CA-23

EPROM

Programer Manual

Jan. 1981

CONTENTS

Introduction to the CA-23

In 1978, a new family of EPROM's were introduced by integrated circuit manufacturers. These parts are quickly becoming the industry standard for several reasons. Previous families of EPROM's required multiple voltages during the read mode and in the program mode. The multiple supply requirement of the old families almost always required an additional system power supply with the single purpose of support for these EPROM's. The new family of EPROM requires +5V only (system supply) in the read mode, thus, eliminating the support power supply. Since the introduction of these EPROM's, the family has grown at an exciting rate becoming not only larger in density to save board "real estate" and parts count but also decreasing the cost per bit. Also, as new manufacturing processes are developed yielding greater density, faster access times, and higher die yield, the die for the early generation of the family are being relegated to the new processes.

The CA-23 board supports this new family of EPROM providing a convenient method of programming and testing them. The program/data to be written to the EPROM may originate from several sources. A master EPROM could be used to make copies or code stored in memor disk may be used.

The CA-23 is designed for the new +5V family only and may not be used to program the older generation of EPROM.

Software Operation

After booting and displaying which version of the EPROM PROGRAMMER SOFTWARE is on-line, "PROG", which is the "basic" interactive program will be run. This program is used to prompt the operator and guide him through the various levels of the program. The actual programming is carried out by a machine code device handler ("HANDLR" on the directory).

The outermost level of the program is the program entry/re-start point. At this time, the CA-23 will be tested for power ON/OFF. If the power is on the operator will be asked to:

SLIDE THE ON/OFF SWITCH TO OFF

This is done to insure that no EPROM's will be inserted or removed while power is applied which could possibly damage the device. The power switch removes "all" power from the CA-23 including the programming voltage. If the power is turned off or is already off, the message

SLIDE THE PROGRAM/READ SWITCH TO READ
HIT RETURN WHEN READY?

will be displayed. The program/read switch is a hardware "fail-safe" switch used to enable/disable the programming voltage (+25V). Slide it to READ and depress carriage return (CR) when ready to continue. If the switch is already in READ, simply depress the CR key.

A list of EPROM's that the program can handle is now displayed. Select the EPROM you are working with and type the appropriate number after the prompt:

PART NUMBER YOU WISH TO WORK WITH?

Other manufacturers' parts can be cross-referenced to these part numbers if the EPROM you wish to program is not listed.

The CA-23 board incorporates a switch to reconfigure connections to the two major pin-out types of +5V only EPROM. This switch is called the TYPE switch which must be correctly set before power is applied to the CA-23 and the EPROM. At this time, the program will prompt the operator to set the switch to either position A or B:

```
SLIDE THE (TYPE) SWITCH TO POSITION (  )
HIT RETURN WHEN READY?
```

When done or if the switch is already in the correct position, depress CR.

There are six major modes of operation. These are now displayed:

```
1 = DUPLICATE
2 = VERIFY
3 = LIST
4 = PROGRAM FROM MEMORY
5 = LOAD MEMORY
6 = EDIT/SAVE MEMORY

WHAT DO YOU WISH TO DO?
```

One of the major modes is now selected by typing the appropriate number and depressing CR.

MODE 1 = DUPLICATE

This mode is used to make DUPLICATE copies of a master EPROM. The master EPROM and the EPROM to be programmed must be of the same part number or cross-referenced as previously discussed. If a copy EPROM is to be programmed and the master is a different type, then MODE 5 should be used to LOAD MEMORY from the master

and MODE 4 should be used to program the copy.

SUPPRESS LISTING DURING THIS ACTION?

Answer yes (Y) or no (N). "N" will instruct the machine code programmer routines to list the location and data being read or programmed. Answering "Y" will suppress the listing (unless an ERROR is encountered), thereby decreasing the amount of time required for an operation to be completed. If an ERROR is encountered, the location and data will be listed regardless of this software switch.

```
SLIDE THE TYPE SWITCH TO POSITION (  )
LOAD THE MASTER EPROM IN THE MASTER SOCKET
LOAD THE COPY EPROM IN THE COPY SOCKET
SLIDE THE ON/OFF SWITCH TO ON
```

A double check is now made to insure that the TYPE switch is in the correct position, and then the EPROM's should be loaded into the correct sockets. The load lever (of the sockets) must be in the closed position before activating the power on switch. When the power on switch is activited, the program will prompt the operator:

SKIP BLANK VERIFY TEST (Y OR N)?

If "N" is answered, the program will test the copy EPROM to insure that it is blank (all $FF) before attempting to program it. In all cases, "N" should be the response unless an EPROM is being edited.

SLIDE THE PROGRAM/READ SWITCH TO PROGRAM

This will apply the (+25V) programming voltage to the software controlled Tri State programming signal which applies the (+25V)

only when required to program the EPROM.

The program will now proceed to transfer the master data to the copy. Any further messages are self-explanatory. The sequence is: check copy for all $FF, load master data in memory, program copy with master data, verify copy = master.

MODE 2 = VERIFY

SUPPRESS LISTING DURING THIS ACTION?

This is the same as explained in MODE 1.

1 = VERIFY COPY TO MEMORY
2 = VERIFY COPY TO MASTER
3 = VERIFY MASTER TO MEMORY

The VERIFY MODE compares an EPROM to another EPROM or memory. A message will be generated to indicate an ERROR if a discrepancy exists and will list the location and data.

MODE 3 = LIST

1 = FROM MASTER SOCKET
2 = FROM COPY SOCKET
3 = FROM MEMORY

When one of the above three is selected, the program will either display memory immediately (#3), or prompt the operator to load the master (#1) or copy (#2) EPROM, and then list the data. In either case, all three will always be listed although they may be invalid (not installed). This is done so that if the operator wished to list all three, (#1) or (#2) could be answered, then the master and copy EPROM's loaded, and all the data for each, including memory, would be listed.

MODE 4 = PROGRAM FROM MEMORY

SUPPRESS LISTING DURING THIS ACTION?

This is explained in MODE 1.

The operator is then asked to load the copy EPROM in the copy socket and apply power and then:

SLIDE THE PROGRAM/READ SWITCH TO PROGRAM
SKIP BLANK VERIFY TEST (Y OR N)?

This is explained in MODE 1. The program will now proceed to program the copy to equal the memory data and then automatically go into the VERIFY MODE.

MODE 5 = LOAD MEMORY

SUPPRESS LISTING DURING THIS ACTION?

This is explained in MODE 1.

1 = FROM MASTER SOCKET
2 = FROM COPY SOCKET
3 = FROM DISK

This mode loads the data memory from one of the three sources. Sources (#1) and (#2) are self-explanatory. Source (#3) is from a disk file (one of seven) which has previously been saved by using the MODE 6 SAVE command. A full 8K will always be loaded regardless of the size of the EPROM as is explained in MODE 6.

MODE 6 = EDIT/SAVE MEMORY

This mode is used to EDIT the memory buffer or SAVE the memory buffer in one of seven files that are available. The computer responds with:

BASE ADDRESS(HEX)?

The computer always stores the EPROM data in a predetermined memory buffer area beginning at $5ØØØ. This actual buffer address could be confusing to the operator. Consider the example where the actual EPROM to be programmed was to be memory mapped in the destination computer to begin at $FCØØ and end at $FFFF (1K EPROM). Now assume that the EPROM data is already resident in the memory buffer area and the user wishes to edit the memory. Normally, an assembly listing would be used to determine the patch(s) and the hexadecimal address(s) of the change(s) as found in the listing. Assume that the EPROM listing begins at $FCØØ and it is desired to change location $FD12. At this point, $FCØØ should be typed. Internally, the computer will now associate $FCØØ with the first location in the buffer which is $(5ØØØ) and $FCØ1 would equal $5ØØ1. If a non-hex character is typed the input will be aborted, or, if an invalid offset is detected such as specifying the base address of a 1K EPROM as $FDØØ.

(O)PEN,(L)IST,(P)RESET,(S)AVE,E(X)IT?

OPEN is used to open a location for change. LIST is used to view memory or list on a printer, PRESET sets memory (RANGE) to an input value, SAVE saves memory to a disk file and EXIT leaves the EDIT MODE and returns to the outermost level of the program.

To EDIT location $FD12, as mentioned earlier, type "O". The program will finish typing open and ask for the location (?). Type FD12. On the left-hand side of the screen, FD12 will be printed, then a space, and then the data in that location is presented. At this time, the data to be entered should be typed in

hexadecimal. If the data is accepted, the location will be changed to the new value and the next location will be opened for change. If a non-hex character is detected, the location will be unmodified and reopened for change. To advance one location without changing the currently open location, hit LINE FEED. To back up one location type (^) which is SHIFT-N on some keyboards. To exit, hit CR.

The other modes are self-explanatory. LIST and PRESET ask for a RANGE to work with and PRESET will ask for data to fill memory with. The SAVE function will save the memory buffer in one of seven files. The full 8K byte buffer is always saved regardless of the size of the EPROM or the actual memory buffer size. In systems with 24K bytes of memory, the actual buffer size is only 4K bytes.

Data Files

Seven data files are resident on the programmer disk. These are maintained as FILE1 through FILE7. On five inch systems, two files are maintained. Each file consists of four tracks, each track storing 2Ø48 bytes of the 8192 byte file. The 2Ø48 limit on eight inch system file tracks retains compatability with five inch systems (eight inch systems are actually capable of saving 3Ø72 bytes per track). The software SAVE command always saves the full 8192 byte RAM buffer ($5ØØØ-$6FFF) on the four tracks of the file regardless of the byte size of the EPROM or the actual buffer size. This is also true of the LOAD/FILE commands. That is, when loading the RAM buffer using the LOAD command, the full 8K RAM buffer is loaded from the four tracks in the file.

The SAVE (SA) command stores the buffer by effectively simulating the operating system SA command. If FILE1 (eight inch) were being saved, the following sequence occurs:

```
A*SA 33,1=5ØØØ/8
A*SA 34,1=58ØØ/8
A*SA 35,1=6ØØØ/8
A*SA 36,1=68ØØ/8
```

This same function could be performed manually by exiting the programmer (this can be done by depressing CR in response to any of the prompts) to the BASIC IMMEDIATE MODE and typing EXIT. The SA command may now be used as explained in detail in the OS-65D manual. The buffer is loaded in a similar manner.

```
A*CA 5ØØØ=33,1
A*CA 58ØØ=34,1
A*CA 6ØØØ=35,1
A*CA 68ØØ=36,1
```

The file handler of the programmer software was implemented in this fashion so that easy manipulation of files is maintained. An example could be that the user has just finished assembling a program (1K bytes) to $8CØØ-$8FFF and wishes to program an EPROM with this data. The assembler should be exited, "EX".

A*

The assembler disk is now removed and the programmer disk inserted in the floppy drive. To SAVE this data in FILE1:

A*SA 33,1=8CØØ/8

Now reset the computer and boot up the programmer disk. Use the LOAD MEMORY function to load FILE1 and then the PROGRAM FROM MEMORY function to program the EPROM copy.

Before programming the copy, it may be wise to use the EDITOR LIST command to insure that the correct data has been transferred by these sequences.

Another example might be the case where two 1K byte EPROM's are to be substituted by one 2K byte EPROM. Using the programmer, load the first EPROM to memory, exit to the IMMEDIATE MODE, and then exit to the operating system.

A*

Tracks 64-76 (eight inch) have been reserved for the following types of operations:

The first EPROM data (1K byte) is still at $5ØØØ-$53FF and could be saved by:

A*SA 64,1=5ØØØ/4

Now type RE BA to return to BASIC, run and LOAD MEMORY with the second 1K byte EPROM, then enter the operating system again and

```
A*SA 65,1=5ØØØ/4
```

The two (1K byte) EPROM's (data) are now saved on tracks 64 and 65. To load the RAM buffer with the 2K byte data:

```
A*CA 5ØØØ=64,1
A*CA 54ØØ=65,1
A*RE BA
RUN
```

The 2K byte EPROM is now programmed using the function PROGRAM FROM MEMORY and the buffer could be saved using the EDIT/SAVE command.

Power Requirements

The CA-23 requires two voltages for full functional operation:

+5V/±5% @ 500MA

+25.6V% @ 500MA

The +25V is required to program an EPROM and is not necessary for read only operations. Most adjustable +24V power supplies can be adjusted to obtain this voltage. Some users have successfully used the +24V supply in the eight inch drive cabinet for this purpose, adjusting it to +25V when programming a part and readjusting to +24V when complete.

The +5V requirement can be satisfied by either a dedicated accessory supply or by connection to the computer system power supply.

There are four zero insertion sockets supported on the CA-23. There are 24-pin and 28-pin master and copy sockets. The master sockets are used to LOAD MEMORY from a master EPROM, VERIFY it or DUPLICATE it. The master socket can never have the voltage required for programming applied to it. Therefore, the master EPROM can't be inadvertently altered by operator error or other means when it is inserted only in the master socket. The copy socket is used for programming EPROM's.

SW1 is used to remove all voltages from the CA-23, so that the EPROM's may be safely inserted or removed from the zero insertion sockets.

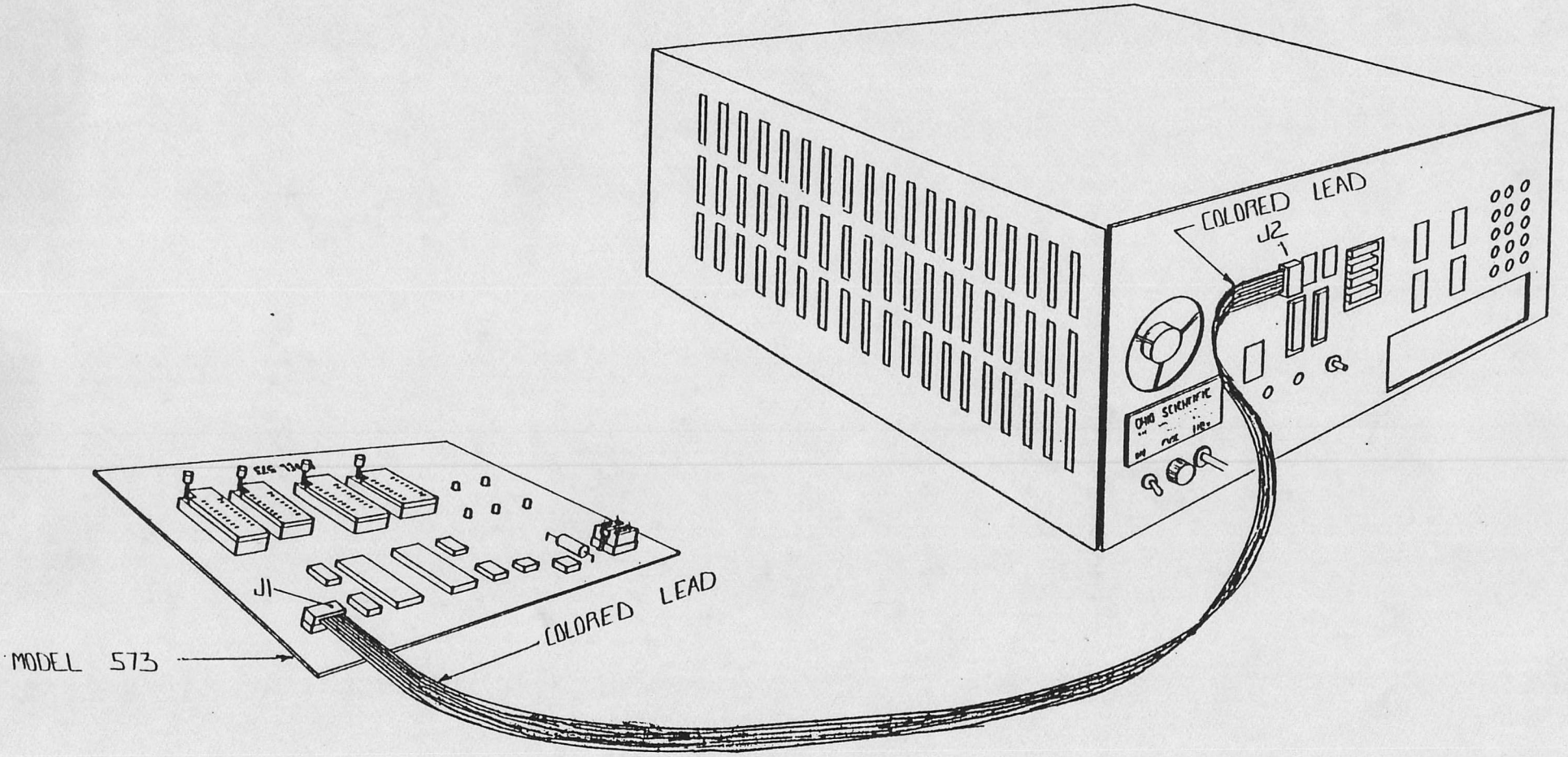
SW2 is used to mechanically inhibit or enable program voltages to the EPROM copy sockets.

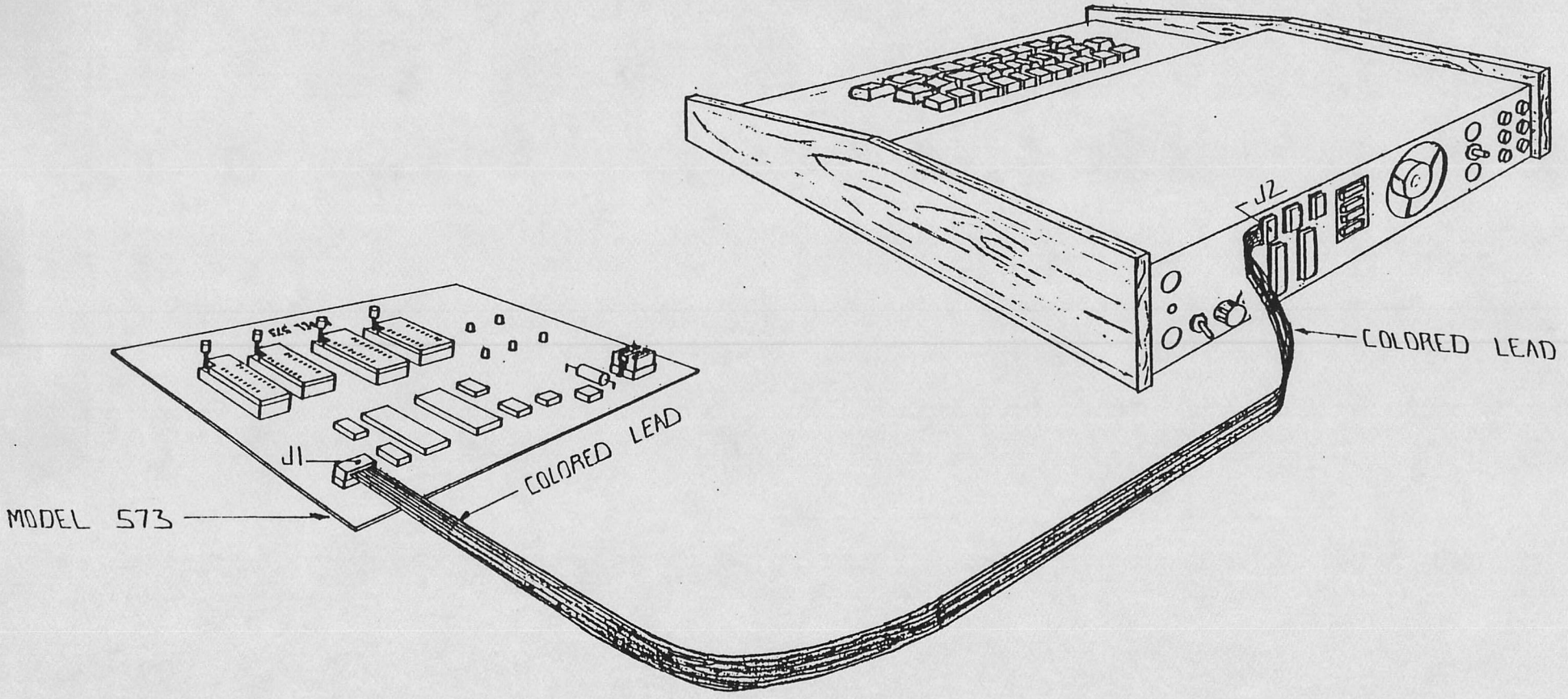
SW3 routes signals to the EPROM sockets in one of two manners depending on the type of EPROM in use.

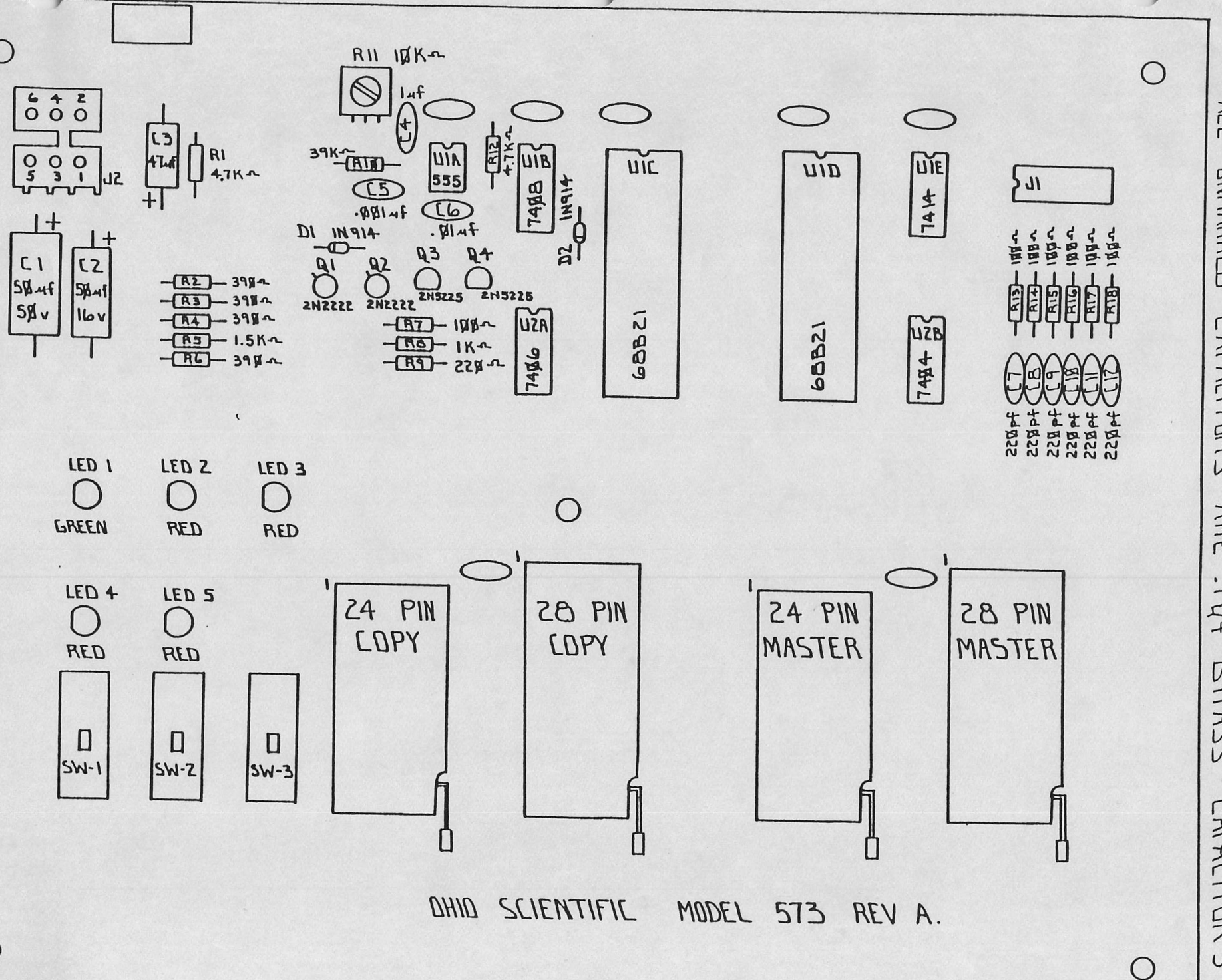
U1D applies addresses (AØ-A7) to the sockets and either provides write data or read data (bidirectional) depending on the mode.

U1C applies addresses (A8-A9) and (A1Ø-A14) where required. U1D also controls several other functions.

The signal VPP(SW) is capable of three levels: +25V, +5V,+.6V. These voltages are required to perform programming and reading of the various types of EPROM's by the CA-23. Q1 sources the +25V, Q2 sources +5V through D1 and U2A pin 12 sinks to ground. U1B pins 13, 12, and 11 logically prevents contention of the three devices.

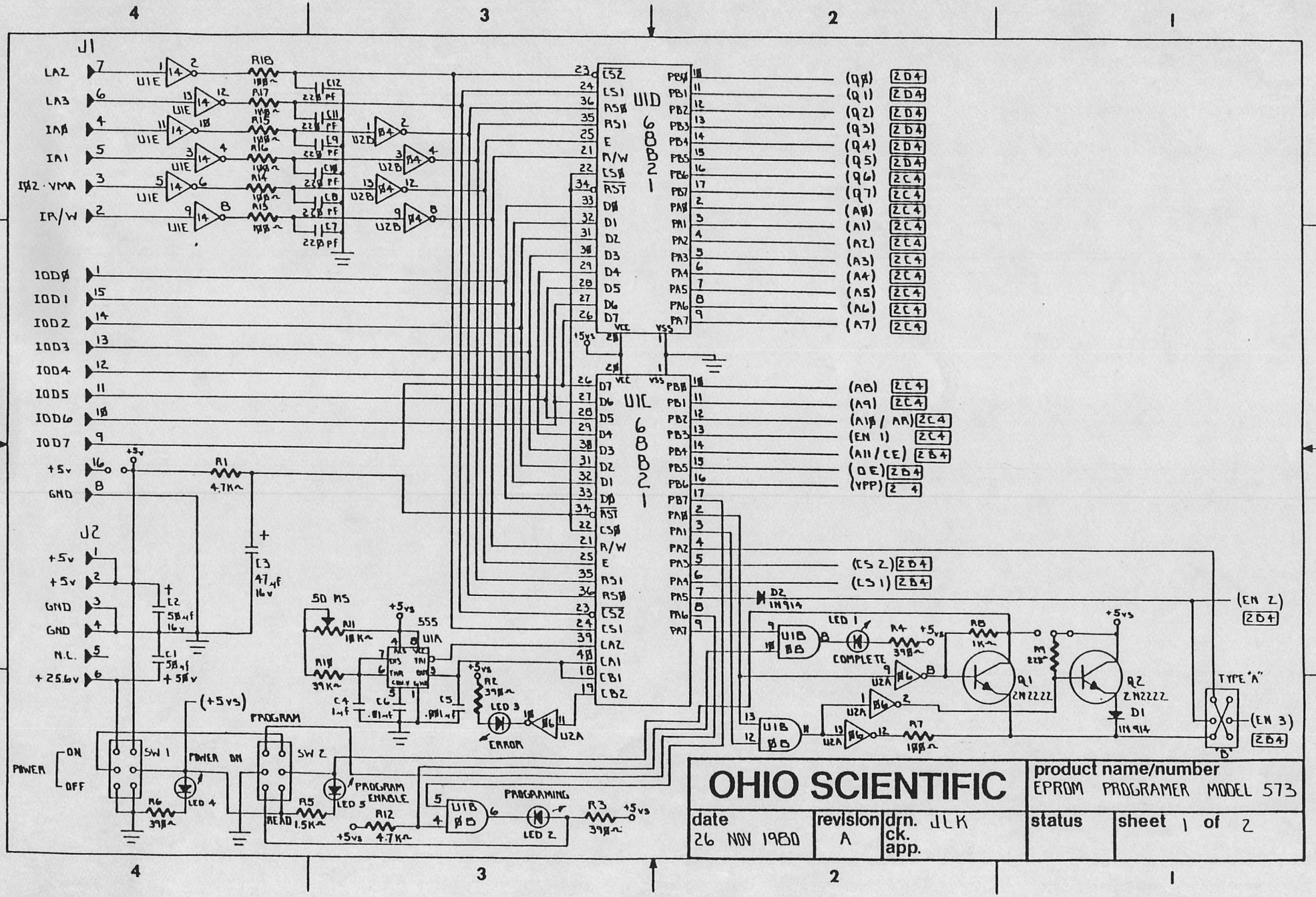

COLORED LEAD
J2
1
J1
COLORED LEAD
MODEL 573

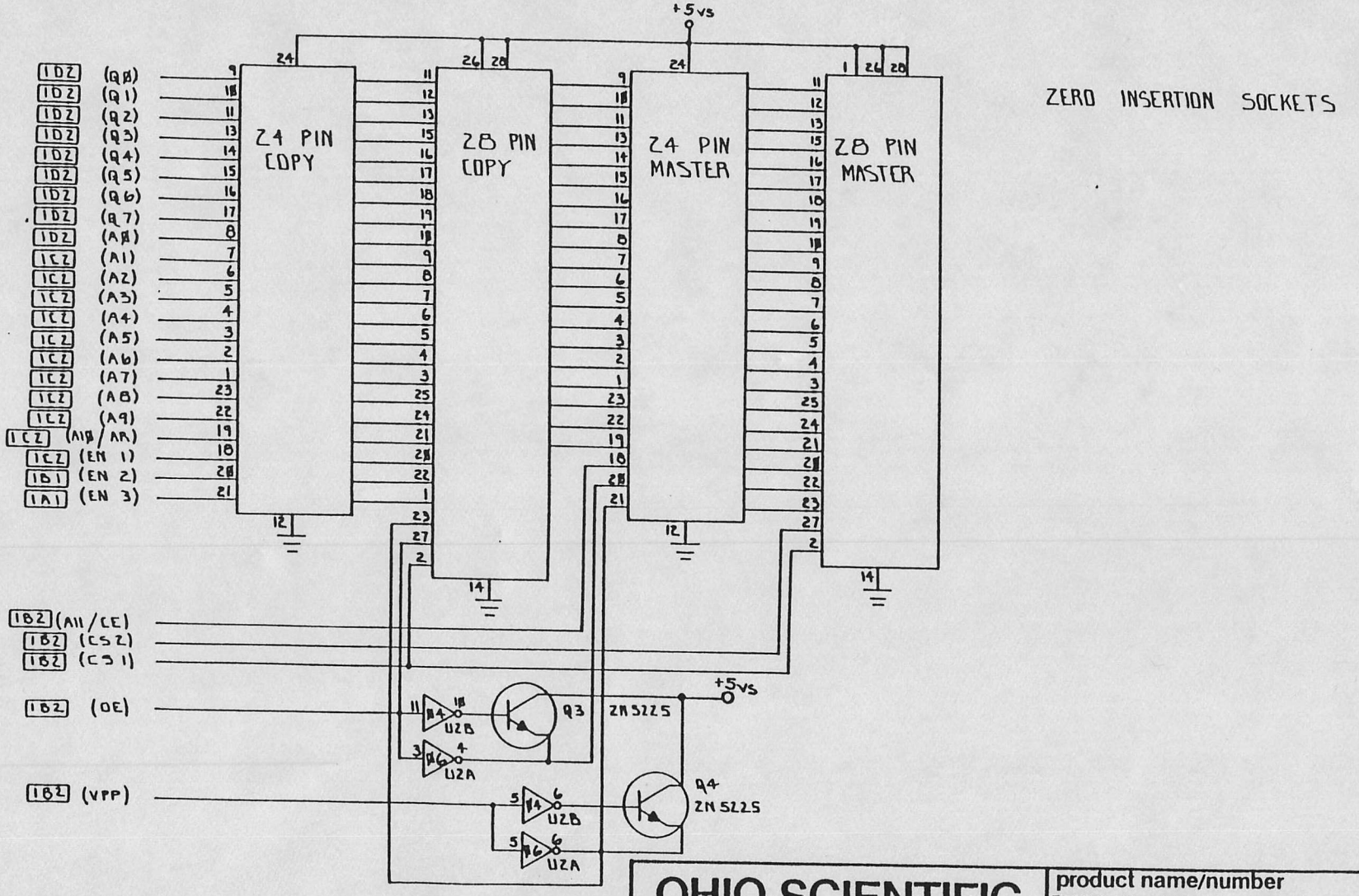

J2
COLORED LEAD
J1
COLORED LEAD
MODEL 573



OHIO SCIENTIFIC MODEL 573 REV A.

ALL UNMARKED CAPACITORS ARE .1uf BYPASS CAPACITORS


J1
LA2
LA3
IA0
IA1
IO2 · VMA
IR/W
U1E
R18
R17
R15
R16
R14
R13
C12
C11
C9
C10
C8
C7
U2B
U1D 6821
U1C 6821
IOD0
IOD1
IOD2
IOD3
IOD4
IOD5
IOD6
IOD7
+5v
GND
J2
N.C.
+25.6v
R1
4.7K
C3
47µf 16v
C2
50µf 16v
C1
50µf 16v
50 MS
R11
10K
R10
39K
555
U1A
C4
1µf
C6
.001µf
C5
.001µf
R2
390Ω
LED 3
ERROR
U2A
(+5VS)
PROGRAM
SW 1
POWER
ON
OFF
LED 4
R6
390Ω
SW 2
READ
R5
1.5K
PROGRAM ENABLE
LED 5
R12
4.7K
U1B
PROGRAMING
LED 2
R3
390Ω
D2
IN914
LED 1
COMPLETE
R4
390Ω
U2A
R8
1K
Q1
2N2222
R9
220
Q2
2N2222
D1
IN914
R7
10K
TYPE "A"
(EN 2)
(EN 3)
(Q0)
(Q1)
(Q2)
(Q3)
(Q4)
(Q5)
(Q6)
(Q7)
(A0)
(A1)
(A2)
(A3)
(A4)
(A5)
(A6)
(A7)
(A8)
(A9)
(A10 / AR)
(EN 1)
(A11 / CE)
(OE)
(VPP)
(CS 2)
(CS 1)
2D4
2C4
2B4
OHIO SCIENTIFIC
product name/number
EPROM PROGRAMER MODEL 573
date
26 NOV 1980
revision
A
drn. JLK
ck.
app.
status
sheet 1 of 2



OHIO SCIENTIFIC			product name/number EPROM PROGRAMER MODEL 573	
date 26 NOV 1980	revision A	drn. JLK ck. app.	status	sheet 2 of 2