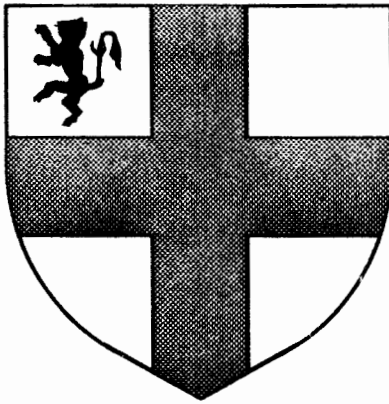

XT-ROM

**Hard Disk Boot ROM for the
CoCo XT and CoCo XT-RTC**



Copyright 1989 by Burke & Burke
P.O. Box 58342
Renton, WA 98058
All Rights Reserved

PUBLISHED BY BURKE & BURKE
P.O. BOX 1283
PALATINE, IL 60078-1283

COPYRIGHT NOTICE

All rights reserved. No part of this manual may be reproduced, copied, or transmitted in any form without prior written permission from Burke & Burke.

This entire manual, any accompanying hardware or computer programs, and any accompanying information storage media constitute a PRODUCT of Burke and Burke. The PRODUCT is supplied for the personal use of the purchaser. Burke & Burke expressly prohibits reproduction of this manual, the accompanying computer programs, and the printed circuit artwork.

Burke & Burke expressly requires, as a condition of providing this PRODUCT and the associated LIMITED WARRANTY to the PURCHASER, that one copy of the PRODUCT be purchased from Burke & Burke for every copy used.

Burke & Burke does not in any way transfer ownership of any computer programs to the PURCHASER. The PURCHASER is granted a limited license to use Burke & Burke computer programs distributed with the product, but only on a single computer.

LIMITED WARRANTY

Burke & Burke warrants the PRODUCT against defects in material or workmanship for a period of ninety (90) days from the date of purchase by the original owner. This warranty is limited to repair or replacement of PRODUCT which proves defective during this period, at the sole expense and discretion of Burke & Burke. This warranty specifically excludes software defects and defects caused by negligence, abuse, accident, or tampering.

DISCLAIMER

Every effort has been made to ensure the accuracy of this manual and the quality of the PRODUCT it describes. Burke & Burke makes no warranties, whether expressed, statutory, or implied, of any kind whatsoever, as to the merchantability of the PRODUCT or its fitness for a particular use, except as set forth above as the LIMITED WARRANTY.

Burke & Burke does not assume any liability for any damages, whether special, indirect, or consequential, resulting from the use of the PRODUCT. The PRODUCT is sold on an AS-IS basis.

TABLE OF CONTENTS

| | |
|-----|-----------------------|
| 1.0 | Introduction |
| 2.0 | Installation |
| 2.1 | Hardware Installation |
| 2.2 | Software Installation |
| A.0 | Manual Pages |
| A-1 | BootPort |
| A-2 | Patch |
| A-4 | TagTrack |
| A-6 | Zap |

1.0 Introduction

XT-ROM is an auto-boot EPROM designed specifically for use with Burke & Burke's CoCo XT and CoCo XT-RTC hard disk interfaces. Once installed, XT-ROM will automatically start up OS9 Level I or OS9 Level II from hard disk whenever you turn on your CoCo with the Multi-PAK set to slot #3.

XT-ROM is also capable of booting OS9 from a floppy disk. This is done automatically if XT-ROM determines that OS9 is not present on your hard disk. You can also force a floppy boot by holding down the SHIFT key during reset or power-up.

The automatic memory test performed by XT-ROM lets you know about memory problems before they have a chance to crash your CoCo. The memory test can be bypassed or aborted at any time by holding down the space bar on the CoCo keyboard.

Holding down SHIFT, the '/' key, and the spacebar during power-up will bypass XT-ROM completely, giving control to Disk BASIC. Of course, you can always get to Disk BASIC by setting the Multi-PAK to slot #4 and depressing the reset button.

There is an entertaining little demonstration program, also accessible from the keyboard, hidden deep within XT-ROM. In your spare moments, you may wish to try and find it.

- . -

XT-ROM is 100% compatible with OS9, RS-DOS, other DOS programs, and both Tandy and 3rd-party floppy disk controllers.

This manual will tell you how to install and use XT-ROM with the Burke & Burke CoCo XT and CoCo XT-RTC hard disk interfaces.

To use XT-ROM you will need:

- Tandy Color Computer, Color Computer 2, or Color Computer 3
- Tandy Multi-PAK interface
- CoCo XT or CoCo XT-RTC hard disk interface
- Hard disk controller (best with socketed BIOS EROM)

2.0 Installation

There are 2 main steps to installing XT-ROM:

- 1) Hardware Installation
- 2) Software Installation

These steps are outlined below.

2.1 Hardware Installation

XT-ROM is a 28 pin EPROM that installs in the hard disk controller's "BIOS ROM" socket (see Figure 1). If your hard disk controller already has a 28 pin IC socket for the BIOS ROM, you can install the XT-ROM hardware as follows:

- 1) Turn off your entire computer system.
- 2) Remove your CoCo XT or CoCo XT-RTC from the Multi-PAK and remove the interface's housing top cover.
- 3) Remove the existing BIOS EPROM or ROM (from the socket on the hard disk controller), and replace it with XT-ROM.
- 4) Several small plastic "jumpers" are supplied with XT-ROM. Use the instructions below to install the jumpers in jumper area S1 on the hard disk controller board.
 - 1) Remove any jumpers from positions 8 and 1-4 of S1.
 - 11) Install jumpers across positions 7, 6, and 5 of S1 according to the table below (In = jumper installed).

| # Heads on Drive | S1-7 | S1-6 | S1-5 |
|---------------------|------|------|------|
| 1 | In | In | In |
| 2 | In | In | Out |
| 3 | In | Out | In |
| 4 | In | Out | Out |
| 6 | Out | In | In |
| 8 | Out | In | Out |
| 10 | Out | Out | In |
| 16 | Out | Out | Out |

Note that some hard disk controllers do not provide jumper area S1. For these controllers you will need to "hard code" the number of heads on your drive into the BOOT module by making a simple patch (see software installation)

- 5) Replace the housing top cover, and re-install the CoCo XT in Multi-PAK slot 3.

- 6) Slide the Multi-PAK's cartridge selector switch to the position marked "3".

The XT-ROM hardware is now installed in your computer. As long as you leave the Multi-PAK selector in position 3, XT-ROM goes into action automatically whenever you turn on your CoCo.

You can disable XT-ROM at any time, simply by sliding the Multi-PAK selector to position 4. This will cause your CoCo to power up in Disk BASIC as it did before installing XT-ROM.

Some hard disk controllers do not provide a socket for the BIOS ROM. On these controllers, the 28 pin ROM is soldered directly to the hard disk controller circuit board. If you have this type of controller, you have 2 options for installing XT-ROM:

- 1) Carefully desolder the BIOS ROM from the hard disk controller board, and replace it with a 28 pin IC socket. This is a superior solution, but Burke & Burke recommends that you do not attempt the procedure yourself unless you are skilled in electronic repair procedures.
- 2) "Piggyback" a low-profile IC socket onto the existing BIOS ROM, as shown in Figure 2 and Figure 3. This requires that you clip one pin off of the BIOS ROM (using diagonal cutters), which neutralizes the ROM *PERMANENTLY*. This procedure also requires some skill at electronic assembly, but is much simpler and less error-prone than desoldering the BIOS ROM.

NOTE: SOME BIOS ROMS REQUIRE A PULL-UP RESISTOR BETWEEN PINS 20 AND 28 OF THE BIOS ROM WHEN INSTALLING A PIGGY-BACK XT-ROM. IF YOUR CONTROLLER DOES NOT SEEM TO RECOGNISE XT-ROM, YOU SHOULD INSTALL THE 4.7K RESISTOR SUPPLIED WITH YOUR XT-ROM. THE RESISTOR CONNECTS TO PIN 20 OF THE BIOS ROM, BUT NOT TO PIN 20 OF THE XT-ROM SOCKET.

Either option provides a BIOS socket on the hard disk controller, allowing you to install XT-ROM.

2.2 Software Installation

XT-ROM allows you to transfer your existing OS9 boot floppy disk directly to hard disk. This is a 2 step procedure:

- 1) Modify the BOOT module to access the hard disk rather than the floppy disk.
- 2) Save the modified kernel & BOOT module to hard disk track #128, and copy the OS9Boot file from floppy to hard disk.

NOTE: NO OTHER TASKS MAY BE RUNNING (IN THE BACKGROUND OR IN OTHER WINDOWS) DURING THE SOFTWARE INSTALLATION PROCEDURE. THE XT-ROM UTILTY COMMANDS MUST HAVE EXCLUSIVE ACCESS TO YOUR HARD DISK DURING SOFTWARE INSTALLATION.

BootPort

Syntax: bootport [opts] source_path dest_path

Function: Copies OS9Boot file and OS9 kernel between hard disks and floppy disks.

Parameters:

| | |
|-------------|---|
| opts | Any combination of the following options may be used to modify the operation of BootPort: |
| -? | Display options summary message. |
| source_path | The name of the device that contains the desired OS9Boot file and OS9 kernel. |
| dest_path | The name of the device to which you are copying the boot information. |

Notes:

- 0 BootPort first verifies that the boot track is not being used by other files. If the boot track is in use, Bootport responds:

WARNING -- Kernel or other data present on track n.
Do you wish to continue without updating the kernel [Y/N]?

If you respond with "Y", BootPort will not update the kernel image stored on the boot track, but will still copy the OS9Boot file from the source device to the destination device. If you respond with "N", BootPort will exit immediately.
- 0 The boot track for floppy disks is track #34. The boot track for hard disks is track #128.

Patch

Syntax: `patch module <path_name`

Function: Overlay a memory-resident module.

Parameters:

| | |
|-----------|--|
| opts | Any combination of the following options may be used to modify the operation of BootPort: -? Display options summary message. |
| module | The name of a memory-resident module that you wish to overlay. |
| path_name | The name of an overlay file. |

Notes:

- Patch copies information from an "overlay file" into the specified module, then recalculates the module's header parity and CRC.
- The patch file contains binary data (it may be produced directly by an assembler) in the following format:

```
<patch record>
.
.
<patch record>
<eof record>
```

The format of the binary data in the records is as follows:

```
<patch record>      $00 <16 bit patch data byte count>
                   <16 bit module offset> <patch data bytes>

<eof record>       $FF $0000 $0000
```

- One use of patch is to allow machine language overlays to be loaded directly into a module, without having to convert the patch to a sequence of DEBUG or MODPATCH commands. For Example:

```
      OPT      M
      FCB      $00                ;patch record preamble
      FDB      Y-X
      FDB      P.OFFSET

X     equ     *
*
Y     equ     *                ;put new code (patch) in here
```



```
FCB      SFF,$00,$00,$00,$00 ;eof record
```

```
END
```

If this file is assembled, the resulting binary file will be suitable for use as a patch file.

- 0 The format of a patch file is identical to the format of an RS-DOS ".BIN" file.

TagTrack

Syntax: tagtrack [opts] source_path track_number

Function: Marks clusters on a particular track as "in-use".

Parameters:

| | |
|--------------|--|
| opts | Any combination of the following options may be used to modify the operation of BootPort: |
| -? | Display options summary message. |
| -e | Mark entire track as in use. If this option is not specified, only those clusters that correspond to the 1st 18 sectors of the track are marked. |
| source_path | The name of the RBF device that contains the track that you wish to "tag". |
| track_number | The (decimal) track number to be "tagged". |

Notes:

- O TagTrack creates a file called "tag000" in the root directory of the source device. The segment list for this file contains exactly one entry, which includes all "tagged" clusters on the specified track.
- O TagTrack will display a message indicating that a particular cluster is allocated to another file, but that cluster will still be allocated to "tag000". The result is a duplicate cluster assignment, which will be detected by the DCHECK utility. This allows TagTrack to be used to detect files that are resident on a particular track, as follows:

```
OS9:tagtrack /d0 34          (look for files on track 34)
.
.
OS9:dcheck -bp /d0
.
.
OS9:zap /d0/tag000          (undo the "tag" -- see ZAP info)
OS9:
```

The output of the DCHECK command will include the names of all files that have clusters in common with "tag000".

- O TagTrack can be used to delete the OS9 kernel from a hard disk or floppy disk by the sequence:

```
OS9:tagtrack /d0 34          (assumes floppy disk)
.
.
```

Zap

Syntax: **zap** [opts] path_name [path_name . . .]

Function: Removes directory entry and deallocates file descriptor sector for each specified file.

Parameters:

| | |
|------------------|---|
| opts | Any combination of the following options may be used to modify the operation of BootPort: -? Display options summary message. |
| path_name | The name of the file that you wish to "zap". Any number of pathnames may be specified. |

Notes:

- 0 Zap does not deallocate the sectors that were specified in the segment list of the file. It simply releases the directory entry and the file descriptor sector.
- 0 One use of Zap is to mark bad sectors in the allocation bit map, so that OS9 will not attempt to use those sectors. This is done by creating a file that includes all of the bad sectors, and then Zap-ing the file.
- 0 Another use of Zap is to deny OS9 access to groups of sectors that are used by other systems (e.g. the boot track). In this context, Zap and TagTrack (see TagTrack) perform opposite functions.

OS9:del /d0/tag000

- 0 Note that since TagTrack may create duplicate cluster assignments, it corrupts the file system in a predictable manner. The effect of TagTrack can be reversed by using the command:

OS9:zap /d0/tag000

The XT-ROM release disk includes a number of utility commands that you should copy to your CMDS directory before starting to install the XT-ROM software. These commands are:

- BootPort -- Copies OS9Boot and kernel between hard disks and floppy disks.
- TagTrack -- Use in conjunction with DCHECK -BP to locate files stored on a given track of a given device.
- Zap -- Delete a file's directory entry without returning it's sectors to OS9.
- Patch -- Allows you to patch memory-resident OS9 modules using the output of an assembler.
- XTR -- A SHELL script file that performs XT-ROM software installation on a newly formatted hard drive, assuming a newly formatted floppy disk is in drive /dl and the desired version of Level II OS9 is running (included for reference).
- L2B.PAT -- a Patch file that installs a hard disk BOOT module in Level II systems.
- L1B.PAT -- a Patch file that installs a hard disk BOOT module in Level I, Version 2 systems.
- L1XB.PAT -- a Patch file that installs a hard disk BOOT module in Level I, Version 1 systems.

Each of the utility commands provides built-in help. If you are unsure of the options for a utility command, enter the command name followed by '-?' and the command will display its legal options. For example:

```
OS9:BootPort -?

Usage: BootPort [-<opts>] <source device> <dest device>
Options:
  -?   Display this message

OS9:
```

2.2.1 Modifying the BOOT module

- 1) Use the normal (floppy disk) boot procedure to boot up the version of OS9 that you want to transfer to the hard disk.

NOTE: THIS VERSION MUST INCLUDE HARD DISK DRIVERS AND DESCRIPTORS IF YOU WANT TO BE ABLE TO ACCESS THE HARD DRIVE FROM OS9!

- 2) Type the patch command that is right for your system:

```
OS9:patch BOOT </d0/cmds/l2b.pat (Level II systems)
```

```
OS9:patch BOOT </d0/cmds/l1b.pat (Level I, Version 2 systems)
```

```
OS9:patch BOOT </d0/cmds/l1xb.pat (Level I, Version 1 systems)
```

- 3) If your controller does not provide the S1 jumper bank (see hardware installation), you must "hard code" your drive configuration into the BOOT module. To do this, simply store your drive's number of heads (xx) at module offset \$140 in the BOOT module, and correct the module's CRC. One way to do this is:

```
OS9:modpatch
l BOOT
c 140 06 xx
v
```

```
OS9:
```

- 4) Save the modified version of OS9 to a newly formatted floppy disk. Assuming you have a newly formatted disk in drive /d1, use:

```
OS9:cobbler /d1
```
- 5) Set this disk (PATCHED BOOT) aside for use later in the installation procedure.

2.2.2 Saving the Kernel and OS9Boot to Hard Disk

If your hard drive is newly formatted, or you have not stored much information on it, you can use the following procedure to install the kernel and OS9Boot on hard disk:

- 1) With your system running, place the PATCHED BOOT disk in floppy drive /d1.
- 2) Enter the command,

```
OS9:bootport /d1 /h0
```

(use the name of your hard disk #0 in place of '/h0')
- 3) BootPort will display its status to you continuously as it copies the necessary boot information to your hard drive.

When BootPort exits successfully, the software installation is complete. Your system will now boot automatically from hard disk whenever you press RESET or power-up the CoCo.

If you have been using your hard disk for a while without XT-ROM, BootPort may generate the following error:

WARNING -- Kernel or other information already present on track 128. Do you wish to continue without updating kernel [Y/N] ?

At this point, you should enter "N" and depress the ENTER key. You will need to move all files off of track #128 (see below) before continuing. Once these files have been moved, run BootPort again to perform the final stages of software installation.

2.2.3 Moving Files off of Track 128

It is possible to move files that use track 128 to another part of the hard disk, using a fairly complicated process. This frees up track 128 for use by the OS9 kernel.

The process has 3 steps:

- 1) Find the files.
- 2) Move them to another location on the hard disk.
- 3) Repeat steps 1 & 2 until no files are found on track 128.

2.2.3.1 Finding Files Stored on Track 128

- 1) Use the TagTrack utility to "mark" all hard disk track 128 sectors that are needed by the kernel (note that output has been redirected to the printer):

```
OS9:tagtrack /h0 128 >/p
```

The printer output includes a series of cluster numbers that will need in order to locate files that are stored on track 128. TagTrack lists these clusters as either "grabbed" or "allocated to another file".

Note: The TagTrack utility temporarily corrupts your hard disk's file system by creating duplicate cluster allocations. All duplicate allocations are collected in a file called 'tag000'. These duplicate cluster allocations are detected by DCHECK. The effects of TagTrack are reversed by the Zap utility.

- 2) Use the standard OS9 DCHECK utility to locate files that use track 128 sectors (note that output has been redirected to the printer, and that this command takes about 10 minutes to run on a half-full 20 meg hard disk):

```
OS9:dcheck -bp /h0 >/p
OS9:
```

The '-bp' tells DCHECK to display the filenames for all files that contain "questionable" clusters. Since duplicate clusters are considered as "questionable", DCHECK will display the names of all files and directories that

use sectors on track 128 (to the printer).

- 3) Cross-reference the information on the TagTrack printout with the information on the DCHECK printout. Use the portion of the DCHECK output that begins:

Displaying pathnames for questionable clusters:

Make a note of any pathnames displayed by DCHECK that indicate clusters also found on the TagTrack output. One such pathname will always be '/h0/tag000'.

2.2.3.2 Relocating Files

- 1) For each DIRECTORY name revealed by the cross-reference:

- i) Create a new directory with the same parent directory, but with a different name.
- ii) DSAVE the entire old directory into the new directory.
- iii) Delete all files from the old directory.
- iv) Use RMDIR to delete the old directory itself.
- v) Rename the new directory to give it the same name as the old directory.

- 2) If any DIRECTORY names were revealed by the cross-reference, perform the following steps:

- 1) Enter the command:

OS9:zap /h0/tag000

This removes the duplicate allocations caused by TagTrack, and deletes the 'tag000' file. Be sure to substitute the name of your hard disk for '/h0'.

- ii) Repeat the entire sequence given in section 2.2.3.1, above, beginning with the TagTrack command.

- 3) For each FILE name revealed by the cross-reference:

- i) Copy the file to a new file in the same directory.
- ii) Delete the old file.
- iii) Rename the copy to have the same name as the original file.
- iv) Enter the commands:

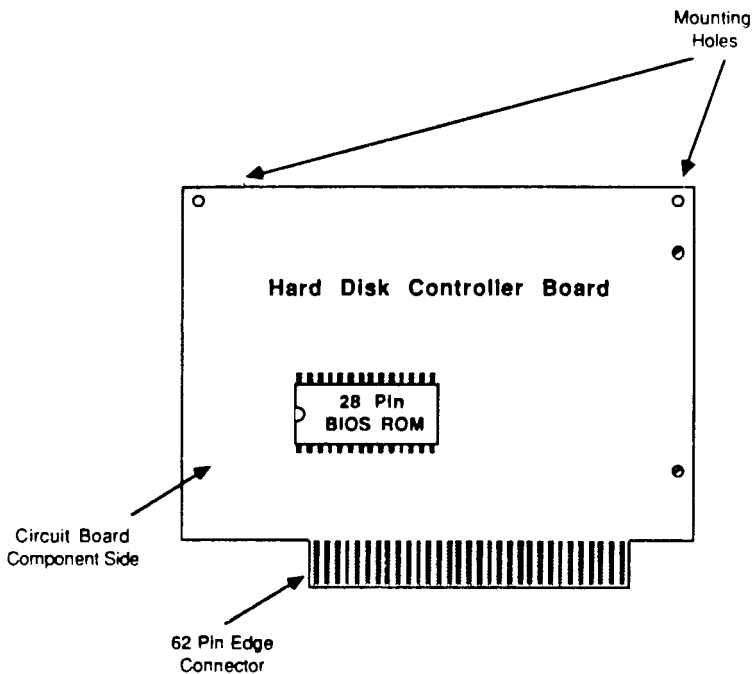

```
OS9:zap /h0/tag000;tagtrack /h0 128
```

(be sure to use the name of your hard disk in place of '/h0')

- 4) If any FILENAMES were revealed by the cross-reference, repeat the sequence from section 2.2.3.1, beginning with the DCHECK command.
- 5) When your cross-reference does not reveal any files or directories on track 128:
 - i) Enter the command:

```
OS9:del /h0/tag000
```
 - ii) Perform the BootPort procedure explained in section 2.2.2. If this is successful, you are done! If not, try again.

DEL is used instead of ZAP here, causing OS9 to free up the area of the disk where the kernel will be stored.



**Figure 1 Typical Hard Disk
Controller BIOS ROM Location**

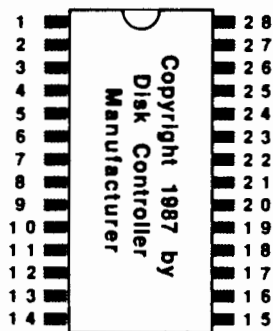


Figure 2 Detail View of BIOS ROM, Showing Pin Locations

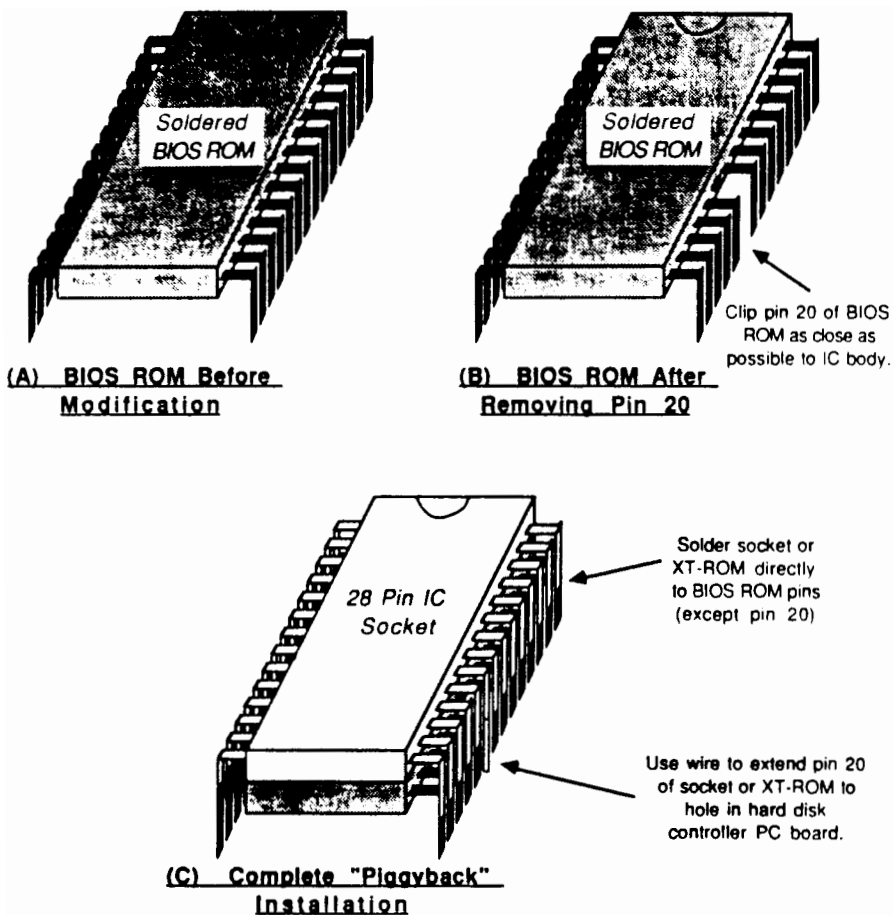


Figure 3 Optional "Piggyback" EPROM Socket Installation