

# FAHRENHEIT 512A

## PROM CF Series Flash Card Programmer

The Fahrenheit 512A CF Series Flash Card Programmer is controlled by the PC program FLASH.EXE. This program resides on your PC and communicates with the programmer. To program a card, you simply place the card in the programmer and then type the command FLASH followed by the name of the file containing the compiled program, i.e., to program a card with a program in a file named "TEST.SBC", type the line

```
FLASH TEST.SBC
```

This assumes you have connected the F-512A programmer to COM1 on your PC. If you have connected it to another port, specify the port number on the command line, e.g., if you have connected the programmer to COM2, type the line

```
FLASH -2 TEST.SBC
```

See Connecting the F512A Programmer to Your PC below.

To obtain a help screen showing program options, type the program name with no arguments, i.e., just type "FLASH".

```
FLASH.EXE - Ver 2.0  
Copyright 1998 PROM Software, Inc. All Rights Reserved
```

```
Usage: flash [-options] (fname) [(ramfname) [varfname]]
```

```
-l for COM1: can be 1, 2, 3 or 4 (default is COM1).  
-#[sernum] to serialize cards being programmed.  
-$(string) to install 'string' in cards being programmed.  
-a to autoinitialize RAM with file named '(fname.ram)'.  
-b/y to suppress sound (errors & when card finished).  
-e to erase cards only (default is to erase & program).  
-i to identify the card in the programmer.  
-m to process multiple cards (default is a single card).  
-q for quiet mode. No sound or unnecessary messages on screen.  
-r to upload RAM initialization file (ramfname).  
-t show command line options in environment variable 'FLASHSW'.  
-u to suppress 'unprotected program' warning message.  
-v to show programmer model and firmware version.  
-x to show progress on screen in hex (default is decimal).
```

```
Push 'y', space-bar or [Enter] to continue at "Next?" prompt.  
Options can be set in environment variable 'FLASHSW'.
```

```
PROM Software Inc Burlington VT 05406-4027 USA 800 843-7766
```

## Description of Options and Arguments:

- (*fname*) is the name of the file containing the compiled program. If no file extension is specified, the extension ".SBC" is used. (*fname*) is required to program a card or to use the RAM initialization feature.
- (*ramfname*) is the name of the RAM initialization file. If no file extension is specified, the extension ".ram" is used. (*ramfname*) is optional. If specified, the program will look for the ram file and report an error if not found.
- (*varfname*) is the name to write the values of all variables in the RAM initialization file. (*varfname*) is optional. See Viewing the Variables in RAM.

Options must be preceded with a single "-" or "/" and can appear in one or more "words" with no embedded spaces. (If you wish to use embedded spaces with the Serialization or String Substitution options, enclose the string in quotes -- See Serializing Flash Cards and String Substitution.)

There must be a space between the "FLASH" program name and the first option, a space between each option and the next, a space between the last option and (*fname*), a space between (*fname*) and (*ramfname*) if *ramfname* is specified, and a space between (*ramfname*) and (*varfname*) if *varfname* is specified. For example to use COM4 and make multiple copies of a program in the file PROGABC.SBC, start FLASH with either of the following lines:

```
FLASH -4M PROGABC or  
FLASH -4 -M PROGABC
```

Note that we did not need to specify the extension because it is ".SBC". Options are not case sensitive, e.g., the letters "M" and "m" are considered identical and use of either will select the multiple copies option.

Options can appear in any order.

- #[*ym[d]*][*serialnum*]  
inserts an incrementing serial number (with an optional date code) starting with [*serialnum*] in each Flash card programmed. The serial number search string must be installed in the program to use this option -- See Serializing Flash Cards.
- \${*ym[d]*}(string)  
inserts 'string' (with an optional date code) in each Flash card programmed. The 'string' search string must be installed in the program -- See String Substitution.
- 1  
is the number of the COM port to which the programmer is connected. It defaults to "1" for COM1, and can be 2, 3, or 4 for COM2, COM3 and COM4 respectively.

- a initializes the variables on the card being programmed using a (ramfname) the same as the program file name but with the extension ".ram". (ramfname) must exist to use this option. See Automatic Variable Initialization below.
- b suppresses all sounds (the two beeps sounded when a card is programmed and the four beeps when an error occurs). (See also '-y' below.)
- e erases but does not program a card. (fname) is not required for this option and any filename passed is ignored.
- i identifies the card in the programmer without erasing or programming it. (fname) is not required or used for this option.
- m is for programming multiple cards. The program will display "Next?" on the screen after each card is programmed. Push the space bar, "Y", or [ENTER] key to continue after you have installed the next card in the programmer. Push any other key to stop.
- q is for quiet mode. No unnecessary messages appear on the screen (except for fatal error messages) and no beeps are sounded. The -m option has no effect if used with this option.
- r makes a RAM initialization file from a Flash card. If you have also specified a Variable file name (varfname), the uploaded RAM file will be decoded and the value of all variables written to (varfname). See Making a Variable Info File.
- t shows the program options set in the FLASHSW environment variable. See Setting the FLASHSW Environment Variable below.
- u suppresses the "unprotected program" message that is issued if the program being downloaded is not password-protected. This option should only be used if you intentionally want to make unprotected cards.
- v shows the programmer model and firmware version number.
- x shows programming progress in hexadecimal on the screen (default is to show progress in decimal).
- y suppresses error beeps only (two beeps sounded when a card is programmed successfully). (See also '-b' above.)

## **Installing FLASH.EXE on your PC**

The program to control the F-512A programmer is named FLASH.EXE on the distribution disk. The program can be used right from the distribution disk, however, we recommend that you copy it to your hard drive.

If you already have the SBC development system and it is installed in the default directory "SBC" on your C: drive, we suggest you copy FLASH.EXE to the "C:\SBC" directory by typing the command:

```
COPY A:FLASH.EXE C:\SBC
```

If the "C:\SBC" directory does not exist, make it by typing the following command:

```
MD C:\SBC
```

and then copy the program to it as shown above. You will also need to add "C:\SBC" to your path statement in your AUTOEXEC.BAT file (located in the root directory of C:). If the path statement already exists, add

```
;C:\SBC
```

to the end of it. Otherwise, add the following statement to your AUTOEXEC.BAT file:

```
SET PATH=C:\SBC
```

Make sure to restart your computer after making the changes to your AUTOEXEC.BAT file so they will take effect. When starting your PC, make sure to turn the F-512A programmer on after you turn your PC on. If the F-512A was already on, turn it off and back on again before you run FLASH.

## **Connecting the F512A Programmer to your PC**

The programmer requires two connections to operate.

The first is the power supply which should be plugged into to any 115vac outlet and the other end connected to the programmer. Make sure to use the power supply furnished with the programmer to assure that the programmer is provided with the correct voltage and current supply. Failure to use the correct power supply can result in damaged or improperly programmed cards.

The second is the data cable. On one end there is a 9-pin serial connector which should be connected to your PC. The other end (with the telephone type connector) should be connected to the programmer.

If your PC does not have a free 9-pin serial connector but does have a free 25-pin serial connector, you will need an adapter to make the connection. (Adapters are free from PROM and can be purchased at any computer store for a few dollars.)

You will need to know the number of the serial port that you connecting the programmer to so you can tell the FLASH program. Serial ports are numbered 1 through 4 (although most systems only have 2). If you can't determine the number, it is easy to try different numbers with the FLASH program.

To do this, turn the programmer on and type the line

```
FLASH -v
```

If the programmer is connected to COM1, you should see the Yellow LED on the programmer come on for a short time and you will get the following message on your screen:

```
F512A Ver: x.x
```

If you get an error message instead, it means FLASH can't find the programmer on COM1, so try the next serial port (COM2) by typing the line

```
FLASH -2v
```

If you still get an error message, try the next 2 ports (3 and 4) with the same command. When you get the right one, the version of the programmer firmware will appear on the screen (see above).

You need to specify the serial port number if it is 2, 3 or 4 on the command line whenever you run FLASH. For example, if you have identified 2 as the correct number, you would program cards by typing "FLASH -2" followed by the program name.. (See Setting the FLASHSW Environment Variable to make the change permanently on your PC.)

## **How to Program a Card**

Type the following command on your PC but don't press the [ENTER] key yet.

```
FLASH (filename)
```

where (filename) is the name of the file containing the compiled program. (Don't forget to specify the filename extension if it is not ".SBC".) If you have connected the programmer to a serial port other than 1, you need to specify the serial port number, e.g., for serial port 2, type the following:

```
FLASH -2 (filename)
```

Insert the card into the programmer and hold it in place while you press the PC [ENTER] key with your other hand. When the card is done (you will hear 2 beeps), remove it. If you are programming multiple cards (see the -m option), put the next card in the programmer and hold it in place, then press the space-bar to program it. (You can also press the "Y" or [Enter] keys.) Do not insert or remove a card while it is being programmed or erased (when either of the red LED's are on).

Programs must be compiled for the type of card on which they will be programmed. Refer to the Application Note Programming CF Series Flash Cards and the SBC Compiler Reference Manual.

### **Resetting the Programmer**

The programmer can be reset by turning it off and back on. When you start your PC, you should turn the PC on first and then the programmer. If you turn them both on at the same time (or turn the programmer on first), you will probably need to reset the programmer before it will work properly. (The reason is that the PC sends one or two spurious characters out the serial port when it is turned on.)

### **Setting the FLASHSW Environment Variable**

Instead of specifying options on the command line each time you run FLASH, you can set an environment variable named FLASHSW equal to the option string you wish to have in effect each time you start FLASH. If you also specify options on the command line, they are passed to the program as well as the ones in the FLASHSW environment variable.

For example, to specify that you wish to use COM3, process multiple cards, and show the programming process in hex, add the following to your AUTOEXEC.BAT file:

```
SET FLASHSW=-3mx          or  
SET FLASHSW=-3 -m -x
```

You will have to restart your computer in order for the change to your AUTOEXEC.BAT file to become effective. Each time you execute the FLASH.EXE program, it will run as if you started it with the "3", "m" and "x" options, i.e., typed "FLASH -3mx (fname)".

(To see what options are currently assigned to the FLASHSW environment variable, run the program with the "-t" option, e.g., type "FLASH -t". The options that have been set by the FLASHSW environment variable will appear on the screen.)

If you use the Serialization or String Substitution options and wish to set the option in the FLASHSW environment variable, make sure to enclose the replacement string in quotes if it has any embedded blanks, e.g.,

```
SET FLASHSW=-m -$"ym LOC#124"
```

will set the String Substitution string to "94NOV LOC#124" for all cards programmed (assuming a current date of November, 1994).

## Serializing Flash Cards

FLASH.EXE can install an incrementing serial number in each Flash card as the card is programmed. The serial number string is programmed into the Flash memory (not the RAM memory) and is therefore permanent and cannot be changed by the user. For each card programmed, the serial number is incremented by 1.

The beginning serial number can be specified on the command line (default is 1) and any series of characters and/or the current date can be prefixed to the incrementing serial number.

In order to use the Serialization feature, bank 0 of the PC-1270 program must be compiled with the serial search string "#SERIAL-+--+--+--". This string must be exactly 16 characters in length and must contain exactly the characters shown. Typically, it would be in a print statement that is used to show the serial number to the user, e.g.,

```
PRINT "#SERIAL-+--+--+--"
```

As a Flash card is programmed, the serial search string is replaced with the actual serial number string. If the actual serial number string is less than 16 characters long, it is padded on the right with spaces.

To program cards with serial numbers starting at 1, issue the command

```
FLASH -# MYPROG.SBC
```

Note that the "-#" option must appear as a separate option on the command line and cannot be combined with other options in the same string. For example, if you are using the COM2 option, want to make multiple copies, and to serialize the copies, issue the command:

```
FLASH -# -3m MYPROG.SBC
```

FLASH.EXE looks for the serial search string in bank 0 and substitutes a "1" in the first card programmed, a "2" in the second card, and so forth. Thus in the cards, the print statements would be changed to

```
PRINT "1           "  
PRINT "2           "  
...
```

The serial number sequence can be started at any number by specifying it on the command line. You can also specify leading letters or words as well as the current date (see below) to be installed with the serial number. For example, to create the serial number series 88-FR101, 88-FR102, 88-FR103, ..., specify the command line as follows:

```
FLASH -#88-FR101 MYPROG.SBC
```

If the serial number string has any embedded blanks, you must enclose the entire string (but not the option switch "-#") in quotes, e.g.,

```
FLASH -#"45 RED 100" MYPROG.SBC
```

which would install the following sequence of serial numbers in cards programmed:

```
45 RED 100  
45 RED 101  
45 RED 102  
...
```

## **String Substitution**

FLASH.EXE can install a string specified on the command line within bank 0 of a program. The string can be prefixed with the current date in either of two formats (See Current Date Option).

The string substitution option can be used to install the customer's name or other information in the program at the time the Flash card is programmed. Because the string is programmed into Flash memory, it is permanent and cannot be changed by the user.

In order to use the string substitution option with a program, the search string "\$STRING-+-+-+--" must appear within bank 0 of the compiled program. This search string must be exactly 16 characters long and contain exactly the characters shown. Typically, it would be installed in the program in a print statement, e.g.,

```
PRINT "$STRING-+-+-+--"
```

The string to be substituted is then specified on the command line when executing FLASH.EXE as follows:

```
FLASH -$"ABC COMPANY" MYPROG.SBC
```

As each card is programmed, the search string "\$STRING-+-+-+--" will be replaced with "ABC COMPANY".

If the search string is found within bank 0 of the compiled program but no string substitution option is specified on the FLASH.EXE command line, blanks are substituted for the string search string "\$STRING-+-+-+--".

The following lines in your PC-1270 program will only display the string if the first character is not a blank:

```
S$(0) = "$STRING-+-+-+--" :  
IF ASC S$(0) > 32 PRINT S$(0)
```

## Serialization versus String Substitution

The string substitution option installs the same string in every card programmed, whereas the serialization option installs a unique serial number in each card.

Thus, if you want to put the same information into every card at the time the card is programmed (such as the customer's name), use string substitution. If you want every card to have a unique serial number, use the serial number method.

You can use both methods but make sure to specify each as a separate option on the command line. For example, suppose we want to serialize cards starting with the number WE-34B100, prefix the year and month to the serial number, and install the customer's name "WEST ELECTRIC" in each card. The RAM initialization file is MYPROG1.RAM. The programmer is connected to COM3 and we are making multiple cards. The command line would appear as:

```
FLASH -#"ym WE-34B100" -$"WEST ELECTRIC" -3m MYPROG.SBC MYPROG1.RAM
```

Note that we have to use quotes for the serial number string and substitution string because both have embedded blanks.

The serial number sequence will be (assuming a current date of November 11, 1994):

```
94NOV WE-34B100
94NOV WE-34B101
94NOV WE-34B102
...
```

## Current Date Option

The current date (date from the PC system clock) can be prefixed to the installed serial number and/or substitution string in either of two formats: *ym* or *ymd*, e.g., 94AUG or 94AUG12.

To include the date format in the substituted serial number, include "ym" or "ymd" (if you want the day number) at the head of the serial number string specified on the command line when executing FLASH.EXE.

For example, to include the year, month and day in the serial number sequence T4R-154, T4R-155, T4R-156, ..., specifying the following:

```
FLASH -#"ymd T4R-154" MYPROG.SBC
```

We need to enclose the entire string in quotes because of the embedded space between 'd' and 'T'. The sequence of serial numbers installed (based on a current date of December 12, 1994) will be:

```
94DEC12 T4R-154
94DEC12 T4R-155
94DEC12 T4R-156
```

The "ymd" or "ym" option must be in lowercase letters and must appear at the head of the serial number string.

Similarly, to prefix the current date to the substituted string, use the same specifiers at the head of the substitution string specified on the command line, e.g., to prefix the current date to the substituted string "TEST VER", issue the command

```
FLASH -$"ymd TEST VER"
```

This will cause the following string to be inserted in the Flash card as it is programmed:

```
94DEC12 TEST VER
```

When using the Current Date Option with either serial number or string substitution, make sure that the overall length of the serial number or string is not more than 16 characters. (An error is reported if this length is exceeded.)

### **Serialization & String Substitution Rules**

Any lower-case letters specified on the command-line will be shifted to upper case. (The PC-1270 can not display lower-case letters.) Note however, that the current date option must be specified in lower case letters, e.g., "ym" or "ymd".

The serial number search string "#SERIAL-+-+-----" must appear in bank 0 of your program (other banks are not searched) if you wish to install serial numbers.

The string substitution search string "\$STRING-+-+-----" must appear in bank 0 of your program (other banks are not searched) if you wish to install a substitute string.

If the serial number search string is found in the program but the serial number option "-#" is not specified on the command line when executing FLASH.EXE, a single "#" will be substituted for the search string "#SERIAL-+-+-----".

If the string search string is found in the program but the string substitution option "-\$" is not specified on the command line when executing FLASH.EXE, blanks will be substituted for the search string "\$STRING-+-+-----".

If the serial number option is specified on the command line but no serial number search string is found within bank 0 of the program, an error is reported.

If the string substitution option is specified on the command line but no string search string is found within bank 0 of the program, an error is reported.

The total length of the serial number (including the optional current date) cannot exceed 16 characters. If the length is close to or exactly 16 characters with the starting serial number, the length may grow to over 16 as the serial number is incremented (an error is reported if this occurs).

The last digits of the serial number string specified on the command line are assumed to be the serial number and are incremented by 1 for each card programmed. If the string contains more than one sequence of numbers, only the last sequence is incremented.

It is recommended that you print the serial number directly as a constant within your program, e.g.,

```
PRINT "#SERIAL-+++++--"
```

If you use a RAM initialization file and assign the serial number to a variable, make sure to assign the card's serial number to the variable before displaying the serial number (failure to do this will result in the serial number used to make the RAM initialization file being displayed).

### **Automatic Card Initialization**

When the card is programmed with the F-512A Flash Card Programmer, the card is automatically initialized by the programmer so it will run in the calculator without having to push [ALL RESET] on the back of the PC-1270. The card must be installed in a PC-1270 calculator within 5 to 10 minutes after being programmed to retain the initialization. (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 initialize it again.)

### **Automatic Variable Initialization**

Variables can be automatically initialized by the F-512A programmer as it is programmed. This will not only establish all the arrays in memory, but also put the desired data into them. The process adds less than 2 seconds to the programming time.

(Pushing [ALL RESET] on the PC-1270 clears all arrays and A-Z variables to 0. Thus, after an [ALL RESET], the program must DIMENSION the arrays and transfer any required data to them before the program can be run. In some programs, this initialization process can take a minute or longer, depending upon the number of arrays and amount of data to be loaded into them.)

### **How to Make a RAM Initialization File**

First, program one card, put it in a PC-1270, and run the initialization routine to configure the card exactly the way you want it. This process loads all the arrays and data into the RAM memory area of the Flash card.

Next, put the card back into the programmer and run FLASH with the "-r" option to make a RAM initialization file on your PC. In addition to the "-r" option, you need to specify both the program file name and RAM file name (see below).

Once you have made a RAM initialization file, you can program cards with all variables (A-Z and arrays) initialized. Specify the name of the RAM initialization file on the command line after the program file name (see below). Each card you make will be configured identically to the first one you used to make the RAM file. You can make several RAM files for the same program, each containing a different configuration.

Use of this feature is optional. If you don't specify a RAM file when you program a card, the card will be initialized (see Automatic Card Initialization above) and all variables will be set to 0. To use the Automatic Variable Initialization feature, you must have Version 1.4 (or later) of FLASH.EXE and the programmer. (If you have an earlier version, contact PROM for a free upgrade.)

### **Example**

These are the steps that need to be done to make and use a RAM initialization file for the program file PGM.SBC.

The first step is to program a card with the program "PGM.SBC". To program the card, type the line

```
FLASH PGM.SBC
```

Put the card in a PC-1270 and run the program to configure it the way you want it to be captured in the RAM initialization file you're making. 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 was used to program the card in the first step above, and PGM.RAM is the name of the RAM initialization file you are making. (The file extension of the RAM file defaults to ".RAM" if not specified.)

The program file must be the same one used to program the Flash card. This security feature prevents anyone who does not have a copy of the original program file from making a RAM initialization file. If the program file matches the program on the card, the arrays and variables will be uploaded and stored in the file "PGM.RAM" on your PC. If the program file does not match exactly, FLASH will report an error and stop. (If you are using the SBC Compiler, make sure you have set the #array\_size directive to the proper value (if it is not set to the proper value, you will get this error.) See the Application Note Programming CF Series Flash Cards and the SBC Reference Manual.)

Then to program cards with the identical RAM configuration, 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 above line as

```
FLASH PGM PGM
```

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

```
FLASH -r PGM.SBC PGM.001
FLASH -r PGM.SBC PGM.002
.
.
.
```

### **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", and it must be located in the same directory as the program file. The program file can have any extension (if not specified, it defaults to ".SBC").

## **Security Considerations**

The automatic variable initialization feature has been carefully designed to maintain the high security in the CF Series of Flash Cards.

In order to make a RAM initialization file, you must have the original program file (or a copy of it) that was used to program the Flash card. If you do not, you cannot make a RAM initialization file.

In determining whether the program on the card matches the disc file, FLASH does not (and cannot) read the program on the card, but compares the size of the program and variable areas. If both sizes match, FLASH will make a RAM initialization file. (The chance that a wrong program file would have the same size program and variable areas as the one on the card is 1 in 60,185,664.)

## **Making a Variable Info File**

If you specify a Variable Info filename (varfname) on the command line as the third argument, FLASH will examine the specified RAM initialization file and print the values of the A-Z variables plus all DIMensioned arrays. To use this feature, you must specify three file names on the command line: the original program file, the RAM initialization file, and the name of the file you want the variable information written to.

(You can use the MS-DOS device "CON" to have the variable information written to the screen, however, if you have used any arrays, there will be more than one screen of data. It is best to write the information to a file and then examine the file with a text editor.)

To make a Variable Info file named "RED1.VAR" from the RAM initialization file named "RED1.RAM" whose original program is in a file named "RED1.SBC", type the following line:

```
FLASH RED1.SBC RED1.RAM RED1.VAR
```

When FLASH detects the third file name (the Variable Info file) on the command line, it does not program the card, but analyzes the RAM initialization file and stores the results in the Variable Info file RED1.VAR. Note that in the above example, we could have typed the line

```
FLASH RED1 RED1 RED1
```

because the program and RAM file have the default file extensions of ".SBC" and ".RAM". The Variable Info file is always created with the extension ".VAR" even if you specify a different one.

If you use the "-r" option to upload a RAM initialization file and specify a Variable Info file name, FLASH will upload the RAM data and create the Variable Info file automatically. The command line would be

```
FLASH -r RED1.SBC RED1.RAM RED1.VAR
```

and in this case the RAM area from the program in the Flash card would be uploaded to the newly created RAM initialization file "RED1.RAM", which would then be analyzed and the information written to the Variable Info file "RED1.VAR".

The tests that apply to making a RAM initialization file also apply to making a Variable Info file, i.e., you must have a copy of the original program in a disc file in order to create a Variable Info file.

If FLASH cannot open the Variable Info file, it will send the variable information to the screen. If the Variable Info file already exists, it will be overwritten with the new information.

### **Error & Warning Messages**

If an error occurs, the program will terminate, return a non-zero value to the calling process or DOS, send an error message to the screen ('stdout'), and terminate if any of the conditions described below are detected.

A warning message will not terminate the program, however, the message will be printed to the screen ('stdout') unless the quiet ("-Q") option or "-U" option are set.

Do not insert or remove a card while either of the red LED's are on. If you need to reset the programmer, turn it off and back on.

Can't open program file  
'filename'.

FLASH can't find or open the specified compiled program file. Check that you have correctly spelled the file name and that the correct path has been specified. This error can also occur if the file is locked by another process.

Can't open COMx.

FLASH can't open the specified COM port. Make sure that the port exists on your system. This error can also occur if more than one port on your system has been configured with the same COM number.

Can't erase the flash card.

The erase function failed. Try another flash card to see if the problem lies with the flash card.

Please reset the programmer.

The programmer should be turned on after you turn on your computer. If it is turned on first, you will likely get this error. Turn the programmer off and back on.

Can't reset the programmer.

This is the error message you get if the programmer isn't connected to the PC, is connected to the wrong COM port, or isn't turned on. The default COM port is 1; if you want to use a different COM port, you have to specify its number (2-4) on the command line. See Descriptions of Options and Arguments.

If the Yellow LED does not come on when you attempt to program a card, it means no data is reaching the programmer.

Check the serial cable connections and make sure they are secure at both ends. Make sure the correct cable is being used. (Use either the cable furnished with the programmer or one wired identically.)

If the Yellow LED does come on but the programmer doesn't operate, turn it off and back on and run FLASH again.

Can't find/initialize programmer on COMx.

See above.

Can't program card at (addr).

This error occurs when the verify function fails. If the card is completely bad, the addresses will likely be zero. If the address is not zero, remove the card and clean the contacts with a regular pencil eraser.

Input file 'filename' is too long.

The specified input file is too long to be a compiled program file.

Input file is not a PC-1270 program.

The specified input file is not a program for a PC-1270.

Can't prepare to program.

The programmer is not responding to the prepare to program request. Check the serial cable connections and make sure they are secure at both ends. Make sure the correct cable is being used. (Use either the cable furnished with the programmer or one wired identically.)

No card in programmer.

No card is in the programmer. If there is a card in the programmer and you get this message, try cleaning the contacts on the card with a regular pencil eraser. You can also try resetting the programmer by turning it off and back on. See Resetting the Programmer.

Unknown card in programmer.

The programmer does not recognize the card and cannot erase or program it. If you get this message when there is a card in the programmer, try resetting the programmer by turning it off and back on. See Resetting the Programmer.

Must restart to switch card types.

This error occurs when you have specified the multiple card option (-m option) and switch card types, e.g., install a RAM card after a Flash card or vice versa. You must exit the multiple card loop by pressing "N" and restart FLASH to switch card types.

Program compiled for a RAM card.

The program being downloaded was compiled for a 2K, 4K or 8K RAM card and cannot be installed on a Flash card. (16K RAM card programs can be installed on a Flash card subject to certain restrictions. See the Programming CF Series Flash Cards Application Note.)

If you are using the SBC compiler, make sure to specify a #CARD\_SIZE of 15 in the source file if the program is for a Flash card.

Program compiled for a Flash card.

The program being downloaded has more than 1 bank and cannot be installed on a RAM card. Single-bank Flash programs can be installed on a 16K RAM card subject to certain restrictions. See the Programs for Flash Cards Application Note.

Program requires a xK RAM card. yK card in programmer.

The RAM card in the programmer is a different size than the size specified in the compiled program being downloaded. The size specified in the compiled program must be the same as the size RAM card on which the program is placed.

You can make two or more separate compilations of a program to be installed on different size RAM cards. For example a program that will fit on an 8K RAM card can also be compiled for a 16K RAM card. Generally, any RAM card that is equal to or larger than the minimum size can be used thus a program for a 2K RAM card can be compiled for any size RAM card (2K, 4K, 8K or 16K).

(model) card is too small for program file (fname).

The program in the file (fname) is too large to fit on the card in the programmer.

This version of FLASH.EXE requires firmware version x.x.

The version of FLASH.EXE you are using requires a different firmware version than the programmer you are using. The programmer needs to be updated.

Can't make a RAM file from a RAM card.

RAM initialization files can only be made for Flash cards.

File (ramfname) is not a valid RAM file.

The file specified as the RAM initialization file does not contain RAM initialization data.

Calculator is not initialized.

You are attempting to upload a RAM initialization file from a flash card that has not been initialized in the PC-1270 calculator.

Arrays not DIMensioned

You are attempting to make a RAM initialization file from a Flash card that has not been initialized in a PC-1270 calculator.

RAM file (ramfname) is not for program file (fname).

The RAM initialization file is for a different program than the one specified in the program file name. See How to Make a RAM Initialization File.

If you are using the SBC compiler and have not set the #array\_size to the correct value for your program, you will get this error message. See the Programming CF Series Flash Cards Application Note and SBC Reference Manual for information about the use of #array\_size.

Out of memory.

FLASH needs more memory than it can obtain from your computer. This error is only likely to appear with very large program files (over 256K). To program 512K Flash cards that are full, you will need about 610K of free memory. For card sizes up to 128K, you need at most 256K of free memory to run FLASH.

No RAM file name specified.

To use the “-r” option to make a RAM file, you need to specify both the program file name and RAM file name.

Can't find RAM file (ramfname).

FLASH can't find the RAM file specified on the command line. If you are using the “-a” option and get this error, FLASH can't find the RAM file named (ramfname.ram). With the “-a” option, the RAM file needs to be in the same directory as the program file. If you don't use the “-a” option and specify the RAM file name, it can be in any directory on any drive.

Program in file (fname) doesn't match one in card.

To make a RAM initialization file with the “-r” option, you must specify the name of the program file used to program the Flash card as well as the name of the new RAM initialization file. This error occurs when the program file specified does not contain the program on the Flash card.

Program and RAM files have same name.

The program file and RAM initialization file cannot have the same file name.

Variable Info file has same name as the program file or RAM initialization file.

The Variable Info file has the same filename as either the program file or the RAM initialization file.

Can't initialize RAM at (addr).

The programmer cannot initialize the RAM memory. Try a different Flash card or try cleaning the contacts on the Flash card with a regular pencil eraser.

More than 1 occurrence of  
'#SERIAL-+-+-+--' found in  
program.

There can be one and only one serial number location in  
a program which must be in bank 0 in a multi-bank  
program.

Command-line serial number 'xxx'  
over 16 characters in length.

The maximum length that a serial number can have,  
including the optional date specification and other letters  
is 16 characters.

'#SERIAL-+-+-+--' search string  
not found in program.

The '-' serial number option is specified on the  
command line and FLASH cannot find the search string  
in the program where the serial number is to be installed.

Serial number 'xxxx' over 16  
characters in length.

The serial number for the current card is longer than 16  
characters, probably because the incrementing serial  
number has added a digit, e.g., '99' to '100'.

Command-line serial number 'xxxx'  
has illegal character(s).

Only characters that are legal in the PC-1270 can be  
used in the serial number specified on the command line.

More than 1 occurrence of  
'\$STRING-+-+-+--' found in  
program.

There can be one and only one string substitution  
location in a program which must be in bank 0 in a multi-  
bank program.

Command-line \$STRING 'xxxx' over  
16 characters in length.

The substitution string specified on the command line,  
including the optional date specification, is too long.

Command-line \$STRING has no  
characters.

The '-' option has been specified with no characters  
following the '\$'.

'\$STRING-+-+-+--' search string  
not found in program.

The '-' string substitution option is specified on the  
command line and FLASH cannot find the search string  
in the program where the string is to be installed.

Command-line \$STRING 'xxxx' has  
illegal character(s).

Only characters that are legal in the PC-1270 can be  
used in the substitution string specified on the command  
line.

Blinking red LED's.

If the two red LED's on the programmer blink alternately,  
the version of FLASH.EXE is not correct for the  
programmer (Version 1.7 or later of FLASH.EXE is  
required in this case).

Error (###).

Other errors are reported by number. Call PROM  
Software to report the error number and the  
circumstances under which it occurred.

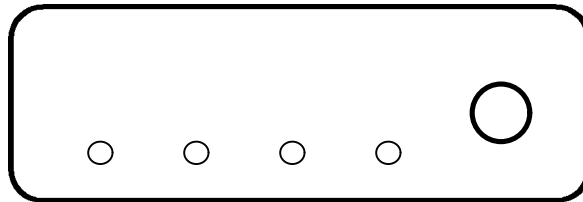
Warning: File 'filename' is NOT  
password protected.

This warning will appear if the program being  
downloaded is not password protected. The SBC  
Compiler automatically password protects a file (unless  
an option is set to make an unprotected copy). You can  
suppress this warning with the "-U" option. This warning  
also does not appear if you use the "-Q" quiet option.

## Serial Cable

The F-512A Programmer serial connector has six connections, numbered 1 through 6 starting at the top of the connector when facing the rear of the programmer. A standard telephone cord can be used in place of the one provided if you desire a different length. There are two types of telephone cords in common use today (both with the same connectors but wired differently). If one cord doesn't work, try another.

Programmer	9-pin PC connector
2	2
3	5
4	no connection
5	3



ERASE ON COM POWER

## Year 2000 Compliant

This program is not date aware and will operate properly in the Year 2000 and beyond. The String Substitution Current Date Option retrieves the current date from the PC's operating system. If the PC's operating system is not Year 2000 Compliant, the correct date may not be retrieved.

**Limited Warranty:** Although this software is believed to be accurate and error free, PROM Software does not so warrant and assumes no liability to any person or persons in conjunction with the use of or the inability to use this software. The media on which this software is furnished will be replaced if defective in manufacture or packaging and is returned to us within the warranty period.

Except for such replacement, the software is provided without warranty or liability.