



Future Technology Devices Intl Ltd.

MProg User Manual

Created by Future Technology Devices Intl. (FTDI) Ltd.

Contents Copyright (c) 2003 FTDI Ltd.

Table of Contents

Part 1	Introduction	1
1.1	MProg Installation	1
1.2	D2XX Driver Installation	2
1.3	FT232/245 EEPROM Functions	5
Part 2	MProg Basics	7
2.1	Modes of Operation	7
Part 3	Edit Mode Functions	9
3.1	Creating a New Programming Template	9
3.2	Modifying an Existing Programming Template	14
Part 4	Program Mode Functions	16
4.1	Scanning for Devices	16
4.2	Erasing Devices	16
4.3	Programming Devices	17
Part 5	Appendix	18
5.1	Customising FTDI's VCP Drivers	18
5.2	Customising FTDI's D2XX Drivers	19

..

1 Introduction

1.1 MProg Installation

IMPORTANT NOTE

Prior to installation, please uninstall any existing FTDI VCP drivers from the target PC using add/remove Programs in the Windows control Panel. MProg uses FTDI's D2XX drivers - a copy of which are installed in the Drivers sub directory of the main program folder for your convenience.

MProg is supplied as a zip file containing a self extracting installation file (setup.exe). Run the setup.exe file and the following screen will appear.



Select which language you wish the setup program to run in and click Next. Follow the on-screen instructions until the software is installed.

USING MPROG FOR THE FIRST TIME.

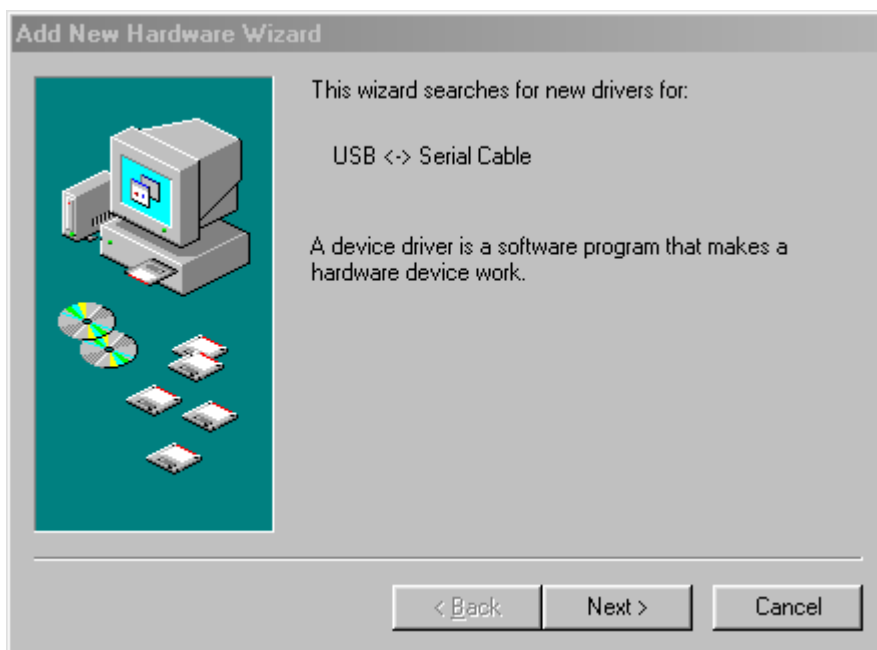
The first time you plug a FT232/FT245 device into the system, it will request USB drivers. If it does not, then FTDI drivers are probably already installed on the target P.C. If these are VCP (Virtual COM Port) drivers, you must uninstall these and install the D2XX drivers instead. Installation of the D2XX drivers are covered in the next section.

1.2 D2XX Driver Installation

The MProg installation program copies a set of FTDI's D2XX drivers into the Drivers subdirectory of the MProg folder. For instance if you installed MProg into path C:\Program Files\MProg 1.0 then the drivers can be found in path C:\Program Files\MProg 1.0\Drivers.

Plugging in a FT232 / FT245 based device into the target PC should result in Windows launching the Add New Hardware wizard (below) . If it does not, then FTDI drivers are probably already installed on the target P.C. If these are VCP (Virtual COM Port) drivers, you must uninstall these and install the D2XX drivers instead.

The following shows the installation procedure for Windows '98. Other Windows versions have a similar installation procedure.



Click on "Next"



Select "Search for the best driverfor your device" and click on "Next"



Select "specify a location" and browse to the folder containing the drivers (see above). Click on "Next"



Click on "Next"



Click on "Finish" to finalise the driver installation.

1.3 FT232/245 EEPROM Functions

FTDI's first generation FT8U232AM USB UART and FT8U245AM USB FIFO devices (and their second generation counterparts, FT232BM and FT245BM respectively) can have some of their default parameters customised by attaching an external 93C46 EEPROM to the device programmed with the replacement parameters.

A convenient way of programming the EEPROM is to use a blank device on the pcb during manufacturing assembly and program it via USB during the manufacturing test process. All these devices support direct programming over USB using a programming utility such as this one.

The parameters that can be customised via EEPROM fall into several catagories -

1. All Devices

- USB Vendor ID and Product ID numbers commonly known as VID and PID respectively.
- Self Powered (has own PSU) or Bus Powered (gets it's power from USB)
- Current Consumption - the maximum current your device draws from the USB Bus during normal operation. For a Bus Powered Device enter the actual operating current of the device, for a Self Powered Device enter zero, however, refer to the notes in the following section which give more details of various limitations which have to be taken into account.
- Manufacturer - the product manufacturer string e.g. "FTDI" - best kept short and abbreviated if possible.
- Product Description - a brief description of the product, maybe including the part number e.g. "US232 USB to RS232 Converter"
- Serial Number - a unique serial number which is programmed into each device. This is used by USB to identify different devices of the same product family, no matter which USB Host or Hub port the device happens to be plugged in to.

2. FT232 USB UART Family

- Windows Plug and Play (PNP) - for general purpose USB to RS232 serial converters it is desirable that the device tries to recognise any attached secondary peripheral that conforms to the Windows Plug and Play standard. These devices are typically modems, or serial mice. On recognition of such a device Windows will attempt to identify and load the drivers for the secondary peripheral. For other application areas PNP is not usually required so this option should be turned off.

3. Wake up from USB suspend (Remote Wake Up)

- FT232 series devices are capable of resuming a PC from the USB suspend (sleep) state via the RI# pin. FT245BM also has this capability via a dedicated pin (SI/WU). FT8U245AM does not have support for remote wake up. To use the remote wake-up capability, this option must be enabled in the EEPROM. Some host PC's also require this option to be enabled in the system BIOS before it will work.

3. BM Series Options

The FT232BM and FT245BM 2nd generation devices have some additional options compared to the AM devices. These options can be controlled via the optional EEPROM.

- USB Version Number - the BM series devices have the option to set the USB version number to USB 1.1 or USB 2.0. This has no effect on the functionality of the device, however for USB compliance testing we recommend that you choose the USB2.0 setting. For AM series devices the version number is hard coded to USB 1.1
- Disable the USB serial number - this allows multiple identical USB devices to be plugged into the same host PC, however, it is then impossible for the USB drivers to uniquely identify the devices. We recommend that for most applications, this option is not selected.
- Pull Down IO Pins in USB suspend - BM devices feature a Power Control Pin (PWREN#). This pin can be used to control power to external peripheral circuitry via a P-Channel MOSFET. When used in this way, select this option to minimize the USB suspend current otherwise leave this option unselected. See the BM data sheets, application schematics and designers guide for more details.
- Isochronous Mode - BM series devices have the option of transmitting / receiving data packets using USB isochronous (streaming) transfer instead of the default mode which uses USB Bulk data transfer. However, special drivers / technical support is required to use this mode. **The web edition of MProg has these options disabled as careless use of this feature can result in a device which can crash the PC under test.** Should you require this feature, please request a copy of MProg with ISO Transfers enabled from FTDI technical support.

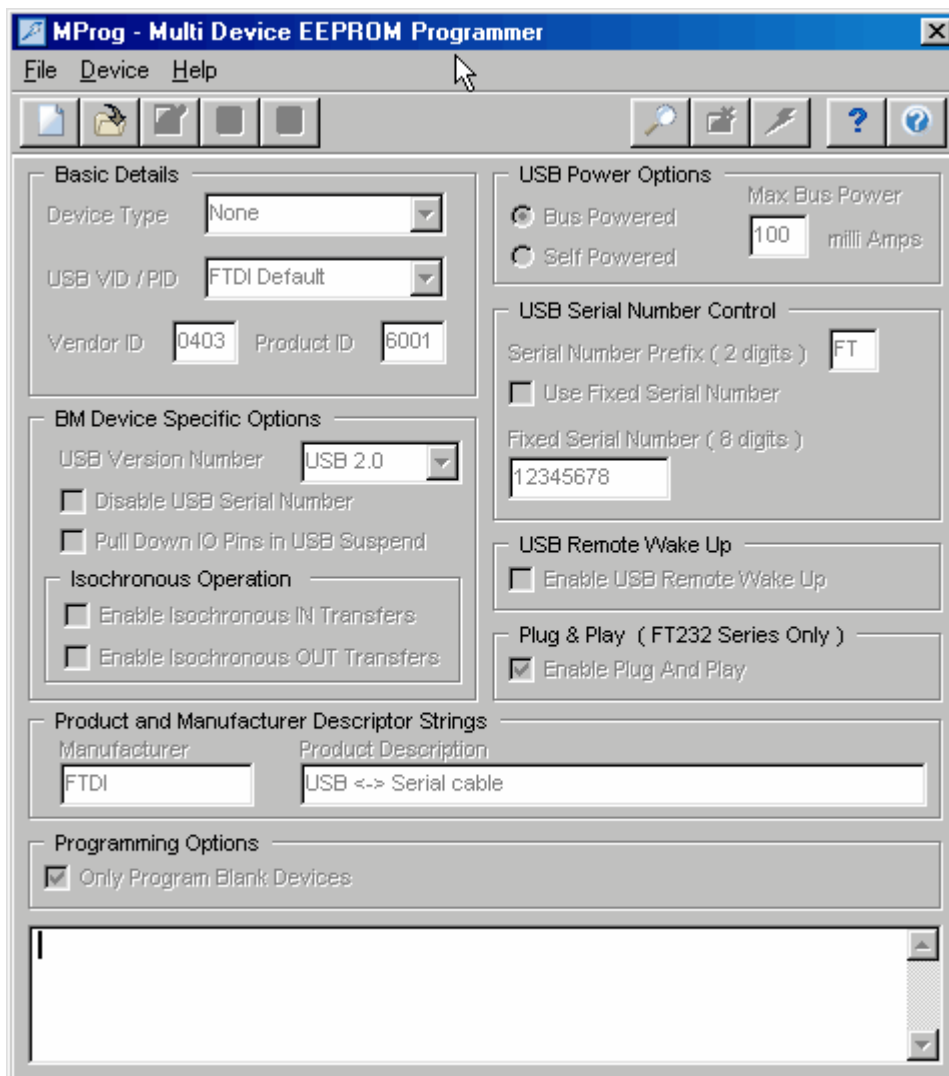
2 MProg Basics

2.1 Modes of Operation

MProg has 3 modes of operation, Idle Mode, Program Mode and Edit Mode. MProg programming parameters are saved in files referred to here as Programming Templates. Once defined, these Programming Templates can be loaded by MProg and used to program EEPROMs.

- Idle Mode is the initial mode of operation when the program is launched.
- Edit Mode is used to edit the settings of a Program Template.
- Program Mode is used to Program and Erase the device EEPROM(s).

When MProg is launched, it is initially in Idle Mode and the screen will look like this.



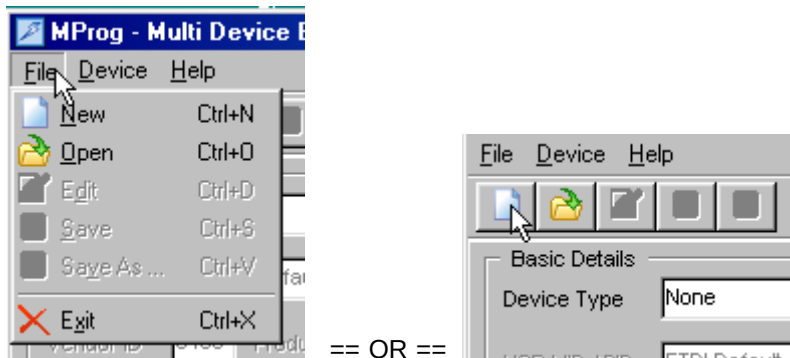
User choices in Idle Mode are -

1. Create a new Programming Template. This will change the mode to Edit Mode and allow the user to select the parameters for his project. The new Programming Template must then be saved to disk which will then put MPROG into Program Mode.
2. Open an existing Programming Template. This will set the MPROG parameters to that of the selected template and put MPROG into Program mode, ready to Program devices.
3. Scan the USB Bus for available devices - MProg will report how many programmed and blank devices are attached to the target PC.

3 Edit Mode Functions

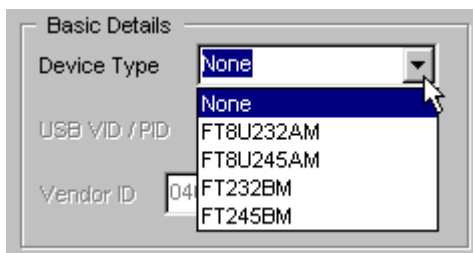
3.1 Creating a New Programming Template

To create a new Programming Template select New from the File Menu, or click on the New Programming Template button on the toolbar.

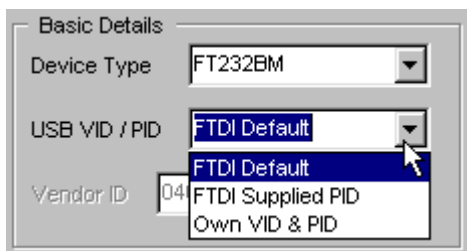


Device Type

Select the device type you wish to program from the Basic Details box.



When selected, this will enable the menu options appropriate to the device you have selected. Next, select the USB Vendor Id (VID) and Product Id (PID) choices for your project as shown ..



If you use the FTDI default values, the VID and PID are fixed and you do not need to enter any values.

If you wish to use FTDI's Vendor Id and FTDI have supplied you with a Product Id for use with your products, please enter the Product Id supplied by FTDI (a 4 digit hexadecimal number) in the

Product Id box provided. If you wish to use FTDI's Product Id and have a unique product Id assigned for you by FTDI then please e-mail support@ftdichip.com with your request. There is no charge for this service to bona-fide FTDI customers.

If you have your own USB Vendor and Product Id for your device, then select the Own VID and PID option and enter both the USB VID and USB PID for your product in the boxes provided.

Power Options

Set the USB Power Options.

Select Bus Powered (the product gets it's power from USB) or Self Powered (the product has it's own power supply) from the power options. The Maximum Bus Power value defaults to 100mA for a Bus Powered product and 0mA for a Self Powered product, however these are only default values and can be changed by entering a new value in the Maximum Bus Power field. A summary of the rules and options are given below -

ALL USB PRODUCTS

1. The maximum current that can be supplied over the USB bus without violating the USB specification is 500mA. Do not enter a value of more than 500 in the Max Bus Power Box.
2. On plug-in, no USB product is allowed to consume more than 100mA from the USB bus until USB enumeration is complete i.e. the product and it's USB drivers are successfully loaded by the host PC. Special care has to be taken in the case of High power Bus Powered Products, however the PWREN# pin of the BM series devices simplifies such designs.

BUS POWERED PRODUCTS

These fall into two categories -

- Low power Bus Powered products. These products consume 100mA or less from the USB Bus and can be plugged into any USB host or hub port.
- High power Bus Powered products. These products consume between 101mA and 500mA from the USB Bus and can be plugged into any USB host, however such products can only be plugged into self powered USB hubs (hub has it's own PSU) as bus powered USB hubs can only supply 100mA per hub port.

SELF POWERED PRODUCTS

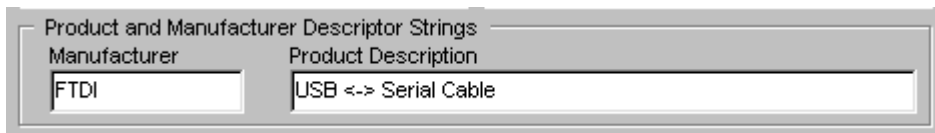
These fall into two categories -

- True Self Powered Products. These products have their own power supply and do not require any current from the USB bus. Enter a value of zero in the Maximum Bus Power field.
- Hybrid Self Powered Products. These products have their own power supply, but also draw some of their operating current from the USB bus. Enter the current draw from the USB Bus in the Maximum Bus Power field.

Product and Manufacturer Description Strings

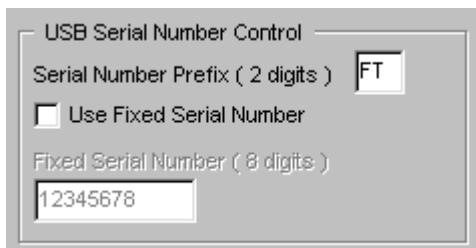
Enter the Product and Manufacturer Description Strings in the fields provided

- Manufacturer - the product manufacturer string e.g. "FTDI" - best kept short and abbreviated if possible.
- Product Description - a brief description of the product, maybe including the part number e.g. "US232 USB to RS232 Converter"



USB Serial Number Control

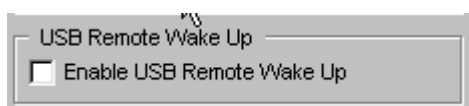
USB Products have the ability to embed a unique serial number for each product into the low level descriptors that are accessed by the host controller drivers when the product is plugged into the USB bus. In turn, the serial number of a product can be used to identify the product



MProg can generate unique serial numbers automatically for you (recommended) . These consist of a 2 letter prefix (e.g. FT) followed by 6 alphanumeric digits generated using an algorithm based on the date and time of programming the device. It is also possible to fix the serial number to a fixed 8 digit alphanumeric string, however we do not recommend this as systems will only recognise the first instance of such a device connected to a PC.

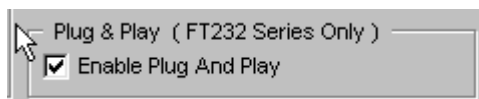
USB Remote Wake Up (FT8U232AM, FT232BM, FT245BM)

FT8U232AM and FT232BM devices are capable of resuming a PC from the USB suspend (sleep) state via the RI# pin. FT245BM also has this capability via a dedicated pin (SI/WU). FT8U245AM does not have support for remote wake up. To use the remote wake-up capability, enable this option by checking the box.



Plug and Play (FT8U232AM, FT232BM)

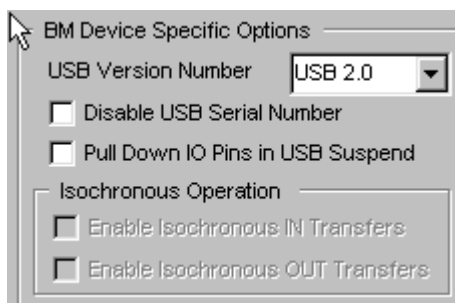
Windows Plug and Play (PNP) - for general purpose USB to RS232 serial converters it is desirable that the device tries to recognise any attached secondary peripheral that conforms to the Windows Plug and Play standard. These devices are typically modems, or serial mice. On recognition of such a device Windows will attempt to identify and load the drivers for the secondary peripheral. For other application areas PNP is not usually required so this option should be turned off.



BM Device Specific Options

The FT232BM and FT245BM 2nd generation devices have some additional options compared to the AM devices. These options can be controlled via the optional EEPROM.

- USB Version Number - the BM series devices have the option to set the USB version number to USB 1.1 or USB 2.0. This has no effect on the functionality of the device, however for USB compliance testing we recommend that you choose the USB2.0 setting. For AM series devices the version number is hard coded to USB 1.1
- Disable the USB serial number - this allows multiple identical USB devices to be plugged into the same host PC, however, it is then impossible for the USB drivers to uniquely identify the devices. We recommend that for most applications, this option is not selected.
- Pull Down IO Pins in USB suspend - BM devices feature a Power Control Pin (PWREN#). This pin can be used to control power to external peripheral circuitry via a P-Channel MOSFET. When used in this way, select this option to minimize the USB suspend current otherwise leave this option unselected. See the BM data sheets, application schematics and designers guide for more details.
- Isochronous Mode - BM series devices have the option of transmitting / receiving data packets using USB isochronous (streaming) transfer instead of the default mode which uses USB Bulk data transfer. However, special drivers / technical support is required to use this mode. **The web edition of MProG has these options disabled as careless use of this feature can result in a device which can crash the PC under test.** Should you require this feature, please request a copy of MProG with ISO Transfers enabled from FTDI technical support.



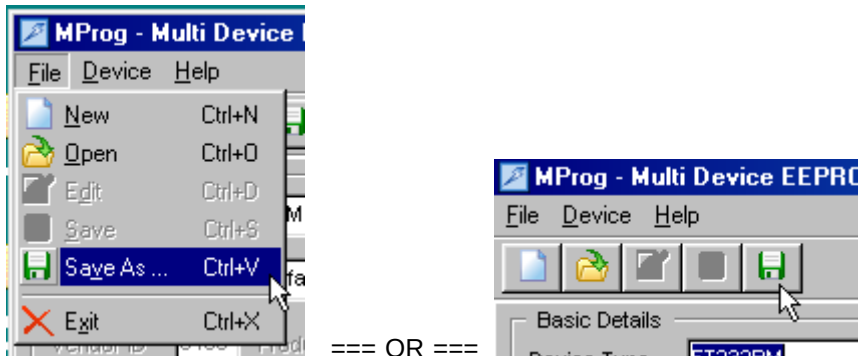
Programming Options

If the "Only Program Blank Devices" Box is checked then MProg will only program any blank devices it finds. If the box is unchecked, MProg will program any blank devices it finds and also re-program any programmed devices it finds.

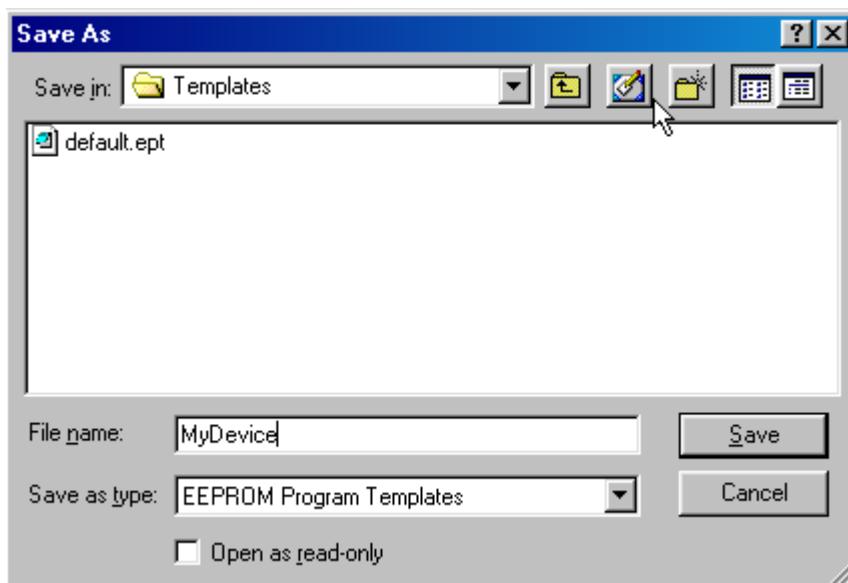


Saving the Programming Template.

When the Programming Template is defined, use the "Save As" Menu Item or Toolbar button..



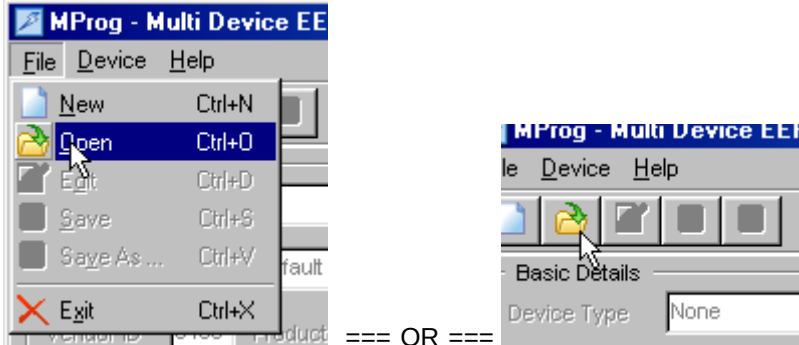
Enter a file name for the programming template, perhaps your product name, and click on "Save"



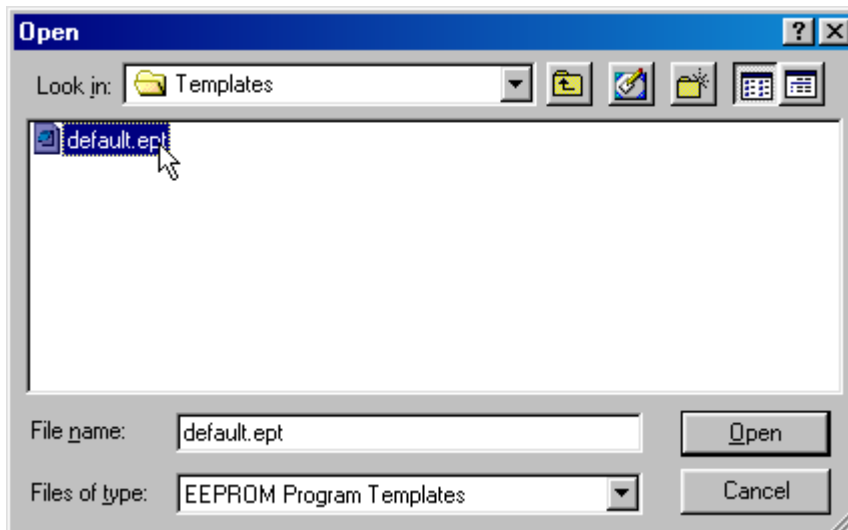
Mprog will save the Programming template and switch into Program mode.

3.2 Modifying an Existing Programming Template

To modify an existing template, use "Open" from the "File" menu or click on the "File Open" button on the toolbar.



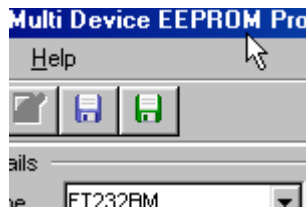
This will launch the File Open dialog box. Select the template you wish to modify and click on "Open"



MProg will load the template parameters, however it will be in the Program mode.



Click on the "Edit Mode" button to switch MProg from Program mode to Edit mode. You will now be able to modify the template parameters.

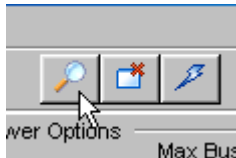


Changing any parameter will activate the "Save" button on the toolbar. When the changes are complete click on the "Save" button to save the changes, or on the "Save As" button to save the resulting template to a different file name. Clicking on "Save" or "Save As" will return MProg to Program mode. To make further changes, click on the "Edit Mode" button again.

4 Program Mode Functions

4.1 Scanning for Devices

To scan the USB bus for available FT232 / FT245 devices, click on the "Scan for Devices" button on the toolbar.

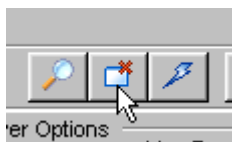


MProg will display the number of blank devices it finds and the number of programmed devices it finds on the host PC, for example 0 and 1 as shown below.

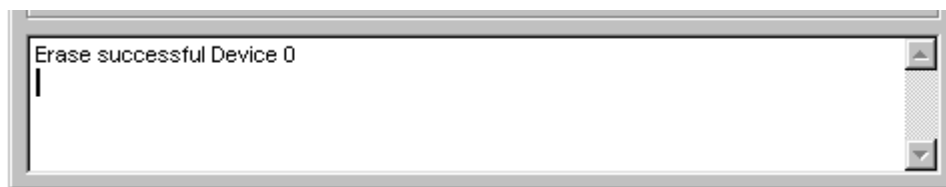


4.2 Erasing Devices

To Erase devices you must be in Program mode. Clicking on the "Erase All Devices" button will erase the EEPROM contents of ALL FT232/FT245 devices connected to the Host PC.

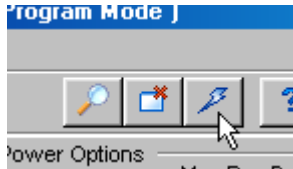


A message will be displayed for each device found and successfully erased. An example is shown below.

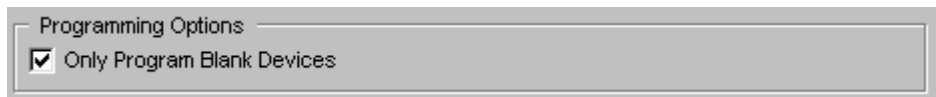


4.3 Programming Devices

To program a device MProg must be in Program mode. Click on the "Program Devices" button on the toolbar.



If the "Only Program Blank Devices" box is checked, then MProg will not try and reprogram any pre-programmed FT232/FT245 devices it finds. If this selection is unchecked then MProg will attempt to program every FT232/FT245 device it finds on the host PC.



Note : to change the "Only Program Blank Devices" box you must first enter Edit Mode by clicking on the "Edit Mode" button. After changing the option, click on the "Save" button which will return you to Program mode.

After programming is complete MProg will display a message showing how many devices it has programmed.



5 Appendix

5.1 Customising FTDI's VCP Drivers

1. Changing the Ftdibus.inf file.

If you are using a USB VID and PID which is different from the default (VID = 0403 hex and PID = 6001 hex) FTDI ones, you must edit the Ftdibus.inf file using a text editor such as NotePad. You will need to add extra lines into the [FtdiHw] and [ControlFlags] sections containing the new VID and PID. You should also add a description of your product into the [Strings] section as shown. *Do not change any other parts of the supplied INF files unless you are an expert and capable of debugging the results for yourself.*

The following example shows a new device with a VID of 1234 (hex) and a PID of 5678(hex) which we shall call "My New USB Device". In this example, the additional lines are shown in red to highlight them.

```
[FtdiHw]
%USB\VID_0403&PID_8372.DeviceDesc%=FtdiBus,USB\VID_0403&PID_8372
%USB\VID_0403&PID_6001.DeviceDesc%=FtdiBus,USB\VID_0403&PID_6001
%USB\VID_1234&PID_5678.DeviceDesc%=FtdiBus,USB\VID_1234&PID_5678
```

```
[ControlFlags]
ExcludeFromSelect=USB\VID_0403&PID_8372
ExcludeFromSelect=USB\VID_0403&PID_6001
ExcludeFromSelect=USB\VID_1234&PID_5678
```

```
[Strings]
Ftdi="FTDI"
DriversDisk="FTDI USB Drivers Disk"
USB\VID_0403&PID_8372.DeviceDesc="USB Serial Converter"
USB\VID_0403&PID_6001.DeviceDesc="USB High Speed Serial Converter"
USB\VID_1234&PID_5678.DeviceDesc="My New USB Device"
WINUN="Software\Microsoft\Windows\CurrentVersion\Uninstall"
FtdiBus.SvcDesc="USB Serial Converter Driver"
```

2. Changing the Ftdiport.inf file.

If you are using a USB VID and PID which is different from the default (VID = 0403 hex and PID = 6001 hex) FTDI ones, you must edit the Ftdiport.inf File using a text editor such as NotePad. You will need to add an extra line into the [FtdiHw] section containing the new VID and PID. You should also add a description of your product into the [Strings] section as shown. *Do not change any other parts of the supplied INF files unless you are an expert and capable of debugging the results for yourself.*

The following example shows a new device with a VID of 1234 (hex) and a PID of 5678(hex) which we shall call "My New USB Device". In this example, the additional lines are shown in red to highlight them.

```
[FtdiHw]
%VID_0403&PID_8372.DeviceDesc%=FtdiPort,FTDIBUS\COMPORT&VID_0403&PID_8372
%VID_0403&PID_6001.DeviceDesc%=FtdiPort232,FTDIBUS\COMPORT&VID_0403&PID_6001
%VID_1234&PID_5678.DeviceDesc%=FtdiPort232,FTDIBUS\COMPORT&VID_1234&PID_5678
```

```
[Strings]
FTDI="FTDI"
DriversDisk="FTDI USB Drivers Disk"
PortsClassName = "Ports (COM & LPT)"
VID_0403&PID_8372.DeviceDesc="USB Serial Port"
VID_0403&PID_6001.DeviceDesc="USB Serial Port"
VID_1234&PID_5678.DeviceDesc="My New USB Device"
FtdiPort.SvcDesc="USB Serial Port Driver"
SerEnum.SvcDesc="Serenum Filter Driver"
```

3. Changing the VCP Uninstaller INI files.

If you are using a USB VID and PID which is different from the default (VID = 0403 hex and PID = 6001 hex) FTDI ones, you must edit the uninstaller Ftdiunin.ini and Ftdiun2k.ini files using a text editor such as NotePad. In the [Uninstall] section of each file, edit the line Device=VID_0403&PID_6001 to reflect the actual VID and PID used.

The following example shows a new device with a VID of 1234 (hex) and a PID of 5678(hex). In this example, the altered line is shown in red to highlight it.

=== In File Ftdiunin.ini ===

```
[Uninstall]
Device=VID_1234&PID_5678
InfFiles=FTDIBUS,FTDIPIPORT,FTSERMOU
Key=FTDICOMM
```

=== In File Ftdiun2k.ini ===

```
[Uninstall]
Device=VID_1234&PID_5678
Converter=FTDIBUS
Serial=FTSER2K
Key=FTDICOMM
```

5.2 Customising FTDI's D2XX Drivers

1. Changing the INF files.

If you are using a USB VID and PID which is different from the default (VID = 0403 hex and PID = 6001 hex) FTDI ones, you must edit the FTD2XX.INF File using a text editor such as NotePad. You will need to add extra lines into the [FtdiHw] and [ControlFlags] sections containing the new VID and PID. You should also add a description of your product into the [Strings} section as shown. *Do not change any other parts of the supplied INF files unless you are an expert and capable of debugging*

the results for yourself.

The following example shows a new device with a VID of 1234 (hex) and a PID of 5678(hex) which we shall call "My New Device". In this example, the additional lines are shown in red to highlight them.

```
[FtdiHw]
%USB\VID_0403&PID_6001.DeviceDesc%=FTD2XX,USB\VID_0403&PID_6001
%USB\VID_0123&PID_4567.DeviceDesc%=FTD2XX,USB\VID_0123&PID_4567

[ControlFlags]
ExcludeFromSelect=USB\VID_0403&PID_6001
ExcludeFromSelect=USB\VID_0123&PID_4567

[Strings]
Ftdi="FTDI"
USB\VID_0403&PID_6001.DeviceDesc="FTDI FT8U2XX Device"
USB\VID_1234&PID_4567.DeviceDesc="My New Device"
FTD2XX.SvcDesc="FTD2XX.SYS FT8U2XX device driver"
WINUN="Software\Microsoft\Windows\CurrentVersion\Uninstall"
DriversDisk="FTDI FTD2XX Drivers Disk"
```

2. Changing the D2XX Uninstaller INI files.

If you are using a USB VID and PID which is different from the default (VID = 0403 hex and PID = 6001 hex) FTDI ones, you must edit the uninstaller FTD2XXUN.INI File using a text editor such as NotePad. In the [Uninstall] section edit the line Device=VID_0403&PID_6001 to reflect the actual VID and PID used.

The following example shows a new device with a VID of 1234 (hex) and a PID of 5678(hex). In this example, the altered line is shown in red to highlight it.

```
[Uninstall]
Device=VID_1234&PID_5678
Converter=FTD2XX
```

Index

- C -

Creating Programming Templates 9

Customising D2XX Drivers 19

Customising VCP Drivers 18

- D -

D2XX Drivers Customising 19

Device Erasing 16

Device Programming 17

Device Scanning 16

- E -

Erasing Devices 16

- F -

FT232/245 EEPROM Functions 5

- I -

Installing MProg 1

Installing the D2XX drivers 2

- M -

Modes of Operation 7

Modifying Programming Templates 14

- P -

Programming Devices 17

Programming Templates - Creating 9

Programming Templates - Modifying 14

- S -

Scanning for Devices 16

- V -

VCP Drivers - Customising 18