

## PROM Application Note Pre-Initializing CF Series Flash Cards

This application note describes the technique to pre-initialize CF Series Flash cards.

### Manual Card Initialization

All Flash and EPROM cards for the PC-1270 must be initialized before the program can be run when the card is installed in the calculator. This involves turning the calculator ON, pushing ALL RESET, pushing [ENTER], turning the calculator OFF and then back on.

### Automatic Card Initialization

The PROM CF Series of cards have two advanced features which allows you to skip the manual initialization process.

When the card is programmed with the CF-256A Flash Card Programmer, the card is automatically initialized by the programmer so it will run in the calculator without having to perform the manual card initialization steps above. Because of the "Data-Saver" feature, the card will stay initialized for about 5 to 10 minutes after being programmed. As long as it is placed into a calculator within that time, the program can be run by just pressing a function key. (If the card is out of a calculator for more than 5 to 10 minutes, you will need to do the manual initialization or reprogram the card to pre-initialize it again.)

### Variable Initialization

The CF-256A Flash Card Programmer and its operation program FLASH.EXE have an advance feature which allows you to pre-initialize all the variables (A-Z and all arrays) to values. If your program has a "setup" (variable initialization) routine and you are programming multiple cards with the same "setup", this feature can save considerable time.

First, program a card and run the "setup" routine to configuring the card exactly the way you want it. Then, using the pre-initialization capabilities, upload the variable storage area of the card to your PC and store it in a file (called a "ram" file). Then, when programming additional cards, you specify the ram file to use to pre-initialize all the variables. Each card you make will be setup identically to the first one you used to make the ram file. You can make several ram files for the same program, each with a different "setup".

### Security Considerations

This pre-initialization process has been carefully designed with several features to maintain the high security in the CF Series of Flash Cards.

- ◆ In order to perform the pre-initialization process, you must have a disc file that contains the same program that is on the card whose variable-storage area you are attempting to upload. If you do not have such a file, you cannot upload the variable-storage area. (To determine whether the program file matches the program on the card, the size of the two programs is compared and must be the same. The programmer does not read the program on the card.)
- ◆ The uploaded variable-storage area is stored in an encrypted disc file.
- ◆ Introduction of this variable initialization feature has not breached the tight security designed into the CF-256A programmer. Although it can read the variables in a card (provided the program disc file matches the program on the card), the programmer cannot read the program itself.

### How to Pre-Initialize Variables

You must have Version 1.1 of FLASH.EXE and the CF-256A Programmer to perform this. (If you have an earlier version, contact P\*ROM for a free upgrade.)

The first step is to program a card with the desired program and run the "setup" routine to configure the variables.

Then put the card back in the programmer and execute FLASH with the following line:

**FLASH -R PGM.SBC PGM.RAM**

(PGM.SBC) is the name of the file that contains an exact copy of the program on the card, and (PGM.RAM) is the name of the file in which you want to store the variable-storage area (called a ram file). The file extension of the ram file defaults to ".ram" if not specified.

If the program file matches the program on the card, the variable-storage area will be uploaded and stored in the specified file.

Then to program additional cards with the exact same "setup", execute the line

**FLASH PGM.SBC PGM.RAM**

The card will be programmed with the program in the file (pgm.sbc), and the variables will be initialized with the data in the ram file (pgm.ram). Note that because the default program file extension is ".sbc" and the default ram file extension is ".ram", we could have entered the line

**FLASH PGM PGM**

We've used the same root file name for both the program and ram file, but we don't have to. We could make several ram files for one program file, each containing different "setup" configurations, e.g.,

```
FLASH -R PGM.SBC PGM.001
FLASH -R PGM.SBC PGM_2.002
.
.
.
```

If you don't specify a ram file name, the card will be programmed and the variables cleared (see Automatic Card Initialization above).

#### **Automatically Generating the RAM File name**

If you execute FLASH with the "-a" option, it will automatically look for and use a (".ram") file with the same root name as the specified program file.

Thus if you type the line

```
FLASH -A PGM
```

or

```
FLASH -A \PROGS\VER1\PGM.C12
```

it is the same as if you typed

```
FLASH -A PGM.SBC PGM.RAM
```

or

```
FLASH -A \PROGS\VER1\PGM.C12 \PROGS\VER1\PGM.RAM
```

Thus the program has taken the root name of the program file ("pgm" or "\progs\ver1\pgm") and added the default file extension ".ram" to generate the ram file name.

With the "-a" option, the ram file must have the same root name as the program file and the file extension ".ram". The program file can have any extension as shown in the above example. (If the program file extension is not specified, it defaults to ".sbc").

**For more information, contact**

**PROM Software, Inc. | [www.promsoft.com](http://www.promsoft.com) | 800 843-7766**