Manuals+ — User Manuals Simplified.

PEmicro CPROG12Z Windows Command Line Version Software User Guide

PEmicro CPROG12Z Windows Command Line Version Software User Guide

Contents

Introduction

CPROG12Z is a Windows command-line version of the PROG12Z software which programs Flash, EEPROM, EPROM, etc. through a PEmicro hardware interface to a supported NXP HC(S)12(X) processor. The hardware interfaces are available from PEmicro.

Once your interface hardware is properly connected between your PC and target device, you may launch the CPROG12Z executable from the command line. In addition to the executable, multiple command-line parameters must also be passed in order to configure which PEmicro hardware interface CPROG12Z should attempt to connect to, and to configure how that hardware interface will connect to the target device. These parameters include the name of the configuration (.CFG) file, as well as startup commands such as the name of the hardware interface or the port to which the interface is connected.

The .CFG file specifies how to program the target as you intend, and it includes standard programming commands and, optionally, configuration commands. The following chapters will provide a detailed explanation of these commands and parameters.

Startup

a. Connect the hardware interface between your PC and the target MCU via the debug ribbon cable.
b. Start the programming software by running it from the Windows Command prompt or by calling the CPROG12Z executable with the correct command line parameters. Allowed command line parameters are CPROG12Z [?/!] [filename] [/PARAMn=s] [io_delay_cnt n] [v] [reset_delay n] [hideapp] [freq n] [Interface=x] [port=y] [showports] [/logfile logfilename] where:

[?/!]	Use the '?' or' '!' character option to cause the command-line programmer to wait and display the result of programming in the PROG12Z window. '?' will always display the result, '!' will di splay the result only if an error occurred. If the user does not use a batch file to test error leve l, this provides a method to display the programming result. This option should be the FIRST command-line option
[filename]	A file containing programming commands and comments, default = prog.cfg. See Section 8 – Example Programming Script File for an example
[/PARAMn=s]	A command-line parameter which can modify the executing script by replacing special tags (/ PARAMn). This can be used to replace any part of the script including programming comman ds, filenames, and parameters. Valid values of n are 0..9. s is a string which will replace any occurrence of /PARAMn in the script file. Section 9 – Using CommandLine Parameters in a Script has an example for usage.
UNIQUEID	Where x is one of the following: (See examples section)
	USBMULTILINK (This setting also supports OSBDM) CYCLONE PARALLEL (Parallel Port or BDM Lightning [Legacy])
[PORT=y]	Where the value of y is one of the following (see the showports command-line parameter for a list of connected hardware; always specify the "interface" type as well):
USBx	Where x = 1,2,3, or 4. Represents an enumeration number for each piece of hardware startin g at 1. Useful if trying to connect to a Cyclone or Multilink product. If only one piece of hardw are is connected, it will always enumerate as USB1.
	An example to select the first Multilink found is: INTERFACE=USBMULTILINK PORT=USB1
#.#.#.#	Ethernet IP address #.#.#.#. Each # symbol represents a decimal number between 0 and 25 5. Valid for Cyclone and Tracelink interfaces.

	Connection is via Ethernet. INTERFACE=CYCLONE PORT=10.0.1.223	
NAME	Some products, such as the Cyclone and Tracelink, support assigning a name to the unit, such as "Joe's Max". The Cyclone may be referred to by it's assigned name. If there are any spaces in the name, the whole parameter should be enclosed in double quotes (this is a Windows requirement, not a PEmicro requirement)	
	Examples: INTERFACE=CYCLONE PORT=MyCyclone99 INTERFACE=CYCLONE "PORT=Joe's Cyclone	
UNIQUEID	USB Multilink products all have a unique serial number assigned to them, such as PE5650030.	
	The Multilink may be referred to this number. This is useful in the case where multiple units are connected to the same PC.	
	Examples: INTERFACE=USBMULTILINK PORT=PE5650030	
COMx	Where x = 1,2,3, or 4. Represents a COM port number. Valid for Cyclone interfaces.	
	To connect to a Cyclone on COM1 : INTERFACE=CYCLONE PORT=COM1	
x	Where x = 1,2,3, or 4. Represents a parallel port number	
	To select a parallel interface on Parallel Port #1 : INTERFACE=PARALLEL PORT=1	
PCIx	Where x = 1,2,3, or 4. Represents a BDM Lightning card number. (Note: this is a legacy product)	

[showports]

The command-line programmer outputs all available ports to a text file and then terminates (regardless of other commandline parameters). This information output to the text file includes the parameters needed to contact attached programming hardware as well as a description of the hardware interface. The default output filename is ports.txt and is created in the same folder as CPROG.

The output can also be directed to a different file.

Example: SHOWPORTS=C:\MYPORTS.TXT

This list does not show parallel port or COM port options which are also available. Below is an example of the output for various hardware interfaces connected to the PC (Note that there are different ways to address the same unit; the data for each interface may be followed by a [DUPLICATE] line which shows a different label for the same interface)

Showports Output Example:

INTERFACE=USBMULTILINK PORT=PE5650030
; **USB1** : Multilink Universal FX Rev A (PE5650030)[PortNum=21]
INTERFACE=USBMULTILINK PORT=USB1
; **USB1** : Multilink Universal FX Rev A (PE5650030)[PortNum=21][DUPLICATE]
INTERFACE=CYCLONE PORT=10.0.9.197
; **10.0.9.197** : Public Cyclone [PortNum=61]
INTERFACE=CYCLONE "PORT=Public Cyclone"
; **10.0.9.197** : Public Cyclone[PortNum=61][DUPLICATE]
INTERFACE=CYCLONE "PORT=Joe's Cyclone"
; **USB1** : Cyclone (Joe's)[PortNum=101]
INTERFACE=CYCLONE PORT=USB1
; **USB1** : Cyclone (Joe's)[PortNum=101][DUPLICATE]

[io_delay_cnt n]

Sets the background debug transfer parameter n where 0<=n<=$7fff. This is the speed of the serial communications on the background debug line. If you start PROG12Z without giving an io_delay_cnt, the software tries to find an appropriate parameter n and shows you what was selected.

for xtal frequencies less than approximately 1 MHz. Parallel cables only.

[v]	Causes the programmer not to check the range of S-record addresses before programming or verifying. This speeds up the programming process. The option should be used with care as all out of range s-records will be ignored
[reset_delay n]	Specifies a delay after the programmer resets the target that we check to see if the part has properly gone into background debug mode. This is useful if the target has a reset driver which hold the MCU in reset after the programmer releases the reset line. The n value is a delay in milliseconds.
[hideapp]	This will cause the command-line programmer to not display a visual presence while running with the exception of appearing on the taskbar. 32-bit applications only
[freq n]	By default, the PROG12Z software tries to determine automatically how fast the target is running by loading a delay routine in the processor and timing how long it takes to execute. On some machines, this may yield inconsistent results which may affect algorithms which program flash internal to an MCU. PEmicro provides a command-line mechanism allowing the user to inform the PROG12Z software exactly how fast the target processor is running. In this way, the timing in the algorithms will be precise. On the command-line, you specify the INTERNAL clock frequency in Hertz following the 'FREQ' identifier. Note that in general if you are using a flash device external to the MCU, this timing parameter is not needed as the flash handles the timing itself.
[/logfile logfile name]	This option opens a logfile of the name "logfilename" which will cause any information which is written to the status window to also be written to this file. The "logfilename" should be a full path name such as c:\mydir\mysubdir\mylog.lo

Command Line Examples:
CPROG12Z C:\ENGINE.CFG INTERFACE=USBMULTILINK PORT=PE5650030

- Opens CPROG12Z with the following options:
- Run the C:\ENGINE.CFG script
- Interface is first USB Multilink Universal with serial number PE5650030
- Autodetect communications frequency (io_delay_cnt not set)

CPROG12Z C:\ENGINE.CFG Interface=CYCLONE Port=209.61.110.251

- Opens CPROG12Z with the following options:
- Run the C:\ENGINE.CFG script
- Interface is Cyclone Pro via the Ethernet Port with an IP address of 209.61.110.251

CPROG12Z C:\ENGINE.CFG Interface=USBMULTILINK Port=USB1

- Opens CPROG12Z with the following options:
- Run the C:\ENGINE.CFG script
- Interface is USB Multilink Universal, first interface detected.

Securing An HC(S)12(X) Device

An HC(S)12(X) device can be secured after it is programmed by adding the SD
command to your programming script. Once a device has been secured, it will not respond to normal background
debug mode commands and you will no longer be able to access the onboard memory.

The device can be unsecured with the UNSECURE_12 utility which can be downloaded from **www.pemicro.com**.

Programming Commands

Programming commands all start with a two character sequence followed by white space (blanks or tabs). Lines
starting with characters which are not commands are listed as REMarks. The term filename means a full DOS
path to a file. Commands use the same two letter codes as used in the interactive programmers PROG12Z. The
same .12P files used by PROG12Z are used to set up for a particular device to be programmed. If a user function
is specified for a particular device, its two character command and the meaning or user_par are specified in the
.12P file.

Note: The command parameters starting_addr, ending_addr, base_addr, byte, word, and user_par use a default
hexadecimal format.

BM	Blank check module
BR starting_addr ending_addr	Blank check range
CHANGEV n.nn	(Cyclone only) Change the voltage provided to the target, where n.nn represents a value between 0.00 and 5.00, inclusive. When the command executes the Cyclone will immediately change to that voltage. If the Cyclone relays are off prior to calling this command, then the relays will turn on and set the new voltage value when this command is executed. Note that too low of a voltage value may put the device into low-power mode which can lose debug communication altogether. Make sure the Cyclone's jumper settings are set correctly to send the power to the right ports.

EB starting_addr ending_addr	– Erase byte range.
EW starting_addr ending_add	Erase word range.
EM	Erase module.
PB starting_addr byte … byte	Program bytes
PW starting_addr word … word	Program words.
PM	Program module
CM filename base_addr	Choose module .12P file. Note: Certainmodules may require a base address to be specified
VM	Verify module
VR starting_addr ending_add	– Verify range.
UM filename	Upload module
UR starting_addr ending_addr filename	Upload range
SS filename	Specify S record.
SM starting_addr ending_addr	Show module.

RELAYSOFF	(Multilnk FX & Cyclone only) Turn off the relays that provide power to the target, including a power down delay if specified. Especially useful for users who want to power cycle their board before running tests, allow their bootloader to run, or have the application code run after programming.
RELAYSON	– (Multilnk FX & Cyclone only) Turn on the relays to provide power to the target, including a power up delay if specified. The voltage supplied will be based on the last voltage setting specified. For Cyclone users, the CHANGEV command can change the voltage value. Especially useful for users who want to power cycle their board before running tests, allow their bootloader to run, or have the application code run after programming.
HE	Help (look at cprog.doc file).
SD	Secure Device, programs the NVOPT (nonvolatile options register) bit to secure the device
QU	Quit
RE	Reset chip.
GO	– Starts device running. Can be used as final command if you want the device to run for testing. Should be immediately preceded by an 'RE' command
DE timeinms	– Delays "timeinms" milliseconds
xx user_par	Only for user function specified in .12P file.

Configuration Commands For Startup

Configuration commands are all processed before the programmer attempts to contact the target. The whole configuration file is parsed for these commands prior to attempting communications. This section gives an overview of using these configuration commands to do different type of configuration.

Note: The default base for configuration command parameters is decimal

An overview of the configuration commands is as follows:

CUSTOMTRIMREF nnnnnnnn.nn
Desired internal reference clock frequency for the "PT; Program Trim" command. This frequency overrides the default internal reference clock frequency. Valid values for "n" depend on the particular device being programmed. Please refer to the electrical specifications of your device for valid internal reference frequency clock range.

Where:

nnnnnnnn.nn: Frequency in Hertz with two decimal places

PROVIDEPOWER n
Determines whether interface should provide power to the target. NOTE: Not all hardware interfaces support this command. Valid values of n are:

0 : Interface does not provide power to target. (default)
1 : Enable Interface provides power to target.
(NOTE: Is the same as legacy option :USEPRORELAYS n)

:POWERDOWNDELAY n

Amount of time to delay when the power to the target is turned off for the targets
power supply to drop to below 0.1v. n is the time in milliseconds.

POWERUPDELAY n
Amount of time to delay when the power to the target is turned on OR the target is reset, and before the software attempts to talk to the target. This time can be a combination of power on time and reset time (especially if a reset driver is used). n is the time in milliseconds.

DEVICEPOWER n
For Cyclone PRO only. This setting defines the target voltage that will be provided to the target if the source of the voltage is derived from the Cyclone PRO's internal power. Valid values of n are:

0 : 5 Volts, Generated/Switched by Cyclone Pro
2 : 3 Volts, Generated/Switched by Cyclone Pro
4 : 2 Volts, Generated/Switched by Cyclone Pro

:POWEROFFONEXIT n
Determines whether power provided to the target should be turned off when the CPROG12Z application terminates. NOTE: Not all hardware interfaces support this command. Valid values of n are:

0 : Turn power off upon exit (default)
1 : Keep power on upon exit

Verification Overview

There are several commands available that can be used to verify the contents of the flash on the device after programming it. The most widely used command is "VC ;Verify CRC of Object File to Module". The "VC" command will instruct CPROG12Z to first calculate a 16-bit CRC value from the chosen object file. CPROG12Z will then load code into the RAM of the device and instruct the device to calculate a 16 bit CRC value from the contents in FLASH of the device. Only valid address ranges in the object file are calculated on the device. Once the 16-bit CRC value from the object file and the device are available, CPROG12Z compares them. An error is thrown if the two values do not match.

Alternatively, the "VM ;Verify Module" command can be used to perform a byte by byte verification between the chosen object file and the device. Typically, the VM command will take longer to perform than VC command since CPROG12Z has to read the contents of FLASH of the device byte by byte. There are also two other commands that can be used for verification. The "SC ;Show Module CRC" instructs CPROG12Z to load code into the RAM of the device and instruct the device to calculate a 16-bit CRC value from the contents of the entire FLASH of the device, which includes blank regions. Once the 16-bit CRC value has been calculated, CPROG12Z will display the

value in the status window. The "VV ;Verify Module CRC to Value" command is similar to the "SC" command. The difference is that instead of displaying the calculated 16-bit CRC value, CPROG12Z will compare the calculated value against a 16-bit CRC value given by the user.

DOS Error Returns

DOS error returns are provided so they may be tested in .BAT files. The error codes used are:

0 – Program completed with no errors.

1. Cancelled by user.
2. Error reading S record file.
3. Verify error.
4. Verify cancelled by user.
5. S record file is not selected.
6. Starting address is not in module.
7. Ending address is not in module or is less than starting address.
8. Unable to open file for uploading.
9. File write error during upload.
10. Upload cancelled by user.
11. Error opening .12P file.
12. Error reading .12P file.
13. Device did not initialize.
14. Error loading .12P file.
15. Error enabling module just selected.
16. Specified S record file not found.
17. Insufficient buffer space specified by .12P to hold a file S-record.
18. Error during programming.
19. Start address does not point into module.
20. Error during last byte programming.
21. Programming address no longer in module.
22. Start address is not on an aligned word boundary.
23. Error during last word programming.
24. Nodule could not be erased.
25. Module word not erased.
26. Selected .12P file does not implement byte checking.
27. Module byte not erased.
28. Word erase starting address must be even.
29. Word erase ending address must be even.
30. User parameter is not in the range.
31. Error during .12P specified function.
32. Specified port is not available or error opening port.
33. Command is inactive for this .12P file.
34. Cannot enter background mode. Check connections.
35. Not able to access processor. Try a software reset.

36. Invalid .12P file.

37. Not able to access processor RAM. Try a software reset.

38. Initialization cancelled by user.

39. Error converting hexadecimal command number.

40. Configuration file not specified and file prog.cfg does not exist.

41. 12P file does not exist.

42. Error in io_delay number on command line.

43. Invalid command line parameter.

44. Error specifying decimal delay in milliseconds.

45. Error in script file.

46. Cable not detected

47. S-Record file does not contain valid data.

48. Checksum Verification failure – S-record data does not match MCU memory.

49. Sorting must be enabled to verify flash checksum.

50. S-Records not all in range of module. (see "v" command line parameter)

51. Error detected in settings on command line for port/interface

52. Error calculating device CRC value

53. Error – Device CRC does not match value given

54. Error – CPROG is already running

55. Error – Must specify both the INTERFACE and PORT on the command line

56. The selected target processor is not supported by the current hardware interface.

Example Programming Script File

The programming script file should be a pure ASCII file with one command per line. This is the CFG file in the previous examples. An example is

CM C:\PEMICRO\9B32_32K.12P 8000	Enable 68HC912B32 Flash at address 8000
EM	;Erase the module
BM	Blank Check the module
SS C:\PEMICRO\TEST.S19	Specify the S19 to use
PM	;Program the module with the S19
VM	Verify the module again

Note: The path names of files that are relative to the CPROG executable can also be

Using Command-Line Parameters in a Script

A command-line parameter in the form of /PARAMn=s can be used to insert text into the script file in place of special tags. This can be used to replace any part of the script including programming commands, filenames, and parameters. Valid values of n are 0..9. s is a string which will replace any occurrence of /PARAMn in the script file.

As an example, the following generic script could be used for programming with exactly the same functionality of the example script in Section 8 – Example Programming Script File:

CM /PARAM1	;Choose Flash Module
EM	;Erase the module
BM	;Blank Check the module
SS /PARAM2	Specify the S19 to use
PM	Program the module with the S19
/PARAM3	;Verify the module again

The following parameters would be added to the CPROG command line:

"/PARAM1=C:\PEMICRO\9B32_32K.12P 8000"
/PARAM2=C:\PEMICRO\TEST.S19
/PARAM3=VM

NOTE: Since the /PARAM1 parameter has a space in its value, the entire parameter needs to be enclosed in double quotations. This indicates to Windows that it is a single parameter. In this instance, a base address of 0x8000 is included on the Choose Module line in the script, therefore /PARAM1 must be specified on the command line like this.

:"/PARAM1=C:\PEMICRO\9B32_32K.12P 8000"

Sample Batch File

Here is an example of calling the command-line programmer and testing its error code return in a simple batch file. Sample batch files are given for both Windows 95/98/XP and Windows 2000/NT/XP/Vista/7/8/10.

Windows NT/2000/Vista/7/8/10:
C:\PROJECT\CPROG12Z C:\PROJECT\ENGINE.CFG
INTERFACE=USBMULTILINK PORT=USB1
if errorlevel 1 goto bad goto good :bad
ECHO BAD BAD BAD BAD BAD BAD BAD BAD :good
ECHO done

Windows 95/98/ME/XP:
START /W C:\PROJECT\CPROG12Z C:\PROJECT\ENGINE.CFG
INTERFACE=USBMULTILINK PORT=USB1
if errorlevel 1 goto bad goto good :bad
ECHO BAD BAD BAD BAD BAD BAD BAD BAD good ECHO done
Note: The path names of files that are relative to the CPROG executable can also be used.

Information

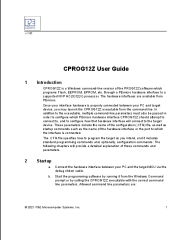For more information on CPROG12Z and PROG12Z please contact us:

P&E Microcomputer Systems, Inc. VOICE: (617) 923-0053

98 Galen St. FAX: (617) 923-0808
Watertown, MA 02472-4502 WEB: **http://www.pemicro.com**

USA

To view our entire library of.12P modules, go to the Support page of PEmicro's website at
www.pemicro.com/support.

Documents / Resources

CPROG12Z User Guide	**PEmicro CPROG12Z Windows Command Line Version Software** [pdf] User Guide CPROG12Z, CPROG12Z Windows Command Line Version Software, Windows Command Line Version Software, Software

References

- **PEmicro | Experts in Embedded tools for Flash Programming and Development**
- **PEmicro | Experts in Embedded tools for Flash Programming and Development**