

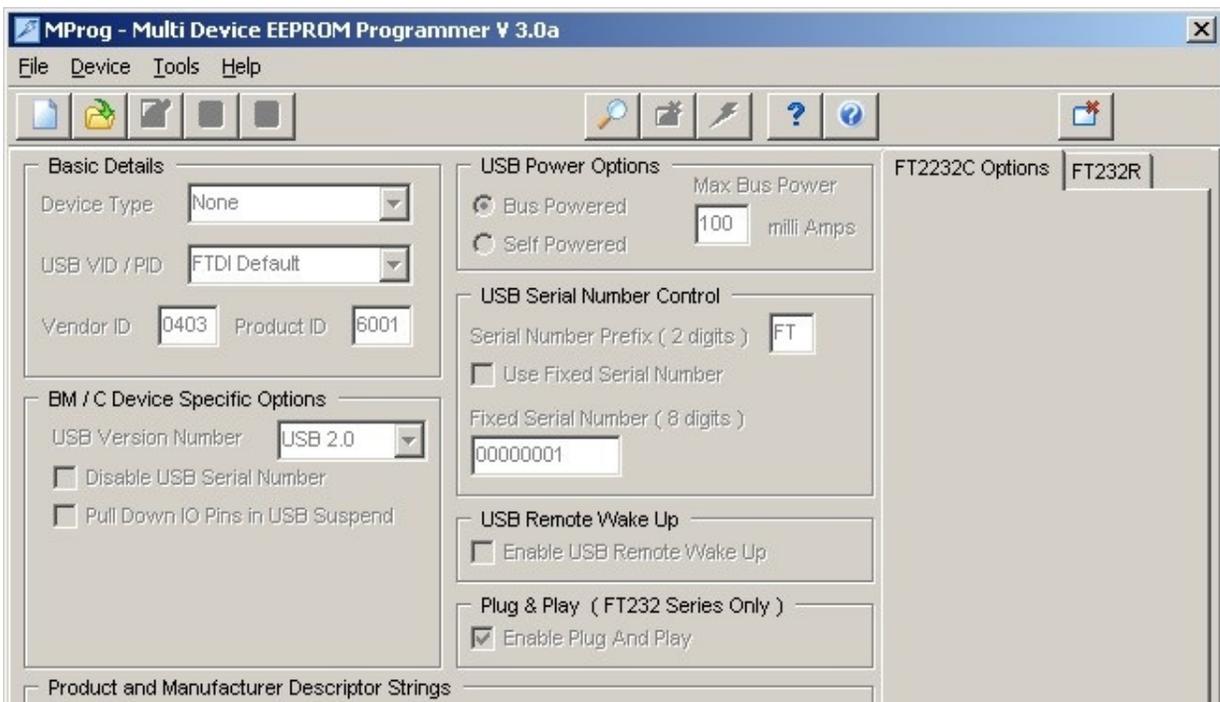
with FTDI devices. Devices supported by the latest version of MProg are:

- [FT232RL & FT232RQ](#)
- [FT245RL & FT245RQ](#)
- [FT2232D, FT2232C & FT2232L](#)
- [FT232BM, FT232BL & FT232BQ](#)
- [FT245BM, FT245BL & FT245BQ](#)
- [FT8U232AM](#)
- [FT8U245AM](#)

MProg is available as a free download from the [Utilities](#) page of the [FTDI website](#).

These devices support 93C46 EEPROMs organised in 16-bit words, except for the R series devices which have an internal EEPROM. The BM and C devices also support 93C56 and 93C66 EEPROMs organised in 16-bit words. When fitted, an external EEPROM allows customisation of the device default parameters such as [VID](#), [PID](#), [Serial Number](#), [Manufacturer Descriptor](#) and [Product Descriptor](#). MProg provides a fast and simple way of programming these 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.



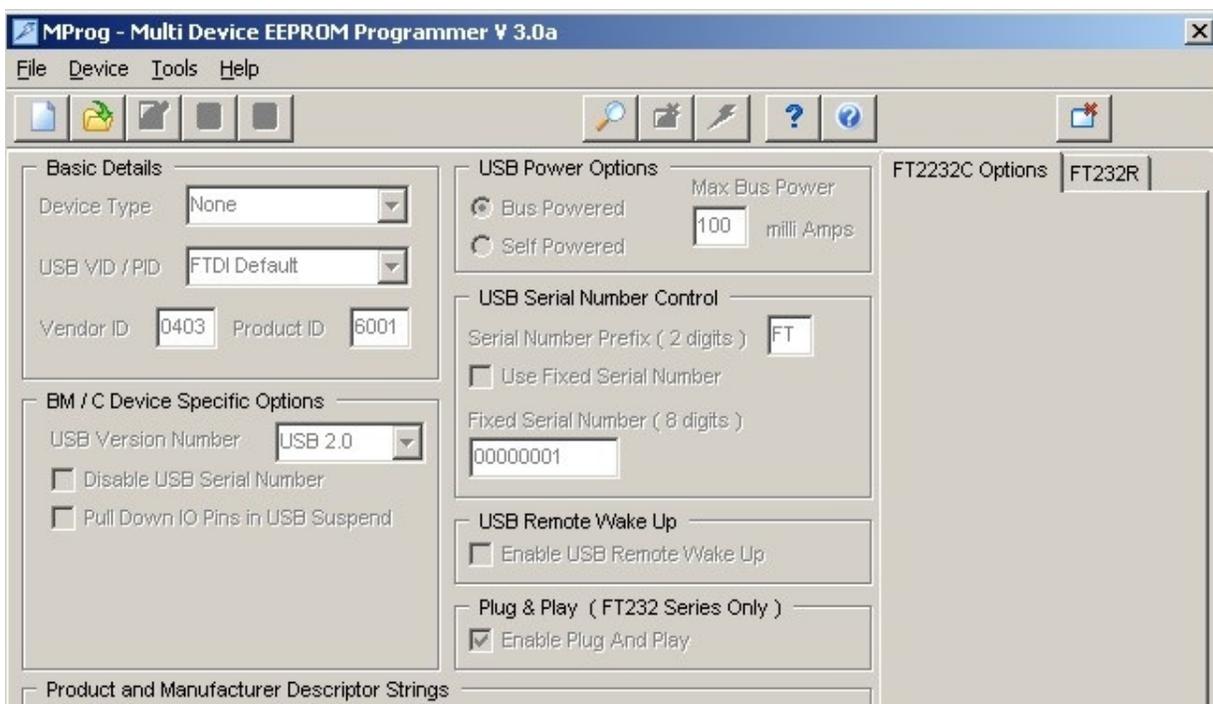
with FTDI devices. Devices supported by the latest version of MProg are:

- [FT232RL & FT232RQ](#)
- [FT245RL & FT245RQ](#)
- [FT2232D, FT2232C & FT2232L](#)
- [FT232BM, FT232BL & FT232BQ](#)
- [FT245BM, FT245BL & FT245BQ](#)
- [FT8U232AM](#)
- [FT8U245AM](#)

MProg is available as a free download from the [Utilities](#) page of the [FTDI website](#).

These devices support 93C46 EEPROMs organised in 16-bit words, except for the R series devices which have an internal EEPROM. The BM and C devices also support 93C56 and 93C66 EEPROMs organised in 16-bit words. When fitted, an external EEPROM allows customisation of the device default parameters such as [VID](#), [PID](#), [Serial Number](#), [Manufacturer Descriptor](#) and [Product Descriptor](#). MProg provides a fast and simple way of programming these 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.

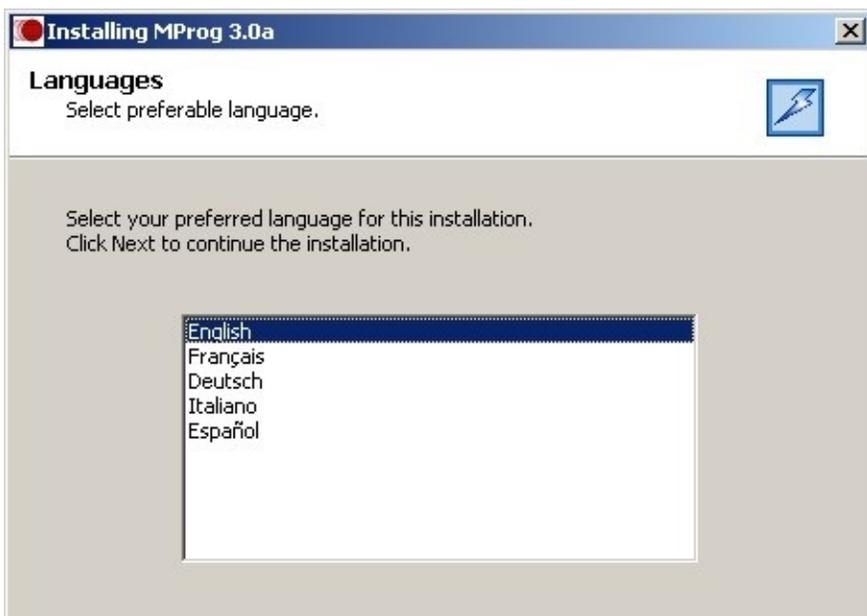


PC using the Add/Remove Programs utility in the Windows control Panel. MProg uses FTDI's D2XX drivers, the current versions of which can be downloaded free of charge from the [Drivers page](#) of the [FTDI website](#). A link to the D2XX drivers page is included in the MProg installation directory.

The latest version of MProg can be downloaded from the [Utilities page](#) of the [FTDI website](#) in a zipped setup executable. Extract the file, run the setup executable file and the following screen should appear:



Click Next and the installation program will prompt you to select the language you want to use for the rest of the setup process. The default value is English.

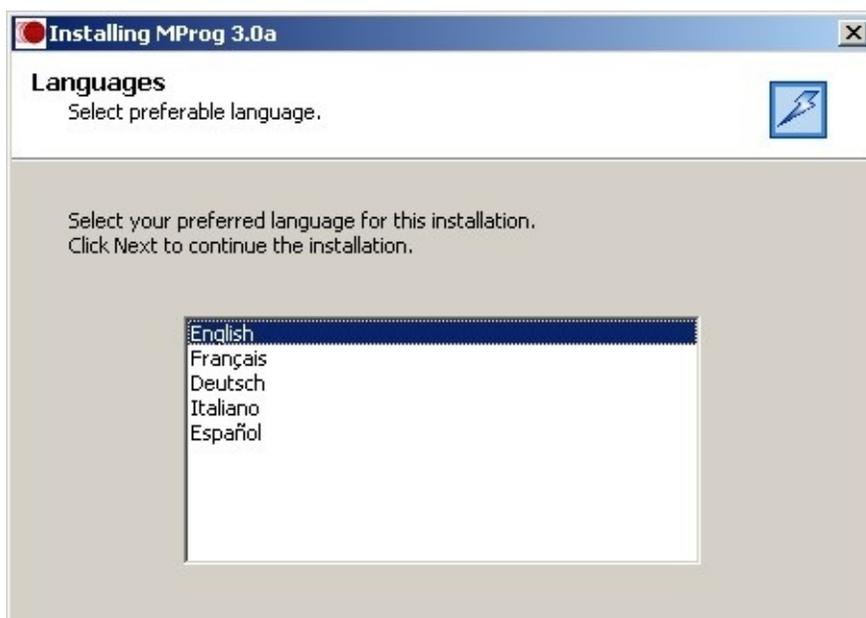


PC using the Add/Remove Programs utility in the Windows control Panel. MProg uses FTDI's D2XX drivers, the current versions of which can be downloaded free of charge from the [Drivers page](#) of the [FTDI website](#). A link to the D2XX drivers page is included in the MProg installation directory.

The latest version of MProg can be downloaded from the [Utilities page](#) of the [FTDI website](#) in a zipped setup executable. Extract the file, run the setup executable file and the following screen should appear:



Click Next and the installation program will prompt you to select the language you want to use for the rest of the setup process. The default value is English.



in the appropriate [driver installation guide](#) available from the [FTDI web site](#). Sample instructions are given below for a Windows 2000 installation, but other Windows versions are very similar.

While MProg uses the D2XX interface, FT232R, FT245R and FT2232 devices that are programmed to load a virtual COM port will also load the COM port from the CDM drivers. As BM and AM series devices do not have an EEPROM setting for this, they will install a COM port automatically.

·If a device of the same type has been installed on your machine before and the drivers that are about to be installed are different from those installed already, the original drivers need to be uninstalled. Please refer to the appropriate [driver installation guide](#) for details of how to do this.

·Download the latest available CDM drivers from the [FTDI web site](#) and unzip them to a location on your PC.

·Connect the device to a spare USB port on your PC. If the device is based on an FT2232 device, the Microsoft composite device driver is automatically loaded first.



·Once the composite driver has been installed the Found New Hardware Wizard will launch. Click "Next" to proceed with the installation.



in the appropriate [driver installation guide](#) available from the [FTDI web site](#). Sample instructions are given below for a Windows 2000 installation, but other Windows versions are very similar.

While MProg uses the D2XX interface, FT232R, FT245R and FT2232 devices that are programmed to load a virtual COM port will also load the COM port from the CDM drivers. As BM and AM series devices do not have an EEPROM setting for this, they will install a COM port automatically.

·If a device of the same type has been installed on your machine before and the drivers that are about to be installed are different from those installed already, the original drivers need to be uninstalled. Please refer to the appropriate [driver installation guide](#) for details of how to do this.

·Download the latest available CDM drivers from the [FTDI web site](#) and unzip them to a location on your PC.

·Connect the device to a spare USB port on your PC. If the device is based on an FT2232 device, the Microsoft composite device driver is automatically loaded first.



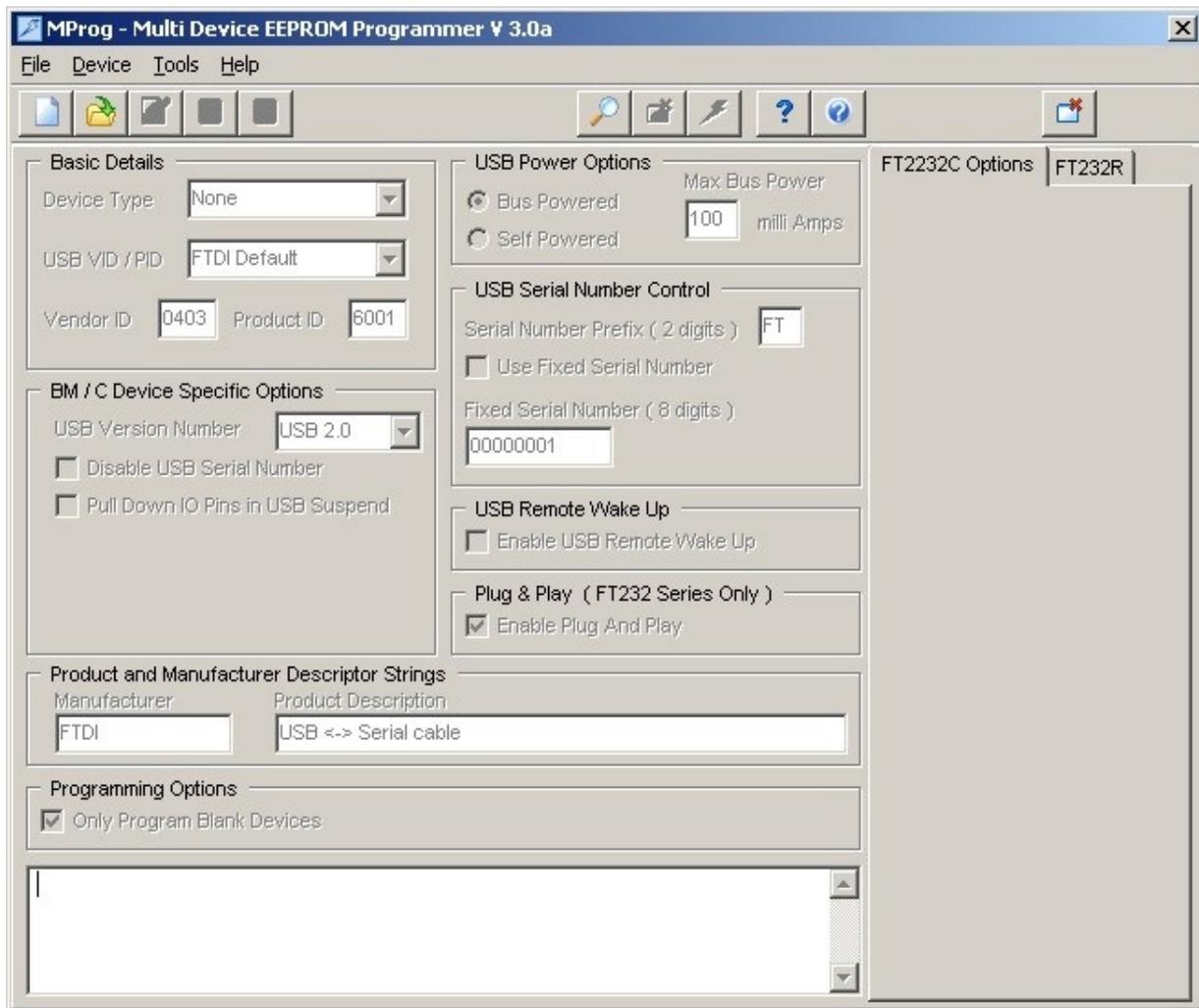
·Once the composite driver has been installed the Found New Hardware Wizard will launch. Click "Next" to proceed with the installation.



MProg programming parameters are saved in files referred to here as EEPROM templates. Once defined, these EEPROM 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 EEPROM 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.



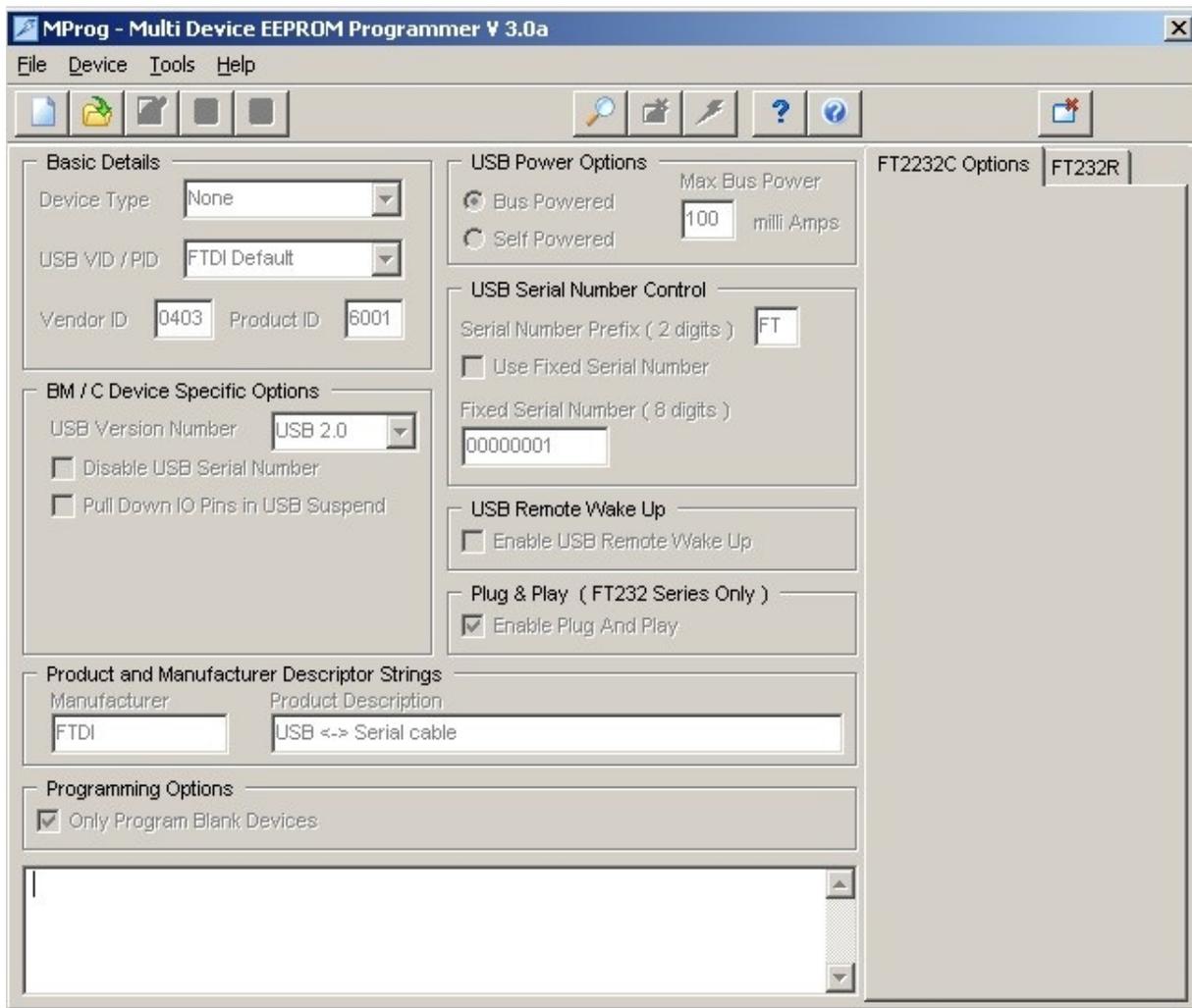
User choices in Idle Mode are:

- Create a new EEPROM template. This will change the mode to Edit Mode and allow the user to select the parameters for their project. The new EEPROM Template must then be saved to disk which will then put MProg into Program Mode.

MProg programming parameters are saved in files referred to here as EEPROM templates. Once defined, these EEPROM 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 EEPROM 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:

- Create a new EEPROM template. This will change the mode to Edit Mode and allow the user to select the parameters for their project. The new EEPROM Template must then be saved to disk which will then put MProg into Program Mode.

Template button on the toolbar



or select New from the File Menu.



The EEPROM template can define parameters in the following categories:

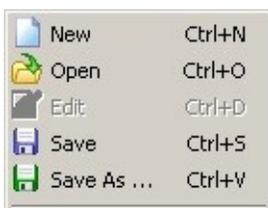
#### Basic Details

- [USB Power Options](#)
- [USB Serial Number Control](#)
- [USB Remote Wake Up](#)
- [Plug & Play \(FT232 Series Only\)](#)
- [BM/C Device Specific Options](#)
- [FT2232C Device Options](#)
- [FT232R Device Options](#)
- [FT245R Device Options](#)
- [FT2232D Device Options](#)
- [Product and Manufacturer Descriptor Strings](#)
- [Programming Options](#)

When the EEPROM template is defined, use the "Save As" toolbar button



or select Save As ... from the File menu.



Template button on the toolbar



or select New from the File Menu.



The EEPROM template can define parameters in the following categories:

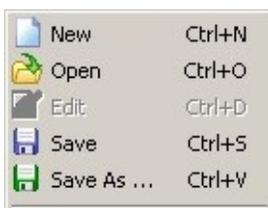
## Basic Details

- [USB Power Options](#)
- [USB Serial Number Control](#)
- [USB Remote Wake Up](#)
- [Plug & Play \(FT232 Series Only\)](#)
- [BM/C Device Specific Options](#)
- [FT2232C Device Options](#)
- [FT232R Device Options](#)
- [FT245R Device Options](#)
- [FT2232D Device Options](#)
- [Product and Manufacturer Descriptor Strings](#)
- [Programming Options](#)

When the EEPROM template is defined, use the "Save As" toolbar button



or select Save As ... from the File menu.



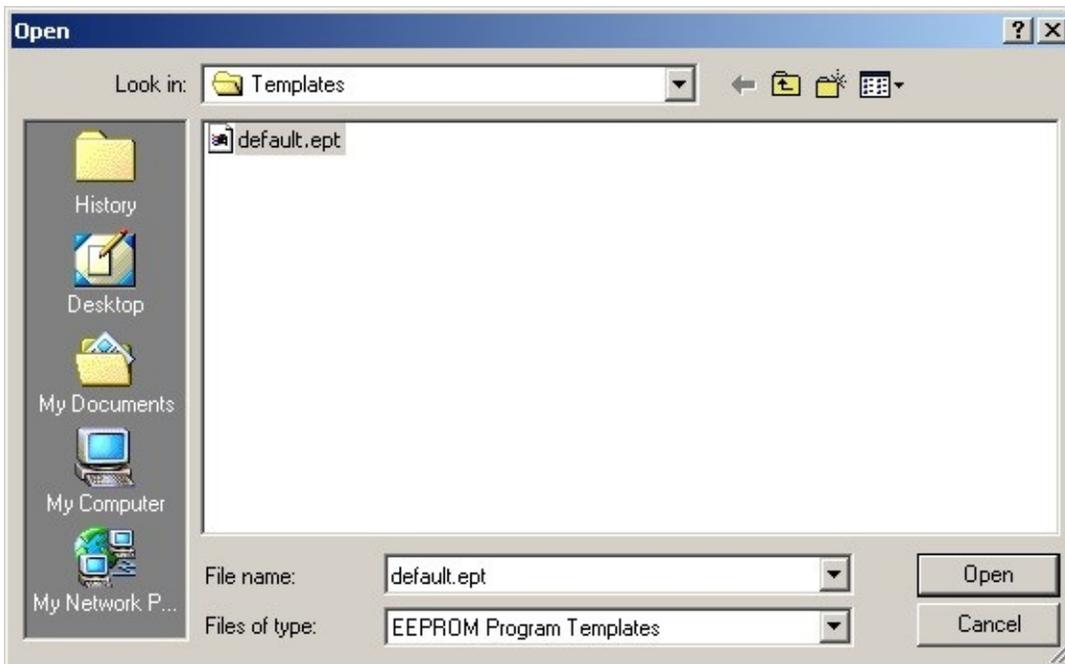
the toolbar



or select "Open" from the "File" menu.



This will launch the 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 on the toolbar 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

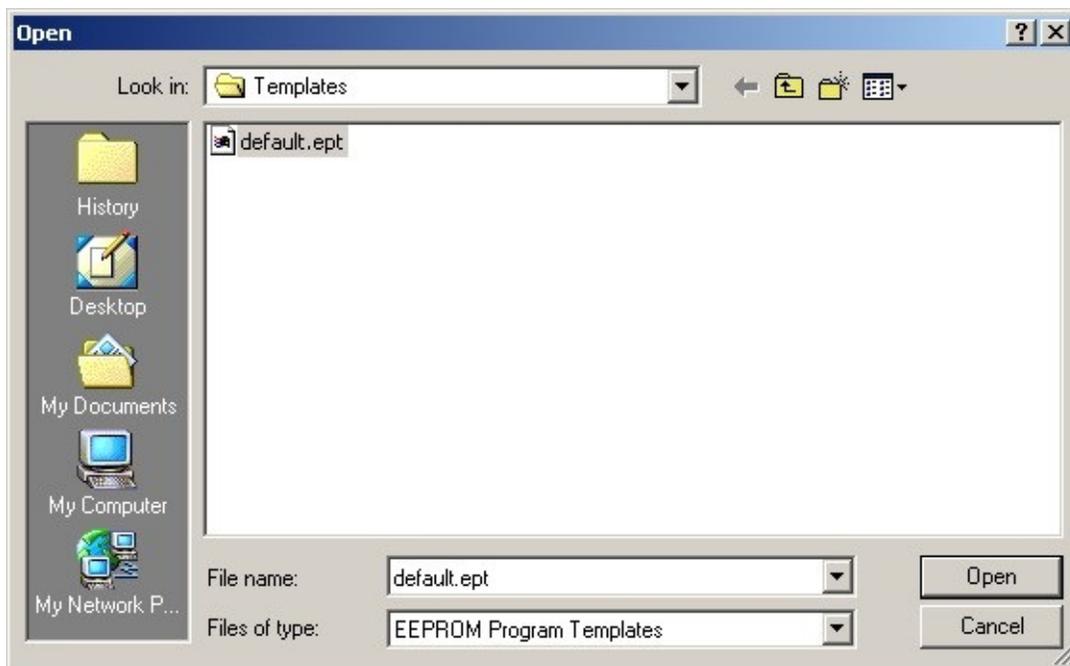
the toolbar



or select "Open" from the "File" menu.



This will launch the 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 on the toolbar 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

section are:

Basic Details

Device Type: FT232RL

USB VID / PID: FTDI Default

Vendor ID: 0403 Product ID: 6001

## 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.

## USB VID / PID

The choices available from the combo-box for VID and PID are

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

- FTDI Supplied PID. 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 do not have a unique Product ID assigned for you by FTDI then please e-mail [support2@ftdichip.com](mailto:support2@ftdichip.com) with your request. There is no charge for this service to bona-fide FTDI customers.

- Other VID & PID. If you have your own USB Vendor ID and Product ID for your device, then this option will allow the user to enter both the USB VID and USB PID for your product in the boxes provided.

**Note that if non-standard VID and/or PID values are used the driver INF and INI files must be modified to accommodate the new VID and PID combination. Please refer to the Appendix of this manual for the modifications that should be made.**

section are:



## 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.

## USB VID / PID

The choices available from the combo-box for VID and PID are

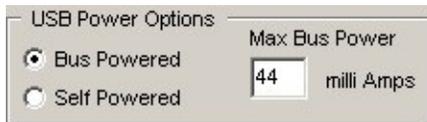
- FTDI Default. If you use the FTDI default values, the VID and PID are fixed and you do not need to enter any values.

- FTDI Supplied PID. 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 do not have a unique Product ID assigned for you by FTDI then please e-mail [support2@ftdichip.com](mailto:support2@ftdichip.com) with your request. There is no charge for this service to bona-fide FTDI customers.

- Other VID & PID. If you have your own USB Vendor ID and Product ID for your device, then this option will allow the user to enter both the USB VID and USB PID for your product in the boxes provided.

**Note that if non-standard VID and/or PID values are used the driver INF and INI files must be modified to accommodate the new VID and PID combination. Please refer to the Appendix of this manual for the modifications that should be made.**

Options section.



Select Bus Powered (the product gets its power from USB) or Self Powered (the product has its own power supply) from the power options. The Max 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 Max Bus Power field. A summary of the rules and options are given below:

### All USB Devices

- 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 500mA in the Max Bus Power field.

- 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 its 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 USB Devices

These fall into two categories:

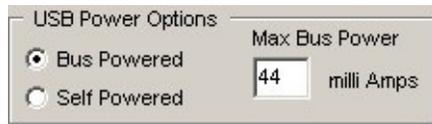
- Low power bus powered products. These products consume 100mA or less from the USB bus and can be plugged into any USB host or hub port.

- 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 (hubs that have their own power supply) as bus powered USB hubs can only supply 100mA per hub port.

### Self Powered USB Devices

These fall into two categories:

Options section.



Select Bus Powered (the product gets its power from USB) or Self Powered (the product has its own power supply) from the power options. The Max 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 Max Bus Power field. A summary of the rules and options are given below:

### All USB Devices

- 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 500mA in the Max Bus Power field.

- 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 its 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 USB Devices

These fall into two categories:

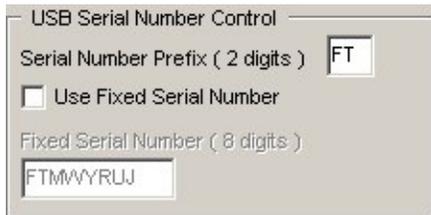
- Low power bus powered products. These products consume 100mA or less from the USB bus and can be plugged into any USB host or hub port.

- 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 (hubs that have their own power supply) as bus powered USB hubs can only supply 100mA per hub port.

### Self Powered USB Devices

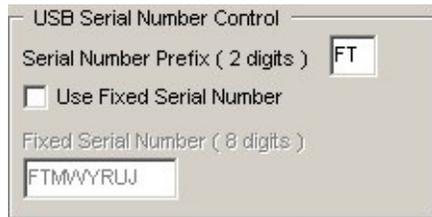
These fall into two categories:

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. The device serial number can be set through the USB Serial Number Control panel.



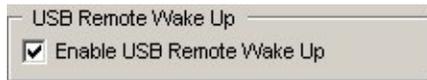
MProg can generate unique serial numbers automatically for you (recommended). These consist of a 2 letter prefix (e.g. FT) followed by 6 alphanumeric digits generated using an algorithm based on the date and time of programming the device. It is also possible to fix the serial number to a fixed 8-digit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

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. The device serial number can be set through the USB Serial Number Control panel.

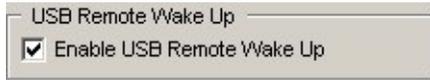


MProg can generate unique serial numbers automatically for you (recommended). These consist of a 2 letter prefix (e.g. FT) followed by 6 alphanumeric digits generated using an algorithm based on the date and time of programming the device. It is also possible to fix the serial number to a fixed 8-digit 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 suspend (sleep) state via the RI# pin. FT245BM also has this capability via a dedicated pin (SI/WU), as does the FT2232C (SI/WUA, SI/WUB). FT8U245AM does not have support for remote wake up. To use the remote wake-up capability, enable this option by checking the box.



USB suspend (sleep) state via the RI# pin. FT245BM also has this capability via a dedicated pin (SI/WU), as does the FT2232C (SI/WUA, SI/WUB). FT8U245AM does not have support for remote wake up. To use the remote wake-up capability, enable this option by checking the box.



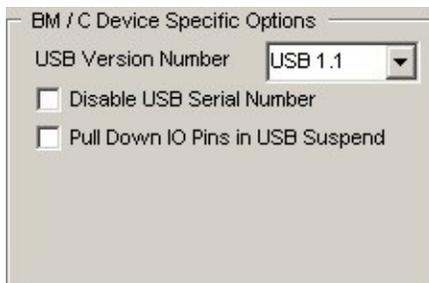
converters it is desirable that the device tries to recognise any attached secondary peripheral that conforms to the Windows Plug and Play standard. These devices are typically modems, or serial mice. On recognition of such a device Windows will attempt to identify and load the drivers for the secondary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



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.



generation device have some additional options compared to the AM devices. These options can be controlled via the optional EEPROM.



·USB Version Number. The BM and C 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 USB 2.0 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 and C 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 product data sheets, application schematics and designers guides for more details.

generation device have some additional options compared to the AM devices. These options can be controlled via the optional EEPROM.



·USB Version Number. The BM and C 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 USB 2.0 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 and C 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 product data sheets, application schematics and designers guides for more details.

[BM/C Device Specific Options](#) section. It also has the following additional options which can be configured in the external EEPROM. These options will only be displayed when the device type in basic details is set to FT2232C.



·Each channel can be independently configured as FT232BM-style USB UART or a FT245BM-style USB FIFO.

·Each channel can also be individually configured in Fast Opto-Isolated Serial Interface mode. See the FT2232C device datasheet (available from the [Datasheets page](#) of the [FTDI website](#)) for more information on these modes.

·High Current I/O's. Setting this option will enable the high output drive level. The UART/FIFO IO pins will drive out at 12mA instead of their normal 4mA.

·The driver that is installed and used by each channel can be set by selecting Virtual COM Port or D2XX Direct option (not currently supported).

[BM/C Device Specific Options](#) section. It also has the following additional options which can be configured in the external EEPROM. These options will only be displayed when the device type in basic details is set to FT2232C.



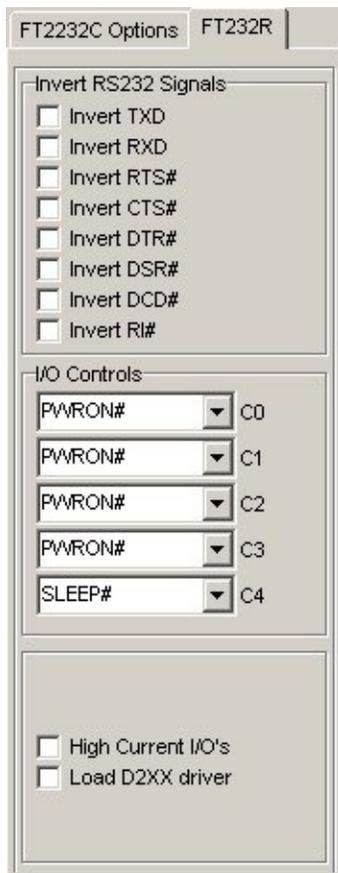
·Each channel can be independently configured as FT232BM-style USB UART or a FT245BM-style USB FIFO.

·Each channel can also be individually configured in Fast Opto-Isolated Serial Interface mode. See the FT2232C device datasheet (available from the [Datasheets page](#) of the [FTDI website](#)) for more information on these modes.

·High Current I/O's. Setting this option will enable the high output drive level. The UART/FIFO IO pins will drive out at 12mA instead of their normal 4mA.

·The driver that is installed and used by each channel can be set by selecting Virtual COM Port or D2XX Direct option (not currently supported).

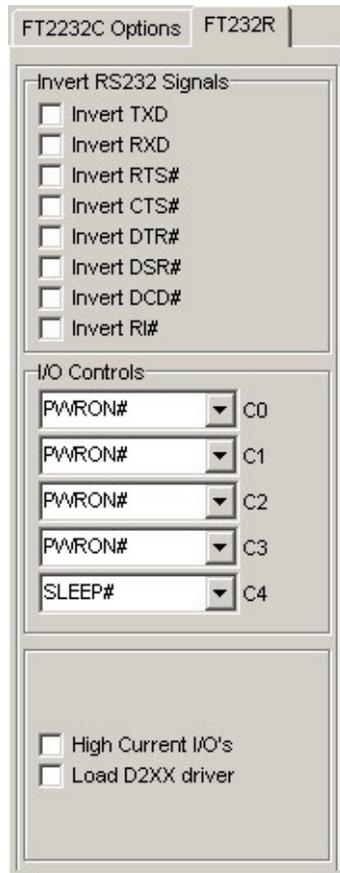
inverted and the CBUS pins to be configured through combo-boxes.



Like the FT2232C, the FT232R supports High Current IOs. Setting this option will enable the high output drive level. The UART IO pins will drive out at 12mA instead of their normal 4mA.

The check box to load the D2XX driver rather than the VCP driver is currently not supported.

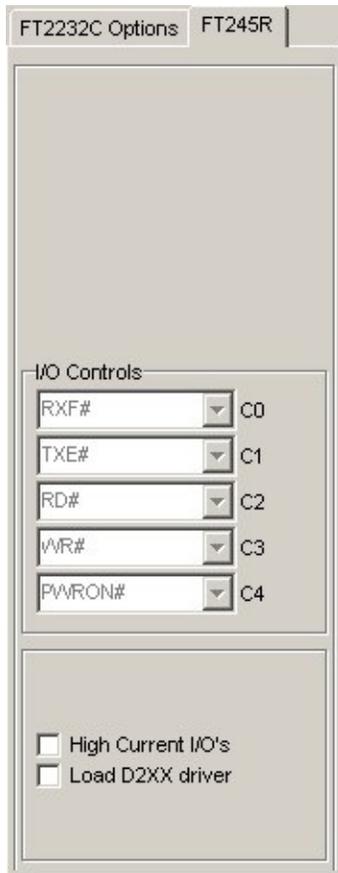
inverted and the CBUS pins to be configured through combo-boxes.



Like the FT232C, the FT232R supports High Current IOs. Setting this option will enable the high output drive level. The UART IO pins will drive out at 12mA instead of their normal 4mA.

The check box to load the D2XX driver rather than the VCP driver is currently not supported.

configuration of the CBUS pins as they are all required.



Like the FT2232C, the FT232R supports High Current IOs. Setting this option will enable the high output drive level. The FIFO IO pins will drive out at 12mA instead of their normal 4mA.

The check box to load the D2XX driver rather than the VCP driver is currently not supported.

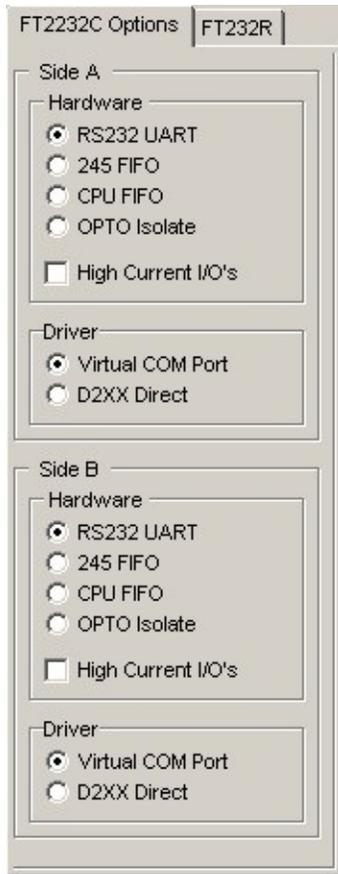
configuration of the CBUS pins as they are all required.

The screenshot shows a configuration window for the FT245R device. At the top, there are two tabs: "FT2232C Options" and "FT245R". The "FT245R" tab is selected. Below the tabs is a large empty rectangular area. Underneath this area is a section titled "I/O Controls" which contains five dropdown menus, each with a label to its right: "RXF#" (C0), "TXE#" (C1), "RD#" (C2), "WR#" (C3), and "PWRON#" (C4). Below the "I/O Controls" section are two checkboxes: "High Current I/O's" and "Load D2XX driver", both of which are currently unchecked.

Like the FT2232C, the FT232R supports High Current IOs. Setting this option will enable the high output drive level. The FIFO IO pins will drive out at 12mA instead of their normal 4mA.

The check box to load the D2XX driver rather than the VCP driver is currently not supported.

the [BM/C Device Specific Options](#) section and all of the [FT2232C Device Options](#), but with an additional "CPU FIFO" option made available.



·Each channel can be independently configured as FT232BM-style USB UART or a FT245BM-style USB FIFO.

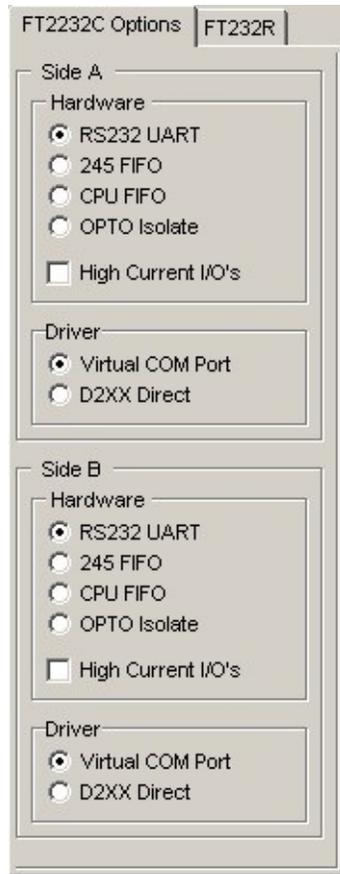
·Each channel can be individually configured in CPU FIFO mode.

·Each channel can also be individually configured in Fast Opto-Isolated Serial Interface mode. See the FT2232C device datasheet (available from the [Datasheets page](#) of the [FTDI website](#)) for more information on these modes.

·High Current I/O's. Setting this option will enable the high output drive level. The UART/FIFO IO pins will drive out at 12mA instead of their normal 4mA.

·The driver that is installed and used by each channel can be set by selecting Virtual COM Port or D2XX Direct option (not currently supported).

the [BM/C Device Specific Options](#) section and all of the [FT2232C Device Options](#), but with an additional "CPU FIFO" option made available.



·Each channel can be independently configured as FT232BM-style USB UART or a FT245BM-style USB FIFO.

·Each channel can be individually configured in CPU FIFO mode.

·Each channel can also be individually configured in Fast Opto-Isolated Serial Interface mode. See the FT2232C device datasheet (available from the [Datasheets page](#) of the [FTDI website](#)) for more information on these modes.

·High Current I/O's. Setting this option will enable the high output drive level. The UART/FIFO IO pins will drive out at 12mA instead of their normal 4mA.

·The driver that is installed and used by each channel can be set by selecting Virtual COM Port or D2XX Direct option (not currently supported).

customisation of the manufacturer and product description strings the device uses.

Product and Manufacturer Descriptor Strings	
Manufacturer	Product Description
FTDI	USB <-> Serial Cable

·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"

customisation of the manufacturer and product description strings the device uses.

Manufacturer	Product Description
FTDI	USB <-> Serial Cable

·Manufacturer: the product manufacturer string e.g. "FTDI" - best kept short and abbreviated if possible.

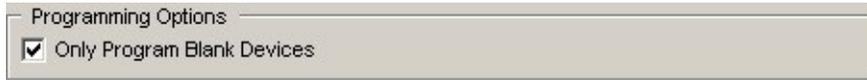
·Product Description: a brief description of the product, maybe including the part number e.g. "US232 USB to RS232 Converter"

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.

Programming Options

Only Program Blank Devices

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.

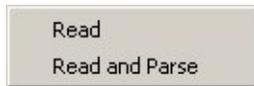


Programming Options

Only Program Blank Devices



Functions for reading the EEPROM contents can be found in the Tools menu.



The functions available through the toolbar are:

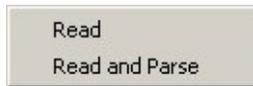
- Scan for Devices
- Erase All Existing Devices
- Program All Existing Devices

The read functions available through the Tools menu are:

- Read
- Read and Parse



Functions for reading the EEPROM contents can be found in the Tools menu.



The functions available through the toolbar are:

- Scan for Devices
- Erase All Existing Devices
- Program All Existing Devices

The read functions available through the Tools menu are:

- Read
- Read and Parse

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.

```
Number Of Blank Devices = 0  
Number Of Programmed Devices = 1  
|
```

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.

```
Number Of Blank Devices = 0  
Number Of Programmed Devices = 1  
|
```

"Erase All Existing Devices" button on the toolbar will erase the EEPROM contents of all FTDI devices connected to the host PC except any FT232R or FT245R devices.



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



"Erase All Existing Devices" button on the toolbar will erase the EEPROM contents of all FTDI devices connected to the host PC except any FT232R or FT245R devices.



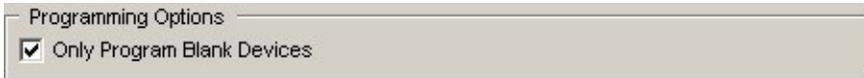
A message will be displayed for each device found and successfully erased. An example is shown below.



"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 FTDI devices it finds. If this selection is unchecked, then MProg will attempt to program every 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.



"Program Devices" button on the toolbar.

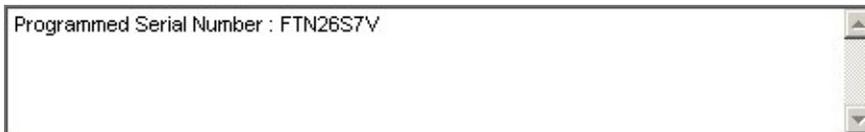


If the "Only Program Blank Devices" box is checked, then MProg will not try and reprogram any pre-programmed FTDI devices it finds. If this selection is unchecked, then MProg will attempt to program every device it finds on the host PC.



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.



its EEPROM has been reprogrammed. This is useful because the device only reads the EEPROM when it is enumerated on USB, so it forces the device to use the new EEPROM contents.



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



its EEPROM has been reprogrammed. This is useful because the device only reads the EEPROM when it is enumerated on USB, so it forces the device to use the new EEPROM contents.



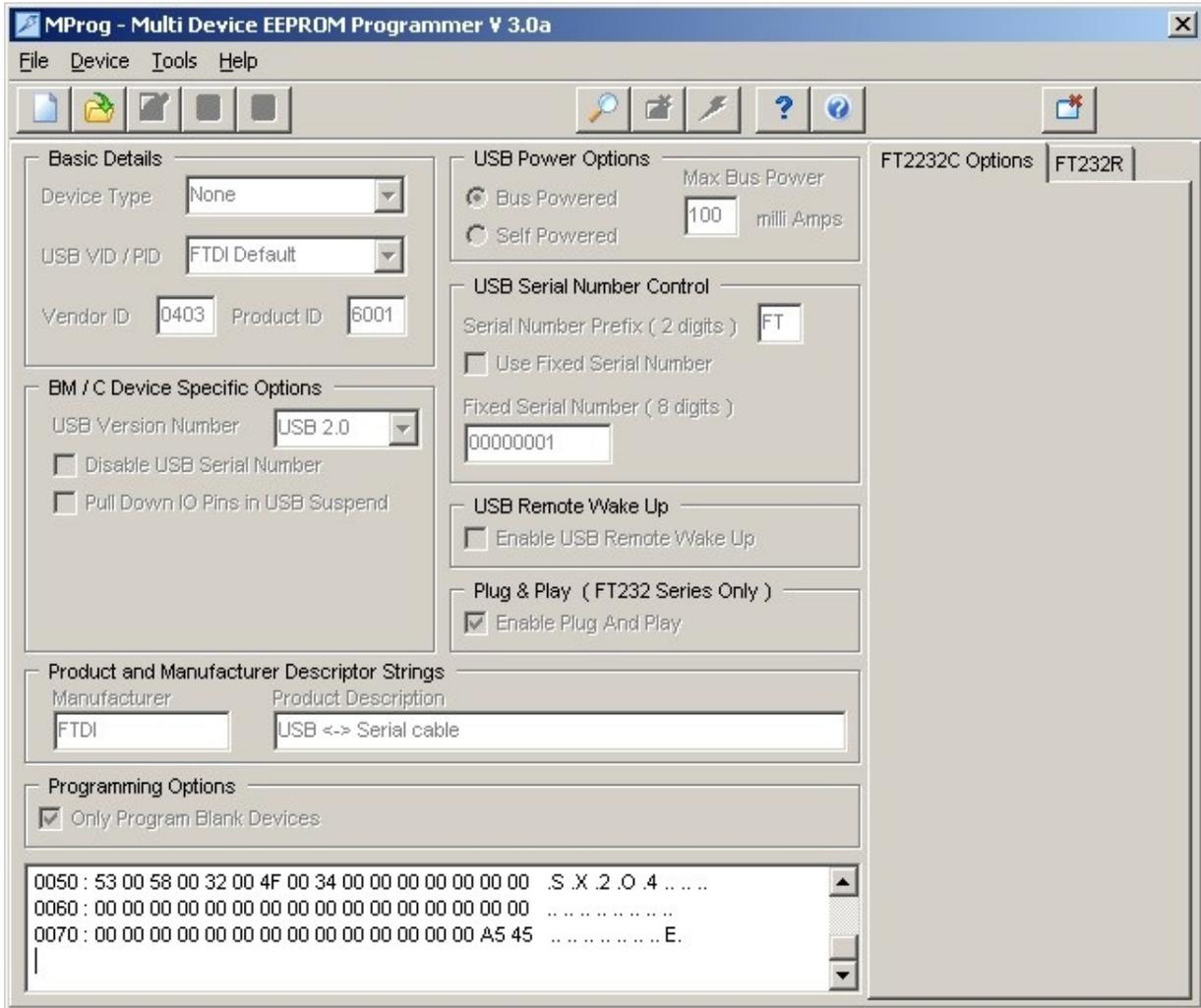
A message will be displayed for each device found and successfully re-enumerated. An example is shown below.



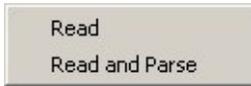
tools menu.



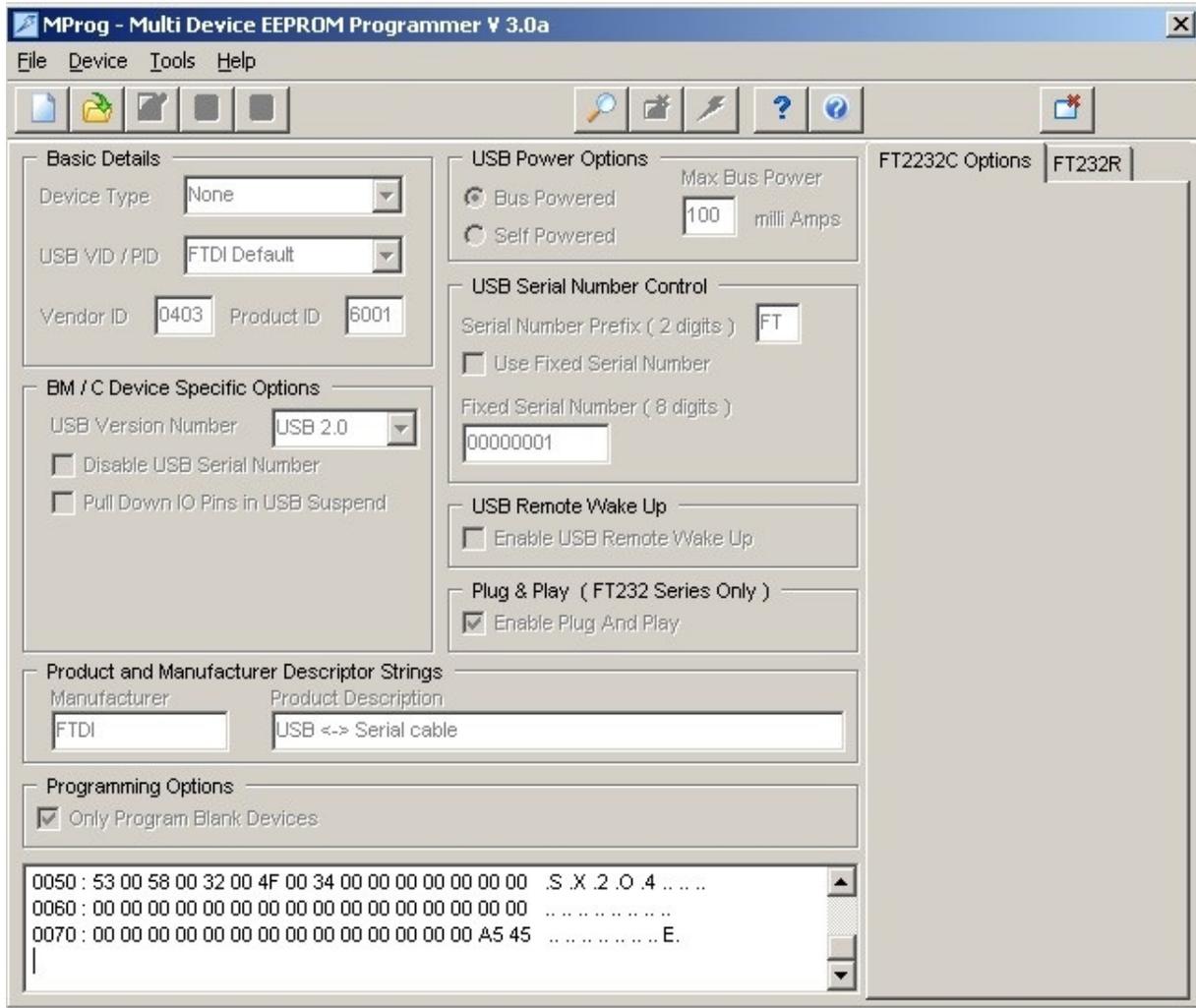
The programmed data will be displayed in the dialog window. For example:



tools menu.



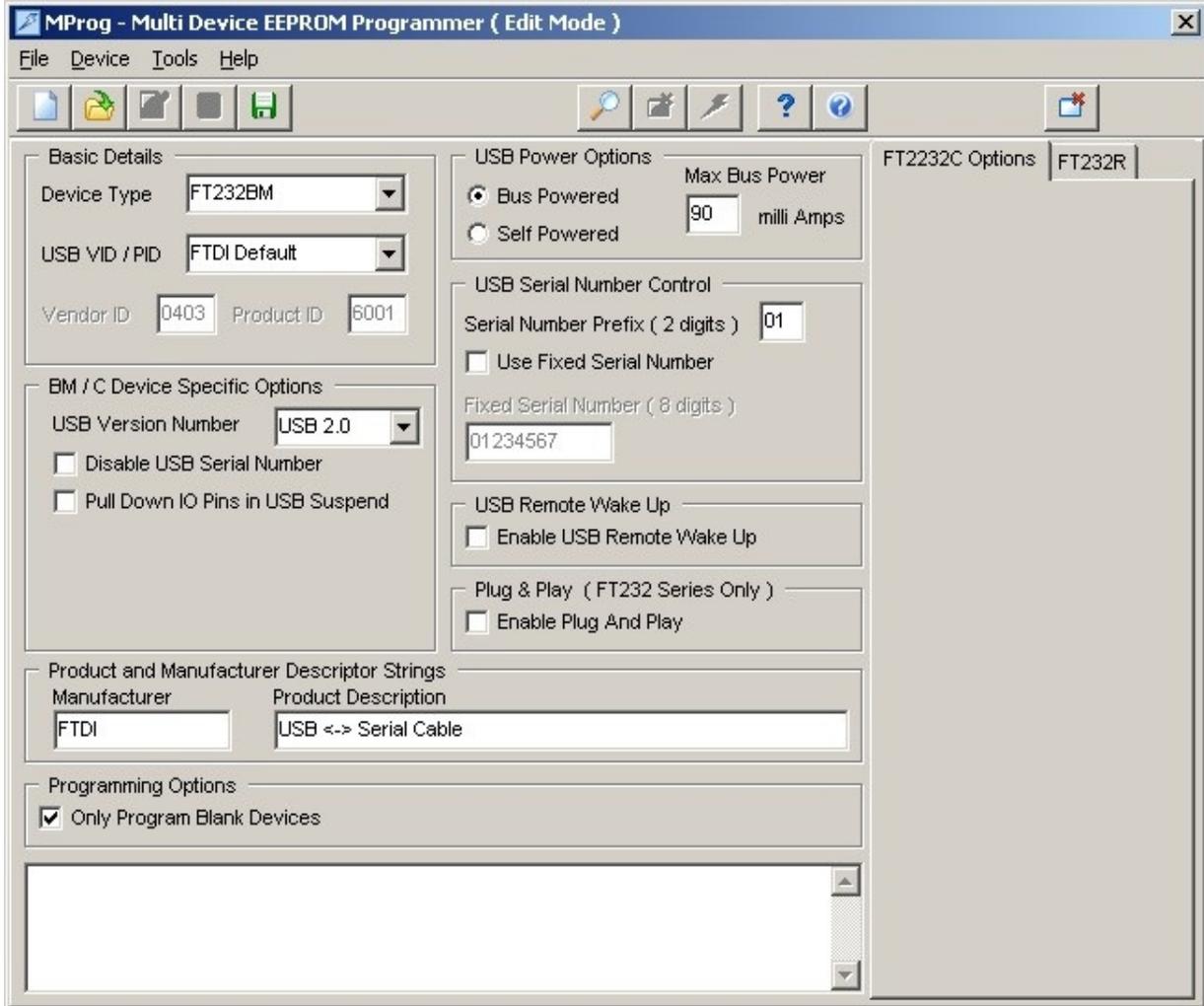
The programmed data will be displayed in the dialog window. For example:



Read and Parse from the tools menu.



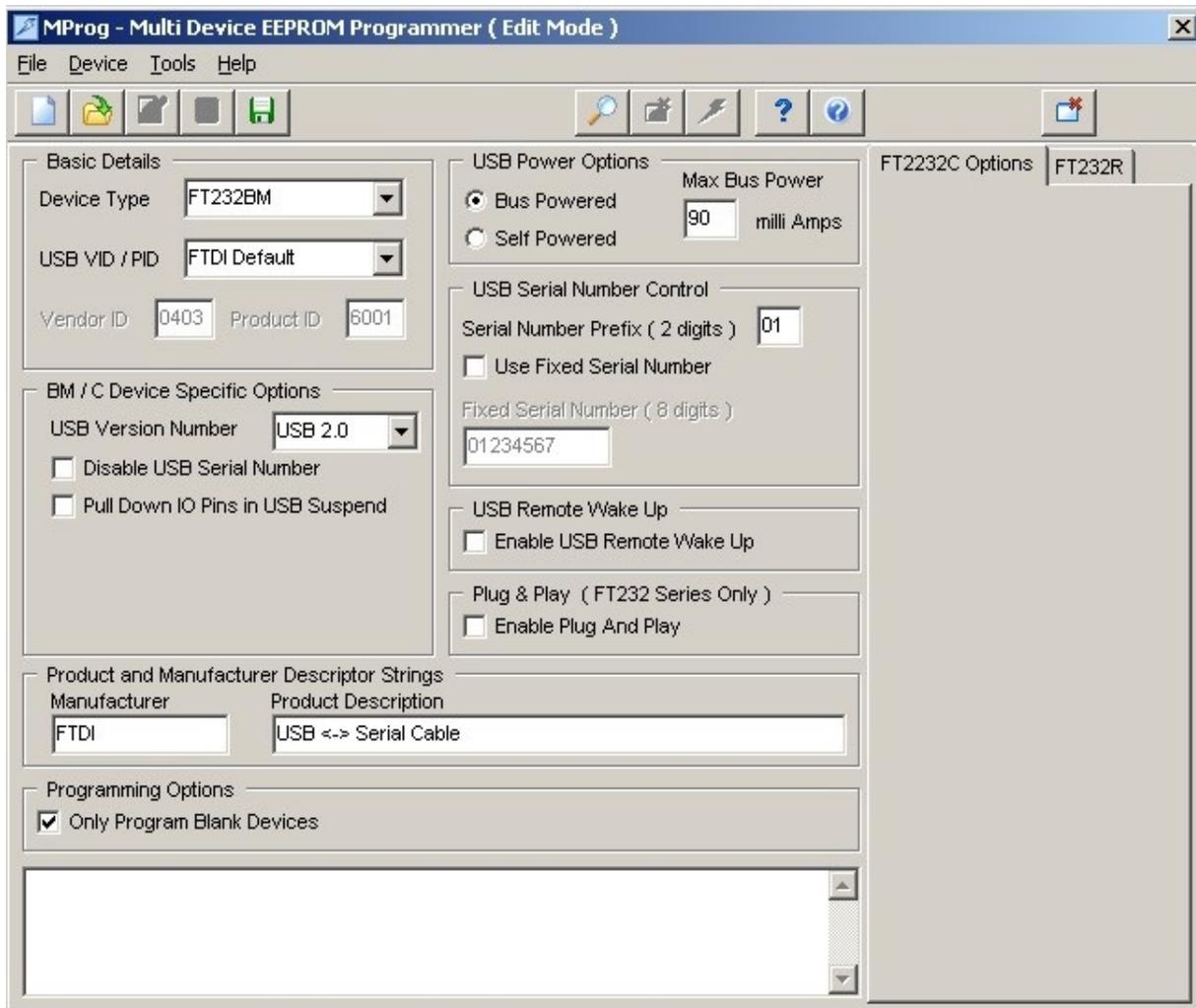
The programmed data will then be shown in each field in MProg. The device serial number will be shown, but will be greyed out.



Read and Parse from the tools menu.



The programmed data will then be shown in each field in MProg. The device serial number will be shown, but will be greyed out.



PID = 0x6001 and 0x6010), you must edit the FTDIBUS.INF file using a text editor like Notepad. You will need to add extra lines into the [FtdiHw] and [ControlFlags] sections to add 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 debugging the results for yourself.**

The following example shows a new device with a VID of 0x1234 and a PID of 0x5678 that shall call "My New USB Device". In this example, the additional lines are shown in red. If the device you are adding is an FT2232C or FT2232D, an entry must be added to the ports section.

[FtdiHw]

```
%USB\VID_0403&PID_6001.DeviceDesc%=FtdiBus,USB\VID_0403&PID_6001
%USB\VID_0403&PID_6010&MI_00.DeviceDesc%=FtdiBus,USB\VID_0403&PID_6010&MI_00
%USB\VID_0403&PID_6010&MI_01.DeviceDesc%=FtdiBus,USB\VID_0403&PID_6010&MI_01
%USB\VID_1234&PID_5678.DeviceDesc%=FtdiBus,USB\VID_1234&PID_5678
```

[Strings]

Ftdi="FTDI"

DriversDisk="FTDI USB Drivers Disk"

USB\VID\_0403&PID\_6001.DeviceDesc="USB Serial Converter"

USB\VID\_0403&PID\_6010&MI\_00.DeviceDesc="USB Serial Converter A"

USB\VID\_0403&PID\_6010&MI\_01.DeviceDesc="USB Serial Converter B"

USB\VID\_1234&PID\_5678.DeviceDesc="My New USB Device"

WINUN="Software\Microsoft\Windows\CurrentVersion\Uninstall"

FtdiBus.SvcDesc="USB Serial Converter Driver"

.6010), you must edit the FTDIBUS.INF file using a text editor such as  
ed to add extra lines into the [FtdiHw] and [ControlFlags] sections containing  
). You should also add a description of your product into the [Strings] section as

**other parts of the supplied INF files unless you are an expert and capable of  
ts for yourself.**

le shows a new device with a VID of 0x1234 and a PID of 0x5678 which we  
USB Device". In this example, the additional lines are shown in red to highlight  
ou are adding is an FT2232C or FT2232D, an entry must be created for each of

```
'ID_6001.DeviceDesc%=FtdiBus,USB\VID_0403&PID_6001  
'ID_6010&MI_00.DeviceDesc%=FtdiBus,USB\VID_0403&PID_6010&MI_00  
'ID_6010&MI_01.DeviceDesc%=FtdiBus,USB\VID_0403&PID_6010&MI_01  
'ID_5678.DeviceDesc%=FtdiBus,USB\VID_1234&PID_5678
```

USB Drivers Disk"

```
)_6001.DeviceDesc="USB Serial Converter"
```

```
)_6010&MI_00.DeviceDesc="USB Serial Converter A"
```

```
)_6010&MI_01.DeviceDesc="USB Serial Converter B"
```

```
)_5678.DeviceDesc="My New USB Device"
```

```
Microsoft\Windows\CurrentVersion\Uninstall"
```

```
USB Serial Converter Driver"
```

= 0x6001 and 0x6010), you must edit the FTDIPORT.INF file using a text editor. You will need to add an extra line into the [FtdiHw] section containing the new device. You should also add a description of your product into the [Strings] section as shown below.

**Do not change any other parts of the supplied INF files unless you are an expert debugging the results for yourself.**

The following example shows a new device with a VID of 0x1234 and a PID of 0x5678. The device shall call "My New USB Device". In this example, the additional lines are shown in red. If the device you are adding is an FT2232C or FT2232D, an entry must be added to the ports section.

```
[FtdiHw]
%VID_0403&PID_6001.DeviceDesc%=FtdiPort232,FTDIBUS\COMPORT&V
%VID_0403&PID_6010.DeviceDesc%=FtdiPort2232,FTDIBUS\COMPORT&
%VID_1234&PID_5678.DeviceDesc%=FtdiPort232,FTDIBUS\COMPORT&V
```

```
[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"
```

, you must edit the FTDIPORT.INF file using a text editor such as NotePad. Add an extra line into the [FtdiHw] section containing the new VID and PID. You add a description of your product into the [Strings] section as shown.

**Never edit any of the supplied INF files unless you are an expert and capable of doing so for yourself.**

The screenshot shows a new device with a VID of 0x1234 and a PID of 0x5678 which we call "My New USB Device". In this example, the additional lines are shown in red to highlight the changes. If you are adding a new device, such as an FT232C or FT232D, an entry must be created for each of

```
01.DeviceDesc%=FtdiPort232,FTDIBUS\COMPORT&VID_0403&PID_6001
10.DeviceDesc%=FtdiPort2232,FTDIBUS\COMPORT&VID_0403&PID_6010
78.DeviceDesc%=FtdiPort232,FTDIBUS\COMPORT&VID_1234&PID_5678
```

```
USB Drivers Disk"
Ports (COM & LPT)"
.DeviceDesc="USB Serial Port"
.DeviceDesc="USB Serial Port"
.DeviceDesc="My New USB Device"
USB Serial Port Driver"
Enumeration Filter Driver"
```

(VID = 0x0403 and PID = 0x6001), you must edit the uninstaller 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 0x1234 and a PID of 0x5678. In this example, the altered line is shown in red to highlight it.

```
[Uninstall]
```

```
Device=VID_1234&PID_5678
```

```
Converter=FTDIBUS
```

```
Serial=FTSER2K
```

```
InfFiles=FTDIBUS,FTDIPORT
```

```
Key=FTDICOMM
```

(VID = 0x0403 and PID = 0x6001), you must edit the uninstaller 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 0x1234 and a PID of 0x5678. In this example, the altered line is shown in red to highlight it.

```
[Uninstall]
```

```
Device=VID_1234&PID_5678
```

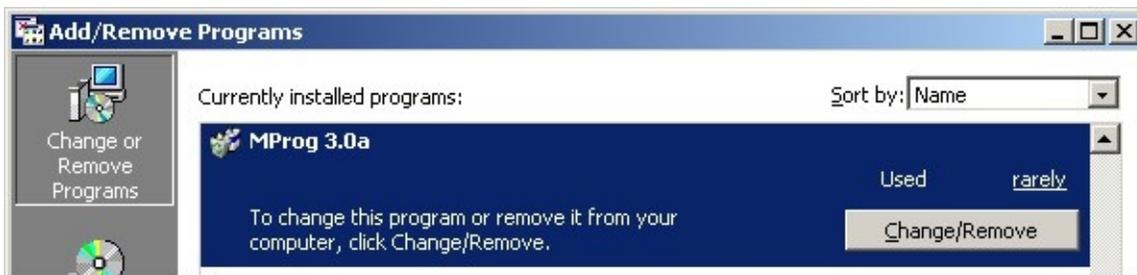
```
Converter=FTDIBUS
```

```
Serial=FTSER2K
```

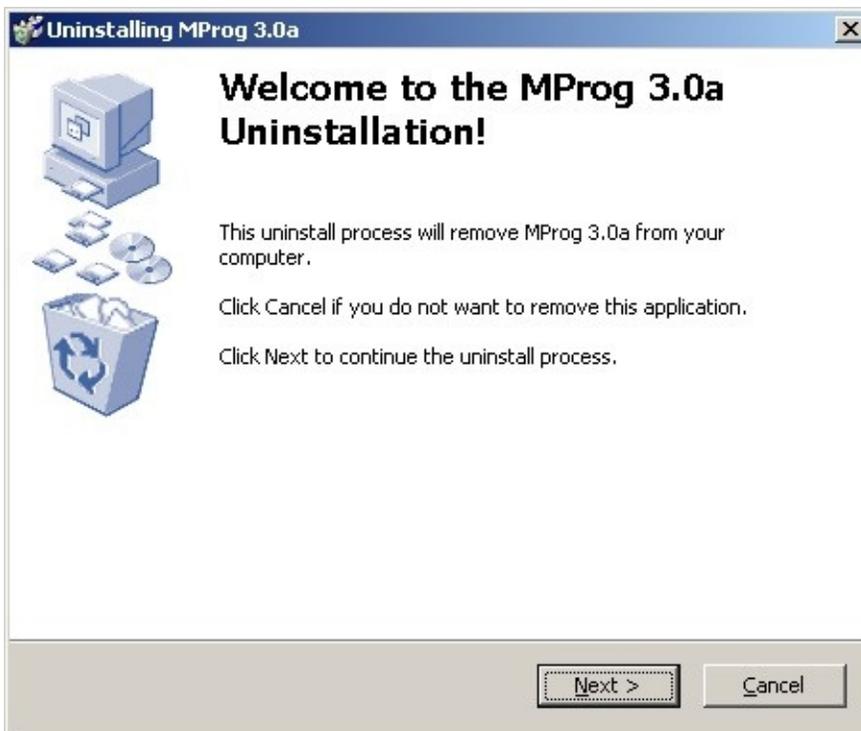
```
InfFiles=FTDIBUS,FTDIPOINT
```

```
Key=FTDICOMM
```

- Disconnect any FTDI devices attached to the PC.
- Open the Add/Remove Programs utility which is located in the Windows Control Panel.
- Select MProg from the list of installed programs and click Change/Remove. This will launch the MProg uninstaller.



- The user will be asked for confirmation to uninstall MProg. Click Next to proceed with the uninstallation or click Cancel to abort.

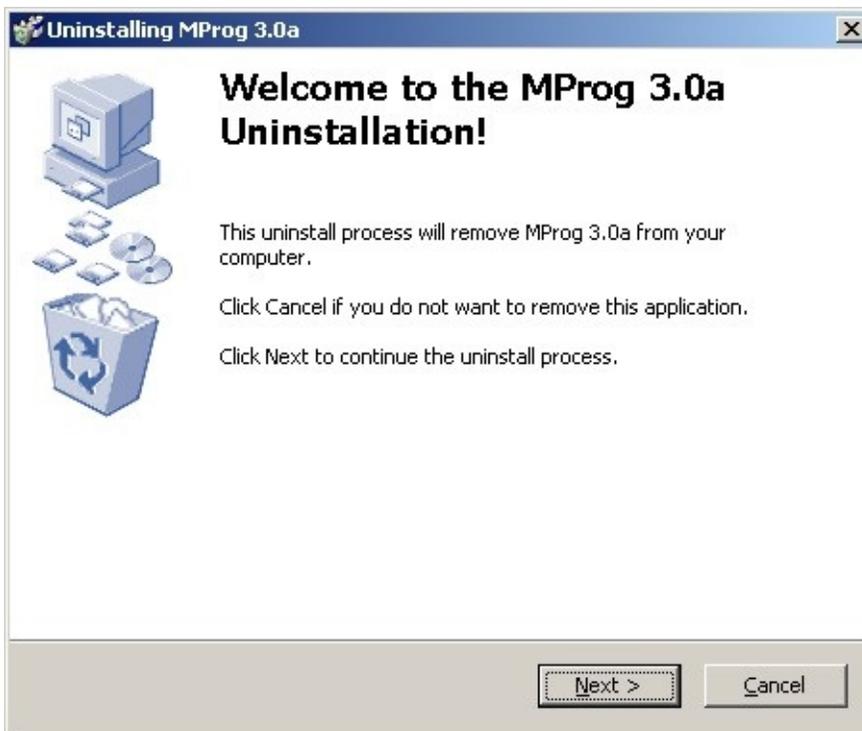


- The files and directories required for MProg will be removed. If there are files other than the original installation files present (e.g. user defined templates) the following screen will be displayed to give the option of deleting them.

- Disconnect any FTDI devices attached to the PC.
- Open the Add/Remove Programs utility which is located in the Windows Control Panel.
- Select MProg from the list of installed programs and click Change/Remove. This will launch the MProg uninstaller.



- The user will be asked for confirmation to uninstall MProg. Click Next to proceed with the uninstallation or click Cancel to abort.



- The files and directories required for MProg will be removed. If there are files other than the original installation files present (e.g. user defined templates) the following screen will be displayed to give the option of deleting them.