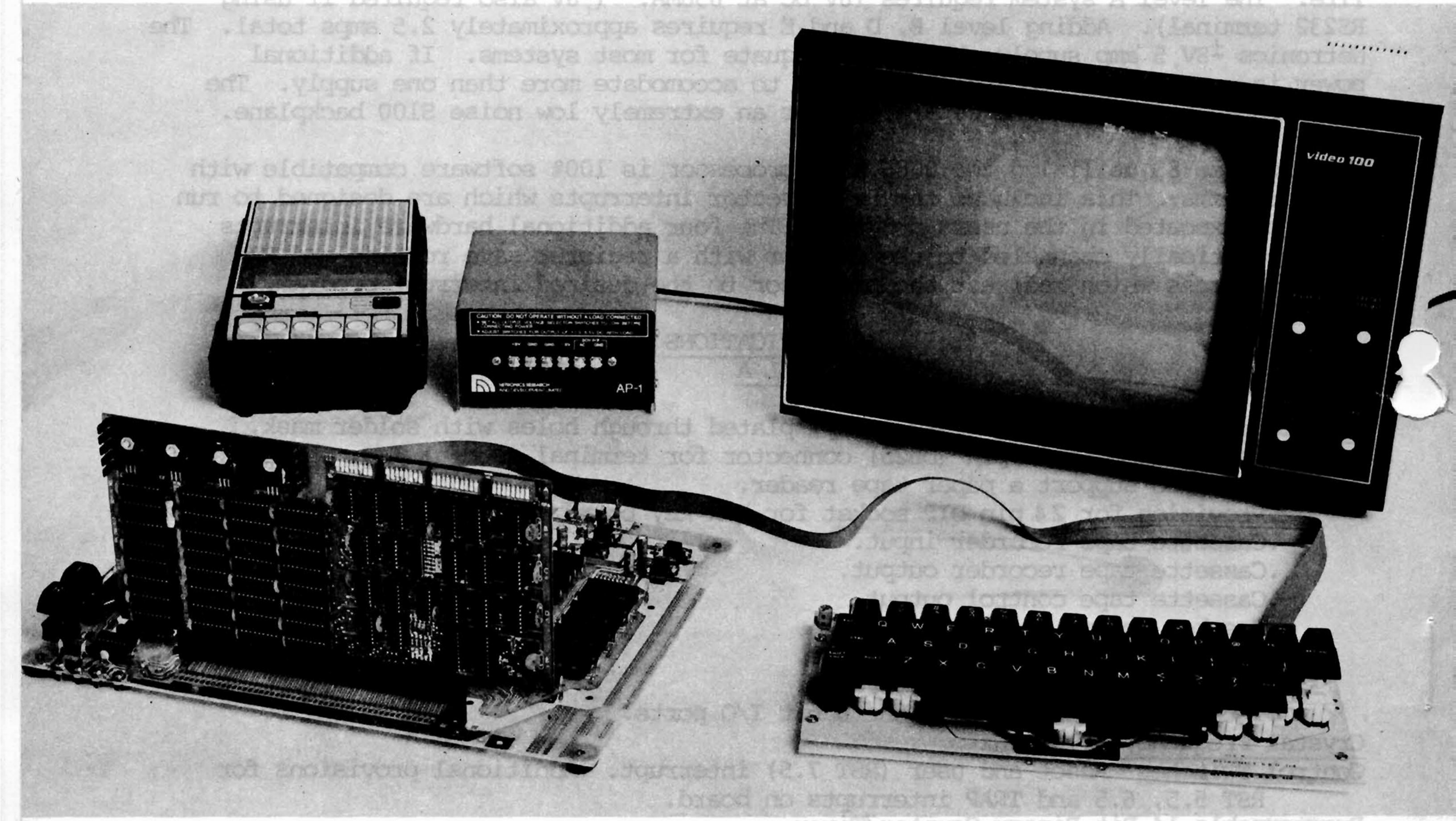
SPECIFICATIONS
LEVEL A

System RAM 256 bytes located at F8ØØ. Ideal for smaller systems and for use as isolated stack area in expanded systems. RAM expandable to 60K via S100 bus or 4K on mother board.

Monitor ROM 2K deluxe system monitor located at FØØØ leaving ØØØØ free for user RAM/ROM. Includes the following features:

- .Tape load with label
- .Tape dump with label
- .Examine/change contents of memory
- .Insert data
- .Warm start (register save input)(useful for break point debugging)
- *Move blocks of memory from one location to another
- *Fill blocks of memory with a constant
- .Examine and change all registers
- .Single step with register display at each break point
- *Display blocks of memory
- .Go to execution address
- *Automatic baud rate selection
- *Variable line length control (1-255 characters/line)
- *Channelized I/O monitor routine with 8 bit parallel output for high speed printer
- *Serial console in and console out channel so that monitor can communicate with serial I/O ports

*Not available in Hex key board monitor version.



SHOWN WITH LEVELS B & D, ASCII KEYBOARD/TERMINAL AND VIDEO MONITOR

Check all components received against the parts list below.

PARTS LIST

DESCRIPTION	QUANTITY
Integrated Circuits	
8085	1
74LS244	1
8755 (8355) (Hex or RS232) (Labeled 1KBD or EX 85)	1
8155	1
74LS00	1
74LS10	2
74LS04	2
74LS20	1
LM3900	1
74LS03	1
74LS21	1
LED (Red)	1
Diode	
IN4001	1
Regulator	
LM340-5 or 7805	1
Transistors	
4355	2
4384	2
Resistors 47 ohm (yellow, violet, black)	1
4.7K (yellow, violet, red)	16
3.9K (orange, white, red)	4
100 ohm (brown, black, brown)	4
390 ohm (orange, white, brown)	1
1000 ohm (brown, black, red)	8
22K (red, red, orange)	3
1 Meg (brown, black, green)	2
10K (brown, black, orange)	1
100K (brown, black, yellow)	1
470 ohm (yellow, violet, brown)	3
200 ohm (red, black, brown)	1
47K (yellow, violet, orange)	1
Capacitors	
.15 Mylar	1
.0068 Mylar	1
10uf 16V PC Electrolytic	2
.1 Mylar	1
1.0 Tantalum	1
.0022 Mylar	1
.01 Disc	13
Relay	
1A12AHH or 1A5AH	1
Switch Momentary	2
Cap	
"R"	1
"I"	1
Sockets	
40 Pin	3
14 Pin	9
Rubber Feet	8
Programming Card (Intel)	1
Screw 6/32 x 3/8	4

F000-F800 ($2K_{10}$)
MONITOR ROM
F800-F900 (256_{10})
RAM

C000-E000 ($8K_{10}$)
BASIC 80 ROM

Continued...

PARTS LIST (Con't.)

DESCRIPTION	QUANTITY
Nut #6/32	4
Crystal 6.144 MHz	1
Printed Circuit Board	1
Instruction Book	1

Check all components received against the Parts List.

NOTE: No sockets are included for the 74LS244 IC (U101). U202,3,4 are not supplied. These buffers may be either inverting or non inverting as required by your printer or other I/0.

ASSEMBLY INSTRUCTIONS

( ) 1. Install 40 pin sockets in locations U100,105, and 106. Note that pin #1 is indicated either by a (#1) or a dot (.) on the foil pattern connected to pin #1. Mark your sockets with a white dot, tape, or other marking to insure proper installation of IC's. A reversed IC will be destroyed and not covered by our warranty. NOTE: The most frequent problem encountered when installing sockets is excess solder which flows to the top side of the printed circuit board causing shorts. Carefully inspect the underside of the socket after soldering all sockets. Solder. DO NOT INSTALL IC'S AT THIS TIME.

( ) 2. Install the 14 pin sockets in locations U107,108,109,110,111,112,113,114,115. Observe all precautions as per Step 1. DO NOT INSTALL IC'S AT THIS TIME. Solder.

( ) 3. Install the two momentary push button switches and buttons in locations I and R as per component layout drawing. Solder.

( ) 4. Install tape control relay as shown. This part is marked A12AHH or 1A5AH. Solder.

( ) 5. Install LM340T regulator Q105 as shown. No heat sink is required. Use 6/32 x 3/8 screw and nut. Part may be marked 7805. Solder.

( ) 6. Install diode IN4001 in location CR101. The band must face direction shown. Solder.

( ) 7. Install 4.7K resistor (yellow, violet, red) in locations R100-112,120,135, 142. Solder.

( ) 8. Install 3.9K resistor (orange, white,red) in locations R113,119,122,125. Solder.

( ) 9. Install 100 ohm resistor (brown,black,brown) in locations R114,116,133,140. Solder.

( ) 10. Install 390 ohm resistor (orange,white,brown) in location R115. Solder.

( ) 11. Install 1000 ohm resistor (brown,black,red) in locations R117,118,121,123, 124,126,127,143. Solder. Omit R117 if using a 20MA current loop input.

( ) 12. Install 22K resistor (red,red,orange) in locations R128,129,130. Solder.

NOTE: Save excess leads from resistors which will be used later for jumpers.

(✓) 13. Install 1 Meg resistor (brown,black,green) in locations R131,132. Solder.

(✓) 14. Install 10K resistor (brown,black,orange) in location R134. Solder.

(✓) 15. Install 100K resistor (brown,black,yellow) in location R136. Solder.

(✓) 16. Install 470 ohm resistor (yellow,violet,brown) in locations R137,139, 144
Solder.

(✓) 17. Install 200 ohm resistor (red,black,brown) in location R138. Solder.

(✓) 18. Install 47K resistor (yellow,violet,orange) in location R141. Solder.

(✓) 19. Install .15uf Mylar capacitor in locations C102. Solder.

(✓) 20. Install .0068uf Mylar capacitor in location C103. Solder.

(✓) 21. Install .1uf Mylar capacitor in location C106. Solder.

(✓) 22. Install 1.0uf tantalum capacitor in location C109. Note polarity. Solder.

(✓) 23. Install .0022 uf capacitor in location C115. Solder.

(✓) 24. Install .01 Disc capacitor in location C107,108,110-114,116-119, 121. Solder.

(✓) 25. Install crystal 6.144MHz as shown. If you wish to bend the crystal, apply tape to the board to prevent electrical contact of the crystal case with the printed circuit board. Solder.

(✓) 26. Install red LED as shown. Note: The longer lead is the + side of the diode. Observe polarity shown.

(✓) 27. Install transistor 2N4355 in location Q100,101. Observe the direction of the flat side of the transistor.

(✓) 28. Install transistor 2N4384 in location Q103,104.
(✓) 29. Install 10uf capacitor in location C105,120. Note polarity. Solder.
NO IC'S SHOULD BE INSTALLED AT THIS TIME.

( ) 30. Connect an 8V DC supply to the 8V input located on the rear of the printed circuit board. Note: The 8V input on the front of the board next to the + and - 16V inputs is used for the S100 bus. Do not connect to these inputs at this time. Measure the voltage at the **output** of the regulator. Across C105 this voltage must be 5.0V$\pm$ 2%. If the voltage is not correct do not proceed until the problem is rectified. Failure to observe this precaution will result in destroying expensive IC's, for which we cannot be responsible.

( ) Check your work again. Check under sockets for excess solder and bridges. Check all components for proper installation.

Install IC's as follows. Note direction in assembly parts layout drawing. Note some IC's have numbers in the corners. Disregard any symbols except the dot next to pin #1 or the slot or indentation as illustrated. Make sure that no pins fold under the socket.

( ) U100, 8085

( ) U101, 74LS244 (Solder) No socket provided or suggested.

( ) U102,3,4 Not supplied or required for Level A.

( ) U105, 8355 or 8755 Note pin #1 may be labeled 20. Follow drawing carefully.

( ) U106, 8155

( ) U107, 74LS00

( ) U108, U115, 74LS10

( ) U109, U112, 74LS04

( ) U110, 74LS20

( ) U111, LM3900

( ) U113, 74LS03

( ) U114, 74LS21

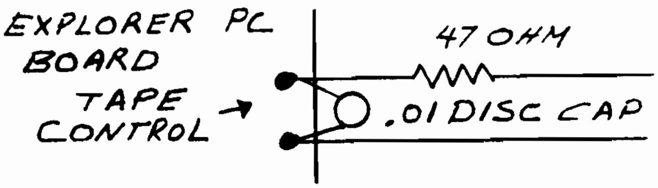
( ) Install jumpers as per jumper installation chart (page 9).

The following wire jumpers are required when installing Level D. We suggest that these additions be made now because no reference is made to them in the Level D instructions.

( ) Connect an insulated jumper between terminal W and X on the component side of the PC board.

( ) Connect an insulated jumper between terminal Z and pin 5 of IC U215 on the bottom of the PC board. Note IC U215 is not to be installed at this time.

( ) If you are using a low-cost cassette tape recorder and are experiencing a sticking relay problem install the following components.



Note the 47 ohm resistor can be installed in either lead going to the tape recorder and is not mounted on the PC board.

( ) If you plan on using Microsoft 8K Basic or our floppy disk drive add a jumper from U105 pin 28 to J-2 pin 22.

JUMPER INSTALLATION CHART

JUMPER	FUNCTION	LEVEL A	COMMENTS
S1	Wait State	Out	Not required.
S2	TRAP Input	A to C	Timer to TRAP for single step
S3	RST 6.5	C to B	Grounded (user can activate)
S4	RST 5.5 { terminal version ~~HEX version~~	C to B ~~A to C~~	Grounded (user can activate) ~~Used for HEX keyboard control~~
S5	Clock to timer input	In	AUX clock may also be used
S6	Input Selector { EIA RS232/C ~~20MA Loop~~	Out ~~In~~	Not required for HEX key pad
S7	Input Selector { EIA RS232/C ~~20MA Loop~~	In ~~Out~~	Not required for HEX key pad
S8	Input Selector { EIA RS232/C ~~20MA Loop~~	Out ~~In~~	Not required for HEX key pad
S9	Input Selector { EIA RS232/C ~~20MA Loop~~	In ~~Out~~	Not required for HEX key pad
S10	Tape input phase	A to C ~~C to B~~	Normal ~~Reverse phase~~
S11		In	For Level A only
S12	NA	Out	
S13	NA	Out	
S14	NA	Out	
S15	NA	Out	
S16	Boot interrupt enable	In	
S17	I/O Port	* ↓	
S18	Input Selector { EIA RS232/C ~~20MA Loop~~	C to A ~~C to B~~	

Omit R117 if you are using a 20MA loop input.

*~~With S17 installed the monitor will automatically set the line limit to 32 characters/line on initialization.~~ USED 1K RESISTOR

With S17 open and pin 25 of U105 connected to +5 (pin 40 of U105) the monitor will automatically set the line limit to 64 characters/line on initialization. The line limit can also be altered by modifying the contents of memory location ~~F8A3~~. The line length constant equals the hex equivalent of the length required.

F8FA 1F=32 3F=64 CHAR

LEVEL A RAM F800 - F900 (F8DE)

EIA TERMINAL CHECK OUT PROCEDURE

The following list of monitor commands and their examples can be used to evaluate the monitor operation on your Explorer 85 and CRT terminal.

POWER UP: When power is supplied to your system the contents of all of the CPU registers and the random access memory (RAM) is random and essentially meaningless. The first thing you should do is press the reset "R" button on the board. This causes the CPU to "boot" or go to the beginning of the monitor. The systems serial output (CR100) LED should turn on. The monitor is now waiting for a signal from your terminal.

The system incorporates automatic baud rate detection over a range of 110 to 9600 baud. To set the baud rate simply press the "space" bar. The monitor will compute the proper baud rate and store the necessary data in the monitor RAM. Then it will transmit the sign-on message to the CRT terminal, followed by a period (.) which is the monitor prompt character. The monitor prompt character is then followed by your terminal's cursor.

In the following pages we will be discussing the use of the monitor commands. In the examples, characters or data output by the Explorer 85 will be shown underlined. The commands themselves are single letters followed by one or more hexadecimal numbers separated by a score, comma, or slash(/). As an example the "Move" command would appear as .MXXXX,YYYY,ZZZZ(CR) where the x's,y's and z's are hexadecimal numbers. The period was supplied by the Explorer 85. Note: No space or other delimiter is allowed between the command and the first number (i.e.M(SB)XXXX is not allowed). The character (SB) indicates pressing the space bar, (CR) indicates the carriage return key and (ESC) indicates the escape key. If you enter a wrong command or wrong number and want to start over, simply press the escape key.

An error in command entry or a non-hexadecimal character in a number field will cause indication (*) and the monitor will disregard the command. When entering hexadecimal numbers, loading zeros are not required. Therefore, entering F alone is equivalent to entering ØØØF.

When you are operating the system, it is important that you DO NOT ALTER THE DATA IN ADDRESSES F8EØ to F8FF. These locations are reserved for the monitor as storage for operating variables. If destroyed the system will not function properly and will have to be reset.

"CHANGE" Command .CXXXX (SB): The change command allows you to change to examine the data in a memory location referenced by XXXX. When the space bar is pressed the computer responds with the contents of the address in hexadecimal format followed by a dash (-). If you wish to make a change, enter the new data followed by a carriage return (CR) which will terminate the command or a space (SB) which will display the contents of the next location. If no number is entered, the memory remains unchanged.

Example #1: Change the data in the memory location referenced by "L LIMIT" to a 1F. This will change the length of the lines sent to the CRT to a maximum of 32 characters. L LIMIT = F8FA

.CF8FA (SB) 3F-(CR)

.

The data at location ~~F8F6~~ F8FA is now A 1F, to check this we use the same command and terminate with (CR).

.CF8FA (SB) 1F-(CR)

.

"FILL" Command .FXXXX,YYYY,ZZ(CR): This command allows the user to fill a number of memory locations with the same value. When executed, the memory addressed by XXXX, to YYYY are filled with the hexadecimal constant ZZ. Note: Do not fill locations F8EØ through F8FF!

Example #2: Fill memory locations F8ØØ through F81F (32 locations) with hexadecimal 77.

```
.FF8ØØ,F81F,77 (CR)
.
```

The 32 locations are now set to Hex 77.

"DISPLAY" Command .DXXXX,YYYY: The display function allows the user to view a block of memory from address XXXX to YYYY on the terminal output device. If your line length is set to less than 64 characters, each line will display eight (8) bytes of data. If it is set for 64 characters or more, sixteen (16) bytes will be displayed on each line (see example #1).

Example #3: Display the contents of memory locations F8ØØ through F81F.

```
.DF8ØØ,F81F (CR)
F8ØØ 77 77 77 77 77 77 77 77
F8Ø8 77 77 77 77 77 77 77 77
F81Ø 77 77 77 77 77 77 77 77
F818 77 77 77 77 77 77 77 77
.
```

Notice that on the left margin, the address of the first charcter in the line is printed. On displays capable of more than 32 characters per line, the line length could be changed to 64 (3F hexadecimal) and the display would appear as shown below.

```
.D F8ØØ,F81F (CR)
F8ØØ  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77
F81Ø  77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77
.
```

"MOVE" Command .MXXXX,YYYY,ZZZZ(CR): The move command copies the data in memory locations XXXX through YYYY to the memory beginning at address ZZZZ. This command is usually used in expanded systems with additional ROM and allows you to store away a program or block of data for later use. Note: If you move a program from one location to another IT WILL NOT RUN IN THE NEW LOCATION. Most 8080 code is memory address dependant.

Example #4: Move the contents of memory locations ØØØØ through ØØ1F to memory locations beginning at F8ØØ.

The command .MØØØØ,ØØ1F,F8ØØ (CR) will accomplish this as will .MØ,1F,F8ØØ (CR).

If you have a minimum system there is no memory in locations Ø through 1F. When the Explorer 85 attempts to read a memory location where there is nothing it will see FF as the data. Therefore, if we examine the results of example #4 using the display command we get the following results:

```
.DF8ØØ,F81F (CR)
F8ØØ FF FF FF FF FF FF FF FF
F8Ø8 FF FF FF FF FF FF FF FF
F81Ø FF FF FF FF FF FF FF FF
F818 FF FF FF FF FF FF FF FF
```

"INSERT Command .IXXXX(CR): The insert command allows the user to insert a string of hex data in the memory beginning at address XXXX. The data must consist of hex characters separated by a space, comma, slash, or carriage return/line feed. For those users with an ASR 33 teletype, this will be an easy way of loading programs via the paper tape reader. To exit the insert command enter the (ESC) key or any non-hexadecimal character.

Example #5: Enter a string of data into memory beginning at address F8ØØ. The data is a program developed at the end of this section.

```
.IF8ØØ (CR)
21,8F,F8,F9,11,89,14,DB (CR)
FØ,F6,8Ø,D3,FØ,Ø6,6B,Ø5 (CR)
C2,ØF,F8,DB,FØ,E6,7F,D3 (CR)
FØ,Ø6,6B,Ø5,C2,1B,F8,1D (CR)
C2,Ø7,F8,15,C2,Ø7,F8,3E (CR)
ØB,3Ø,FB,C3,2B,F8       (CR)(ESC)*
.
```

The fact that the data actually has been entered can be verified with the display command by entering:

```
.DF8ØØ,F82E (CR)
F8ØØ 21 8F F8 F9 11 89 14 DB
F8Ø8 FØ F6 8Ø D3 FØ Ø6 6B Ø5
F81Ø C2 ØF F8 DB FØ E6 7F D3
F818 FØ Ø6 6B Ø5 C2 1B F8 1D
F820 C2 Ø7 F8 15 C2 Ø7 F8 3E
F828 ØB 3Ø FB C3 2B F8
.
```

"EXAMINE" Command .X(reg) (SB) or .X (CR): The examine command allows the user to examine and/or change the contents of any CPU register. The data you see when using this command is not what is instantaneously in the CPU, but rather the contents which will be placed in the register by the monitor before you begin executing your program with the "GO" or "STEP" commands.

To examine all the registers enter "X" followed by a carriage return. The system will respond with a complete list with the register identification followed by the register data nn.

A=nn	Accumulator
B=nn	B register
C=nn	C register
D=nn	D register
E=nn	E register
F=nn	Processor flags
I=nn	Interrupt status and mask
H=nn	H register
L=nn	L register
S=nnnn	Stack point
P=nnnn	Program counter

Example #6: Set the user stack pointer to F87F and the program counter to F800. To accomplish this we enter the following:

```
.XS (SB) nnnn- F87F (SB) nnnn- F800 (CR)
.
```

Notice that since the program counter appears in the X(CR) command after the stack points we can go from setting the stack pointer to setting the program counter by simply entering a space (SB). The current contents of the registers are random and shown as "nnnn".

"STEP" Command .SXXXX(CR) or .S(CR): The step command allows the user to execute a program one instruction at a time. And, if necessary, examine the registers. Programs can be stepped from a specific address XXXX or from the current address in the users PC. Once you enter the command (S) and an address followed by (CR) or just a (CR), the current instruction will execute and the program counter contents will be output. To execute successive instructions press the space bar (SB). Each time the new program counter will be output. To examine the current contents of the registers, press carriage return (CR). This will print the register contents but WILL NOT execute the next instruction. To stop stepping, enter (ESC).

Example #7: Step through the first 3 instructions of the sample program and examine the contents of the D & E and the H & L registers.

```
.SF800 (CR)
F803 (SB)
F804 (SB)
F807 (ESC)
. (CR)
```

Since we did not plan on continuing our stepping, we used (ESC) to get out of the single mode and the X command to check the registers. Note that the first step in the program was to load the H & L registers with F88F which has been done. Then in the second instruction this number is placed in the stack pointer. Instruction #3 loads the D & E registers with the number 1489 and we see that this has been accomplished.

Note: Before using the single step command you MUST, absolutely MUST, have a valid RAM location in the user stack pointer using the X command. (i.e. 1XS(SB)nnnn-F885 (CR)). Failure to have a valid stack could cause the loss of all your program data and require re-booting the system. Sometimes this will cause all memory locations to be filled with "00390039....".

"GO" Command .GXXXX(CR) or .G(CR): Now, the moment of truth is at hand. Time to run the sample program full speed. The go command transfers control of the processor to the users program beginning at either the address following the command or the current address in the users PC. To check the output of the program connect a speaker or scope at the speaker output on the mother board.

Example #8: Run the sample program. First, check the user stack pointer.

```
.XS (SB)F88F-(CR)
.
```

Since the stack pointer value is valid, we next enter:

```
.GF800(CR)
BEEP!(Tone for 5 seconds then stops)
```

Note that the monitor prompt character . does not appear. The program is looping at F82B. We have, however, initialized the RST 7.5 vector interrupt, which will branch the program back to the warm start input of the monitor (saving the registers), issuing the sign-on message followed by (.) the monitor prompt. To execute the RST 7.5 interrupt simply depress the user interrupt button.

The next section of the monitor deals with recording and reloading programs using an inexpensive cassette tape recorder. There are a great variety of cassette recorders on the market. The system is designed to operate using a minimum quality recorder (we found most $30 units perform very satisfactorily) and a relatively high grade tape. The problem with tape, as you can well imagine, is drop outs. This phenomena is perfectly satisfactory in music because no one will recognize that a particular tone is missing. However, when you are recording a program that missing note or drop out will cause the program to be improperly loaded. The monitor program is designed so that it checks the incoming signals to ensure that all of the bits have been received and if an error should occur the computer will automatically recognize the error and issue a tape error message on the screen. To spare yourself a great deal of agony, make sure that the tape heads in your machine are clean and that you use a premium quality tape. There is one other consideration to be dealt with when using the low cost variety tape recorders, and that is that the output phase may vary from model to model or for that matter even between two similar units. If you continuously have trouble loading the programs it may be advisable to invert the phase of the tape input signal. This is conveniently done via a jumper on your Explorer 85 mother board. Please refer to the jumper table in the assembly instruction section.

The Explorer 85 features a tape control circuit which will automatically turn on the tape recorder provided you have it in the play or record mode, and it will automatically turn off the tape recorder when either the program has been successfully recorded, loaded or a tape read error occurs. It will be necessary to remove this remote control jack from the tape recorder when rewinding or repositioning the tape. This feature is useful when loading programs and also for custom programs which control tape recorders accessing data only when the program requires it. For example, you might consider a program of a mailing list on cassette tape where the program automatically turns on the tape recorder, takes in 50 names, turns the tape recorder off, prints the 50 names, turns the tape recorder on for another 50, etc.

"RECORD" Command .RXXXX,YYYY,ZZ(CR): The record command allows you to save a program on cassette tape for later use. Place your recorder in the "rec" mode. The tape control relay will inhibit the recorder. Enter the (R) command, the beginning address of the data to be output (XXXX), the ending address (YYYY) and assign it a program number (ZZ). All three must be entered before (CR) or an error will be generated. The monitor will then turn the recorder on and wait three seconds for the tape to get up to speed. Next about 15 seconds of leader tone is generated followed by the data and about one second of exit tone after which the recorder is turned off and control is passed to the monitor for another command.

Example #9: Store the sample program on tape as program #2. Note the program exists in memory locations F8ØØ to F82D. The following command is used:

.RF8ØØ,F82D,02(CR)
CXX (CR)   The monitor will return a CXX when recording is complete.
(XX may be any random number)

"LOAD" Command .LXX (CR): The load command allows you to recover a program from tape storage. To retrive a program enter "L" followed by the program number you wish to load. The monitor will turn on the tape and scan for a program with the proper number. Then load it into the appropriate memory (same locations as when it was recorded) and stop. In reading the tape a parity check is done on each byte to avoid errors. If an error is detected, the system will print "tape error" on the CRT and stop. Otherwise, when the program is loaded the recorder is stopped and the monitor returns to the command mode.

Note: Since all cassette recorders are not created equal it will be necessary for you to experiment and find the proper volume setting before reliable tape usage can begin. A good starting spot is about midpoint on the volume control. If tape errors persist, try increasing or decreasing the playback level. If this does not help reverse the input phasing (see tape phase jumper section) and try again.

Example #10: Fill the memory with Ø's and load program #2 from the tape. First we use the fill command to set the memory field to ØØ.

```
.FF8ØØ,F82D,ØØ(CR)
.
```

You can verify this with the display command.

Next we load the cassette with our program on it, rewind to the beginning, connect the control jack, press the "play" button and enter the command:

```
.LØ2 (CR)
```

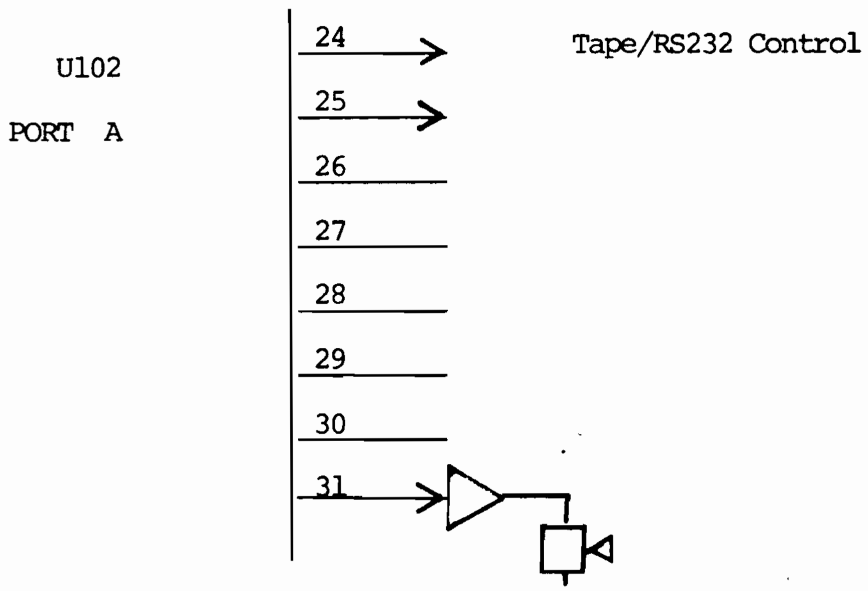
When the recorder has stopped, if no tape errors have been detected, you can use the display command to verify that the data has been loaded.

```
.DF8ØØ,F829 (CR)
F8ØØ 21 8F F8......etc.
```

Note that some random characters may be printed on the screen at the end of a tape load. Simply enter a (CR) or (ESC) to get back to the monitor.

SAMPLE PROGRAM DEVELOPMENT

Output a 1K Hz tone at the speaker output for 5 seconds then stop. First we note that the speaker is connected to bit #7 of the system control port (I/0 address FØ) and that the tape/RS232 control is located on the same port. This means that we'll have to be careful not to disturb the other bits of the port.

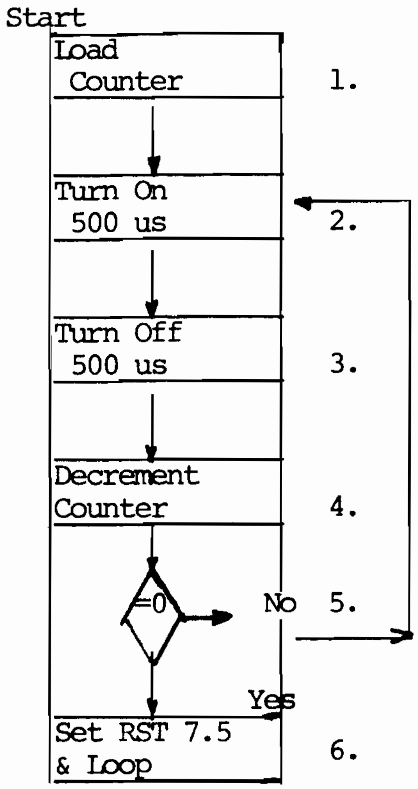


The first thing to do to generate a flow chart diagram of what we want to do.

1. To generate the 5 seconds of audio we will use a counter and decrement it each time we output one frequency cycle. Therefore, 5 seconds at 1000 Hz=a count of 5000, or hexadecimal 1388.

2. To generate the 1K Hz signal we must turn the bit on for 500 us and then off for 500 us. This will require another counter whose contents will be determined later.

3. In the flow chart we first load the 5 second counter (1). Then we begin to output the signal by making the appropriate bit a 1 and waiting 500 us (2). Next we make the same bit a zero and wait for 500 us again (3). This completes one cycle so now we decrement our main counter (4) and if it has not gone to zero jump back for another cycle (5) otherwise set RST 7.5 interrupt and loop (6).



4. Fortunately, the I/0 ports on the Intel 8355/8755 are constructed so that the user can read (input) from the port to see what has been stored there. So, to set the speaker bit to a 1 the following code is used.

```
DB  IN  FØ  This gets the port contents to the accumulator          DB FØ
F6  ORI 8Ø  This sets the speaker bit to a 1 (80 hex = 1ØØØØØØØ)    F6 8Ø
D3  OUT FØ                                            speaker bit   D3 FØ
```

To set the speaker bit to a Ø we do the following:

```
DB  IN   FØ                                   DB FØ
E6  ANI  7F    7F HEX = Ø111 1111 Binary      E6 7F
D3  OUT  FØ            ↑speaker bit           D3 FØ
```

5. To generate a 500 us delay we will decrement the B register to zero before continuing. To do this we use the instructions: (loop is the address of the DCR B).

```
06  MVI B,XX      06 XX
Ø5  DCR B         05
C2  JNZ LOOP      C2 PPQQ
```

NOTE: The 8085 data book says that DCR B takes 4 "states" (or clock cycles) and the JNZ instruction takes 7 to 10 states (7 if the zero flag is set and 10 if it is not). The statement MVI B,XX loads the B register with the number to be counted. We can compute that each loop takes 4.66 us (14 states * clock period 08.333 us), and, therefore, we will need 107 loops for 500 us (107 decimal-6B Hex) so the XX gets replaced with 6B.

6. We can now turn the speaker on and off for one cycle by using the following code:

```
          IN    FØ       DBFØ
          ORI   8Ø       F68Ø  Set Bit = 1
          OUT   FØ       D3FØ

          MVI   B,6B     066B        107₁₀
 Loop 1   DCR   B        05    Wait 500 us
          JNZ   LOOP 1   C2 PPQQ                         One Cycle

          IN    FØ       DBFØ
          ANI   7F       E6 7F  Set Bit = Ø
          OUT   FØ       D3FØ

          MVI   B,68     066B
 Loop 2   DCR   B        05    Wait 500 us
          JNZ   LOOP 2   C2  PPQQ
```

The task could be accomplished by writing this code 5000 times into memory but there is a much easier method. That is using another counter and executing the same code 5000 times.

7. Since an 8 bit register can only have a count of 256 we'll have to use at least 2 registers to accomplish this task. The register map of the 8085 shows 7 registers plus the status flags, the program counter and the stack pointer. Since we are using the accumulator (A) for the I/O operations and the B register for the 500 us counter we'll have to use two other registers for our 5000 counter. Let's use the D and E registers !!

There are a group of "register pair" commands in the 8085 instruction set to build a 5000 count counter. We might be tempted to use the following:

```
         LXI   D,1388    (1388 Hex = 5000 Decimal)  11, 1388
Loop X   DCX   D         1B
         JNZ   Loop X    C2 PPQQ
```

Where LXI D,XXXX loads both the D and E registers like they were one 16 bit register and DCX D decrements them as a pair.

But wait, unlike the DCR instruction the DCX does not effect any flags. Therefore, the JNZ (jump if the zero flag is not set) instruction will not work. This is a common mistake among novice programmers. To make this work we can use the DCR instruction on both the D and E registers individually using the following code:

```
         LXI   D,1388   21 1388
Loop X   DCR   E        1D
         JNZ   LOOP X      When E goes to zero we decrement    C2 PPQQ
         DCR   D        15 D and jump back to decrement E again.
         JNZ   LOOP X                                          C2 PPQQ
```

It's close...But this will not work either. First, if a register=Ø and we decrement it it becomes FF. That works well, but since we decrement before testing we will have to add 1 to each register to make sure we do the proper count. So now our count becomes 1489.

8. Now let's put this whole thing together one continuous program:

```
          LXI   D,1489      21 1489; Load D & E with proper count
Loop 1     IN   FØ          DBFØ
          ORI   8Ø          F68Ø
          OUT   FØ          D3FØ   ; Set the speaker bit to 1
          MVI   B,6B        066B
Loop 2    DCR   B           05
          JNZ   LOOP 2      C2 PQ; Wait 500 us
           IN   FØ          DBFØ
          ANI   7F          E67F
          OUT   FØ          D3FØ   ; Set speaker bit to a zero
          MVI   B,6B        066B
Loop 3    DCR   B           05
          JNZ   LOOP 3      C2 PQ; Wait 500 us
          DCR   E           1D
          JNZ   LOOP 1      C2 PQ
          DCR   D           15
          JNZ   LOOP 1      C2 PQ; Do it 5000 times                 3E ØB
ØØØØ1Ø11  MVI   A,ØØØØ1ØØ11        ; Unmask RST 7.5 interrupt
          SIM               3Ø
          EI                FB     ; Interrupt enable
          JMP   SELF        C3 PPQQ
```

9. This looks good, but any good programmer should always make sure the stack pointer is properly positioned (even though we don't use it here). So at the beginning we'll put the instructions LXIH, F88F selected arbitrarily to be somewhere in RAM where it will not interfere with our program.

10. Now we're ready to assemble the code into hexadecimal machine language. We will begin at location F8ØØ which is the start of the monitor RAM.

ADDRESS	HEX CODE	LABEL	MNEMONIC	OPERAND
	~~F88F~~ 21-8F-F8			
F8ØØ	21~~8FF8~~	SET SP F88F	LXI H	F88F
F8Ø3	F9	MOVE HL TO SP	SPH L ←	
F8Ø4	118914 11-89-14		LXI D	1489
F8Ø7	DB FØ	Loop 1	IN	FØ
F8Ø9	F6 8Ø		ORI	8Ø
F8ØB	D3 FØ		OUT	FØ
F8ØD	Ø6 6B		MVI B	6B
F8ØF	Ø5	Loop 2	DCR B	
F81Ø	C2ØFF8 C2-ØF-F8		JNZ	Loop 2 (F8ØF)
F813	DB FØ		IN	FØ
F815	E6 7F		ANI	7F
F817	D3 FØ		OUT	FØ
F819	Ø6 6B		MVI B	6B
F81B	Ø5	Loop 3	DCR B	
F81C	C21BF8 C2-1B-F8		JNZ	Loop 3 (F81B)
F81F	1D		DCR E	
F820	C2Ø7F8 C2-Ø7-F8		JNZ	Loop 1 (F8Ø7)
F823	15		DCR D	
F824	C2Ø7F8 C2-Ø7-F8		JNZ	Loop 1 (F8Ø7)
F827	3EØB		MVI A	ØB
F829	30		SIM	
F82A	FB		EI	Enable Interrupts 7.5
F82B	C32BF8		JMP SELF	F82B

EIA TERMINAL VERSION
SYSTEM RAM/ROM MAP DECODES

Unusable (RAM Overlay)	FFFF	
	F9ØØ	
System Monitor Uses This Area	F8FF	
	F8FD,FE	D-out (monitor initializes to console out)
	F8FB,FC	D-in (monitor initializes to console in)
	F8A3	Unconditional jump address for RST 6.5
	F8FA	Line length limit-monitor initializes to 3F (64) characters/line, or to 1F for 32 characters/line, max.=225 characters/line.
	F8DC	
User Memory Area	F8DB	
	F8ØØ	
Monitor ROM Area	F7FF	
	F5ØC	CO (console out) with line limit and CR/LF
	F4D2	CO (console out) no line limit or CR/LF
	F47F	CI (console input)
	FØØØ	Call FØ24 for Warm Start, then use carriage return for register display
User RAM Area	EFFF	
	ØØØØ	

I/O MAP DECODES

Included in Level A System	FF	CMMD CNTRL (hex keyboard command control) 8279
	FE	DISPLAY (hex keyboard display) 8279
	FD	TIMER HI (timer high 8 bits) 8155
	FC	TIMER LO (timer low 8 bits) 8155
	FB	PORT C (6 bit) 8155
	FA	PARALLEL INPUT PORT B 8155
	F9	PARALLEL OUTPUT PORT A 8155
	F8	CSR (command status register) 8155
	F4-7	I/O OVERLAYS FØ-3 (not usable)
	F3	PORT B DDR 8355/8755
	F2	PORT A DDR 8355/8755
	F1	PORT B (user port) 8355/8755
	FØ	PORT A (system port) 8355/8755
User Available	EF	
	ØØ	

NOTE: The same decoding is used for both memory and I/O operations.

If you wish to patch your own software into the Netronics system monitor a monitor source listing is now available. Please specify (Hex Keyboard or Terminal) version. Price: $25.00 postpaid.

WARRANTY: All components of this kit are warranteed for six months from the date of shipment. Defective components will be replaced free of charge if returned within six months with $1.00 each to cover testing and return postage. Return parts in a suitable package and ship insured to Netronics Research & Development Limited, Route 202, New Milford, Connecticut 06776, attention: Service Department, with a letter explaining the defect. Any parts received damaged due to poor packaging will be returned. (i.e., DO NOT ship IC's in envelopes via the mail).

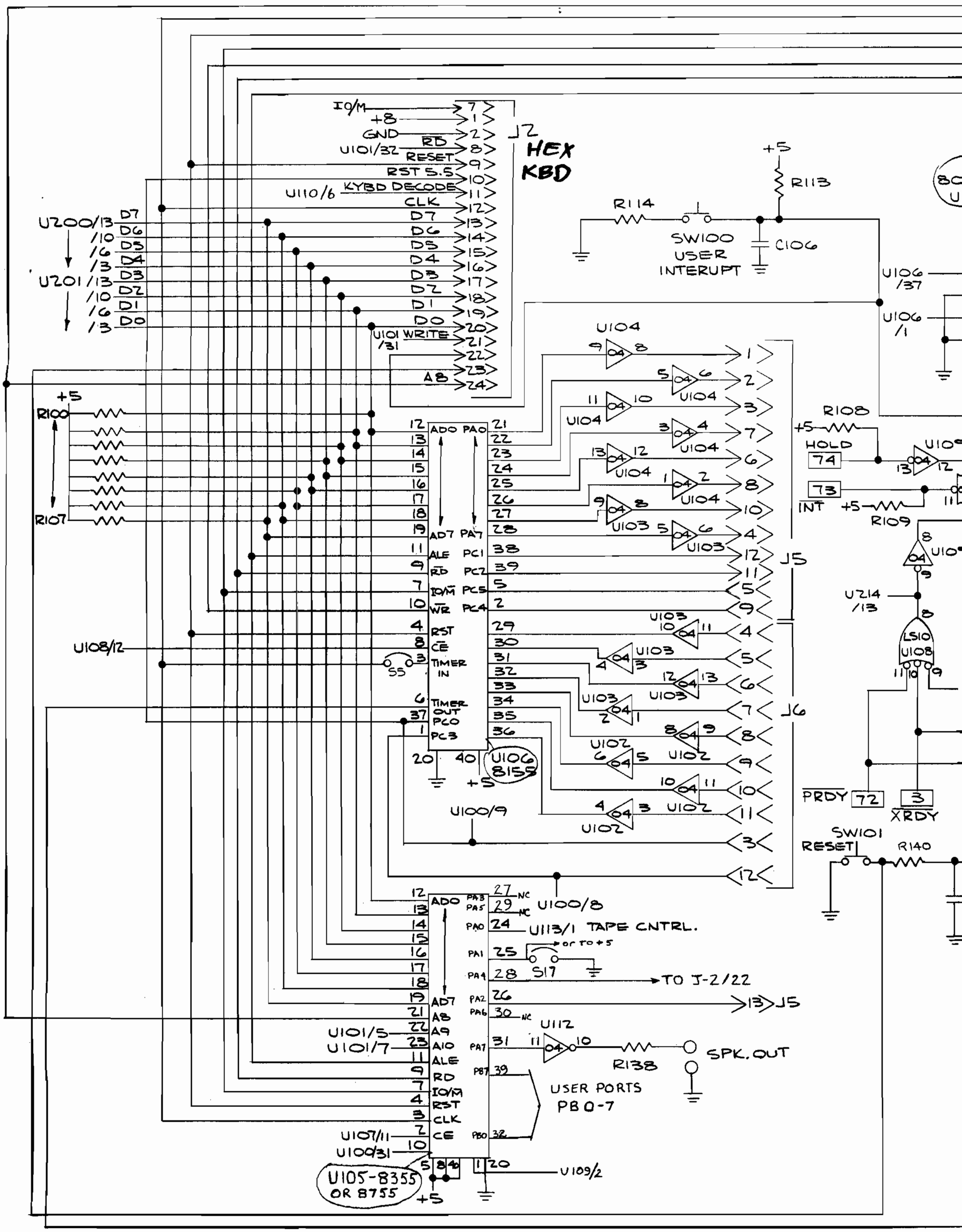
IN CASE OF DIFFICULTY: After having carefully checked your work and you still have difficulty getting your Explorer to work, the Factory Service Department will repair, fully test, and return your system for a flat fee (see below). This covers all parts, except parts destroyed by your negligence, (i.e., IC installed backwards, broken, etc.), and return postage. Package the unit (less cabinet) carefully and return insured with a letter describing the difficulty.

If your system includes other level components, please see the instruction book for fees which apply to the further expanded systems. If any components are added which are not part of a Netronics kit you will be advised of the service charge prior to any work being done. If you have added any "Levels" to your system (using your own parts) it would be advisable to purchase the appropriate assembly manual, which will contain any factory modifications or updates, prior to returning your unit.

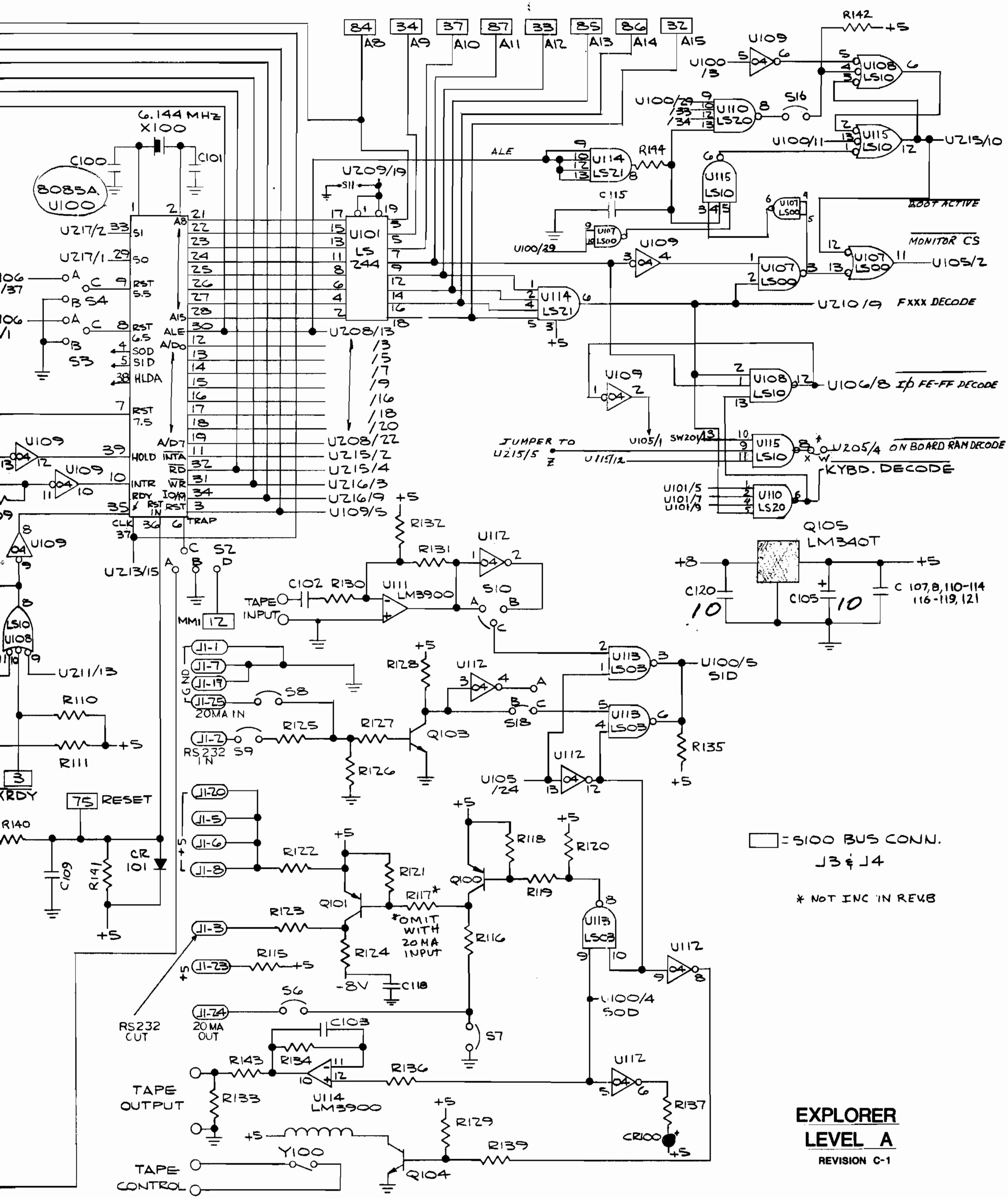
SCHEDULE OF IN WARRANTY FACTORY TROUBLESHOOTING PRICES*

	FLAT FEE
Level A	$12.50
Hex Keypad/Display	7.00
Level A + B	16.50
Level A + B + D and/or E	20.00
Power Supply	6.50
ASCII Keyboard	7.00
Video Display Board	9.50

* Covers cost of all parts except those destroyed by the customer. These prices are not valid for levels added using parts not obtained from Netronics. If you have parts not supplied by Netronics send your unit and request a quotation.



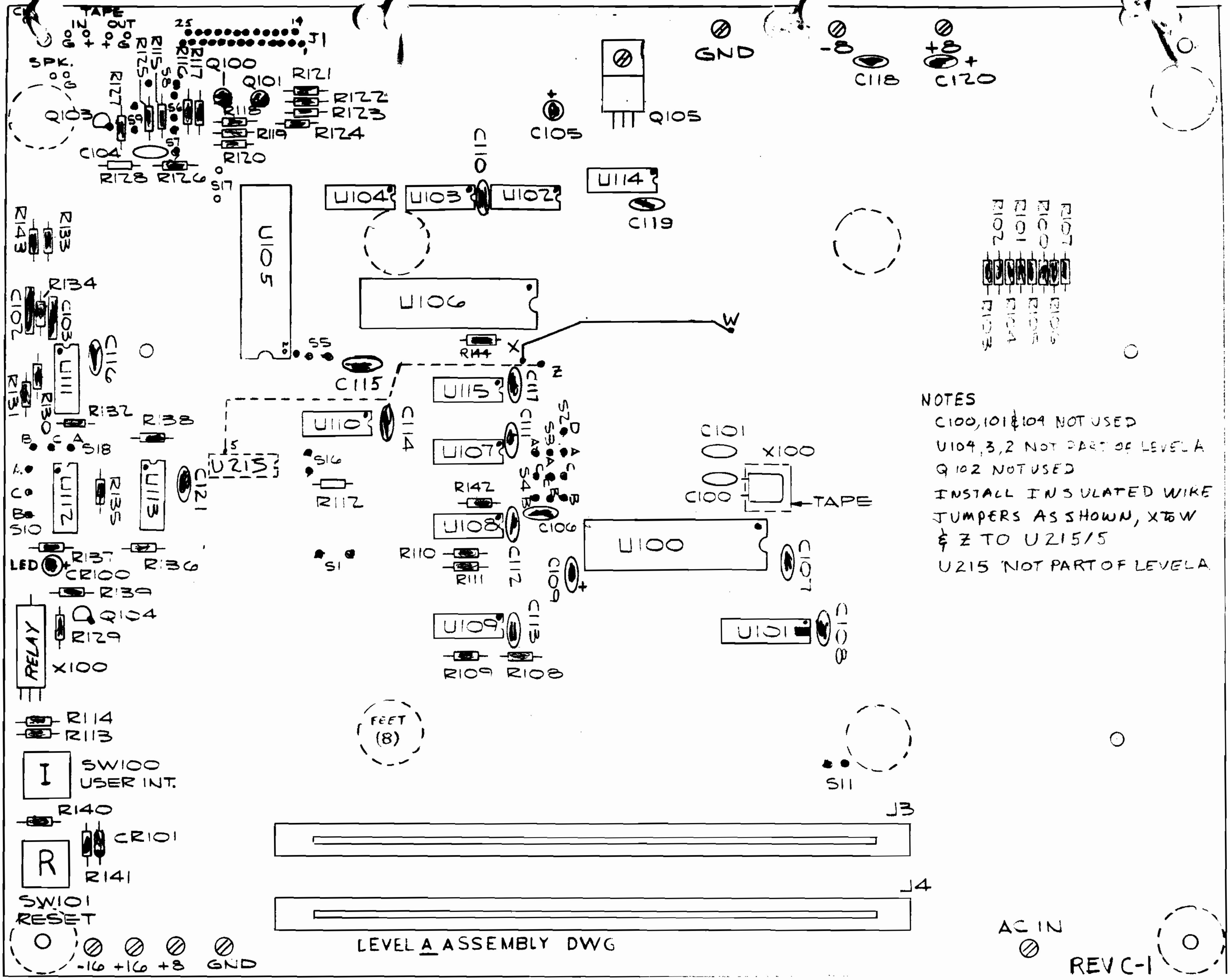
HEX KBD
J2
IO/M
+8
GND
RD
U101/32
RESET
RST 5.5
U110/6
KYBD DECODE
CLK
U200/13
U201/13
D7 D6 D5 D4 D3 D2 D1 D0
U101 WRITE /31
A8
+5
R100
R107
R113
R114
SW100 USER INTERUPT
C106
U106 /37
U106 /1
U104
U103
U102
J5
J6
R108
HOLD
74
73
INT
R109
U214 /13
LS10 U108
PRDY 72
3 XRDY
SW101 RESET
R140
ADO PAO
AD7 PA7
ALE PC1
RD PC2
IO/M PC5
WR PC4
RST
CE
TIMER IN
TIMER OUT
PCO
PC3
U108/12
S5
U106 8155
U100/9
U100/8
U113/1 TAPE CNTRL.
or to +5
S17
TO J-2/22
13 J5
U112
R138
SPK. OUT
USER PORTS PB0-7
U101/5
U101/7
U107/11
U100/31
U105-8355 OR 8755
U109/2



EXPLORER

LEVEL A

REVISION C-1



LEVEL A ASSEMBLY DWG

REV C-1

NOTES

C100,101 & 104 NOT USED

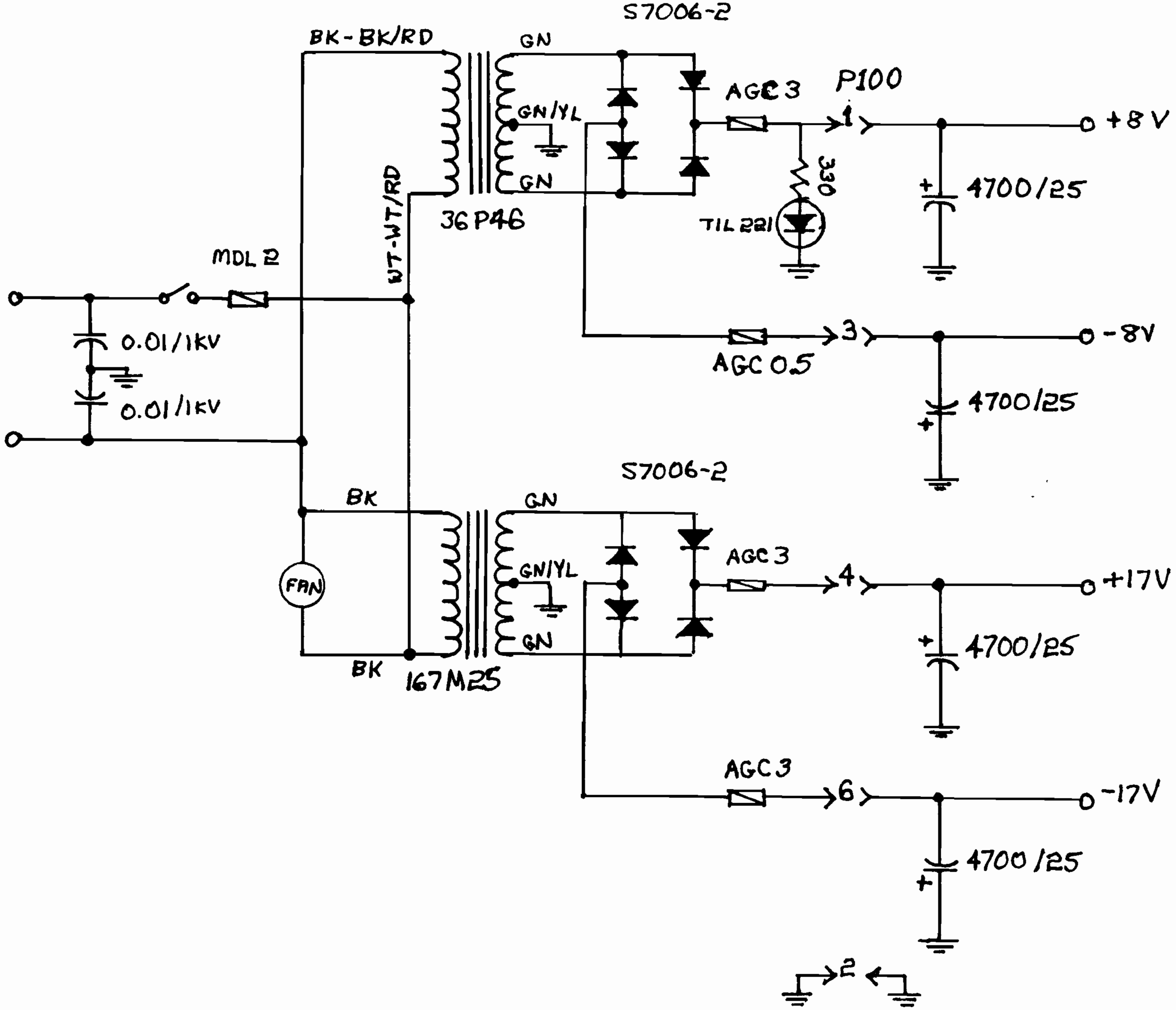
U104,3,2 NOT PART OF LEVEL A

Q102 NOT USED

INSTALL INSULATED WIRE JUMPERS AS SHOWN, X to W & Z TO U215/5

U215 NOT PART OF LEVEL A

EXPLORER 85 POWER SUPPLY



MAY 21, 1980

EXPLORER 85 INTERFACE CONNECTOR J1

1	BRN	GND
2	RED	RS232C RECFIVE
3	ONG	RS232C TRANSMIT
4		
5	YEL	+5V
6	GRN	+5V
7	BLU	GND
8	VIO	+5V
9		
10		
11		
12		
13		
14		
15		
16		
17	SLT	GND
18		
19		
20	WHT	+5V
21		
22		
23	BLK	+5V
24	BRN	20mA TRANSMIT
25	RED	20mA REACFIVE