OISTO

SUPER PRODUCTS

M PROM ADAPTER

C.R.C COMPUTER inc
C.R.C ORDINATEUR inc

THE DISTO SUPER MPROM ADAPTER

Congratulations on the purchase of your new MPROM adapter. The quality materials and workmanship used in this product insures years of trouble free use. The MPROM Adapter will work only with the DISTO SUPER CONTROLLER. Only one DISTO Super Adapter will fit into the controller at a time. With the proper software (included) this adapter will allow you to program, verify, read and save a 2764, 27128 and 27256 EPROM.

INSTALLATION INSTRUCTIONS.

1 - Turn your computer, and any another devices connected to it, off. Remove the controller from the computer and remove the drive cable. Remove the cover of the controller by removing the 2 screws on each side and locate the 17 pin connector J1 on the controller board. See the Diagram #1 in the controller's user's manual.

2 - Locate the single row female connector on one edge of the adapter. Hold the adapter so that the components face towards the controller. Insert the adapter into the connector so that the solder side of the adapter faces upwards and the adapter completely fits inside the controller case. Make sure that the connector is properly positioned so that there are no pins sticking out on eihter side. As you can well see, you must operate this adapter with the controller cover off. PLug the controller and MPROM adapter back onto the computer.

3 - Supplied with the MPROM adapter is a battery adapter. Insert 3 9-volt batteries (alkaline recommended) into the adapter. Insert the adapter into the 2 pin connector on far side of the adapter, making sure that the white dots corresponds to each other. To save your batteries,

only connect them when you are ready to program an EPROM. If you don't want to use batteries, any unregulated DC adapter that can deliver 25 to 40 volts at 100 ma. will do. A minimum of 32K of memory is required to run the MPROM software.

4 - The regulated voltage to program an EPROM is 21.0 volts. Some of the newer EPROMs (sometimes noted by the letter A on the part number) require only 12.5 volts to program. In that case you must change the regulated voltage of the MPROM adapter. To do that, locate the three pins, to the left of the battery connector. Move the jumper from covering the top and center pins to cover the center and the bottom pins. This will change the programming voltage to 12.5 volts.

SOFTWARE

In order to program an EPROM, data is required. This data can be anything from a customized DOS, to an often used utility, to a fun game. (In some cases, a knowledge of machine language programming may be necessary). Whatever the data, it must be in memory before you LOAD and EXEC the MPROM software. The software expects the data to start at memory location 12288 or $3000 in hex. All numbers in hex will preceded by a '$'. In the case of a 2764 the data ends at 20479 ($4FFF). For a 27128 the end address is 28671 ($6FFF). In the case of the 27256 only half of the EPROM can be programmed at a time. In order to program a 27256 you must program both halves, one at a time, making sure that you are programming the right data. In both cases, the data start and end locations are the same as a 27128.

After the data to be programmed is loaded into memory, load the MPROM software;

```
LOADM "MPROM" : EXEC (ENTER)
```

You will be greated with the title page and the option to work with a 2764, a 27128 or one of the two halves of of a 27256 EPROM. Choose the proper EPROM and hit (ENTER). Note, if you insert the wrong EPROM into the socket, you will not damage the EPROM or the adapter, but it will not program or verify properly. You can also use the MPROM auto loader program;

```
    RUN "MPROM
```

This Basic program will ask you for a file to load in. It will check to see if it is a machine language file, and load it with the required offset (if any) to reside at 12288 ($3000). It then shows the original load address and offset required. It will also tell you what EPROM you will need. Since the file length for a 27128 or half of a 27256 are the same, this program cannot tell the difference. Hitting any key after this will bring you to the title page of the MPROM software.

The next screen to appear is the MAIN menu. The following is a detailed description of each item in the main menu;

1 - EPROM ERRASED? Selecting this item from the menu will verify that the EPROM is completely erased. Initialy and after each erasure, all bits of the EPROM are in the logical "1" state. The status line will start of by printing;

```
            CHECKING FOR FF's
```

If all bits are erased, the status line will then say " - OK !" at the end. If a location is not properly erased, the status line will appear as such,

```
    AT XXXX EXPECTED FF FOUND YY
```

where XXXX is the EPROM address (starting from 0), FF is the expected data (in this case it is all 1's or 255, $FF in hex) and YY is the actual data it found. If there is more than one address not erased, then the last non-erased address will appear on the status line. When checking an EPROM to make sure that is erased, the " - OK" must appear on the status line. If just one bit is missing, the EPROM is not fully erased and cannot be programmed properly. To erase a used EPROM, an ULTRAVIOLET light must be used. Refer to the EPROM ERASER's manual, for more details.

2 - PROGRAM EPRPOM. Selecting item #2 from the main menu will start the programming process. This process will transfer the data that is in memory, to the EPROM. Before starting this process, make sure that the proper, erased, EPROM is placed in the socket. Pin one of the EPROM goes to pin one of the socket. Pin one of the socket is in the lower right hand corner as seen from sitting in front of the keyboard when the MPROM and controller are plugged into the computer. Also make sure that the battery pack is properly connected before starting. The process starts by displaying the following on the status line;

```
PROGRAMMING ADDRESS: XXXX YY
```

where XXXX is the current EPROM address being programmed and YY is the program pulse width. Let me explain. In the good ol' days, it was believed that each location of an EPROM had to be programmed for about 50ms. or about 1/20 of a second. It doesn't sound like a lot of time, but when you have a 16k EPROM to do, it takes over 14 minutes. Today's EPROMs do not have to be programmed so long. A 1ms progam pulse is given to the address being programmed. If it proves to be enough, a 4ms. insurance pulse is given. If the address is not true, more 1ms. pulses are given until it is, or a maximum of 15 is reached. Then an insurance pulse of 4 times the amount of tries is given, hence the value of YY changes. If after all this, the EPROM address is still not true, the program gives up and issues a status line of;

EPROM NOT PROGAMMED PROPERLY AT ADDRESS: XXXX

where XXXX is the address at which the EPROM failed. At this point, try to program the EPROM again. If it fails at the same address, there is a problem with the EPROM and should not be used. If it fails at the next or so address, chances are that your batteries are weak and should be replaced. Caution, never plug or unplug the battery pack with an EPROM in the socket. Always remove the EPROM first. When all the addresses of the EPROM are properly programmed, the status line will read " - OK!". 0 — 3Fxx

3 - VERIFING EPROM. Though verifying an EPROM is done during the programming stage, it is sometimes necessary to verify an EPROM without programming it, like when comparing two EPROMs. Choosing #3 from the main menu will do just that. Verifing the EPROM is done by comparing the data found in the EPROM to that in memory. The status line will read;

VERIFING EPROM

If no difference is found, then the status line will read the now familiar " - OK!" If a difference is found the status line will change to;

AT XXXX EXPECTED YY FOUND ZZ

where XXXX is the address at which the discrepancy was found, YY was the data it expected to find and

ZZ is the actual data found. If more discrepancies are found, only the last one will be shown on the screen.

4 - MOVING EPROM TO MEMORY. It is sometimes necessary to transfer data from one EPROM to another or from EPROM to disk. In any case data, must first be transfered from EPROM to memory. Selecting this item from the main menu will do just that, transfer the contents of the EPROM into memory. The data will reside in memory starting from 12288 ($3000) and will be 8k or 16k long, for a 2764 and 28128 respectively. Eighter half of a 27256 is 16K long. When activeated, the status line will read;

```
MOVING EPROM TO MEMORY
```

and then issue the " - OK!" when finished. No verify is done when reading the EPROM, verifying the EPROM after a move is done using the VERIFY command.

5 - RETURN TO BASIC. When all EPROM activity is done, selecting #5 on the main menu will return you to Basic. The MPROM software is still intact and you can return to it with just a simple EXEC, providing that you have not changed the EXEC address. Also intact is the EPROM data. It is still in memory, and not changed. Now is the time to save EPROM data to disk. For a 2674, the proper format for saving data is;

```
SAVEM "FILENAME",&H3000,&H4FFF,&HA027 (ENTER)
```

&H6FFF

where FILENAME is the standard file name, extention and drive number and the next three values are start, end and exec addresses. The exec address has been chosen so that a warm start to basic will not crash into unwanted data. To save a

27128 or either half of a 27256 EPROM data, change the end address value from &H4FFF to &H6FFF.

8 grans !!

6 - EXAMIN/CHANGE MEMORY. This command is used when it is necessary to change some data in memory or to verify that the data you expect is there. When selected, the status line will ask for a start address. The address must be entered in hex values only. When finished, hit enter. The following line will show the address requested and the contents of that memory location. You now have three choices. 1) Use the up or down arrows to increase or decrease the address value by one. 2) Enter a 2-digit hex value, which will replace the old value automactically. 3) Or, hit the enter key to return to the main menu.

HINTS & TIPS

A: To make a copy of a DOS already in the DISTO controller, use these steps.

1 - Use the DOS command (or POKES) to switch to the desired DOS. Insure that the DOS's logo you want to copy is on the screen.

2 - Save the DOS on disk;

```
8K - SAVEM "FILENAME",&HC000,&HDFFF,&HA027
16k - SAVEM "FILENAME",&HC000,&HFEFF,&HA027
```

3 - With the MPROM adapter properly plugged in, Reload the DOS with an offset;

```
LOADM "FILENAME",&H7000
```

4 - LOAD and EXEC the MPROM software. The offset used in step #3 will automatically put that DOS at address 12288 ($3000).

5 - Select the 2764 EPROM for 8K DOS, or the 27128 EPROM for a 16K DOS.

6 - Insert the proper EPROM into the ZIF socket.

7 - Insure that the EPROM is blank by choosing #1 on the main menu.

8 - Program the EPROM by selecting #2 on the menu.

9 - Verify that the EPROM is properly programmed by selecting #3 on the main menu.

10 - Remove the EPROM from the ZIF. Turn the computer off and insert the EPROM into your DOSTO controller.

To make a copy of a DOS in RAM, loaded from disk, such as ADOS or SPECTRUM DOS, follow the instructions supplied with the DOS on how to prepare that DOS for EPROM. Then follow steps 2 to 10 above.

B: Some of the more experienced hardware buffs may want to fit a 2764 into a 24 pin socket. The following is a step by step procedure to do so. Soldering experience is required.

1 - Move the two jumpers on the solder side of the MPROM board over to cover the center pin and other pin. That will swap A11 and A12 for the 24 pin socket.

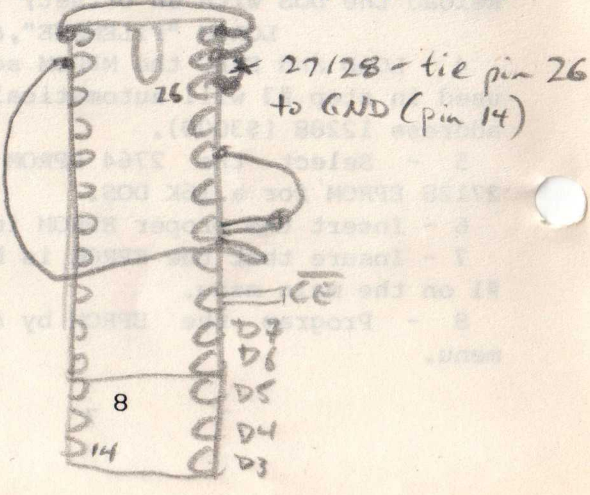
2 - Program the 2764 as usual.

3 - Bend up pin 20 on the EPROM. Solder a short piece of wire from this pin to pin 22. 27128 - bend up 26

4 - Solder pins 1, 28, 27 and 26 together.

5 - Solder a piece of wire to pin 2 of the EPROM. Insert the other side of this wire into the socket hole left by pin 20.

6 - Inset the 2764 into the 24 pin socket. Pin 3 of the 2764 must go into pin 1 of the socket, pin 4 of the 2764 into pin 2 of the socket, and so on.



Slot Switching Addendum

The Mprom Adapter and Real Time Clock software is made to run without the Multi-Pak Interface. If you are using a MPI and the add-on you are using is not in a Controller, there is still a way that you can make it work.

The controller's I/O area (drive selects) and the adapters I/O area both use the same area known as the SCS. In the MPI, this area is software switchable between the four slots. If your adapter is not in the same slot as the controller, i.e. Ramdisk or MEB adapter, you must first switch access, from the controller slot, to the slot that your adapter is in. To do this follow these simple instructions.

1 - Do all loading or saving on disk that you will need for that adapter.

2 - Type in one of the following lines according to the slot that the adapter is in;

```
POKE &HFF7F,&H30 ' FOR SLOT #1
POKE &HFF7F,&H31 ' FOR SLOT #2
POKE &HFF7F,&H32 ' FOR SLOT #3
```

Note that the controller must be in slot #4.

3 - Now use the software related to that adapter. At this point, you cannot do any disk access.

4 - When you are finished, in order to regain access to the disk drive, type in the following command;

```
POKE &HFF7F,&H33 ' FOR SLOT #4
```

At this point, your disk will work normally and you can load or save again. If you need access to your adapter again, repeat steps 2 to 4. These commands can be done within a BASIC program if so desired.