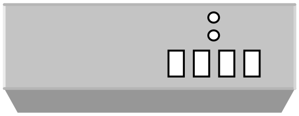
# 40 Embedded Systems

“Charm can be deceiving, and beauty fades away.”

Our useful programs can be "embedded" in chips. These chips can be used in creating different electronic devices. So programming for embedded systems is considered to be one of the interesting topics for the people who are from Electronics background.

## 40.1 PROM

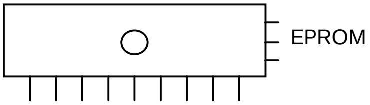
Our program can be embedded in PROM (Programmable ROM) category ROM. PROMs are usually available in sizes 1KB to about 2MB. It is identified by part number. Usually PROM's part number will be 27xxxx, where 27 denotes TL type PROM, xxxx denotes size in Kilo bits. For example, the widely used PROM got part number 27512, which indicates that the size is 512K bits, or 64KB. The blank PROM is the one, which is preloaded with 1's (not 0’s). The 1's of PROM corresponds to the fuses present in it. So if you burn the fuse, it represents 0. And so programming ROM is very often referred as *burning*. This burning is achieved with a hardware device known as Device Programmer. Device Programmer is also referred as PROM burner or PROM programmer. The term “Programmer” in “PROM programmer” refers to hardware device, not a person! PROM Programmer helps us to embed our program in the PROM chip. PROMs are OTP (One Time Programmable). Programmed or burned PROMS are widely used in electronic devices like billing machines, washing machines, cell phones, etc.



PROM burner or PROM programmer

## 40.2 EPROM

An Erasable PROM or EPROM is available with the same part numbering scheme of PROM. EPROM has a clear quartz crystal window for allowing UV rays to erase the contents. The UV rays erase the chip by a chemical reaction that melts the fuses back together. EPROM chips should be handled with care as sunlight has UV rays.



## 40.3 EEPROM

Electronically Erasable PROM or EEPROM is a kind of EPROM that doesn’t require UV rays for erasing the contents. Instead it uses electricity for erasing. Nowadays we have Flash

ROMs. Flash ROM is a type of EEPROM, which can be programmed without removing it from the system and without any special devices.

## 40.4 Programming for Embedded Systems

On our nornal PC, the *boot sector* contains code to load the rest of the OS using the BIOS which is in ROM and always available. In embedded systems, we burn the BIOS and OS in ROM in addition to our program. The reboot vector of embedded systems usually points to the BIOS initialization routines that are also embedded in the chip. Many embedded systems make use of an embedded operating system like QNX, VxWorks, Linux and rarely DOS. In this case our program gets advanced features such as device drivers and operating systems constructs built in to it and we don't have to write such code ourself.

In embedded systems, it is sometimes necessary not to have an Operating System or BIOS. In other words, we are in need of “stand alone” programs. “Stand alone” programs are the one, which don’t use Operating System or BIOS. For stand alone programs, we burn a chip with reboot vector pointing to the code we want to run instead of the BIOS initialization routines. If we want to write such a stand alone program in Turbo C, we have to modify the startup routines, because these routines are dependent on the DOS and they load libraries that are dependent on DOS. Thus making a C program embeddable (i.e., ROMable) in a standalone manner requires all dependencies on DOS be removed.

We cannot embed all our programs in the chip. Only ROMable codes can be embedded which means we need a specialized code. Certainly we cannot use relocatable (or ordinary EXE) codes. In embedded system programming, we use void for `main( )` as `void main( )`, because there is no place for the code to get returned.

## 40.5 Locate utility

Locate is the utility that written by **Mark R. Nelson** found on CD💿. It helps us to remove relocations in EXE files and does other functions that are necessary for a ROMable code. It helps us to produce ROMable code without any overhead.

## 40.6 ROMable Code

As I pointed out earlier, only ROMable code can be embedded. Since this topic will be useful and interesting only to the people from Electronics background, I don’t want to harp more in this topic. If you want to know more about this topic, please refer the programs found on CD💿.

## 40.7 Applications

Programming ROM has so many applications including the creation of chips used in washing machine, creation of chips used in cars for monitoring the performance etc.