NEC

User's Manual

*flash*MASTER V 3.02

Flash Programmer

Document No. U13598EE3V2UM00
Date Published June 2000

CE

***flash*MASTER complies with the EMC protection requirements**

Warning

This is a ´Class A´ (EN 55022: 1994) equipment. This equipment can cause radio frequency noise when used in the residential area. In such cases, the user/operator of the equipment may be required to take appropriate countermeasures under his responsibility.

EEDT-ST-001-10

Caution

This equipment should be handled like a CMOS semiconductor device. The user must take all precautions to avoid build-up of static electricity while working with this equipment. All test and measurement tools including the workbench must be grounded. The user/operator must be grounded using the wrist strap. The In-Circuit Emulator probe target connector plug and/or its adapter pins should not be touched with bare hands.

EEDT-ST-004-10

MS-DOS and MS-Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
PC/AT and PC DOS are trademarks of IBM Corp.

The related documents in this publication may include preliminary versions. However, preliminary versions are not marked as such.

The export of this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by the customer. The export or re-export of this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

M5 2000.03

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.l.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Vélizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Spain Office
Madrid, Spain
Tel: 91-504-2787
Fax: 91-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 65-253-8311
Fax: 65-250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

NEC do Brasil S.A.
Electron Devices Division
Rodovia Presidente Dutra, Km 214
07210-902-Guarulhos-SP Brasil
Tel: 55-11-6465-6810
Fax: 55-11-6465-6829

J99.1

Revision History

Date	Revision	Chapter	Description
22.05.1998	V 1.0		First release
30.06.1998	V 1.1	all	Rework of document layout
30.11.1998	V. 2.00	all	Rework of document layout
20.02.2000	V. 3.00	all	Rework of document layout
02.05.2000	V. 3.01	3	Table 3.42, page 28
05.06.2000	V. 3.02	2.2, 8.6	Chapter 2.2, page 15 Chapter 8.6, Table 5, page 88, Table 6, page 89

Table of Contents

List of Figures

List of Tables

1. Introduction

*flash*MASTER is a tool for programming data into, erasing data from, and verifying data in flash ROM of the NEC microcontrollers.

Main features of *flash*MASTER

- On-board programming
  *flash*MASTER enables programming without removing the MCU from the user target system.
- Stand-alone programming
  Program information can be stored in the *flash*MASTER 1MB SRAM or 1MB flash memory in order to program target systems without host connection of *flash*MASTER
- A 2048 bit EEPROM stores programming parameters
- Serial and parallel interface
  Serial interface (RS232C) connection to host systems and parallel interface for optional fast program download available.
- I/O port configuration
  Parallel interface can be configured to control *flash*MASTER remotely.
- Universal host connection
  A Windows 95, Windows 98 and Windows NT User interface is provided as well as ASCII command line interface for terminal connection.
- CSI, UART, I^2C or Port Mode interface supported for data transfer to flash device.
- Programming Voltage can be selected between 5 and 10.5 V.
- Voltage supply (Vdd)
  *flash*MASTER can supply up to 100 mA. (2.5 - 5 V)

This manual will familiarise you with hard- and software of the *flash*MASTER package.

Chapter 2 describes the hardware components and the installation of *flash*MASTER.

Chapter 3 describes the installation of the GUI and the required steps to upgrade *flash*MASTER firmware, if necessary. This chapter is very important if you have *flash*MASTER prior to version 3.00 in use already.

Chapter 4 describes the GUI software.

Chapter 5 explains in a sample session how to program flash devices using the GUI.

Chapter 6 lists all flash and host commands which are available for operating *flash*MASTER in terminal mode.

Chapter 7 shows in a sample session how to program flash devices using a terminal program.

Chapter 8 describes connectors and cables of the *flash*MASTER package.

Chapter 9 and chapter 10 contain design proposals for user systems.

It is strongly recommended to read the **README.TXT** file additionally to this manual. The README.TXT contains last minute information and may be useful for any kind of operation of *flash*MASTER. The README.TXT is available on disk 1 of the software package.

2. Hardware Installation

2.1 System requirements

HOST PC	A PC supporting Windows 95, Windows 98 or Windows NT is required for *flash*MASTER user interface program. For terminal operation of *flash*MASTER any terminal program may be used.
Host interface	Serial (RS232C) interface capable to handle communication at 9600 (minimum) baud up to 38400 baud.
File formats	Program files must be available in Motorola S format or Intel HEX file format.
GUI	The graphical user interface (GUI) is available for Windows 95, Windows 98 or Windows NT operating systems. About 10 Mbytes of free hard disk space is required to install the graphical user interface (GUI) software.
Compatible MCU	All NEC devices with flash ROM can be programmed. A detailled list of programmable devices is available in the README file. Please read this information carefully.

2.2 Package contents

Please verify that you have received all parts listed in the package contents list attached to the *flash*MASTER package. If any part is missing or seems to be damaged, please contact the dealer from whom you purchased your *flash*MASTER.

Note: Updates to this User's Manual, additional documentation and/or device parameter file(s) for *flash*MASTER, if available, may be downloaded from the NEC WEB page(s) at

http://www.nec.de/updates

2.3 System configuration and components

The *flash*MASTER system configuration is given in the diagram below:

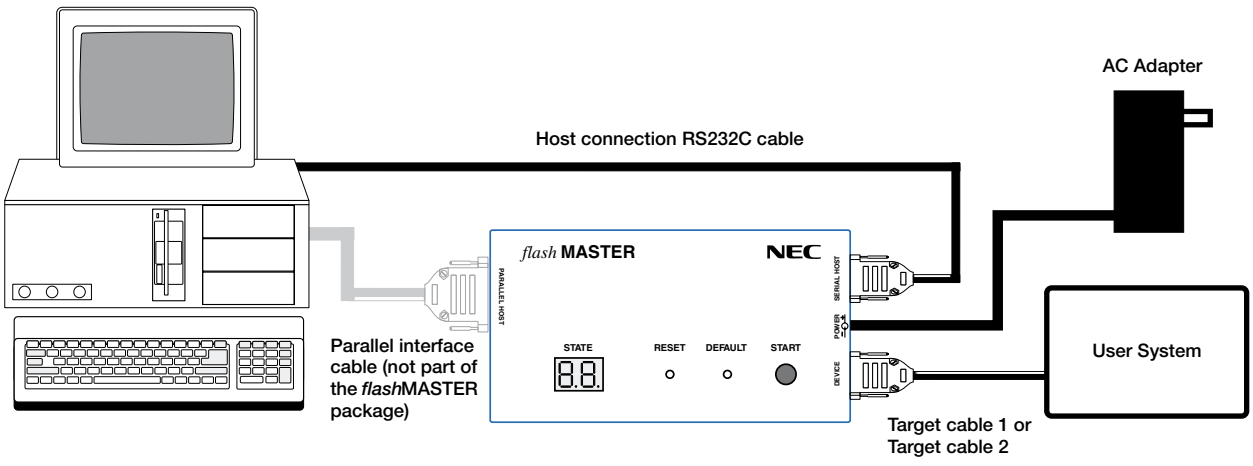


Figure 1: *flash*MASTER system configuration

*flash*MASTER is connected to the host system via RS232C serial interface cable. An optional parallel interface connection to the hosts system printer port may be established using the centronics interface connection of *flash*MASTER. This interface cable is not part of the *flash*MASTER package. *flash*MASTER is connected to the user system by either target cable 1 or target cable 2 depending on the layout of the pin connector of the user system. For any detailed specification of target cable 1 or target cable 2 please refer to the chapter "Connectors and Cables" of this documentation.

2.3.1 Host computer

A PC is used to communicate with *flash*MASTER. The PC must support any terminal program to communicate to the *flash*MASTER command line interface. For GUI operation of *flash*MASTER Windows 95, Windows 98 or Windows NT must be available.
It also must be equipped with a serial interface. An optional parallel interface may be used for fast program download to *flash*MASTER.

*2.3.2 flash*MASTER

2.3.2.1 *flash*MASTER control panel

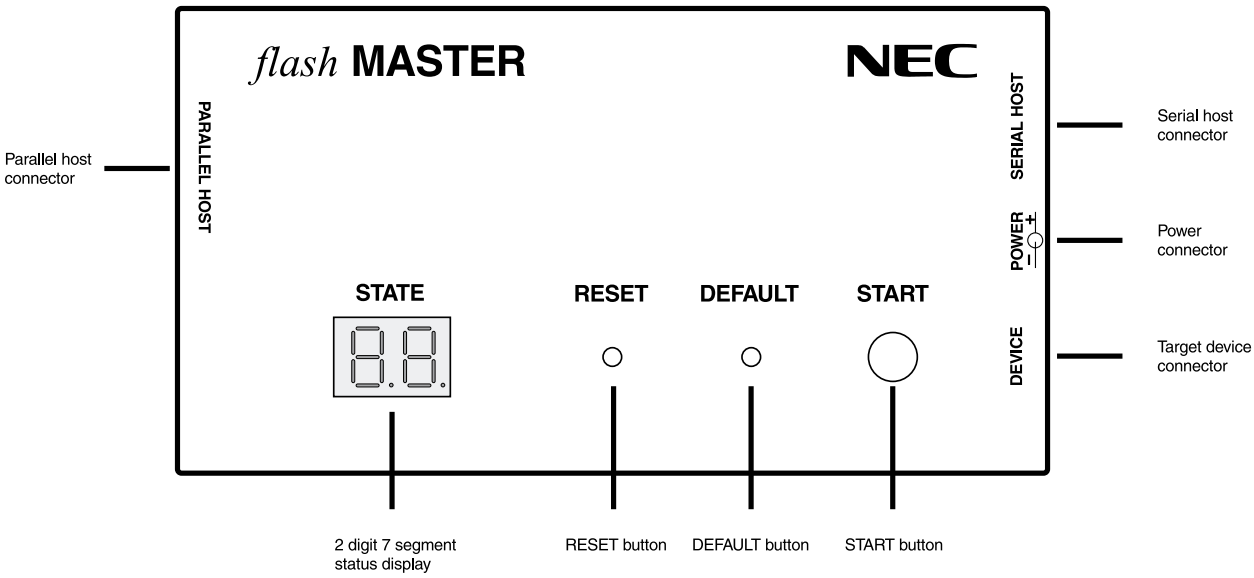


Figure 2: *flash*MASTER top view

The control buttons are used in stand alone operating mode of *flash*MASTER.

RESET button	The RESET button resets the *flash*MASTER internal MCU.
DEFAULT button	The DEFAULT button will reset the host communication to 9600 bps.
START button	The START button will initiate the EPV (Erase / Program / Verify) command sequence.
Status display	The status display informs about the operating modes. It is mainly used for *flash*MASTER s stand alone operating mode.

2.3.2.2 *flash*MASTER connectors

The serial connector, the target connector and the power connector are located on the right side of *flash*MASTER.

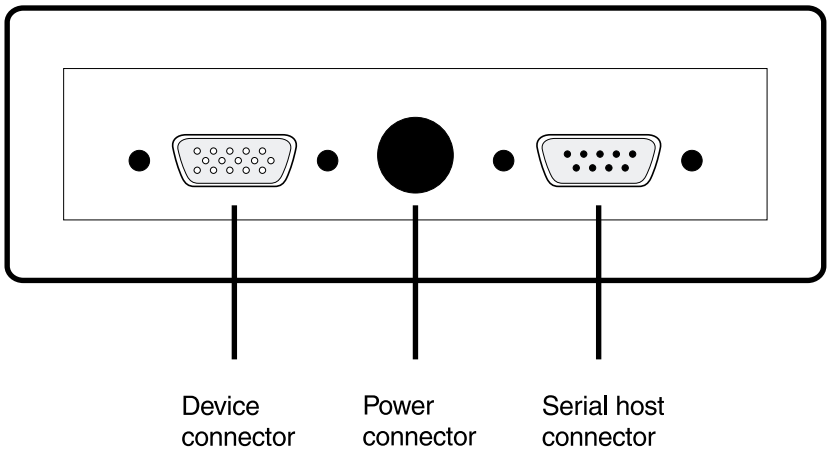


Figure 3: *flash*MASTER connectors

The parallel connector is located on the left side of *flash*MASTER. The parallel interface connector may be configured as centronics interface (default) or I/O port.

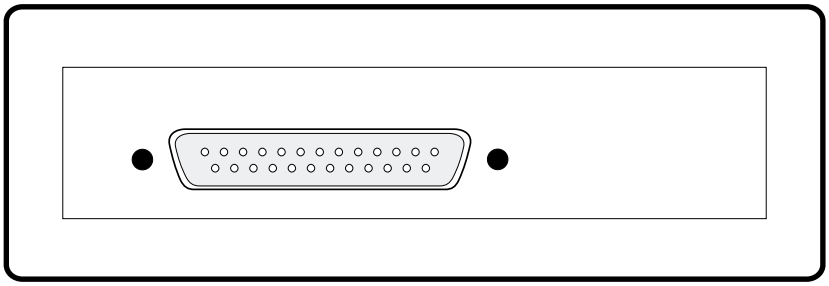


Figure 4: *flash*MASTER parallel host interface

2.3.3 User system

The user system must be equipped with a device interface according to target cable 1 or target cable 2 specification. For any detailed specification please refer to the chapter “Connectors and Cables” of this document.

2.3.4 Power supply

The power supply FW7201/12 is equipped with a DC-plug 2.1 x 55 x 14 and may be connected to mains using one of the available AC-plugs Euro, UK or USA / Japan.

For a specification of the power supply jack please refer to the chapter “Connectors and Cables” of this document.

Note: Connect only the provided AC adapter to the power supply jack!

2.3.5 RS232 Host connection

The RS232 host interface enables communication to the *flash*MASTER. A terminal program or the Windows 95 / Windows 98 / Windows NT GUI may be used to operate *flash*MASTER, which is connected to the serial port. RS232 data transfer starts at 9600 baud, 8 data bits, 1 stop bit, no parity and no hardware handshake.
The baudrate may be selected from 9600 bps (default), 19200 bps or 38400 bps.

For a detailed specification of the host interface please refer to the chapter “Connectors and Cables” of this document.

2.3.6 Parallel host connection

The parallel host connection may be configured in one out of two possible configurations

1. Centronics Interface configuration
   Configured as centronics interface program data can be downloaded via high speed parallel port from the host system.

2. I/O port configuration
   Configured as I/O port *flash*MASTER can be controlled remotely. Via I/O port start signal an ERASE-PROGRAM-VERIFY sequence can be started and *flash*MASTER signals status information as device connected, busy, OK and error via I/O signal lines.

For a detailed specification of the parallel interface please refer to the chapter “Connectors and Cables” of this document.

2.3.7 Target cable 1

The target cable 1 is compatible with all NEC flash programming adapters FA-XXX-YY.

XXX: 20 pin, 28 pin, 30 pin, 42 pin, 44 pin, 64 pin, 80 pin, 100 pin and 144 pin.
YY: package type GC, GF, GJ, GK, GS and CT.

For a detailed specification of target cable 1 please refer to the chapter “Connectors and Cables” of this document.

2.3.8 Target cable 2

The target cable 2 is compatible with the programming layout of the EB-V853STARTER.

For a detailed specification of target cable 2 please refer to the chapter “Connectors and Cables” of this document.

3. Software Installation

3.1 Graphical User Interface

The *flash*MASTER graphical user interface software allows easy and most comfortable access to all features of the *flash*MASTER flash programmer. It requires MS Windows 95, Windows 98 or MS Windows NT, V4.0 or higher versions as operating system.

This software allows the user:

- Setup the connection between *flash*MASTER and PC
- Setup all device specific options of the *flash*MASTER
- Download a program to the *flash*MASTER
- Erase, program and verify all 75XL, K0S, K0, K4 and V850 Series flash devices
- Run automatic Erase -> Program -> Verify procedures

All current device specific options are stored in the *flash*MASTER hardware itself. When the program is opened these options are loaded from the hardware. So the last used options are present.

Any terminal program may be used as well to operate *flash*MASTER. The command interface is not that comfortable as the GUI and it requires much more command input.

3.2 GUI Installation

The installation program is located on the first of the two floppy disks, which are delivered with the *flash*MASTER package.

To install the GUI software, please perform following steps:

⇒ Insert floppy disk 1 into your floppy drive (normally drive a:)
⇒ Click on the Windows **START** button
⇒ Click on **RUN**
⇒ Type in or select ‘a:\setup’ (If your floppy drive has another letter, change the entry accordingly)
⇒ Click on the **OK** button.

Now the setup program is starts. After a short while of initialisation the welcome screen appears:

Figure 5: Welcome screen

Click on **Next** to continue the installation.

The installation start window appears:

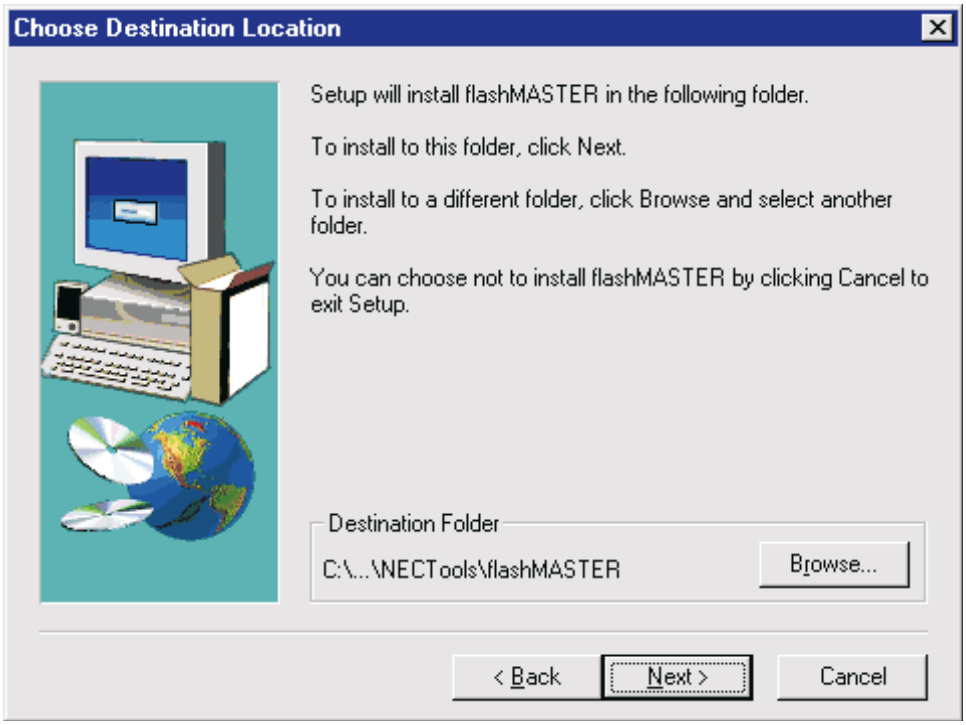


Figure 6: Installation start window

You might exit the installation with the **Cancel** button.
You can click on the **Browse** button to be able to change the installation destination path.
Pressing the **Next** button will proceed to the *Setup Program Folder* selection.

When activating the **Browse** button**,** the *Choose Folder* window appears:

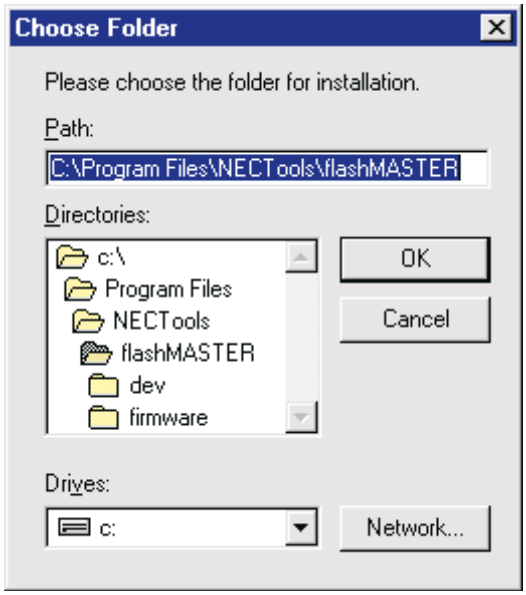


Figure 7: Choose Folder window

Here you can change the destination path. Then click on **OK** to accept the changes or click on **Cancel** to undo the changes. You will return to the Installation start window (Figure 7).

When the folder window is closed, press the **Next** button in the installation start window. The setup procedure opens the *Program Folder* selection window. By default a program group NEC Tools will be created for the *flash*MASTER executable. You may change the folder name by selecting one of the existing folders or by typing a new folder name in the edit field.

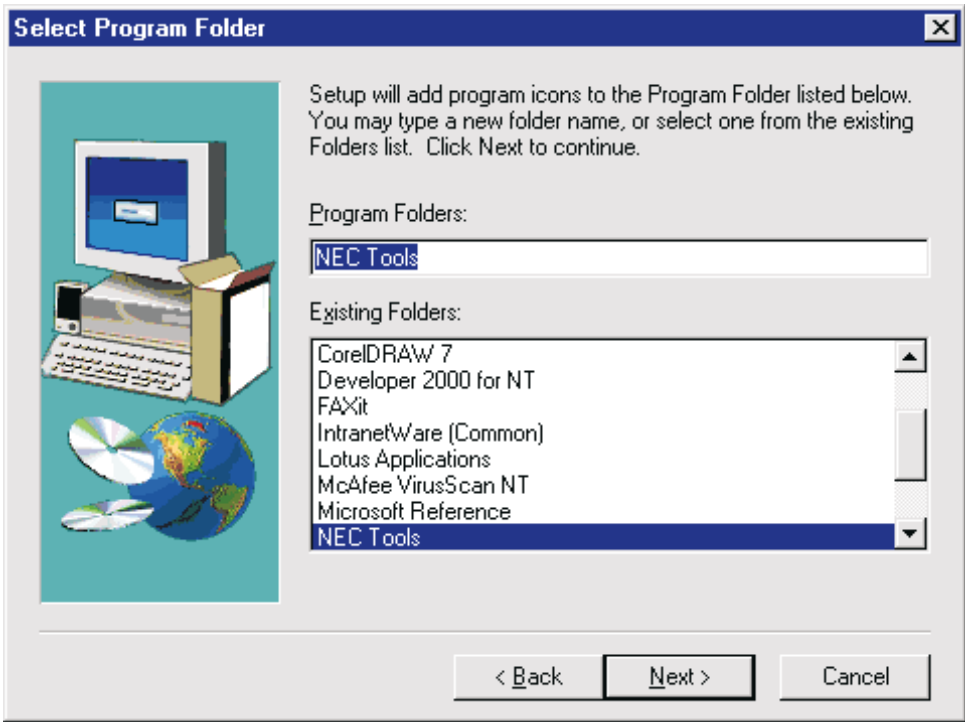


Figure 8: Select Program Folder window

When the correct folder has been selected, the **Next** button will start file copying to your hard disk. On the installation progress window you can follow the installation. You will be prompted to change the floppy disk.

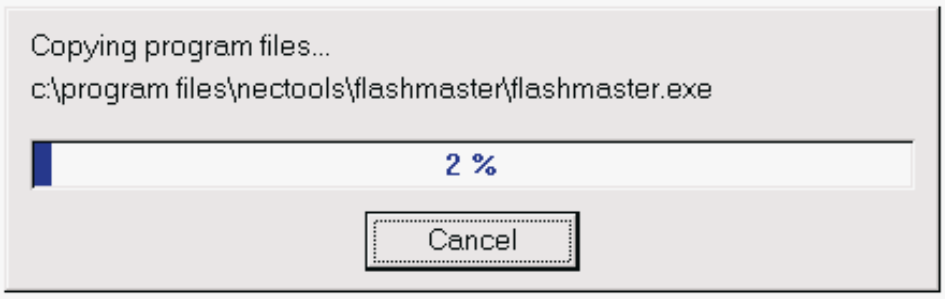


Figure 9: Setup progress window

You can always abort the installation with the **Cancel** button.

When the installation is terminated, the following message window appears:

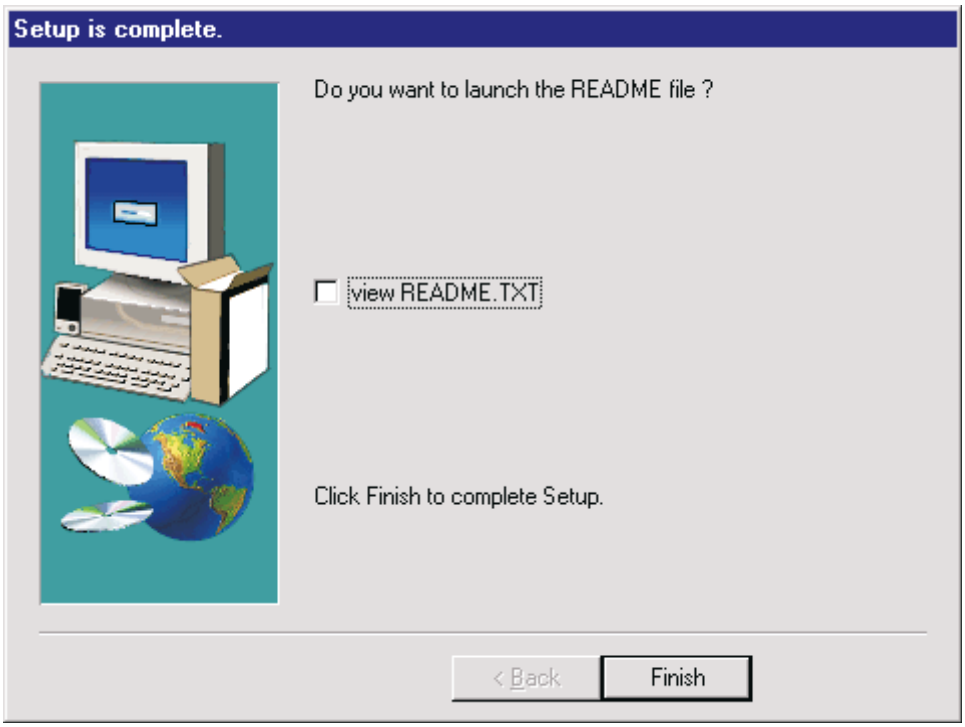


Figure 10: Setup is complete

The program has been installed successfully and an entry has been made into the program menu to start the *flash*MASTER software. Select the view README.TXT checkbox to start Notepad to show you the README.TXT file. Since this file is copied into your *flash*MASTER directory anyway you may review it at later times.
Press the **Finish** button to exit the installation.

The installation process is complete now and the setup utility has installed a new program folder which holds the *flash*MASTER GUI Software and a *flash*MASTER unInst icon which allows to delete the *flash*MASTER GUI software if it is not used any more.

Figure 11: Program folder after installation

The setup program has installed following files on your hard disk:

`C:\...\NECTools\flashMASTER`	
`flashMASTER.EXE`	*flash*MASTER GUI.
`fMASTER.ini`	Customisation of the GUI as well as storage of most recently used settings.
`Readme.txt`	Last minute information.
`Uninst.isu`	Database for proper use of unInst.
`\dev\*.dev`	Target device initialisation information if GUI is used.
`\std\*.std`	Target device initialisation information if terminal program is used.
`\cust\*.dev`	user customised device settings
`\prg\*.prg`	bootstrap files
`\firmware`	
`FIR_V300.rec`	*flash*MASTER firmware program file
`EXT_V300.rec`	*flash*MASTER external functions program file
`SP_V300.rec`	*flash*MASTER selfprogramming library

3.3 Terminal installation

If a terminal program is used as communication interface only some data files will be needed for the software package enclosed to the *flash*MASTER package.
In subdirectory `\std` of disk 2 some device initialisation files are available which may be used with the `loaddev` command. Please copy these files to your harddisk or wokstation for later use.

Start communication with *flash*MASTER using 9600 bps, 8 data bits, 1 stop bit no parity and no hand-shake. Once communication is established you may switch to 19200 bps or 38400 bps.

Make sure that communication is done in lower case letters only.

3.4 Firmware Installation

In order to guarantee proper operation of *flash*MASTER programmer it is mandatory that the correct firmware version is available in the programmer's internal memory. The GUI software checks the firmware version automatically when connecting to the programmer. In case the firmware is not up to date you may update the firmware by the *flash*MASTER GUI software or any terminal communication software.

Firmware installation consists of 3 steps which are executed automatically if you update the firmware using the GUI software.

Step 1: Installing firmware for internal functions

Internal functions provide rudimentary operations of the programmer. Mainly they support communication of the programmer with the host system and the installation of external functions. Updating internal functions is the most sever state in the installation process.
The program code file for the internal functions is located in the subdirectory `firmware` in your *flash*MASTER directory. It needs to be downloaded.
After successful download it needs to be selfprogrammed. The selfprogramming algorythm itself is located in the directory `firmware` as well.
Although the whole process of installing new firmware should not be interrupted, the selfprogramming state is the most sensitive one. Switching off power or resetting the programmer may damage the programmer seriously!

Step 2: Installing firmware for external firmware functions

External functions support all the commands useable for setting up the programmer to the needs of the target device.
The file containing external firmware functions is located in the subdirectory "firmware" in your *flash*MASTER directory. After downloading the file the programmer will restart automatically with the just downloaded new functionality.

Step 3: Setting default values for all parameters in the programmer

Attention:

Installing new firmware is a rudimentary operation for the programmer!

PLEASE READ THE FOLLOWING CHAPTER VERY CAREFULLY AND TAKE YOUR TIME FOR UPDATING YOUR *flash*MASTER FIRMWARE!

3.4.1 Firmware update using *flash*MASTER GUI software

When you start the *flash*MASTER GUI software it checks the version number information from the programmer's hardware automatically.

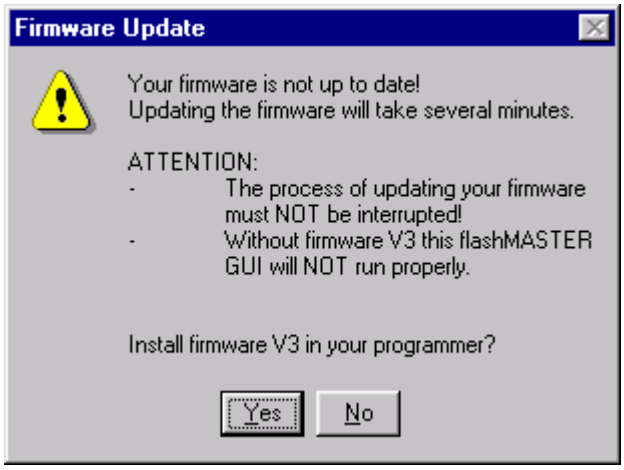


Figure 12: Firmware Update Window

You should click the **Yes** button in order to install the new firmware version. Otherwise *flash*MASTER will not operate correctly!

During all operations of *flash*MASTER the status window will inform about the current activities:

Figure 13: GUI status window

After downloading the new firmware it needs to be programmed into *flash*MASTER's internal memory.

An additional message box will inform that from now on, *flash*MASTER **must not be switched off or reset**:

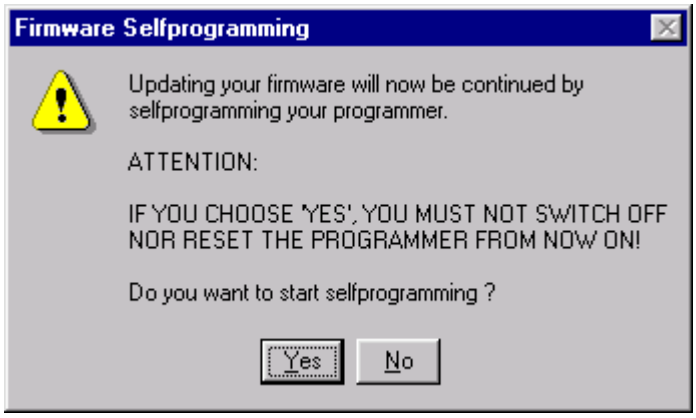


Figure 14: Firmware Selfprogramming Window

You will be notified, if the new firmware is installed successfully in your programmer:

Figure 15: Firmware update successful message

In case selfprogramming failed GUI will warn and recommend counter measures:

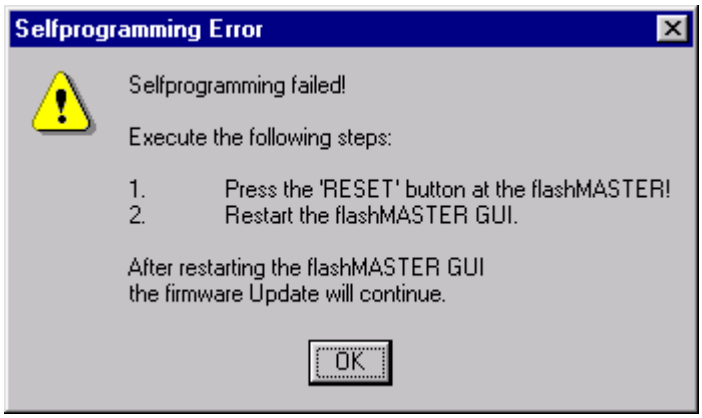


Figure 16: Selfprogramming failed message

Please follow the recommendation given and reset *flash*MASTER and restart the GUI software again.

In case the installation of the new firmware in your programmer failed the following message will be displayed:

Figure 17: Firmware update error message

This message will also appear if any of the firmware update steps failed and *flash*MASTER holds invalid fragments of firmware data.

3.4.2 Firmware update using terminal communication software

If *flash*MASTER is used without the GUI software firmware can also be upgraded using a terminal program. After installation of the hardware and startup of a terminal program select 9600 baud as communication speed and no flow control as handshake. *flash*MASTER will start up showing following screen:

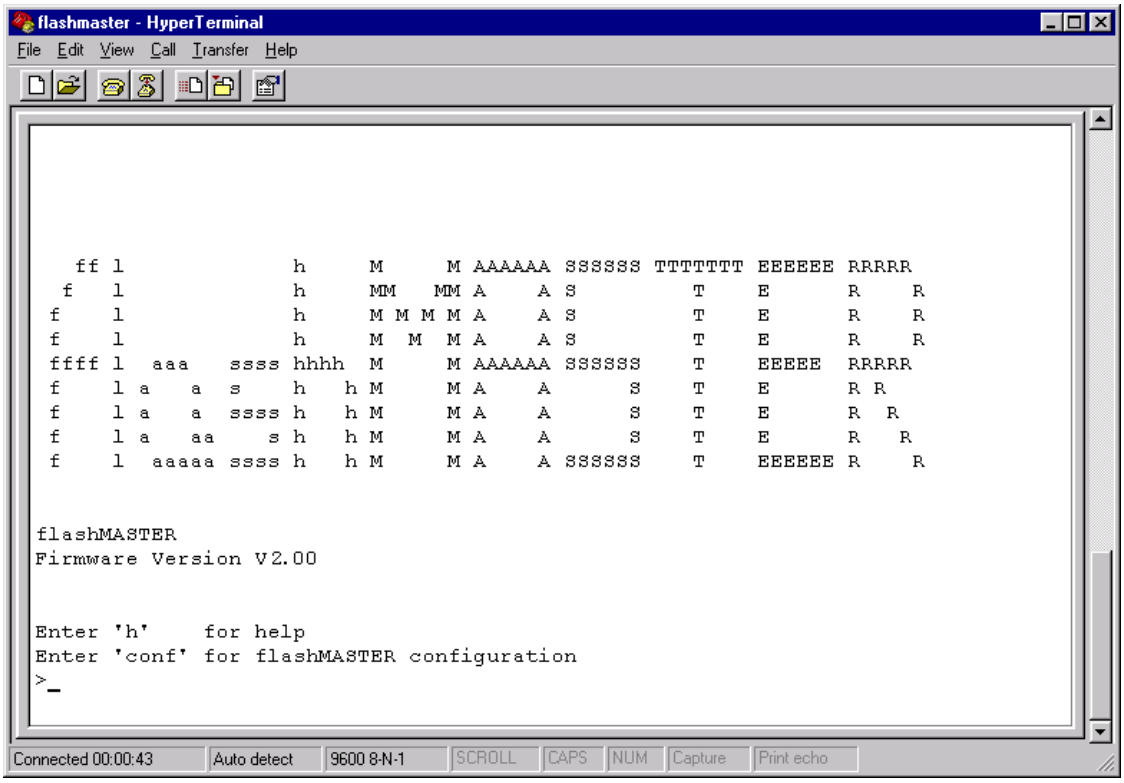


Figure 18: Initial screen before firmware upgrade

All data files necessary for the firmware update are located in the directory \firmware of disk 2 of the software package. It is recommended to copy the files from this directory on to your hard disk.
For firmware upgrade, please enter below given command sequence.
Note that *flash*MASTER will accept only lower case characters in terminal mode!

Command	Description	LED
`set mem r` ↵	Select *flash*MASTER internal memory	
`set vpp 10300` ↵	Set the correct programming voltage	
`ls ff` ↵	Preset internal memory with 0xFF and start program download sequence for the internal part of the firmware When `ls` command is accepted, *flash*MASTER waits for new program data in Motorola S Format to be sent via serial line. Select Send Text File from your terminal program and specify `FIR_V300.REC` as program filename. *Flash*MASTER will display several dots on screen while download is in progress. Please wait until the message `Done` is shown. `FIR_V300.REC` is available in the subdirectory firmware of disk 2 of the software package enclosed to the *flash*MASTER package.	

`ls` ↵

Download the selfprogramming function library `SP_V300.REC`.
When `ls` command is accepted, *flash*MASTER waits for new program data in Motorola S Format to be sent via serial line.
Select Send Text File from your terminal program and specify `SP_V300.REC` as program filename.
*Flash*MASTER will display several dots on screen while download is in progress. Please wait until the message `Done` is shown.
`SP_V300.REC` is available in the subdirectory firmware of disk 2 of the software package enclosed to the *flash*MASTER package.

ATTENTION:

Selfprogramming the *flash*MASTER firmware is a rudimentary operation!

Interrupting the selfprogramming process, switching off or resetting *flash*MASTER will seriously damage the programmer!

`g 120000` ↵

Program the new firmware into *flash*MASTER.
*Flash*MASTER will display several dots on screen during self programming. Please wait until the message `Done` is shown.
The self programming may take up to 10 minutes.
After successful programming, *flash*MASTER will startup automatically showing the new startup screen.

`lu` ↵

Start program download sequence for external functions.
When `lu` command is accepted, *flash*MASTER waits for new program data in Motorola S Format to be sent via serial line.
Select Send Text File from your terminal program and specify `EXT_V200.REC` as program filename.
*flash*MASTER will display several dots on screen while download is in progress. Please wait until the message `Done` is shown.
`EXT_V200.REC` is available in the subdirectory firmware of disk 3 of the software package enclosed to the *flash*MASTER package.

`conf` ↵

Change to external firmware.

`init` ↵

Reset firmware program constants and write new parameters to *flash*MASTER s EEPROM.
The communication speed to the host will be set to 9600 bps. If you are connected to *flash*MASTER with some different communication speed, make sure to adjust the port settings after this command.

`+++` ↵

Reset *flash*MASTER to ensure that all internal variables are in their initial state.

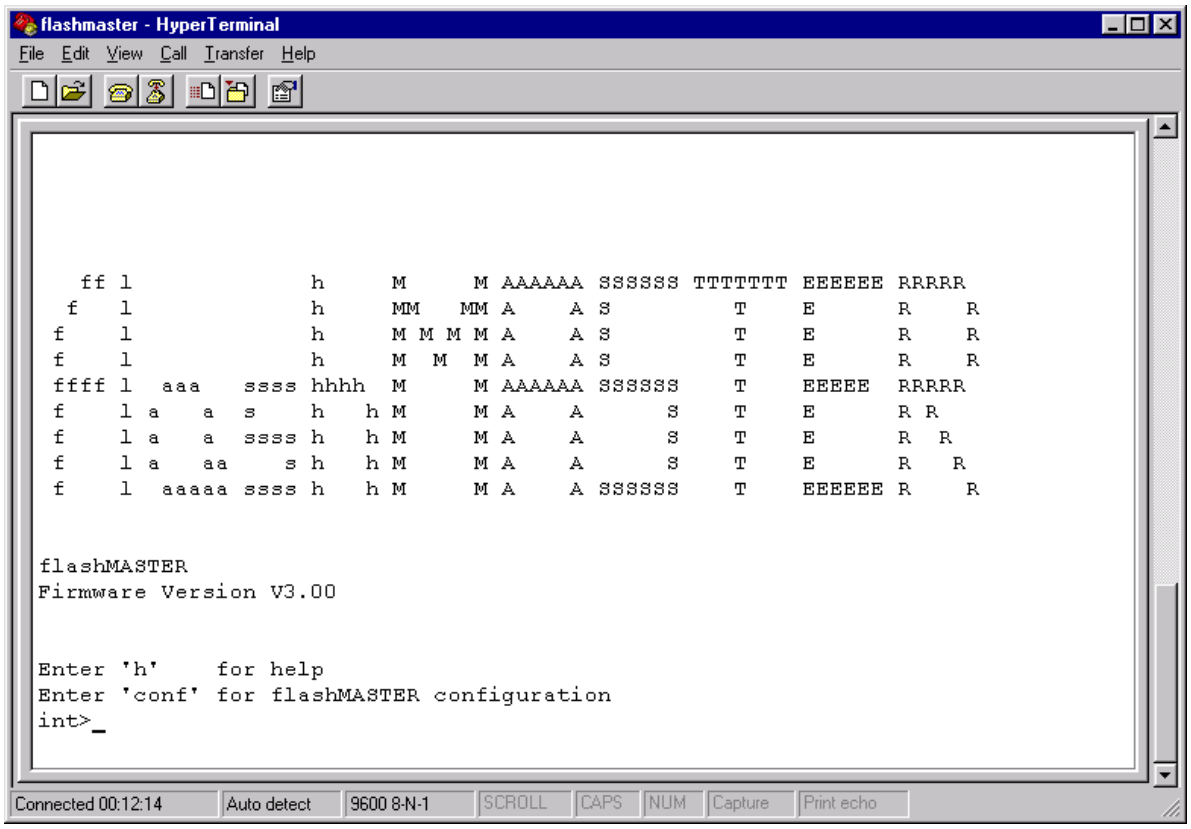


Figure 19: Initial screen after firmware upgrade

In case any error occurs during firmware upgrade, repeat the command until it returns successful.

Note: If there is an error during self programming, try again after pressing the RESET button at the *flash*MASTER box. If this does not help to restart *flash*MASTER, please get in contact with your NEC sales representative.

4. *flash*MASTER operation using GUI

4.1 Getting started

After unpacking *flash*MASTER please connect *flash*MASTER to your host computer using the provided serial interface cable and the 'Serial Host' connector. *flash*MASTER uses at startup the default baudrate, which is 9600 bps.
After plugging in the power supply, the 7-segment LED will show an 'r' indicating that the SRAM is used to store the download data.

4.2 Program Start

When being started, the following entry window appears. This window is also shown by the menu item *Help* → *About*.

Figure 20: Help - About Window

After several seconds or by clicking on the Micro picture the window disappears and the programs main window is displayed.

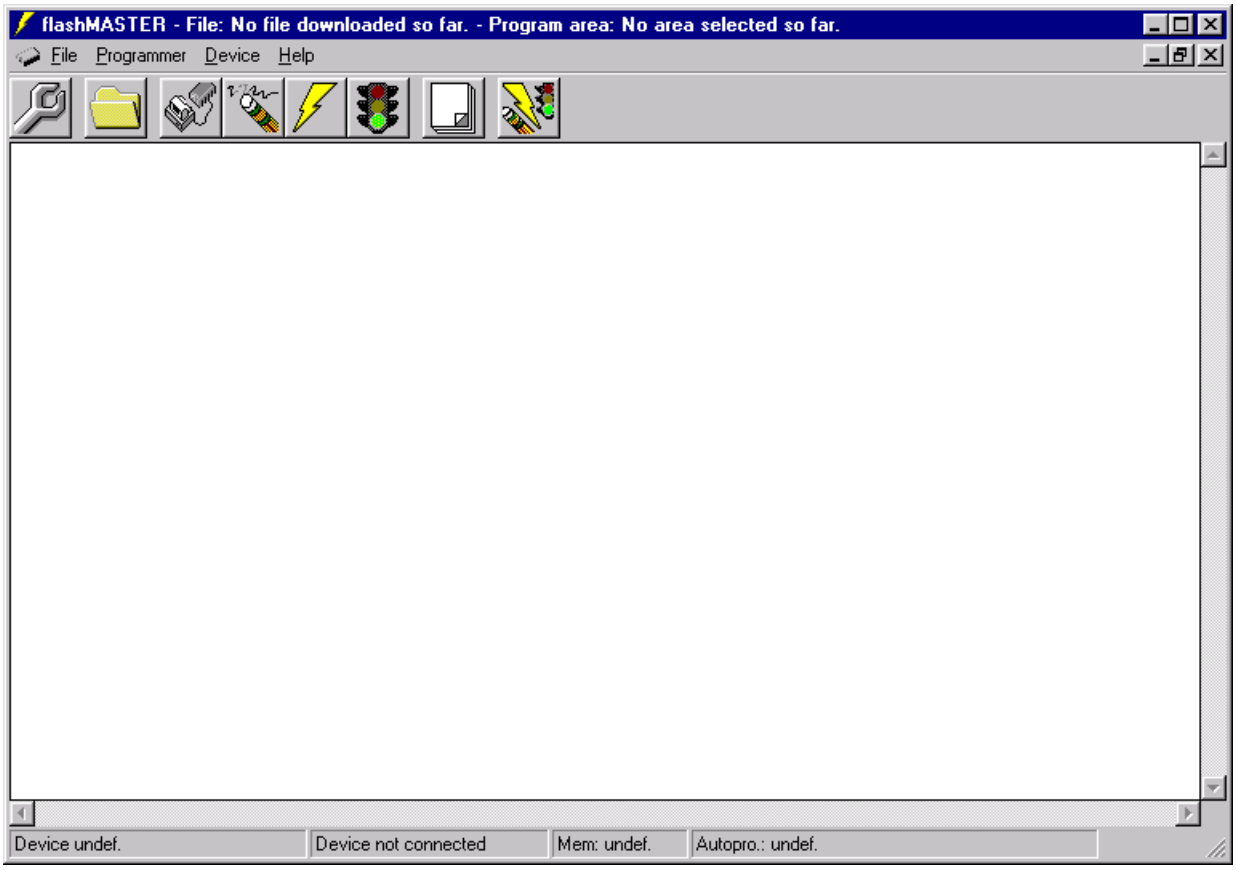


Figure 21: Main window with communication window opened

The programs main window consists of

- The menu (top of the window)
- The toolbar (menu below) with buttons for all most important program options.
- The status bar (bottom of the window) with device name, device status, memory selection information and information about the programming options.
- The title bar, which contains the name of the most recently downloaded file and the selected programming area if multiple areas are enabled.

Note: Version 3.00 of the *flash*MASTER GUI software requires firmware version 3.00 installed in the programmer. If firmware version 3.00 has not been installed yet you may update the firmware either by *flash*MASTER GUI software or by any terminal program. Please refer to the corresponding chapter(s) of this document to learn about the necessary steps of operation.

4.3 The Menu Items

Depending on the actual device status and depending on the setting of several keys in the initialisation file fMASTER.INI, some menu items may be enabled or disabled, i.e. the menu *Device* → *Blankcheck* is only enabled when a device is connected.

4.3.1 File Menu

The file menu allows to select and download a program file in various formats to *flash*MASTER so that it can be programmed into the device's flash memory. A checksum can be calculated over the downloaded program file and *flash*MASTER parameter settings can be dumped.

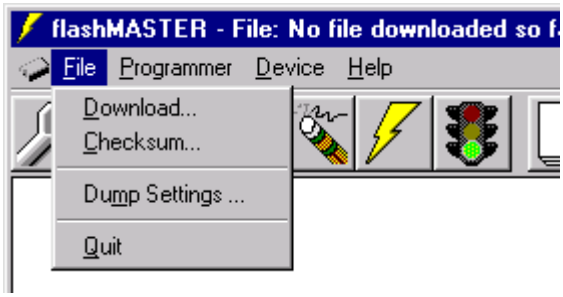


Figure 22: Menu item *File*

4.3.1.1 Download Menu

This menu item allows to select and download a program file into the *flash*MASTER memory. Before downloading a program file you may select whether to download into *flash*MASTER's internal SRAM or internal flash memory. Please refer to the section **Memory Selection**. After downloading the program file may be programmed into the device's flash memory.

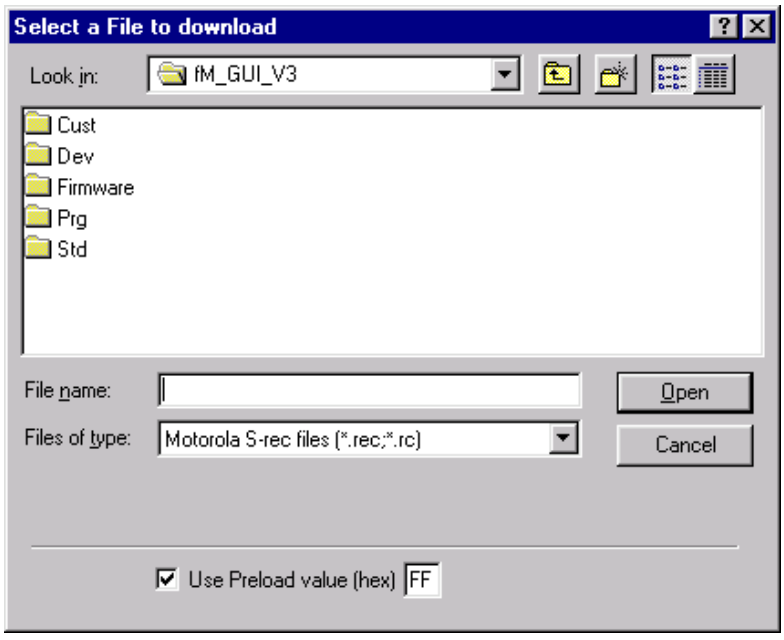


Figure 23: File selection window

In the appearing window you can

- Select a program file for download.
- Select the download file format (Intel HEX or Motorola S-Record format)
- Select a preload value in hexadecimal format to initialise unused memory locations.

The most recently used directory a file has been downloaded from will be offered in this download menu. The directory name will be saved in the key `FileDownloadDirectory` of the `[GUI]` section of fMASTER.INI file.
After program download a CRC will be calculated covering *flash*MASTERs complete flash / SRAM memory area and the CRC will be stored in the key `FileDownCrcSum` of the `[Programmer]` section of fMASTER.INI file. The CRC will be used to verify *flash*MASTERs memory contents before an autoprogramming is started, provided the key `SaveMode` of the `[GUI]` section is set.

The **Open** button starts the download procedure.
The **Cancel** button closes the window without downloading the program.

4.3.1.2 Checksum Menu

The menu item *Checksum* may be used to verify if *flash*MASTERs flash / SRAM memory area contains the correct download file.

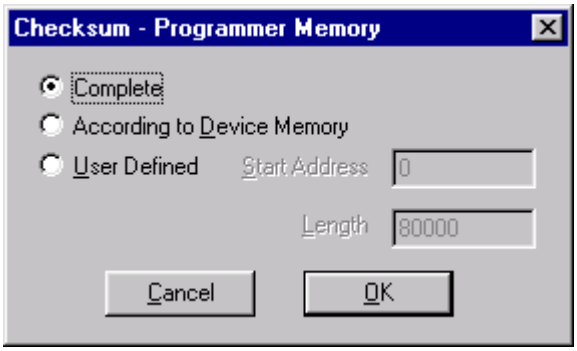


Figure 24: Checksum dialogue window

Select **Complete** if a 32 bit CRC shall be calculated covering the complete *flash*MASTER's programming memory area.
Select **According to Device Memory** if the target device's memory shall define the memory area for the CRC calculation. If the device has not been connected so far, connection will be established automatically to read the target device's memory area.
Select **User Defined** if a CRC shall be calculated over any other memory area. As soon as this option is enabled, the memory **Start Address** and memory **Length** need to be specified.

4.3.1.3 Dump Settings… Menu

The *Dump Settings…* menu displays all GUI settings as well as the *flash*MASTER configuration.

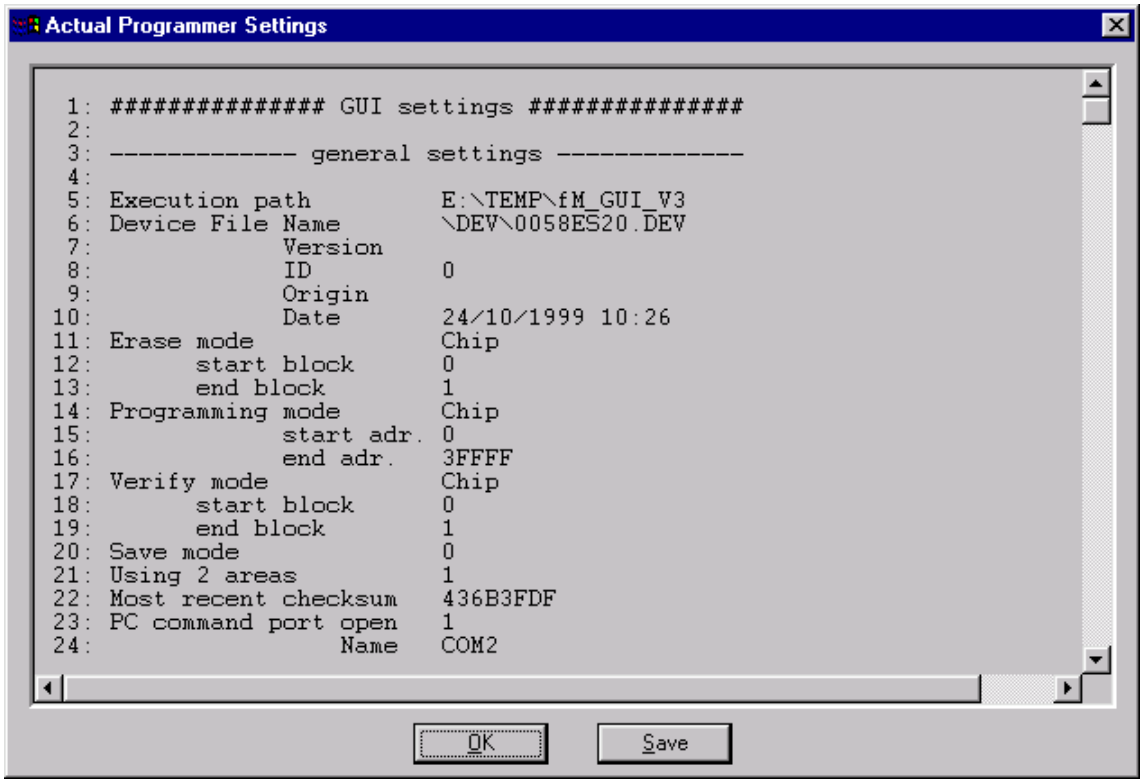


Figure 25: Dump settings dialogue

The **Save** button allows to save the window's contents to a file for later investigation.

The **OK** button closes the window.

4.3.1.4 Quit

The menu item *Quit* terminates the interface program and returns control to the operating system.

4.3.2 Programmer

In the menu item *Programmer* it is possible to select communication parameters and to open a serial protocol window to monitor the commands sent to *flash*MASTER. Memory selection (SRAM / flash memory) as well as multiple programming selections may be specified as well.

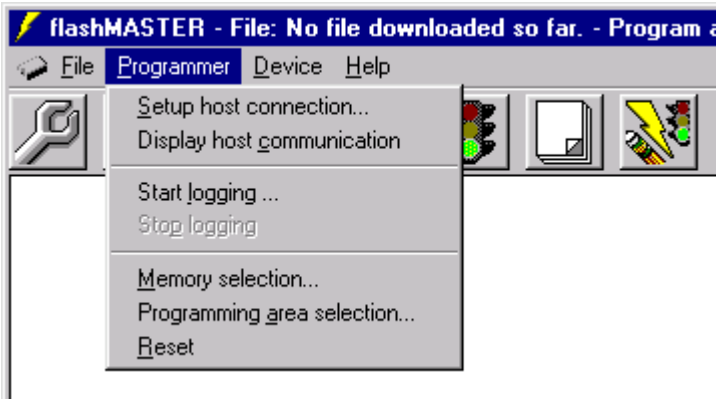


Figure 26: Menu item *Programmer*

4.3.2.1 Setup host connection

At startup the GUI automatically tries to connect to *flash*MASTER using the parameters `HostConnectionSpeed` and `HostConnectionPort` of the section `[GUI]` of fMASTER.INI.
If this connection cannot be established, the GUI tries to connect via COM1 … COM6, using 9600 baud, 19200 baud and 38400 baud on each port until it succeeds.

Additionally, manual selection of the comport settings for the connection between *flash*MASTER and the PC can be done in this dialogue:

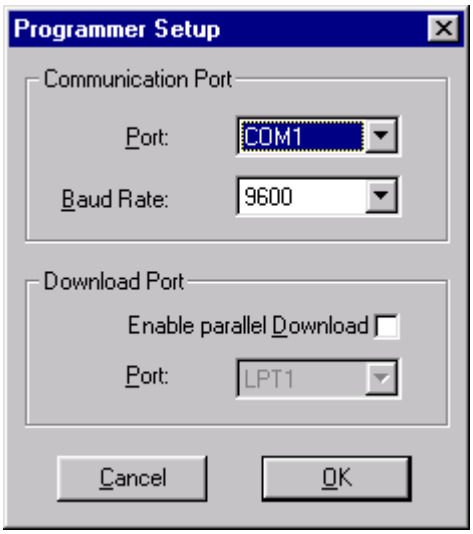


Figure 27: Setup host connection

You can select comport number and communication speed.
If **Enable parallel Download** is selected, data download to *flash*MASTER is performed via the selected parallel interface. Despite of this selection, command communication will still be done via the serial interface.

Note: The parallel interface cable is not part of the *flash*MASTER package!

Using the parallel port requires that the LPTx port is not captured by the operating system! For details, please refer to the documentation of your operating system.

Clicking the **Cancel** button closes the window without changes.
By clicking the **OK** button the software tries to establish a connection between *flash*MASTER and PC using the selected communication parameters.

4.3.2.2 Display host communication

Activating this menu item shows a window, which monitors the communication between the GUI software and *flash*MASTER. You may enter flash commands or host commands described in chapter 6 in this window. The window is shown in figure 22.
The communication between programmer and the PC is ASCII based. Lower case command input is required!

The communication window area may be used to copy and paste command sequences from / to other files for any purpose in addition to the logging feature of the GUI.

4.3.2.3 Start logging…

The *Start logging…* menu allows to create an ASCII log file which protocols the complete communication between *flash*MASTER and the PC.

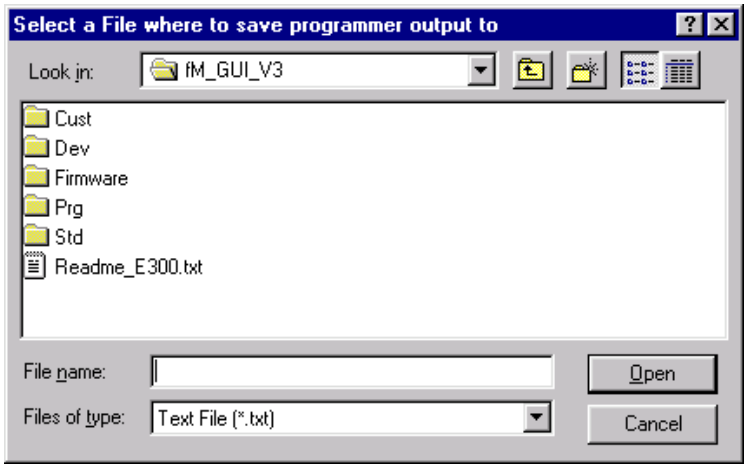


Figure 28: Start logging window

A file name needs to be specified to keep the communication protocol data.

4.3.2.4 Stop logging

The *Stop logging* menu stops any logging activity of the *flash*MASTER GUI.

4.3.2.5 Memory selection

This menu item opens a dialogue box where you can specify the destination memory of the *flash*MASTER. Program files will be downloaded into the *flash*MASTER's memory according to your selection.

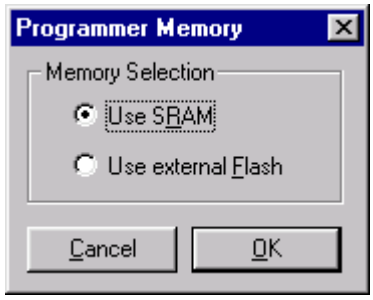


Figure 29: Programmer Memory selection

Select **Use SRAM** when SRAM memory shall be used to hold download files.
Select **Use external Flash** when *flash*MASTER's internal flash memory shall be used to hold download files.

4.3.2.6 Programming area selection

This menu item allows to select one out of two different programming areas of *flash*MASTER.
The *flash*MASTER internal memory area can be split into two independent memory areas which may hold two different application programs for flash programming.

Area 0: 0x00000 – 0x3FFFF
Area 1: 0x40000 – 0x7FFFF

This feature needs to be enabled by the key `UseMultipleProgramArea` of the `[Programmer]` section of fMASTER.INI file.

4.3.2.7 Reset

By clicking on this menu item, you can do a software reset to the programmer. The communication window will show the current version of *flash*MASTER firmware after the reset.

4.3.3 Device Menu

This menu item offers commands to control *flash*MASTER. Most items of this menu are only available if communication to *flash*MASTER is established!

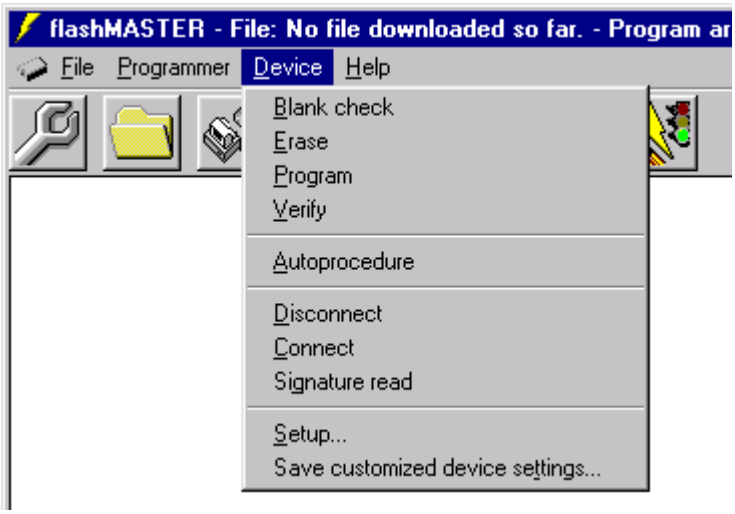


Figure 30: Menu item *Device*

4.3.3.1 Blank Check

This *Blank Check* command initiates the blank check. When it is finished, the result is displayed.

If the device is erased, the blank check is terminated successfully. Then a message window appears with the message OK.

Figure 31: Positive blank check result

If the chip is not or not completely erased the following error message is displayed. The chip has to be erased first before programming may start.

Figure 32: Possible negative blank check result (Example)

4.3.3.2 Erase

This menu item starts the erasing procedure.

As erasing needs some time (some seconds up to several minutes, depending on the device) a progress window is displayed, indicating the programmers activity.

Figure 33: Erase procedure progress window

Note: You may abort the erase procedure by the **Abort** button but additionally you have to press the Reset button on the programmer.

When the erase procedure is finished successfully an OK message window appears.

In case of errors, the concerning error is displayed in the message window.

4.3.3.3 Program

This menu item starts the programming procedure. The program, which has been downloaded to the programmer hardware, is programmed into the erased device.

As writing needs some time (some seconds up to several minutes, depending on the device) a progress window is displayed, indicating the programmers activity. The progress windows shows the percentages of the program already programmed into the device.

Note: You may abort the erase procedure by the **Abort** button in the progress window but additionally you have to press the Reset button on the programmer.

When the program procedure is finished successfully an ***OK*** message window appears.

In case of errors, the concerning error is displayed in the message window.

4.3.3.4 Verify

This menu item allows you to compare the program stored in the programmer with the program programmed in the device.

As verifying needs some time (some seconds up to several minutes, depending on the device) a progress window is displayed. In case the device needs to be verified completely, the percentages already verified are displayed. In case only several blocks of the device need to be verified, a stopwatch is displayed indicating the programmers activity.

Note: You may abort the erase procedure by the **Abort** button in the progress window but additionally you have to press the Reset button on the programmer.

When the verify procedure is finished successfully an ***OK*** message window appears.

In case of errors, the concerning error is displayed in a message window.

4.3.3.5 Autoprocedure

This menu item starts an automatic procedure which may be configured by the user. The commands **blank check**, **erase**, **program** and **verify** may be executed, depending on which of them have been defined in the **Device Setup** menu.
When the key `SaveMode` of section `[GUI]` of fMASTER.INI is set, the recently made device settings of the GUI are compared with the actual programmer settings before the epv command is executed. This includes comparing the programmer memory CRC with the most recently calculated CRC which has been saved before in fMASTER.INI file after downloading a file.
As the concerning steps need some time (some seconds up to several minutes, depending on the device) a progress window is displayed, indicating the programmers activity. Regarding the selected commands, their execution behavior and their possible messages, please refer to the corresponding chapter(s).

Note: You may abort the erase procedure by the **Abort** button in the progress window but additionally you have to press the Reset button on the programmer.

When all steps have been finished, a message window appears showing the result of each of the steps.

4.3.3.6 Disconnect

This menu item will disconnect the target device from the *flash*MASTER. This menu item should be executed before disconnecting *flash*MASTER and target hardware from each other.

4.3.3.7 Connect

This command will try to establish a connection between *flash*MASTER and target hardware. In case there is already a connection to the device, *flash*MASTER will first disconnect and then connect again to the device.
When you setup the device specific options and *flash*MASTER could not get a connection to the device with this settings, you will have to change the settings and reconnect.

4.3.3.8 Signature Read

By means of this command the device name and flash block structure will be displayed. This command may also be activated automatically by the host-PC, e.g. when changes of block settings are performed.

4.3.3.9 Setup

This menu item allows to setup all device specific options of the programmer. Each time the *Setup* dialogue is opened the actual settings of the programmer are read. After closing the Dialogue using the **OK** button all changed items will be written to the programmer. This behavior of reading and writing will be controlled by the `SaveMode` key of section `[GUI]` of the fMASTER.INI file. If `SaveMode = 1` is set, the GUI will write all settings to the programmer regardless if they have been changed or not.

4.3.3.9.1 Standard Setup

The following window is opened:

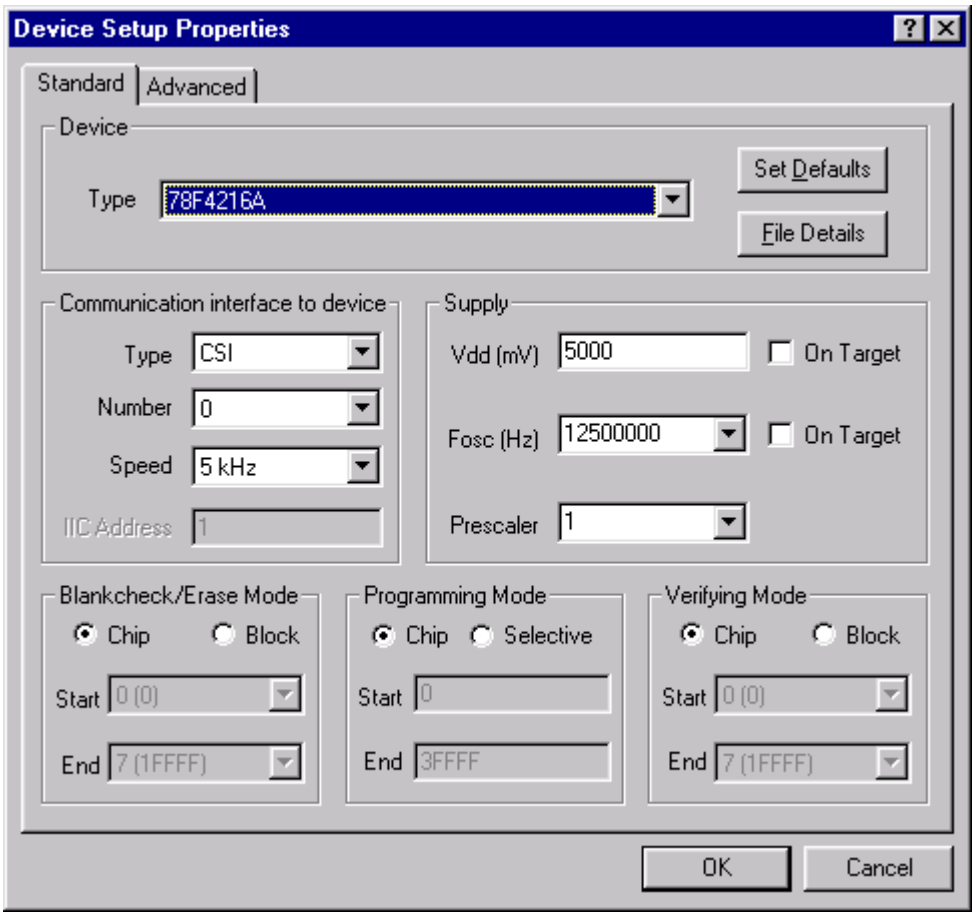


Figure 34: Device setup window – Standard view

This window shows all basic options which might be set depending on your target device. Also specific device options which are enabled/disabled in the device parameter file may force the Standard Setup window to look different in some specific areas. The following sections of the manual will highlight these areas.

4.3.3.9.1.1 *Setup Device*

The device to be programmed can be selected from the device list box. Standard device parameter files and customised parameter files will be offered for selection.
Customised device parameter files are those which have been modified according to specific needs and which may differ from the standard parameter files delivered by NEC.

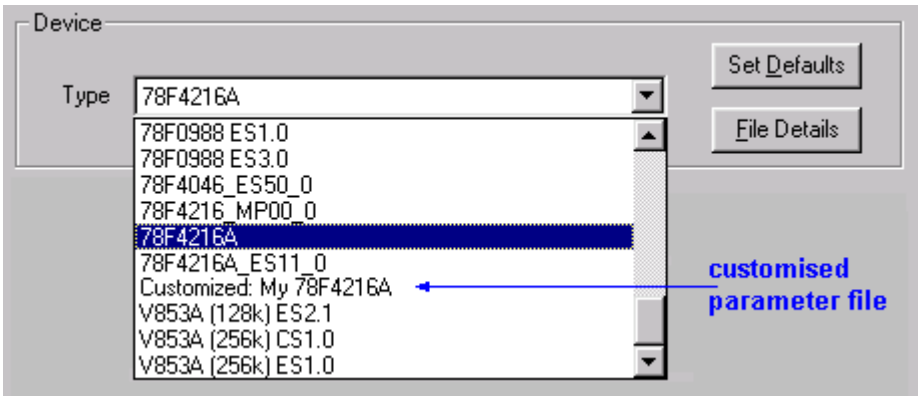


Figure 35: Setup window - Device type selection

After changing the device type selection all parameters will be read out of the device file. The **OK** button will activate the download to *flash*MASTER.

The **Set Default** button may be used to change all device parameters to the default setting as given in the device parameter file.
The **File Details** button will open a dialogue to inform about details of the device file selected:

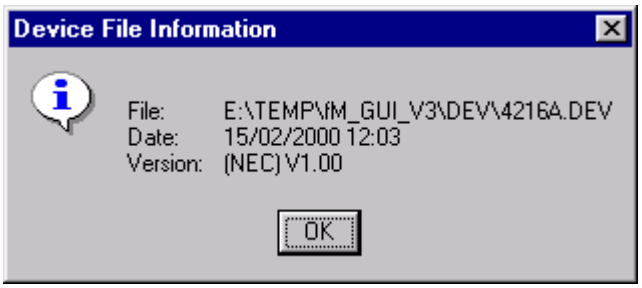


Figure 36: Device file information dialogue

4.3.3.9.1.2 *Setup of power supply Vdd*

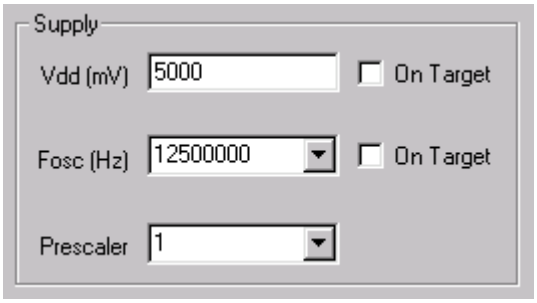


Figure 37: Setup window - Supply selection

You can select, whether the programmer shall supply Vdd or if Vdd is available on you target board. For this selection simple click the check box near to the Vdd selection.

4.3.3.9.1.3 *Setup of device clock frequency Fosc and prescaler*

You can select, whether the programmer shall supply the clock or if it is available on your target board. For this selection simple click the check box near to the frequency selection.
If you select that the clock shall be provided by the programmer, then the frequency is no longer free selectable but in steps. The text box changes to a drop down box.
However, frequencies below 1 MHz can only be entered via direct keyboard input.
With the prescaler selection you can set-up the device internal frequency division or multiplication (by PLL). Possible factors are ½, 1, 2, 4, 5 or 10, where 5 means device internal frequency multiplication by 5.
Example: f_{Quarz} = 5 MHz, prescaler selection 5 ⇨ f_{OPR} = 25 MHz.

4.3.3.9.1.4 *Setup of Blankcheck/Erase Mode, Programming Mode and Verifying Mode*

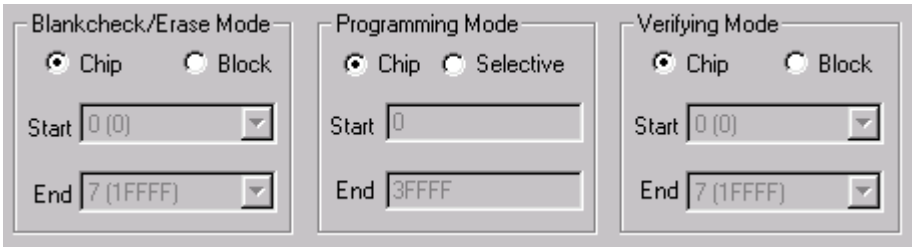


Figure 38: Setup window – BlankcheckErase Mode, Programming Mode and Verifying Mode 1

In most devices the flash memory is divided into several blocks or areas. If you select block or area mode, you can blankcheck, erase, program and verify single/multiple blocks/areas or all blocks/areas. Before this selection is available, connection between *flash*MASTER and the device must be established to receive the block/area structure which is filled into the drop down boxes **Start** and **End**.

When changing Start or End block the corresponding entry in the Program Mode will change accordingly.

The operation mode can be selected for blankcheck/erase, programming and verify separately.

In chip mode the block/area and address selection don't apply. If you select chip mode, the required operation affects the complete flash memory of the device.

For those devices where separate selection of Programming Mode / Verifying Mode cannot be made, the above shown part of the standard setup dialogue will look as follows:

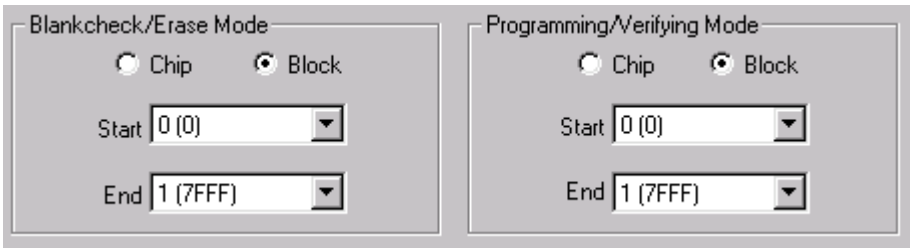


Figure 39: Setup window – BlankcheckErase Mode, Programming Mode and Verifying Mode 2

4.3.3.9.1.5 *Setup the communication interface to the device*

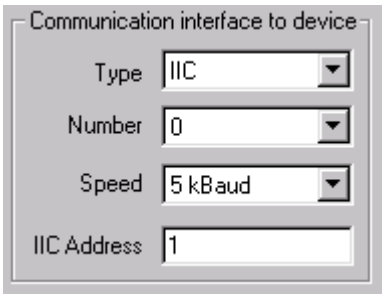


Figure 40: Setup window - Communication interface

Depending on the target device and the electrical connection between device and programmer you can select

- The communication interface to the flash device
  - CSI
  - CSI0 with handshake
  - UART
  - PORT
  - I^2C
- The number of the interface
- The speed of the communication via this interface
  - CSI: 5 KHz – 1000 KHz
  - UART: 4800 Bd – 76800 Bd
  - PORT: 100 Hz – 2000 Hz
  - I^2C: 5 KBd – 30 KBd
- I^2C address if I^2C communication interface is selected

4.3.3.9.2 Advanced Setup

By clicking on *Advanced* tab some advanced device specific options are displayed.

Most of the advanced options cannot be changed and are loaded with the device parameter file.

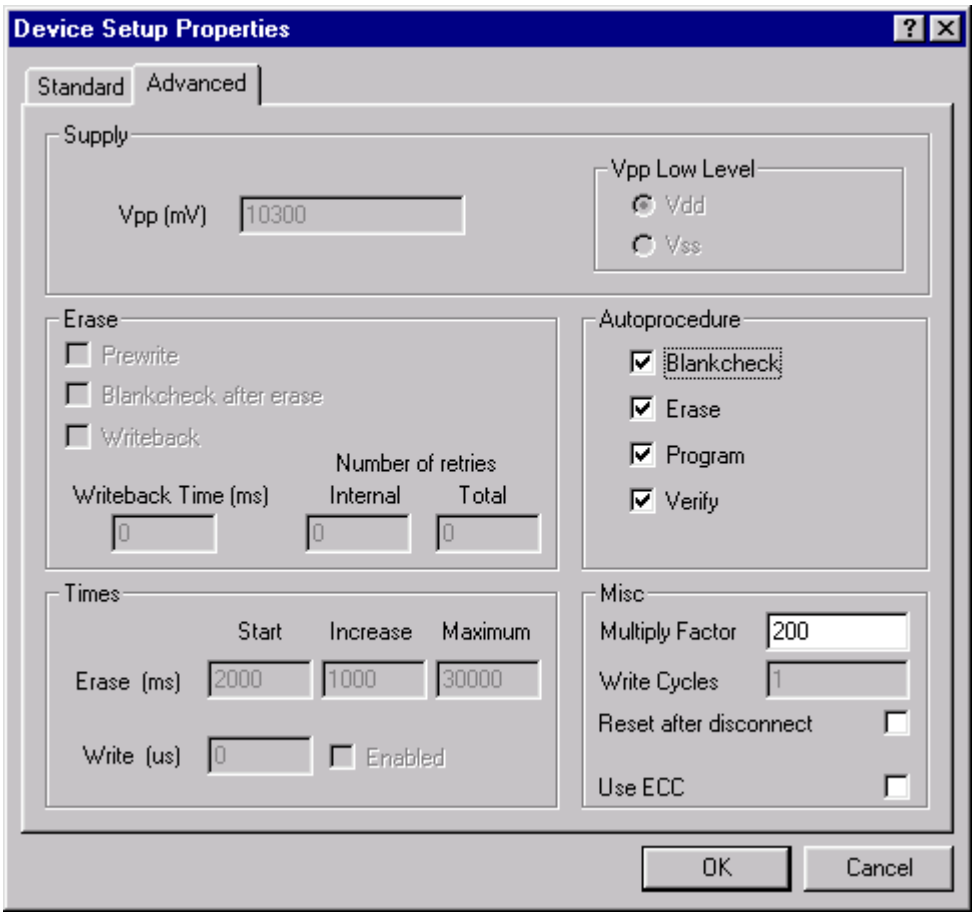


Figure 41: Device setup window – Advanced view

4.3.3.9.2.1 *Setup Autoprocedure options*

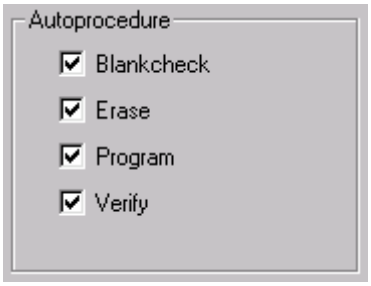


Figure 42: Setup window - Autoprocedure

The Autoprocedure checklists fix the command sequence to be executed during an epv command.

- Blankcheck
  - When checked, a blankcheck of the device is to be performed when executing the epv command.
- Erase
  - When checked, erasing the device is to be performed when executing the epv command. Regardless the block settings always the whole chip is erased.
- Program
  - When checked, programming the device is to be performed while executing the epv command. Regardless the block settings always the whole chip is programmed.
- Verify
  - When checked, erasing the device is to be performed when executing the epv command. Regardless the block settings always the whole chip is verified.

4.3.3.9.2.2 *Setup misc*

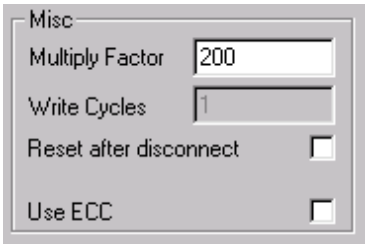


Figure 43: Setup window - Misc

The **Multiply Factor** is determined by the performance of the device. It can be changed by typing in a value in the text box.
The **Reset after disconnect** may be set to release the reset signal after disconnecting the device from *flash*MASTER.
The **Use ECC** may be set to use the ECC code area. This field is available only if the device supports ECC!

4.3.3.10 Save customized device settings… menu

If the key `CustomerDeviceFileCreation` key of the section `[GUI]` of fMASTER.INI is set and changes have been made in the Setup dialogue, the *Save customized device settings…* menu item becomes enabled and allows to save these settings in a new device parameter file in the subdirectory \cust.
A dialogue opens to enter the file name and any other directory:

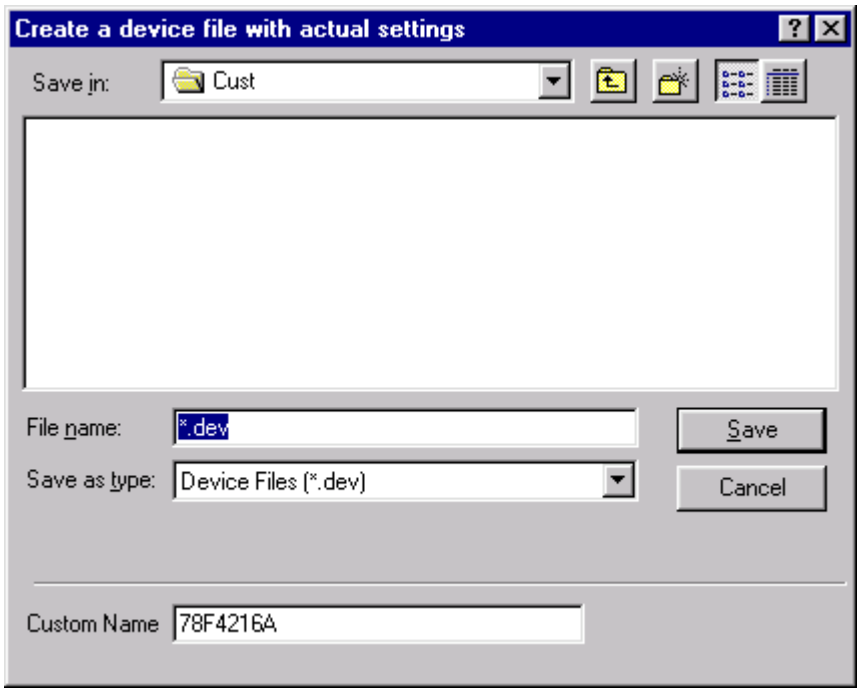


Figure 44: Create customized device parameter file

Additionally to file name a **Custom Name** can be entered which may consist of up to 20 characters. The Custom name will be displayed in the parameter file selection box with the preceeding “Custom:” indicator to distinguish the customized device parameter file from those delivered by NEC.
Only customised device file having the extension `*.dev` and located in the subdirectory `cust` will be displayed during device parameter file selection.

4.3.4 Help Menu

The help menu item displays the about box only.

Figure 45: Menu item *Help*

4.3.4.1 About

This opens the program entry window (Chapter *Program Start*).
It is displayed until you click on the Micro picture.

4.4 The Toolbar

The toolbar contains the most important procedures of the *flash*MASTER.
You can activate all these procedures by the menu but the common used items can be activated faster by this tool bar.


Menu item *Device* → *setup*

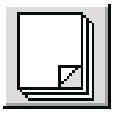

Menu item *File* → *Download*


Menu item *Device* → *Connect* / *Disconnect* button


Menu item *Device* → *Erase*


Menu item *Device* → *Program*


Menu item *Device* → *Verify*


Menu item *Device* → *Blank check*


Menu item *Device Erase* → *Program* → *Verify (Autoprocedure)*

4.5 *flash*MASTER initialisation file

*flash*MASTER can be customised by specific settings in an initialisation file. The initialisation file fMASTER.INI is located in the same directory as FLASHMASTER.EXE itself.

The initialisation file consists of several sections and keys in these sections. In case a key is not available in the corresponding section the default setting for that key will be assumed.

*flash*MASTER GUI will maintain most of the keys of fMASTER.INI to keep actual settings and start up with the same settings as valid during a previous session.
Only those keys which are marked (*) will not be written by the GUI itself.

Default key setting(s) are marked as bold.

4.5.1 Section [GUI]

Key name	Value	Description
`StartWithMaximizedMainWindow`	0	At startup of flashmaster, the main window will open with default window size.
	1	At startup of flashmaster, the main window will open with maximised window size.
`StartWithCommunicationWindow`	0	At startup of flashmaster, the communication window will not be opened.
	1	At startup of flashmaster, the communication window will be opened.
`CustomerDeviceFileGeneration (*)`	1	The menu item **Device → Save device settings** will be enabled to save device parameters in a specific parameter file.
	0	The menu item **Device → Save device settings** will not be disabled.
`SaveMode=0 (*)`	**1**	**Device Setup:** Send all settings from GUI to the programmer. **Autoprogramming:** Compare all actual *flash*MASTER settings and verify checksum of most recent downloaded file before epv.
	0	**Device Setup:** Send settings from GUI to the programmer only for changed settings. **Autoprogramming:** Send epv command only.
`HostConnectionPort`	**COM1**	Specifies the communication port.
`HostConnectionSpeed`	**9600**	Specifies the communication speed. Other settings may be 19200 or 38400.

4.5.2 Section [Device]

Key name	Value	Description
DevFile	<value>	Specifies the most recently used device parameter file name. If no file name is given, *flash*MASTER will use the first file found in the subdirectory \DEV.
EraseMode	**Chip**	Specifies the way of erasing the device in use. Chip specifies the complete device.
	AreaBlock	Specifies the way of erasing the device in use. AreaBlock specifies the address range given in the **Device → Setup** dialog..
ProgrammingMode	**Chip**	Specifies the way of programming the device in use. Chip specifies the complete device.
	AreaBlock	Specifies the way of programming the device in use. AreaBlock specifies the address range given in the **Device → Setup** dialog..
VerifyMode	**Chip**	Specifies the way of verifying the device in use. Chip specifies the complete device.
	AreaBlock	Specifies the way of verifying the device in use. AreaBlock specifies the address range given in the **Device → Setup** dialog..
ProgrammingStartAddress	<value>	Most recently used program start address given in hexdecimal notation. Default is **0**.
ProgrammingEndAddress	<value>	Most recently used program end address given in hexdecimal notation. Default is the last possible flash address.
BlankcheckEraseStartBlockAreaNumber	<value>	Most recently used block/area number for erase start of the device. Default is **0**.
BlankcheckEraseEndBlockAreaNumber	<value>	Most recently used block/area number for erase end of the device. Default is the last possible flash block/area.
VerifyStartBlockAreaNumber	<value>	Most recently used block/area number for verify start of the device. Default is **0**.
VerifyEndBlockAreaNumber	<value>	Most recently used block/area number for verify end of the device. Default is the last possible flash block/area.

4.5.3 Section [Programmer]

Key name	Value	Description
`UseMultipleProgramAreas (*)`	1	Enable/disable the usage of multiple program areas and display the menu item **Programmer → Programming area selection ...**.
	0	Single programming area only.
`FileDownloadDirectory`	<value>	Specifies the most recently used directory program files have been downloaded from. Default directory is the directory `FLASHMASTER.EXE` is located in.
`FileDownCrcSum`	<value>	Saves the most recently used CRC as hexadecimal value of the complete memory after user or prg-file download. In case of errors **Failed** will be written into this key.

5. Sample programming session using GUI software

As an example, a 78F4126A microcontroller will be used to show a typical programming sequence using *flash*MASTER GUI software.

1. Use the provided serial cable to connect flashMASTER to your host computer

2. Choose the target cable 1 from your *flash*MASTER package and connect *flash*MASTER to the target hardware. This may either be a NEC programming adapter or your own target hardware.

3. Select the suitable AC-plug (EURO, UK or US/JAPAN) and connect it to the AC adapter.

4. Connect the AC adapter to *flash*MASTER's power connection and mains.

5. If not yet done, install the *flash*MASTER GUI software on your PC as described in the chapter "Software Installation".

6. Start *flash*MASTER GUI software.

7. Select the appropriate communication speed you want *flash*MASTER to operate at. Communication speed shall be set to 9600 bps.

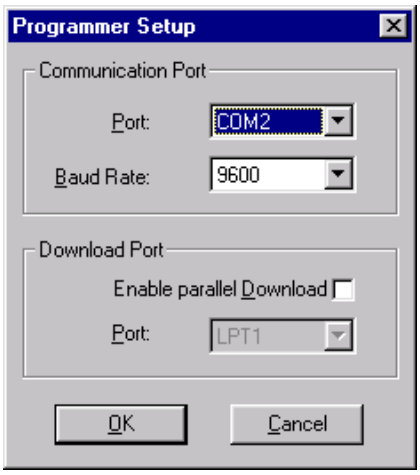
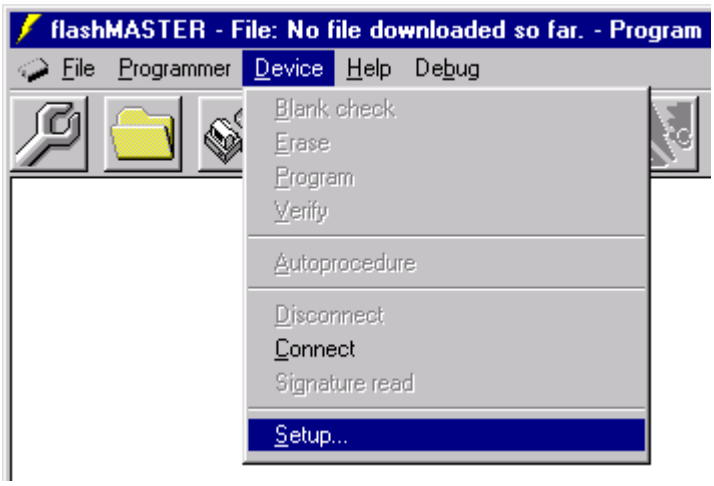


Figure 46: Setup communication parameters

8. Select **OK** to activate the port settings.

9. Select the menu item **Device→Setup**



Toolbar:

Figure 47: Device setup menu

10. The Dialogue for device setup will be activated.

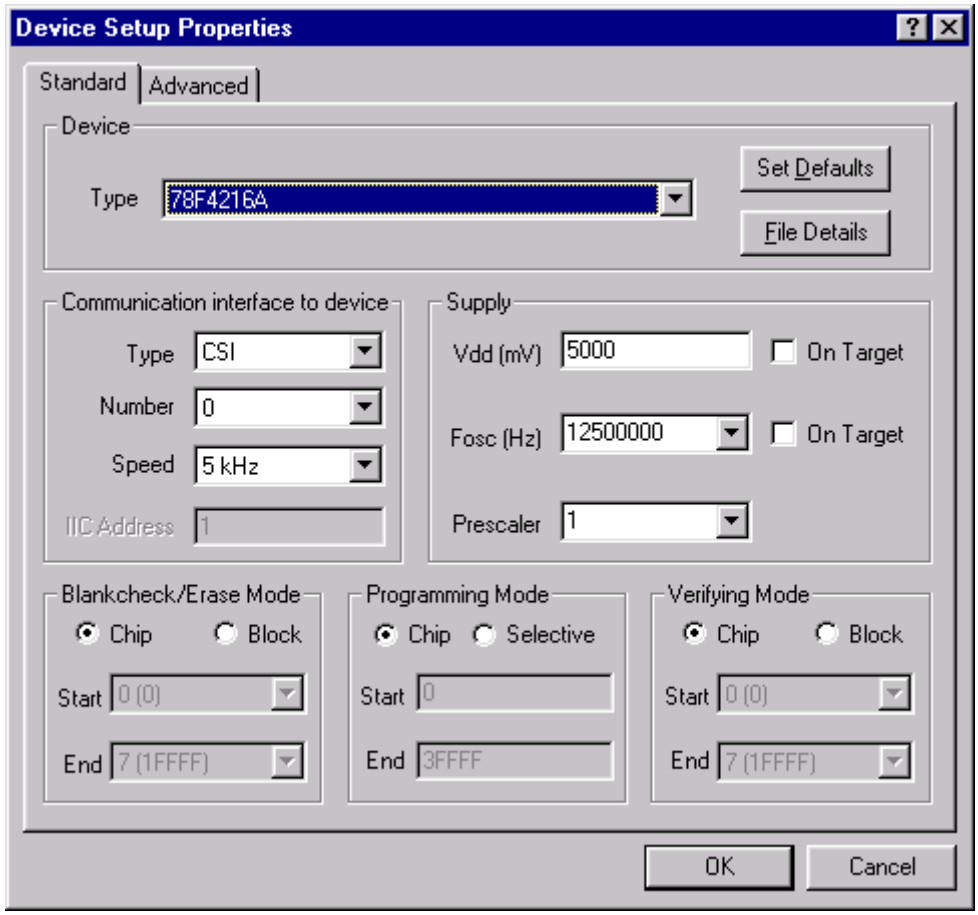


Figure 48: Device Setup Dialogue

11. Select 78F4216A in the **TYPE** drop down box.

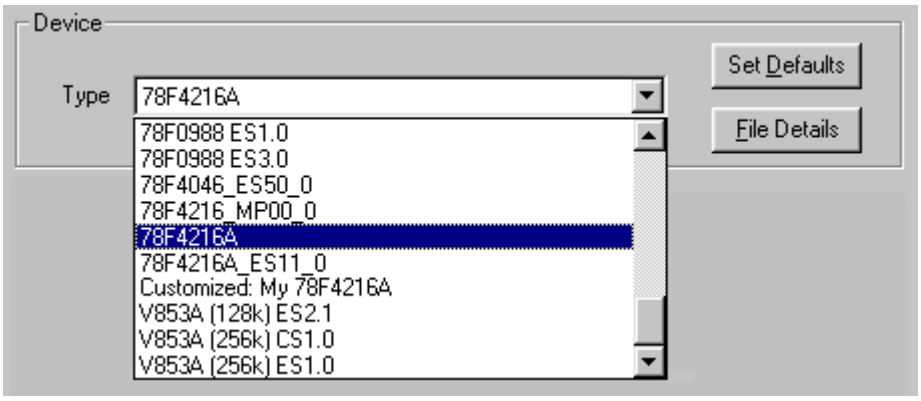


Figure 49: Device type selection

12. After device selection *flash*MASTER will select the default settings for this new device.

13. Verify that all settings shown in this dialogue correspond to the needs of your target hardware.

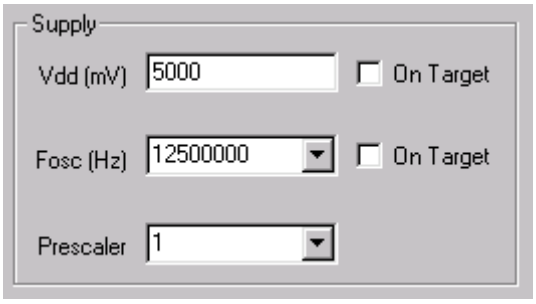


Figure 50: Device supply data

Especially V_{DD} and F_{OSC} must be set to the correct values.

14. Switch to the Advanced dialogue.

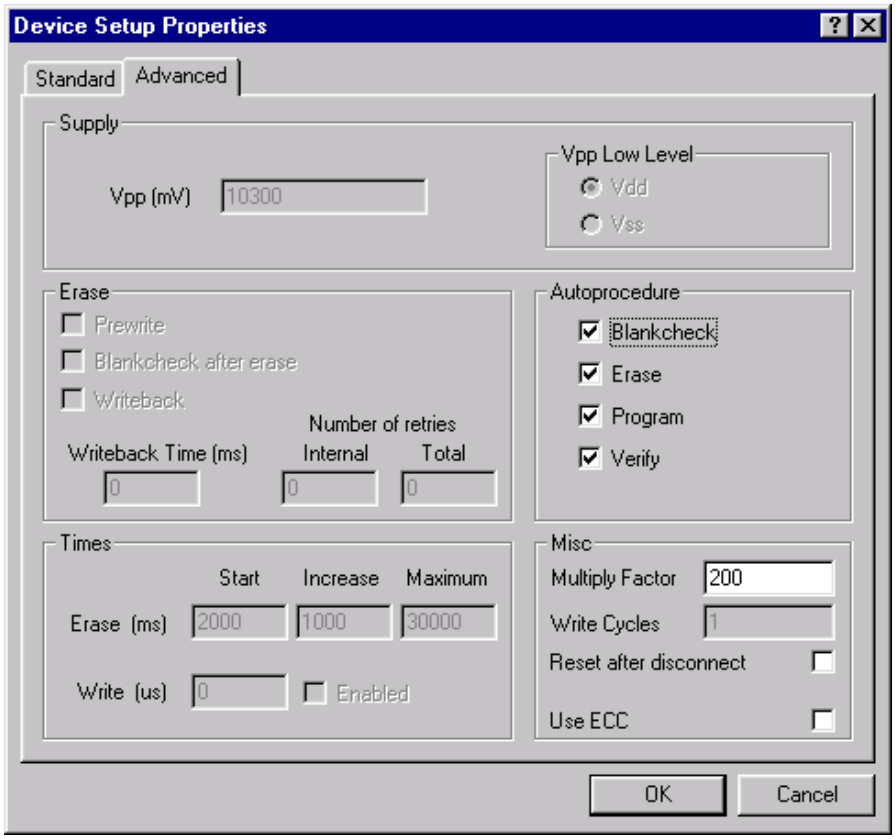


Figure 51: Advanced device setup dialogue

15. Mark Blankcheck, Erase, Program and Verify as required actions for the autoprocedure:

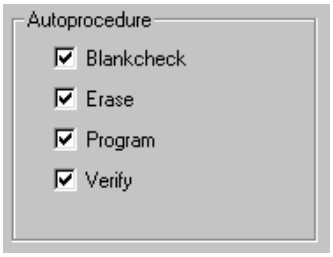


Figure 52: Erase/Program/Verify selection

16. Click **OK** to activate all settings of the device setup.

17. Activate the **Programmer→Memory** menu item.

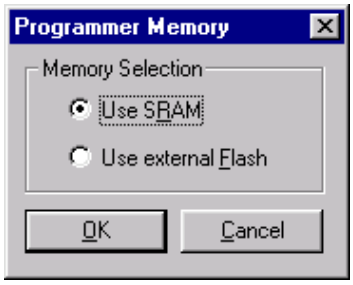
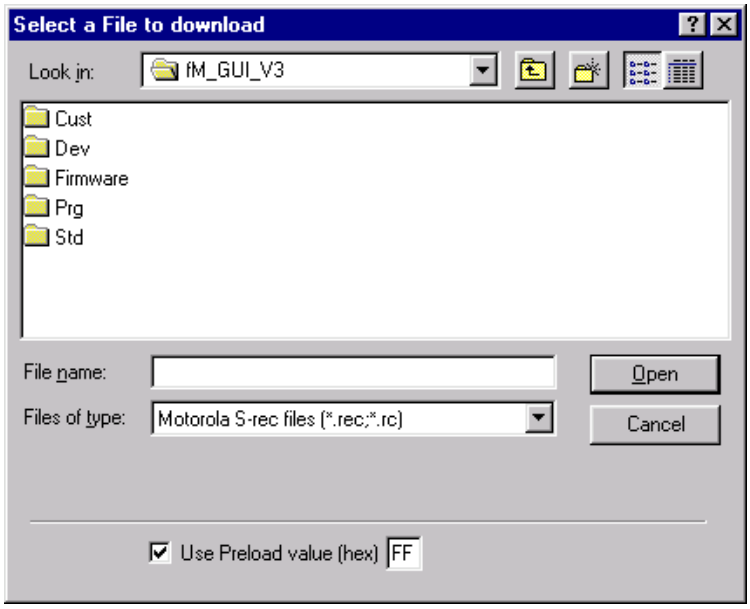


Figure 53: Programmer memory selection

18. Select the destination memory of *flash*MASTER where you would like to download your program files to and click **OK** to activate the selection.

19. Select the menu item **File→Download**

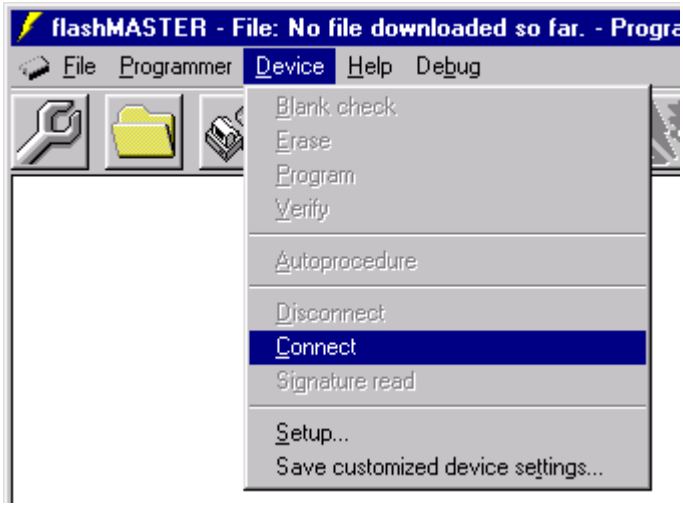


Toolbar: 

Figure 54: Open a program file for download

20. Select the file type (Intel-HEX or Motorola S) and the file name you would like to download. Click **OK** to start the download sequence. During program download a progress window will show the progress of downloading.

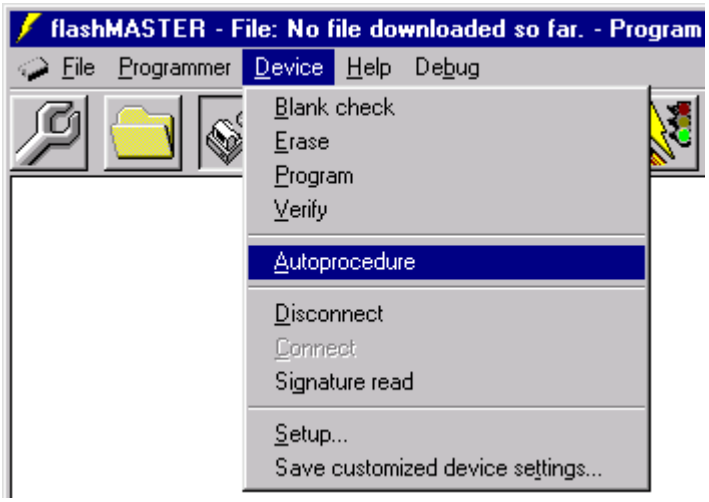
21. Select the **Device→Connect** menu item to connect flashMASTER to the device.



Toolbar: 

Figure 55: Connect *flash*MASTER to target device

22. Select the **Device→Autoprocedure** menu item.

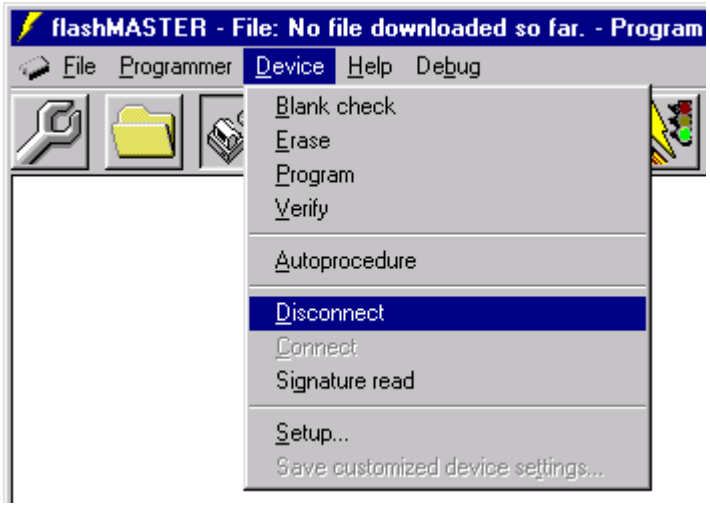


Toolbar: 

Figure 56: Erase→Program→Verify menu

The selections made in 16. will now be activated and the 78F4216A will be blankchecked, programmed and verified.

23. Select the **Device→Disconnect** menu item.



Toolbar: 

Figure 57: Device→Disconnect menu

Your target hardware may now be disconnected from *flash*MASTER.

24. If necessary, connect a new device to be programmed to *flash*MASTER and repeat from 22.

25. If no other device needs to be programmed exit *flash*MASTER GUI. All settings made during this programming session will be saved so that they can be reused GUI software is started up next time. The file fMASTER.INI will keep the window layout as well as the communication settings. All target device settings are saved in an EEPROM inside *flash*MASTER. They will be reloaded when connecting next time to *flash*MASTER.

6. *flash*MASTER operation using terminal communication

After unpacking *flash*MASTER please connect *flash*MASTER to your host computer using the provided serial interface cable and the 'Serial Host' connector.
Start communication with *flash*MASTER using 9600 bps, 8 data bits, 1 stop bit no parity and no handshake. Once communication is established you may switch to 19200 bps or 38400 bps. Make sure that communication is done in lower case letters only.

After plugging in the power supply the host screen will show the standard output of *flash*MASTER showing the firmware version number. The 7-segment LED will show an 'r' indicating that the SRAM is used to store the download data.

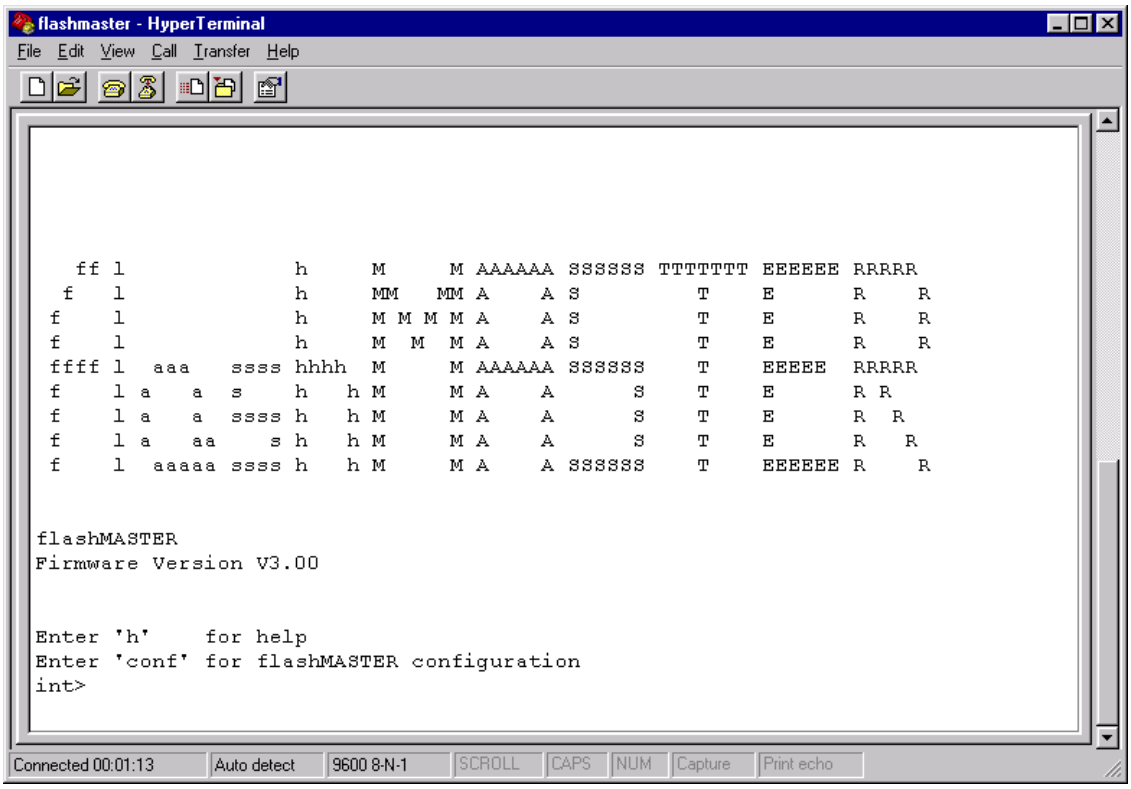


Figure 58: Initial information screen of *flash*MASTER

Following steps are necessary to connect a device to *flash*MASTER:

- The device has to be connected to *flash*MASTER using one of the provided target cables. The function of both cables is the same, only the pin layout is different and provides compatibility with previous NEC programming tools. For details of the target cable layout please refer to the chapter "Connectors and Cables" of this document.
- Load the device default settings with the `loaddev` command.
- V_{DD} have to be set to the correct values using the 'set command'. V_{DD} has to be 0 if the target board has its own power supply. Important: the entered values for the voltage are mV (10300 for 10.3 V V_{PP}).
- Please check the oscillator frequency, the multiply factor and if *flash*MASTER has to provide the clock signal.
- Check the used device interface (*flash*MASTER will use the CSI interface by default)
- Check the communication speed to the device
- Use the 'Connect Command' to connect to the device.
- If the connection has been established you will find the device name on the screen. If anything went wrong an error message will be output.

In order to use the parallel download feature, enter `set centronics 1` and use the `ls` or `li` command. As soon as *flash*MASTER responds with the "*now loading*" message, open a DOS command shell and enter:

`COPY <file name> LPT`*n* ↵ *n*: parallel port number

The data will be sent via parallel interface and *flash*MASTER will captured on the centronics port. Be sure that the parallel port you use is not captured by the operating system.

*flash*MASTER will store all parameters and continue to use those.

6.1 Stand-alone Mode

*flash*MASTER provides all necessary features to allow a stand-alone programming mode:
- On-board flash memory to hold the downloaded data.
- Start button (pressing this button executes the 'epv' command sequence.
- Parallel port input to start the 'epv' command via TTL signal.
- Two 7-segment LED to show the status.

6.2.1 Setting up the stand-alone mode

Before the stand-alone mode can be used, a couple of simple steps have to be performed:
- A `*.STD` file according to the device selection needs to be downloaded to *flash*MASTER.
- All device parameters (see device parameter settings) have to be set to the proper values. Those values will be stored in the EEPROM and used during the following operations. The parameters to be set include the following
    - V_{DD} (0 for external supply)
    - Target device port and speed
    - Target clock supply
    - 'epv command' options
- Set the memory mode to Flash 'set mem f'
- Download the data to be written to the device (see `ls / li` command).

After those steps the host connection is not necessary any more. The 'Start' button or the parallel port I/O input can now start a complete programming of the Flash device.

6.2 Flash commands supported by *flash*MASTER

6.2.1 Blank Check Command

This command verifies if the flash memory of the device is erased.

Command	b ↵	The whole flash will be checked.
	b a ↵	The whole flash will be checked block wise.
	b *block number* ↵	The specified block will be checked.

Status	**Screen Output**	**LED**
On success	`Blank Check Block` *block number* : OK	
On error	`Blank Check Block` *block number* : ERROR	

There is no progress information returned from *flash*MASTER during blank check operation. Please wait until *flash*MASTER send the prompt character before you enter another command.

6.2.2 Connect Command

This command connects *flash*MASTER to the device. Before connecting *flash*MASTER to the target device, the corresponding `*.STD` file needs to be downloaded first using the `loaddev` command. The `*.STD` files contains necessary device settings for flash programming.
*flash*MASTER checks during the connect command if the correct `*.STD` file has been downloaded and refuses connection if not.
Also V_{DD} (if selected) and V_{PP} are applied to the device during connect command execution. Please check V_{DD} and V_{PP} before using the connect command.

Command c ↵

Status	**Screen Output**	**LED**
On success	The initial screen is reprinted including the information from the device connected. Those are the block structure and the device name. The displayed prompt is changed in a way that it includes the device name (if available the nickname).	
On error	If the device could not be connected an error message will be displayed.	

6.2.3 Disconnect Command

This command disconnects from the device. Vdd and Vpp are switched off.

Command dc ↵

Status **Screen Output**

The prompt is changed back to its default appearance.

6.2.4 Erase Command

This command erases the flash. The specified times (see Set Erase Time Command) are used. The erase will always start with a prewrite, if this is selected (see 'set prewrite' command).

Command e a ↵ The whole flash will be erased.

e *block number* ↵ The specified block will be erased.

Status	**Screen Output**	**LED**
During operation		
On success	The status of the erase is output.	
On error	The status of the erase is output.	

6.2.5 Erase-Program-Verify Command

By setting the EPV options (see 'set progoptions xxxx') the user can decide on what operations should be performed during the epv command execution. All operations include the whole chip, it is not possible to erase blocks or write to specific addresses.
The following operations can be included:
b: blank check before erase
e: erase (prewrite is defined by prewrite settings)
p: program
v: verify

Command epv ↵ Executes the e-p-v sequence.

The output of this command sequence is the same as for the single functions.

Status **Screen Output**

Same as for single functions.

6.2.6 Silicon Signature Command

This command reads the silicon signature of the device.

Command sg ↵

Status	**Screen Output**
	Silicon Signature, including device name.

6.2.7 Verify Command

Verifies the contents of the flash. In case of no parameter is passed, the complete flash is verified.

Command	v ↵	The complete flash will be verified.
	v a ↵	The complete flash will be verified block wise.
	v *block number* ↵	The requested block will be verified.

Status	**Screen Output**	**LED**
During operation	During operation the progress of verification is shown as percentage.	
On success		
On error		

6.2.8 Write Command

This command writes the data stored in the *flash*MASTER buffer to the device flash.

Command	w ↵	The complete flash will be written.
	w *startaddr number* ↵	Write *number* of bytes from *startaddr* to flash.

Status	**Screen Output**	**LED**
During operation	During operation the progress of verification is shown as percentage. If an error occurs this will be displayed in the terminal window.	
On success		
On error		

6.3 Host Commands

This chapter describes the host commands which are supported by *flash*MASTER.

6.3.1 Checksum Command

This command calculates are CRC32 checksum of the *flash*MASTER memory using the specified address range.

Command	Description	Screen Output
checksum ↵	Calculate CRC covering the whole flash memory as per device selection.	`CRC Checksum xxx-yyy: cccc`
checksum startaddress length ↵	Calculate CRC from *startaddress*, *length* bytes.	`CRC Checksum xxx-yyy: cccc`

6.3.2 Configuration Command

This command shows all the settings of *flash*MASTER.

Command	conf ↵	Displays the current configuration of *flash*MASTER.

6.3.3 Dump Memory Command

This command dumps the memory contents of the *flash*MASTER buffer to screen.

Command	d *address length* ↵	Dumps *length* memory bytes from *address* to screen
	d *address* ↵	If *length* is omitted, memory dump continues to display a block of the internal buffer memory.
	d ↵	If *address* and *length* are omitted, memory dump continues to display a block of the internal buffer memory, starting from the end location of the previous memory dump command.

Status	Screen Output
	Shows the memory contents.

6.3.4 Fill Memory Command

This command is used to alter the contents of the *flash*MASTER buffer manually. This command should not be used when the on-board flash is selected.

Command	f *address length data* ↵	Fills *length* memory bytes from *address* with *data* byte
	f a ↵	Fills the complete memory with 0xFF.

6.3.5 Loaddev Command

This command is used to download device specific data to *flash*MASTER and initialise its EEPROM. For each flash device which can be programmed with *flash*MASTER a separate initialisation file is available in the subdirectory `\std` of disk 2 of the software package enclosed to the *flash*MASTER package.
It is necessary in all cases to download the correct `*.STD` file before executing any flash command. The `*.STD` files contain target specific settings for *flash*MASTER which are essential for correct flash programming operation.
The data file `*.STD` may be downloaded using the *Send Text File* menu of the terminal program.

Command loaddev ↵ Download device specific data to *flash*MASTER.

Status	**Screen Output**	**LED**
	. During download.	

In order to activate the new settings, either reset command or RESET button at flashMASTER must be activated.

Beside internal flashMASTER intialisation, the device specific data affected by downloading the `*.STD` file are:

Initialised item:	**Detailed command description given in section:**
prewrite flag	`set prewrite`
blankcheck flag	`set blankcheck`
writeback flag	`set writeback`
supply voltage V_{DD}	`set vdd`
programming voltage V_{PP}	`set vpp`
level of V_{PP} pulses	`set pulselevel`
device erasetime	`set erasetime`
device writetime	`set writetime`
number of writecycles	`set writecycles`
writeback time	`set writebacktime`
writeback retry numbers	`set wbretries`
device waitfactor	`set waitfactor`
device prescale factor	`set prescale`

6.3.6 Help Command

This command is used to inform about all available terminal commands and their options.

Command h ↵ Displays all available functions with the necessary parameter (indicated by <>) and optional parameter (indicated by ())

6.3.7 Help Command for set command

This command is used to inform about all available set commands and their options.

Command helpset ↵ Displays help on all available set commands.

6.3.8 Init Command

This command initialises the *flash*MASTER. This command is used only if the *flash*MASTER is programmed for the first time. All used EEPROM parameters are set to the default values.

Command init ↵

Output The initial screen is printed using the new values.

6.3.9 Load to Memory Command

Loads data from a Motorola S Format file to the *flash*MASTER buffer. This can either be the SRAM or the on-board flash depending on the selection switch. It is possible to load the data via the serial connection or the parallel interface. The reception of the data using the parallel interface can either be started by sending the command via the serial or the parallel connection.
When on-board flash is selected, the flash is automatically erased and rewritten.

Command ls ↵ Loads data from a Motorola S Format file to the *flash*MASTER buffer.

ls *xx* ↵ Loads data from a Motorola S Format file to the *flash*MASTER buffer. Before loading, memory will be filled with hexvalue *xx*.

Status	**Screen Output**	**LED**
	. During download.	

6.3.10 Load to Memory Command (INTEL Hex Format)

Loads data from an INTEL HEX Format file to the *flash*MASTER buffer. This can either be the SRAM or the on-board flash depending on the selection switch. It is possible to load the data via the serial connection or the parallel interface. The reception of the data using the parallel interface can either be started by sending the command via the serial or the parallel connection.
When on-board flash is selected, the flash is automatically erased and rewritten.

Command	li ↵	Loads data from an INTEL HEX file to the *flash*MASTER buffer.
	li *xx* ↵	Loads data from an INTEL HEX file to the *flash*MASTER buffer. Before loading, memory will be filled with hexvalue *xx*.

Status	**Screen Output**	**LED**
	. During download.	

6.3.11 Load Update Command

Loads data from a Motorola S Format file into a reserved memory area of *flash*MASTER. This data file contains all device specific operations.

Command	lu ↵	Loads data from an Motorola S format file to the *flash*MASTER buffer.

Status	**Screen Output**	**LED**
	. During download.	

6.3.12 Reset Command

The reset command restarts the *flash*MASTER firmware program from reset conditions. The reset command has the same effect as pressing the reset button.

Command +++ ↵ restarts *flash*MASTER firmware.

Status	**Screen Output**	**LED**
	Initial information screen as shown in Figure 58.	 or  depending on memory selection

6.3.13 Set Command

This command is used to set parameters on the *flash*MASTER, which are written to the EEPROM.

6.3.13.1 Set autocon Command

The `set autocon` command sets the autoconnect flag. This flag defines if *flash*MASTER tries to connect automatically to a device. Due to safety reasons, this is only done if the target board supplies Vdd.

Command	Description	Screen Output

set autocon ↵	Displays the current setting
	If autoconnect is enabled:
	If autoconnect is disabled:
		
		`Autoconnect : ON`
		`Autoconnect : OFF`

set autocon 1 ↵	Enables autoconnection, if Vdd is set to 0.	`Autoconnect : ON`
set autocon 0 ↵	Disables autoconnection	`Autoconnect : OFF`

6.3.13.2 Set blankcheck Command

The `set blankcheck` command sets the blankcheck flag: This flag decides if an additional blankcheck is performed after an erase command and, if the blankcheck fails, an additional erase is added.

Command	Description	Screen Output
set blankcheck ↵	Displays the current setting	
	If the blankcheck and possible erase is performed	`Blankcheck after erase : ON`
	If the blankcheck and possible erase is not performed	`Blankcheck after erase: OFF`
set blankcheck 1 ↵	Enables blankcheck after erase.	`Blankcheck after erase : ON`
set blankcheck 0 ↵	Disables blankcheck after erase.	`Blankcheck after erase: OFF`

Note: It is highly recommended not to use this command directly!
The target specific device settings are available in the device corresponding `*.STD` file and can be downloaded using the loaddev command.

6.3.13.3 Set centronics Command

The `set centronics` command sets the functionality of the parallel interface. This is by default set to centronics and can be changed with this command.

Command	Description	Screen Output
set centronics ↵	Displays the current setting	
set centronics 1 ↵	Parallel interface will be set to centronics.	`Parallel Port used as centronics`
set centronics 0 ↵	Parallel interface will be set to I/O port.	`Parallel Port used as I/O port`

6.3.13.4 Set clock Command

The `set clock` command defines if the CPU clock is provided by the target hardware or by *flash*MASTER.

Command	Description	Screen Output
set clock ↵	Displays the current setting	
	If clock will be provided by *flash*MASTER.	`Clock source: Programmer`
	If clock has to be provided by the target system.	`Clock source: Target`
set clock 1 ↵	Clock will be provided by *flash*MASTER.	`Clock source: Programmer`
set clock 0 ↵	Clock has to be provided by the target system.	`Clock source: Target`

6.3.13.5 Set comdevcsi Command

The `set comdevcsi` command sets the baudrate for device communication using the CSI. The baudrate input in kbps.

Command	Description	Screen Output
set comdevcsi ↵	Displays the current setting	`Current CSI baudrate is xxxx kbps`
set comdevcsi *xxxx* ↵	Baudrate will be set to *xxxx* kbps. Range 5 - 1000.	`Current CSI baudrate is xxxx kbps`
	If invalid value has been selected:	`Value not correct, please check entry`

6.3.13.6 Set comdeviic Command

The `set comdeviic` command sets the baudrate for device communication using the I²C.

Command	Description	Screen Output
set comdeviic ↵	Displays the current setting	`Current IIC speed is : xx kbps`
set comdeviic xx ↵	Baudrate will be set to xx kBps. Steps: 5, 10, 20, 30	`Current IIC speed is : xx kbps`

6.3.13.7 Set comdevport Command

The `set comdevport` command sets the baudrate for device communication using the port The baudrate input in bps.

Command	Description	Screen Output
set comdevport ↵	Displays the current setting	`Current Port baudrate is xxxx bps`
set comdevport *xxxx* ↵	Baudrate will be set to *xxxx* bps. Range 100 - 2000.	`Current Port baudrate is xxxx bps`
	If invalid value has been selected:	`Value not correct, please check entry`

6.3.13.8 Set comdevuart Command

The `set comdevuart` command sets the baudrate for device communication using the UART.

Command	Description	Screen Output
set comdevuart ↵	Displays the current setting	`Current UART baudrate is xxxx bps`
set comdevuart 9600 ↵	Baudrate will be set to 9600 baud	`Baudrate to device will be set to 9600 bps`
set comdevuart 19200 ↵	Baudrate will be set to 19200 baud	`Baudrate to device will be set to 19200 bps`
set comdevuart 38400 ↵	Baudrate will be set to 38400 baud	`Baudrate to device will be set to 38400 bps`

6.3.13.9 Set comhost Command

The `set comhost` command sets the baudrate to communicate with the host. This is by default set to 9600 bps and can be changed with this command.

Command	Description	Screen Output
set comhost ↵	Displays the current setting	`Current baudrate is xxxx bps`
set comhost 9600 ↵	Baudrate will be set to 9600 baud	`Baudrate will be set to 9600 bps`
set comhost 19200 ↵	Baudrate will be set to 19200 baud	`Baudrate will be set to 19200 bps`
set comhost 38400 ↵	Baudrate will be set to 38400 baud	`Baudrate will be set to 38400 bps`

Note: The baudrate changes directly after command input!

The baudrate may be changed as well using the DEFAULT button at *flash*MASTER. It will force *flash*MASTER to switch back to default settings which are 9600 bps, 8 data bits, 1 stop bit, no parity and no handshake.

6.3.13.10 Set devstatdisconnect Command

This command sets the status of the reset line after the disconnect command was executed. This can be used to test the device with *flash*MASTER still plugged in, when the power is provided by the target board.

Command	Description	Screen Output
set devstatdisconnect ↵	Displays the current setting.	`Device status after disconnect: xxxx`
set devstatdisconnect 0 ↵	Device will be reset after disconnect.	`Device status after disconnect: Reset`
set devstatdisconnect 1 ↵	Reset will be released after the disconnect.	`Device status after disconnect: Active`

6.3.13.11 Set erasetime Command / set et Command

The `set erasetime` command sets the device erase time.
The erase begins with the start value and continues if necessary with the increment value until the flash is erased or the maximum time is reached.
For long data input, `set et` may be selected as well.

Command	Description	Screen Output
set erasetime ↵ set et ↵	Displays the current erasetime.	`Erase time (ms): ss ii mm`
set erasetime *SS II MM* ↵ set et *SS II MM* ↵	Sets the new starttime *SS*, incremental time *II* and maximum time *MM*.	`Erase time (ms): ss ii mm`

Note: It is highly recommended not to use this command directly!
The target specific device settings are available in the device corresponding `*.STD` file and can be downloaded using the loaddev command.

6.3.13.12 Set frequency Command

The `set frequency` sets the oscillator or crystal frequency.

Command	Description	Screen Output
set frequency ↵	Displays the current frequency setting	`Oscillator (Crystal) frequency (Hz): xxxx`
set frequency *xxxxx* ↵	Sets the new frequency	`Oscillator (Crystal) frequency (Hz): xxxx`

The `set frequency` and `set clock` commands are related to each other. The frequencies generated by *flash*MASTER may be 0 … 1.000.000 Hz, 3.125.000 Hz, 6.250.000 Hz or 12.500.000 Hz. Trying to set other frequencies than the above listed while clock is provided by *flash*MASTER will not be accepted. There is no such restrictions when external clock is specified as clock source (`set clock 0`).

6.3.13.13 Set iicaddress Command

The `set iicaddress` command sets the I²C address of the device after a successful connect was done.

Command	Description	Screen Output
set iicaddress ↵	Displays the current I²C address of the device	`IIC Address: xx`
set iicaddress xx ↵	I²C address will be set to xx. Range: 0 0x7F	`IIC Address: xx`

6.3.13.14 Set mem Command

The `set mem` command selects the *flash*MASTER internal memory which holds the target program.

Command	Description	Screen Output	LED
set mem ↵	Outputs the current memory selection.	`Flash selected`	
		`RAM selected`	
set mem f ↵	selects Flash memory	`Flash selected`	
set mem r ↵	selects RAM memory	`Flash selected`	

6.3.13.15 Set port Command

The `set port` command defines the port used to communicate with the device.

Command	Description	Screen Output

set port ↵	Displays the current setting
	If UART is selected:
	
	If CSI is selected:
		
		`Device connection : UART n with xxxx bps`
		`Device connection : CSI n with xxxxx bps`

set port csi0 ↵	Selects CSI0	`Device connection : CSI 0 with xxxxx kbps`
set port csi1 ↵	Selects CSI1	`Device connection : CSI 1 with xxxxx kbps`
set port csi2 ↵	Selects CSI2	`Device connection : CSI 2 with xxxxx kbps`
set port csihs ↵	Selects CSI0 with additional handshake lines	`Device connection : CSI 0 (Handshake) with xxxx kbps`
set port iic0 ↵	Selects IIC0	`Device connection : IIC 0 with xxxx kbps`
set port iic1 ↵	Selects IIC1	`Device connection : IIC 1 with xxxx kbps`
set port iic2 ↵	Selects IIC2	`Device connection : IIC 2 with xxxx kbps`
set port iic3 ↵	Selects IIC3	`Device connection : IIC 3 with xxxx kbps`
set port uart0 ↵	Selects UART0	`Device connection : UART 0 with xxxx bps`
set port uart1 ↵	Selects UART1	`Device connection : UART 1 with xxxx bps`
set port uart2 ↵	Selects UART2	`Device connection : UART 2 with xxxx bps`
set port uart3 ↵	Selects UART3	`Device connection : UART 3 with xxxx bps`
set port port0 ↵	Selects PORT0	`Device connection : PORT 0 with xxxx bps`
set port port1 ↵	Selects PORT1	`Device connection : PORT 1 with xxxx bps`

6.3.13.16 Set prescale Command

The `set prescale` command sets the prescale factor. This factor defines the PLL factor or divider of the device.

Command	Description	Screen Output
set prescale ↵	Displays the current prescale factor.	`Prescale factor : xxxx`
set prescale *xxxx* ↵	Sets the new prescale factor. (1/2 is used for division by 2)	`Prescale factor : xxxx`

6.3.13.17 Set prewrite Command

The `set prewrite` command sets the prewrite flag: This flag decides if a prewrite is performed before an erase or not.

Command	Description	Screen Output
set prewrite ↵	Displays the current setting	
	If prewrite is selected.	`Prewrite : ON`
	If prewrite is not selected.	`Prewrite : OFF`
set prewrite 1 ↵	Prewrite is selected.	`Prewrite : ON`
set prewrite 0 ↵	Prewrite is not selected.	`Prewrite : OFF`

Note: It is highly recommended not to use this command directly!
The target specific device settings are available in the device corresponding `*.STD` file and can be downloaded using the loaddev command.

6.3.13.18 Set progarea Command

If the two area option is activated by `set twoarea 1`, this command defines which area of the two possible ones is to be used. This setting is used for downloading of the program from host to *flash*MASTER as well as for the programming of the device.
When two areas are used the *flash*MASTER application memory is devided as follows:

Area 0: 0x00000 ... 0x3FFFF

Area 1: 0x40000 ... 0x7FFFF

Command	Description	Screen Output
set progarea ↵	Displays the current setting.	
set progarea 0 ↵	Program area 0 is used.	`Program area 1 not activated`
set progarea 1 ↵	Program area 1 is used.	`Program area 1 activated`

6.3.13.19 Set progoptions Command

The `set progoptions` command sets the programming options for the EPV command sequence. These options are used either for the EPV Command or the Start button.

Command	Description	Screen Output
set progoptions ↵	Displays the current option setting	`Autoprogramming Options : xxxx`
set progoptions *xxxx* ↵	Set the autoprogramming options. b: blankcheck before erase e: erase p: program v: verify	`Autoprogramming Options : xxxx`

6.3.13.20 Set pulselevel Command

The `set pulselevel` command sets the level of the V_{PP} pulses. With this command the level of Vpp during the pulses which define the selected device port can be defined.

Command	Description	Screen Output
set pulselevel ↵	Displays the current setting.	
	If the low level of the pulse is going down to the Vss level	`Vpp pulses (low level): Vss`
	If the low level of the pulse is going down to the Vdd level	`Vpp pulses (low level): Vdd`
set pulselevel 0 ↵	Sets the low level of the pulse to the Vss level.	`Vpp pulses (low level): Vss`
set pulselevel 1 ↵	Sets the low level of the pulse to the Vdd level.	`Vpp pulses (low level): Vdd`

Note: It is highly recommended not to use this command directly!
The target specific device settings are available in the device corresponding `*.STD` file and can be downloaded using the loaddev command.

6.3.13.21 Set twoarea Command

The `set twoarea` command allows keeping two different program codes in the *flash*MASTER memory at the same time. Which program code is actually programmed to the device is selected with the `set progarea` command.

Command	Description	Screen Output
set twoarea ↵	Displays the current setting.	
set twoarea 0 ↵	One program area (512K max. size) is used.	`Two program areas not activated`
set twoarea 1 ↵	Two program areas (256K max. size each) are used.	`Two program areas activated`

6.3.13.22 Set V_{DD} Command

The `set vdd` command selects the supply voltage.

Command	Description	Screen Output
set vdd ↵	Outputs the current Vdd in mV	`Vdd: xxxx mV`
set vdd *xxxx* ↵	Selects Vdd in mV and decimal. Range 0 - 5000 mV.	`Vdd: xxxx mV`

Note: Due the fact that an 8 bit D/A converter is used, the entered value is automatically converted to a value which can be output by the D/A converter.

6.3.13.23 Set V_{PP} Command

The `set vpp` command selects the programming voltage.

Command	Description	Screen Output
set vpp ↵	Outputs the current Vpp in mV	`Vpp: xxxxx mV`
set vpp *xxxxx* ↵	Selects Vpp in mV and decimal. Range 0 - 10500 mV.	`Vpp: xxxxx mV`

Note: Due the fact that an 8 bit D/A converter is used, the entered value is automatically converted to a value which can be output by the D/A converter.

Note: It is highly recommended not to use this command directly!
The target specific device settings are available in the device corresponding `*.STD` file and can be downloaded using the loaddev command.

6.3.13.24 Set waitfactor Command

The `set waitfactor` command sets a waitfactor. This is an input that describes the general communication speed (NOT baudrate) of the *flash*MASTER. This factor is needed to adapt the writer to different devices, which have different performances even with the same CPU clock.

Command	Description	Screen Output
set waitfactor ↵	Displays the current setting.	`Factor : xxx`
set waitfactor *xxx* ↵	Sets the new wait factor.	`Factor : xxx`

6.3.13.25 Set wbretries Command

The `set wbretries` sets maximum retry number for the writeback command.

Command	Description	Screen Output
set wbretries ↵	Displays the current setting	`Number of WB retries: 0` `Total number of WB retries: 0`
set wbretries *iii ttt* ↵	Sets the new values for internal retries (*iii*) and total retries (*ttt*)	`Number of WB retries: iii` `Total number of WB retries: ttt`

Note: It is highly recommended not to use this command directly!
The target specific device settings are available in the device corresponding `*.STD` file and can be downloaded using the loaddev command.

6.3.13.26 Set writeback Command

The `set writeback` command sets the writeback flag. This flag defines if *flash*MASTER starts writeback operation after erase if necessary.

Command	Description	Screen Output
set writeback ↵	Displays the current setting	
	If writeback is enabled:	`Writeback : ON`
	If writeback is disabled:	`Writeback: OFF`
set writeback 1 ↵	Enables writeback, if Vdd is set to 0.	`Writeback: ON`
set writeback 0 ↵	Disables writeback	`Writeback: OFF`

Note: It is highly recommended not to use this command directly!
The target specific device settings are available in the device corresponding `*.STD` file and can be downloaded using the loaddev command.

6.3.13.27 Set writebacktime Command

The `set writebacktime` command sets the writeback time used, if writeback is enabled.

Command	Description	Screen Output
set writebacktime ↵	Displays the current setting	`Writebacktime: xxx ms`

set writebacktime *xxx* ↵	Sets writebacktime in ms.
	Range 0 - 1000 ms.
	If value is out of range:
		`Writebacktime: xxx ms`
		
		`Value to large`

Note: It is highly recommended not to use this command directly!
The target specific device settings are available in the device corresponding `*.STD` file and can be downloaded using the loaddev command.

6.3.13.28 Set writecycle Command

The `set writecycle` command sets a wait cycle. This factor is used to define how often a Flash has to be written. There are some devices that require several writes.

Command	Description	Screen Output
set writecycle ↵	Displays the current setting. It shows the current number of writes that will be performed when using the write command	`Number of writes: xx`

set writecycle *xx* ↵	Sets the new number of write cycles.
	Range 1 - 10.
	If invalid range is selected
		`Number of writes: xx`
		
		`Number to large`

Note: It is highly recommended not to use this command directly!
The target specific device settings are available in the device corresponding `*.STD` file and can be downloaded using the loaddev command.

6.3.13.29 Set writetime Command

The `set writetime` command sets the device write time.

Command	Description	Screen Output
set writetime ↵	Displays the current writetime.	`Write time (µs):xxxx`
set writetime *xxxx* ↵	Sets the new writetime (ms).	`Write time (µs):xxxx`

Note: Specific devices like 78K0 and 78K4 do not allow specifying the writetime. In this case the writetime must be set to 0!

Note: It is highly recommended not to use this command directly!
The target specific device settings are available in the device corresponding `*.STD` file and can be downloaded using the loaddev command.

7. Sample programming session using terminal communication program

As an example, a 78F0034A micro-controller will be used to show a typical programming sequence using the Windows HyperTerminal software.

1. Use the provided serial cable to connect *flash*MASTER to your host computer.

2. Choose target cable 1 from your *flash*MASTER package and connect *flash*MASTER to the target hardware. This may either be a NEC programming adapter or your own target hardware.

3. Select the suitable AC-plug (EURO, UK or US/JAPAN) and connect it to the AC adapter.

4. Connect the AC adapter to *flash*MASTER's power connection and mains.

5. If not yet done, copy all *.STD files in the subdirectory \dev on disk 2 of the *flash*MASTER GUI installation package to your harddisk.

6. Start HyperTerminal.

7. Activate **File → Properties** menu to set the appropriate port number you connected *flash*MASTER to.

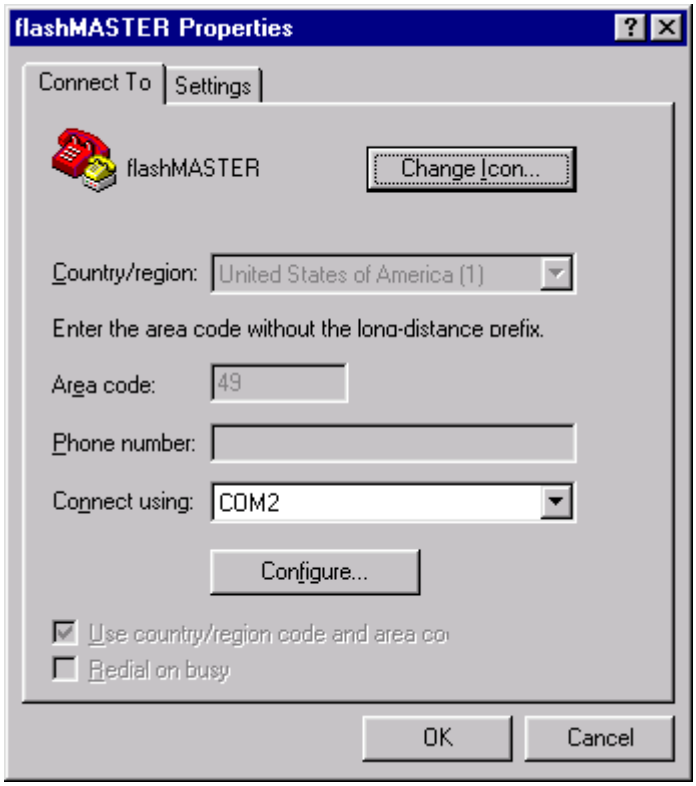


Figure 59: HyperTerminal properties window

By precssing **Configure** you proceed to the communication settings for the selected port.

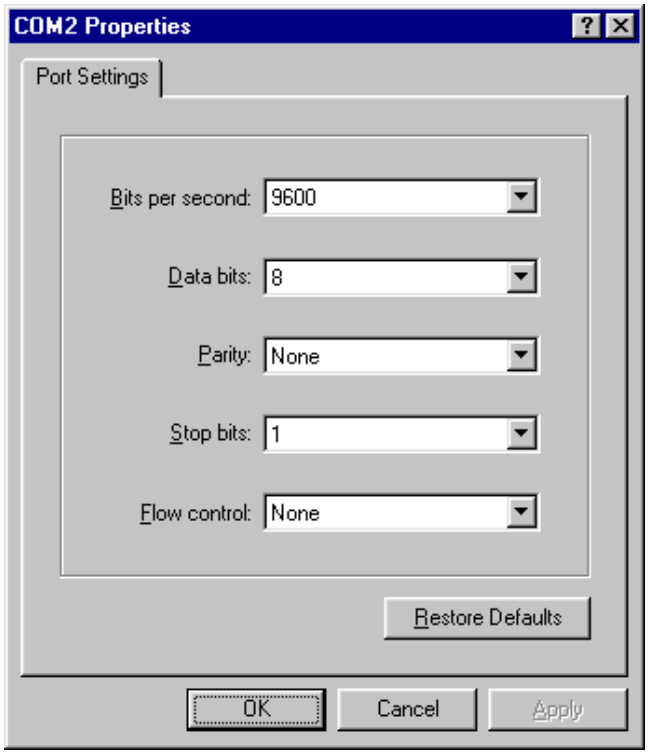


Figure 60: Hyperterminal Port Setup

The communication speed shall be set to 9600 bps. Select 8 data bits, no parity, 1 stop bit and no flow control in this window.

8. Select **Call →Disconnect** and **Call → Connect** menus to activate the communication settings made in step 7. You will now be connected to the *flash*MASTER.
   If you would like to change the communication speed you may use the `set comhost` command in the HyperTerminal communication window to select any other speed setting. Then return to step 7 to adjust the HyperTerminal settings.

9. Type `loaddev` in the HyperTerminal communication window, activate **Transfer → Send Text File** menu. Select the file 0034AHP.STD in the directory you have chosen in step 5. Press **Open** to send this file to *flash*MASTER.
   By resetting *flash*MASTER the device specific settings for the 78F0034A are initialised.

10. Depending on your target hardware define the port to be used by *flash*MASTER to communicate with the target system by one of the following commands:

```
set port csi0..2  ↵
```
or
```
set port uart0..3  ↵
```
or
```
set port iic0..3  ↵
```

11. Define the baud rate for communication between your target hardware and *flash*MASTER by one of the following commands (depending on the port you have chosen in 10.):

    `set comdevcsi 5..1000` ↵ (measured in KHz)
    or
    `set comdevuart 4800..38400` ↵ (measured in Baud)
    or
    `set comdevport 100..2000` ↵ (measured in Hz)
    or
    `set comdeviic 5..30` ↵ (measured in KBaud)

    If comdeviic is selected, the command `set iicaddress 0…7F` is also required to specify the I²C address.

12. Define the clock source depending on your target hardware by one of the following commands:

    `set clock 0` ↵ clock is provided by your target system
    or
    `set clock 1` ↵ clock is provided by *flash*MASTER

13. Define the appropriate oscillator or crystal frequency for your target hardware with the following command:

    `set frequency xxxxx` ↵ (measured in Hz)

14. Define the supply voltage of your target hardware by the following command:

    `set vdd 0..5000` ↵ (measured in mV)

    Use

    `set vdd 0` ↵

    in case V_{dd} is supplied by your target system.

15. Define the destination memory for your target program in *flash*MASTER by one of the following commands:

    `set mem r` ↵ SRAM will hold your target program
    or
    `set mem f` ↵ onboard flash memory will hold your target program

16. Enter

    `set progoptions epv` ↵

    in the HyperTerminal communication window to set programming options for the EPV-command sequence.

17. Use the

    `conf` ↵

    command to verify that all settings are correct.

18. Start downloading your program by using the

    `ls` ↵

    command for Motorola S files or the

    `li` ↵

    command for INTEL HEX files. When Now loading is displayed on the screen, select **Transfer** → **Send Text File** in HyperTerminal. Highlight the file to be send and press **Open**. While downloading is processed, several '.' will be displayed.

19. Connect to the device by entering

    `c` ↵

    in the HyperTerminal window. If the connection has been established you will find the device name shown on the screen.

20. Use the

    `epv` ↵

    command to program your target device. What operations will be performed during the epv command execution depends on the setting made in 16. with the set progoptions command.

21. Enter the disconnect-command

    `dc` ↵

    Your target hardware may now be disconnected from *flash*MASTER.

22. If necessary, connect another device to be programmed to *flash*MASTER and repeat from step 19.

23. If no other device needs to be programmed exit HyperTerminal. All settings made during this programming session will be saved in the *flash*MASTER so that they can be reused next time.

8. Connectors and Cables

8.1 Power supply jack

The specifications of the power supply jack are listed below:

Figure 61: Power supply jack

Power supply specification:

Compatible plug: EIJA class 4
Polarity: Center GND

Note: Connect only the provided AC adapter to the power supply jack of *flash*MASTER!

8.2 Serial host connector

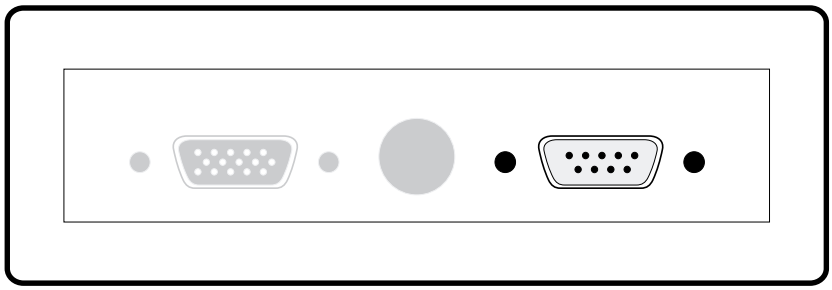


Figure 62: Serial host connector

Serial Host	Signal name at *flash*MASTER
1	NC
2	RxD
3	TxD
4	NC
5	GND
6	NC
7	NC
8	NC
9	NC

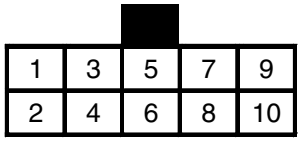
Table 1: Pinout of serial host connector

8.3 Target cable 1

The target cable 1 is compatible with all NEC flash programming adapters FA-XXX-YY.

HD-SUB 15 (male)

Target Connector:



(View from the solder side of the target system)

Figure 63: Target cable 1

Target cable 1	*flash*MASTER HD-Sub 15	Signal name at *flash*MASTER
1	15	GND
2	4	/RESET
3	2	SI / RxD
4	8, 9	V_{DD}
5	1	SO / TxD
6	11	V_{PP}
7	3	SCK
8	7	(*)
9	14	CLK
10	13	(*)

Table 2: Pinout of target cable 1

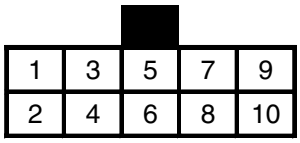
(*) : Reserved for future use, leave open!

8.4 Target cable 2

The target cable 2 is compatible with the programming layout of the EB-V853STARTER.

HD-SUB 15 (male)

Target Connector:



(View from the solder side of the target system)

Figure 64: Target cable 2

Target cable 2	*flash*MASTER HD-Sub 15	Signal name at *flash*MASTER
1	2	SI / RxD
2	1	SO / TxD
3	3	SCK
4	4	/RESET
5	11	V_{PP}
6	8, 9	V_{DD}
7	7	(*)
8	15	GND
9	14	CLK
10	13	(*)

Table 3: Pinout of target cable 2

(*) : Reserved for future use, leave open!

8.5 HD-Sub 15 device connector

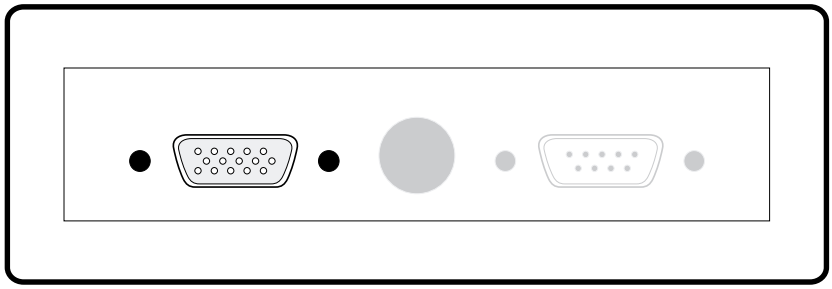


Figure 65: HD-Sub 15 connector

*flash*MASTER HD-Sub 15	Signal at *flash*MASTER
1	SO / TxD
2	SI / RxD
3	SCK
4	/RESET
5	(*)
6	(*)
7	(*)
8	VDD
9	VDD
10	(*)
11	VPP
12	(*)
13	(*)
14	CLKOUT
15	GND

Table 4: Pinout of HD-Sub 15 connector

(*) : Reserved for future use, leave open!

8.6 Parallel host connector

The parallel host connector can be configured as centronics interface or as I/O port. Depending on the selection, one of the following signal setting is available.

8.6.1 Parallel port configured as centronics interface

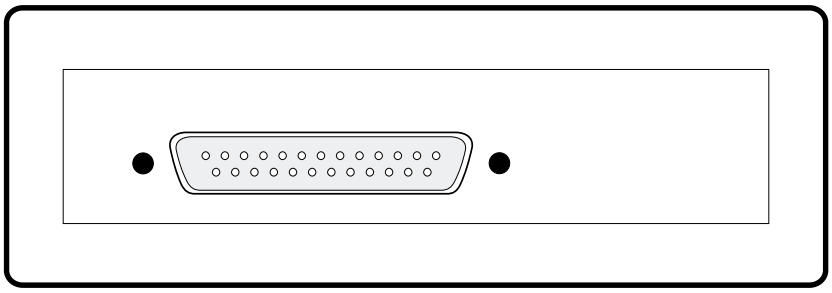


Figure 66: Parallel port

If the parallel port is configured as centronics interface, below listed signals are available.

Parallel Port	Signal at *flash*MASTER
1	/STROBE
2	D0
3	D1
4	D2
5	D3
6	D4
7	D5
8	D6
9	D7
10	/ACK
11	BUSY
12	GND
13	NC
14 - 17	V_{cc}
18	NC
19 - 25	GND

Table 5: Pinout of centronics interface

NC: Not connected.

8.6.2 Parallel port configured as I/O port

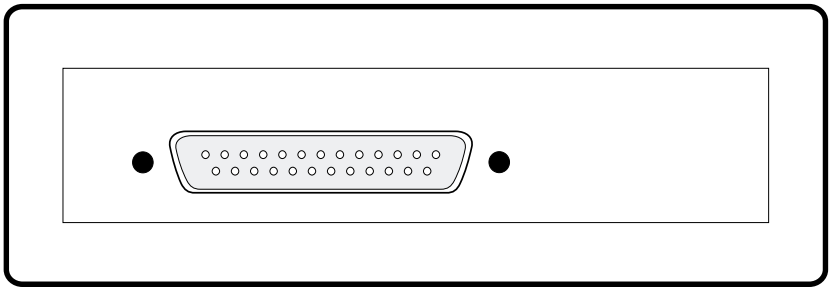


Figure 67: Parallel port

If the parallel port is configured as I/O port, below listed signals are available. The signal I/O is TTL level (74LS241).

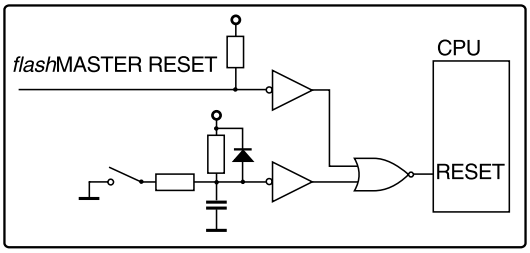
Parallel Port	Signal at *flash*MASTER	
1	IC	
2	Start EPV command, area 0	INPUT, active low
3	Start EPV command, area 1	INPUT, active low
4	IC	
5	IC	
6	Device connected	OUTPUT, active high
7	Busy	OUTPUT, active high
8	OK	OUTPUT, active high
9	Error	OUTPUT, active high
10	IC	
11	IC	
12	GND	
13	NC	
14 - 17	Vcc	For reference only!
18	NC	
19 - 25	GND	System ground reference

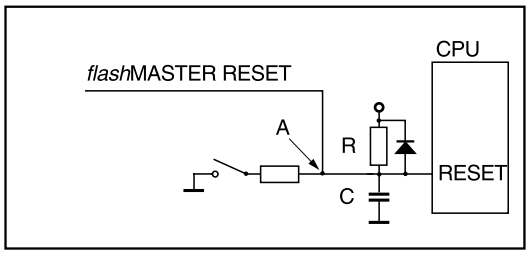
Table 6: Pinout of I/O port

NC: Not connected.
IC: Internally connected, leave open!

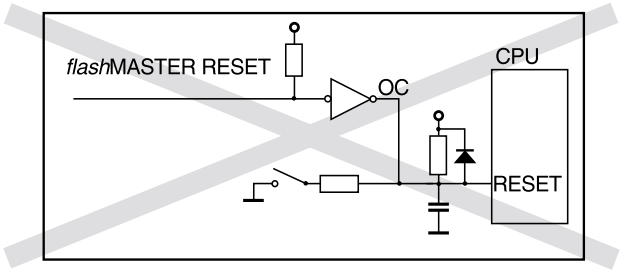
9. Design proposals for user systems

This chapter explains the user system design proposals for rewriting the flash ROM in the microcontroller using *flashMASTER*.

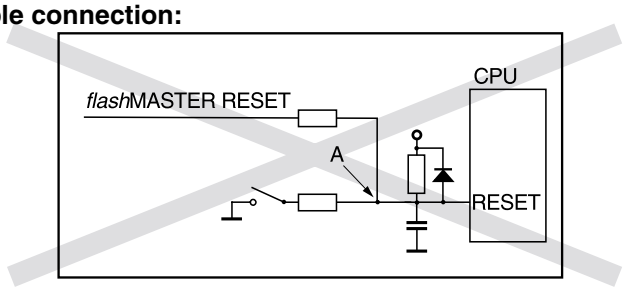
<table>
<tr><th>CPU pin</th><th>Design proposal</th></tr>
<tr><td>/RESET</td><td>Do not generate RESET while flashMASTER is connected. Be particularly careful to observe this requirement when using a system with an external watchdog timer.</td></tr>
<tr><td></td><td>Connect the RESET signal of flashMASTER at a point where the status of the programmer RESET signal and that of the CPU RESET pin are the same.
Acceptable connection:
 



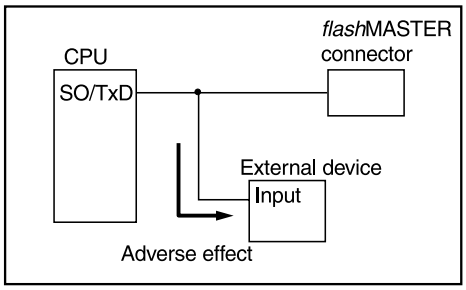
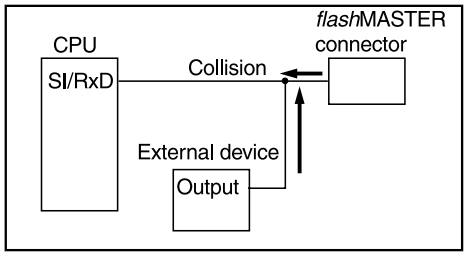
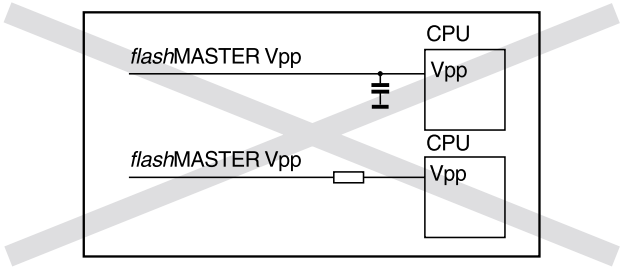
R must be 1 kΩ or more. The value for C must be chosen that the rise time between V_{IL} and V_{IH} does not exceed max. spec of the device.

Avoid the following RESET signal connection:
• Connection to a point where the target CPU RESET rise time is slower than the flashMASTER RESET rise time.
Unacceptable connection:
 

It takes time for the CPU RESET pin to go high after the flashMASTER RESET level goes from low to high.

• Connection to a point where the target CPU RESET pin cannot be driven to low level by flashMASTER RESET signal.
Unacceptable connection:
 

When the flashMASTER RESET is driven low, the level of the voltage at point A does not fall.</td></tr>
</table>

NMI	Do not generate NMI while the *flash*MASTER is connected. Be particularly careful to ensure that this requirement is satisfied for a user system with an external watchdog timer.
Maskable interrupt	All maskable interrupts are ignored while *flash*MASTER is connected.
Ports	While the *flash*MASTER is connected, all those ports that are not being used by the *flash*MASTER will be in the input (floating) state. When an external device connected to such a port does not allow the port to enter the floating state, pull the port up or down. When the CPU port used by the *flash*MASTER is also connected to the input of an external device, and if that device malfunctions, disconnect the external device. Example:  While the CPU port used by the *flash*MASTER is also connected to the output of an external device, and if a sign al collision occurs, disconnect the external device. Example: 
V_{PP}	Keep the wiring between the *flash*MASTER connector and the CPU Vpp pin as thick and short as possible. Insert neither capacitors nor resistors into the V_{PP} line. **Unacceptable circuit:** 
Others	The CPU may have WAIT, MODE , and other pins whose handling depends on the CPU. For an explanation of such signals, refer to the related CPU manual.

10. User system interface circuits

This chapter describes the user system interface circuits of the *flash*MASTER (TTL level).

10.1 SO/TxD, SCK, RESET

10.1.1 V_{DD} supplied by *flash*MASTER

For programming flash devices, V_{DD} may be supplied by *flash*MASTER. The signal lines SO/TxD, SCK and RESET will have TTL level voltage.

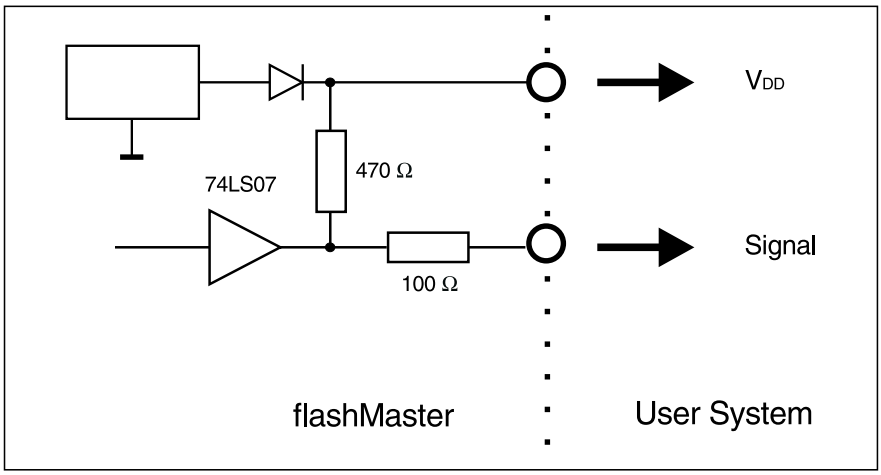


Figure 68: *flash*MASTER output signal level

10.1.2 V_{DD} supplied by User System

Alternatively, V_{DD} may be supplied by the User System. The *flash*MASTER internal voltage regulator is protected so that user V_{DD} will only affect the signal lines SO/TxD, SCK and RESET.
The terminal command `SET VDD 0` ↵ will activate this selection.

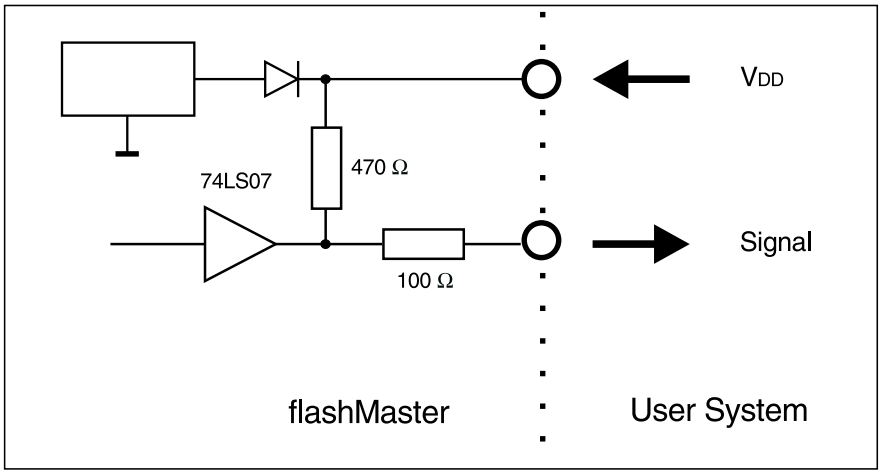


Figure 69: *flash*MASTER output signal level

10.2 SI/RxD

The SI/RxD input signal must not exceed TTL level voltage.

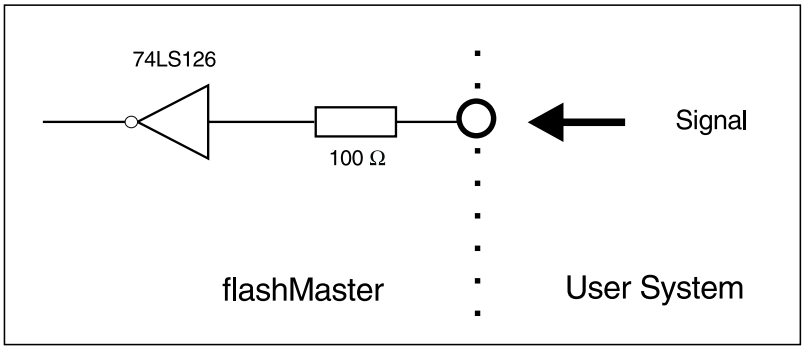


Figure 70: *flash*MASTER input signal level

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6465-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: 02-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: 044-548-7900

I would like to report the following error/make the following suggestion:

Document title: _____

Document number: _____ Page number: _____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	☐	☐	☐	☐
Technical Accuracy	☐	☐	☐	☐
Organization	☐	☐	☐	☐