



# Application Note

## AN\_124

# User Guide for FTDI FT\_PROG Utility

Version 1.6

Issue Date: 2016-04-06

This application note explains how to use the FT\_Prog utility which provides a user interface to access any EEPROM used with FTDI devices.

FT\_Prog replaces the previous FTDI utility used for this purpose called MPROG

Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold FTDI harmless from any and all damages, claims, suits or expense resulting from such use.

### **Future Technology Devices International Limited (FTDI)**

Unit 1,2 Seaward Place, Glasgow G41 1HH, United Kingdom  
Tel.: +44 (0) 141 429 2777 Fax: + 44 (0) 141 429 2758  
E-Mail (Support): [support1@ftdichip.com](mailto:support1@ftdichip.com) Web: <http://www.ftdichip.com>  
Copyright © Future Technology Devices International Limited

## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
<b>2</b>	<b>Downloading and Installing FT_Prog .....</b>	<b>4</b>
<b>3</b>	<b>FT_Prog Basics .....</b>	<b>5</b>
3.1	Modes of Operation .....	5
<b>4</b>	<b>Edit Mode Functions .....</b>	<b>7</b>
4.1	Create a New EEPROM Template .....	7
4.2	Modify an Existing EEPROM Template .....	8
4.3	Save a Template.....	9
<b>5</b>	<b>Device EEPROM Parameters.....</b>	<b>10</b>
5.1	Chip_Details .....	10
5.2	USB_Device_Descriptor .....	10
5.3	USB_Config_Descriptors.....	11
5.4	USB_String_Descriptors .....	12
5.5	FT232R Hardware_Specific.....	13
5.6	FT245R Hardware_Specific .....	14
5.7	FT2232D Hardware_Specific.....	15
5.8	FT2232H Hardware_Specific.....	16
5.9	FT4232H Hardware_Specific.....	17
5.10	FT232H Hardware_Specific.....	18
5.11	FT4222H Hardware Specific.....	20
5.12	FT260 Hardware Specific .....	21
5.13	FT-X Hardware Specific .....	23
<b>6</b>	<b>Program Mode Function .....</b>	<b>25</b>
6.1	Scan for Devices .....	25
6.2	Erase Existing Devices.....	26
6.3	Use an Existing EEPROM Template .....	27
6.4	Program Existing Devices .....	29
6.5	Cycle Port .....	31
<b>7</b>	<b>Command Line Programming.....</b>	<b>32</b>

---

<b>7.1 Batch File Programming .....</b>	<b>36</b>
<b>8 Vinculum Programmer .....</b>	<b>37</b>
<b>9 Definition of Error Messages.....</b>	<b>41</b>
<b>10 Contact Information.....</b>	<b>44</b>
<b>Appendix A - References.....</b>	<b>45</b>
Document References .....	45
Acronyms and Abbreviations .....	45
<b>Appendix B – Revision History.....</b>	<b>46</b>

## 1 Introduction

FT\_Prog is a free EEPROM programming utility for use with FTDI devices. It is used for modifying EEPROM contents that store the FTDI device descriptors to customize designs.

FT\_Prog also includes the capability of programming the VNC1L and VNC2 firmware using a UART interface.

FT\_Prog is an enhanced version (and a replacement for) the previous utility used for this function called MPROG as well as the VProg application.

Devices supported by FT\_Prog are:

- [FT-X](#)
- [FT232RL & FT232RQ](#)
- [FT245RL & FT245RQ](#)
- [\\_FT2232DFT232BM, FT232BL & FT232BQ](#)
- [FT\\_245BM, FT245BL & FT245BQ](#)
- [FT8U232AM](#)
- [FT8U245AM](#)
- [FT2232H](#)
- [FT4232H](#)
- [FT232H](#)
- [FT4222H](#)
- [FT260](#)
- [VNC1L](#)
- [VNC2](#)

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

FTDI devices support 93C46 EEPROMs organized in 16-bit words, except for the R series of devices which have an internal EEPROM and H series which supports the larger devices. The BM, D and H devices also support 93C56 and 93C66 EEPROMs organized in 16-bit words.

When fitted, an external EEPROM allows customization of the device default parameters such as VID, PID, Serial Number, Manufacturer Descriptor and Product Descriptor. FT\_Prog provides a fast and simple way to program these parameters.

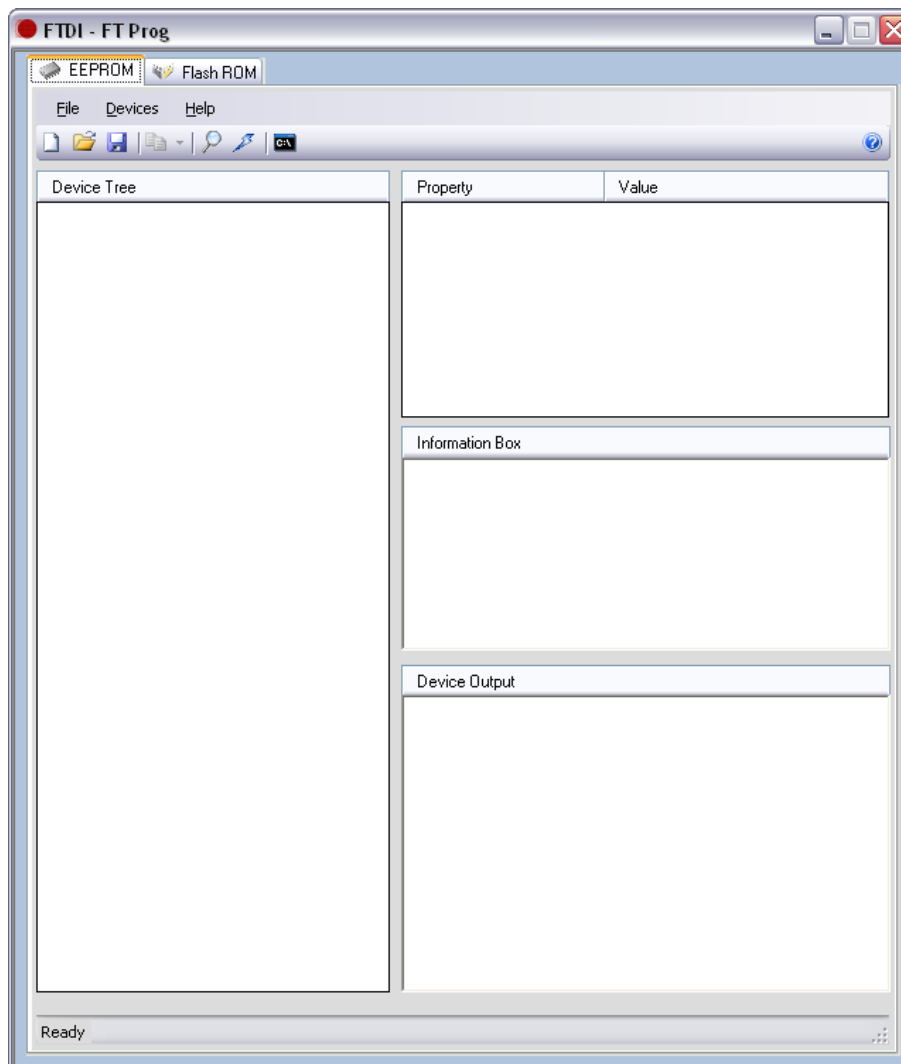
A convenient way of programming the EEPROM is to populate a blank device on the PCB during manufacturing assembly and then program it via USB during the manufacturing test process. All these devices support direct programming over USB using the FT\_Prog programming utility.

## 2 Downloading and Installing FT\_Prog

The latest version of FT\_Prog can be downloaded from the [Utilities](#) page of the [FTDI website](#) in a zipped executable file. Extract the file using an extraction tool. There are many available on the web such as Winzip or WinRAR. Most versions of Windows e.g. Win 7 come with these utilities pre-loaded. The application requires the latest version of the D2XX drivers installed, an installation guide can be found on the FTDI website under the [Install Guides Section](#). For Win 7 please refer to [AN 119](#), for Win 8 refer to [AN 234](#), and for Win 10 please refer to [AN 396](#).

**Note:** FT\_Prog requires the Microsoft .NET Framework 3.5 (or later) which can be downloaded from the [Microsoft website](#). If your system does not have .Net 3.5 installed you must download and install this from the above link before running the application.

Run the executable file by double clicking on the .exe and the following screen appears:



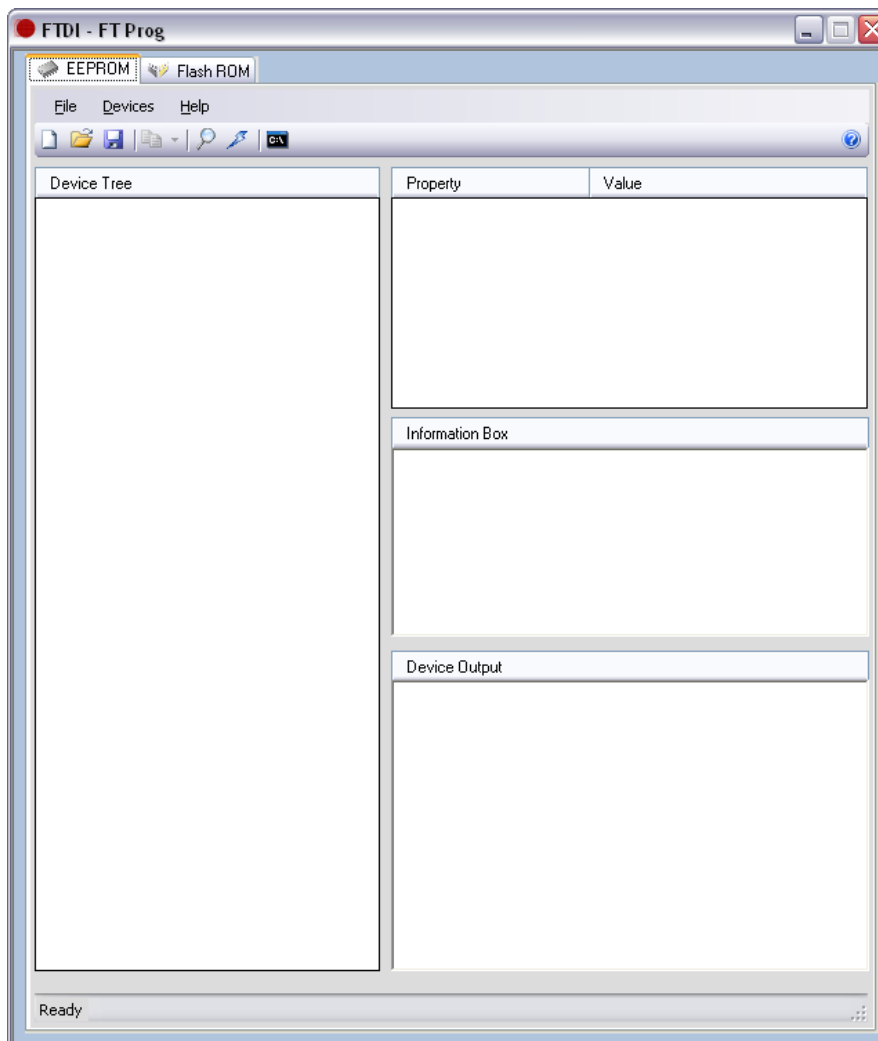
## 3 FT\_Prog Basics

### 3.1 Modes of Operation

FT\_Prog has 3 modes of operation: Idle Mode, Program Mode and Edit Mode. FT\_Prog programming parameters are saved in files referred to here as EEPROM templates. Once defined, these EEPROM templates can be loaded by FT\_Prog 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 an EEPROM template.
- Program Mode is used to Program and Erase the device EEPROM(s).

When FT\_Prog is launched, it is initially in Idle Mode and the screen looks like the one given below :



User choices in Idle Mode are:

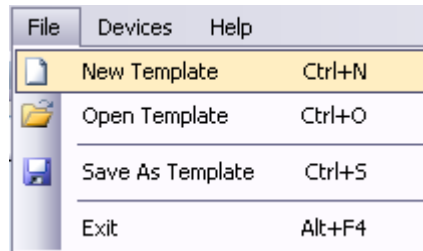
- Create a new EEPROM template. This will change the mode to Edit Mode and allow the user to select the parameters for their device.
- Open an existing EEPROM template.
- Scan the USB Bus for available devices. FT\_Prog will report how many programmed and blank devices are attached to the target PC.

## 4 Edit Mode Functions

### 4.1 Create a New EEPROM Template

To create a new EEPROM template click on the Create New Programming Template button on the toolbar

or select  **"New Template"** from the **"File"** Menu.



The EEPROM template can define parameters in the following categories:

- Chip\_Details
- USB\_Device\_Descriptor
- USB\_Config\_Descriptors
- USB\_String\_Descriptors
- FT232R Hardware Specific
- FT245R Hardware Specific
- FT2232D Hardware Specific
- FT2232H Hardware Specific
- FT4232H Hardware Specific
- FT232H Hardware Specific
- FT4222H Hardware Specific
- FT260 Hardware Specific

Please see [Section 5](#) for a detailed description of template parameters.

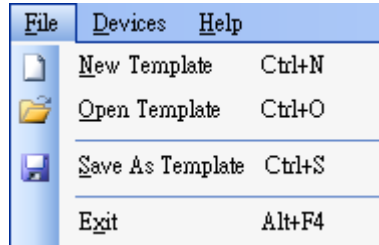


## 4.2 Modify an Existing EEPROM Template

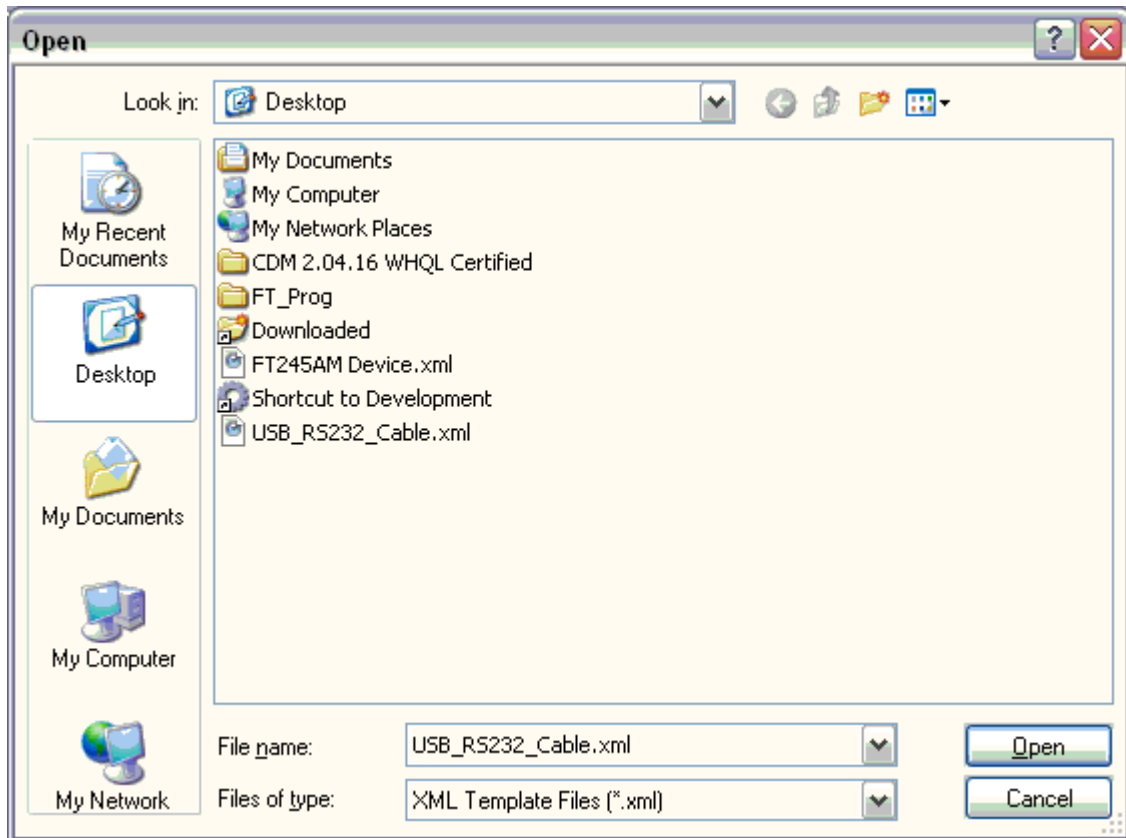
To modify an existing EEPROM template, click on the **"Open Template"** button on the toolbar Or select,



**"Open Template"** from the **"File"** menu.




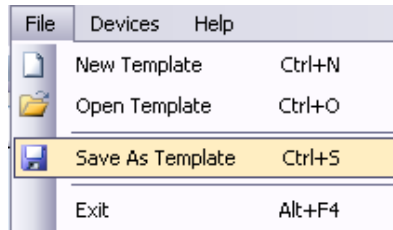
This will launch the Open dialog box. Select the template to be modified and click on **"Open"**.



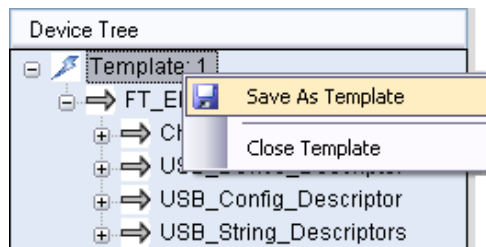
FT\_Prog will load the template into the **"Device Tree"** window, making it available to be edited.

### 4.3 Save a Template

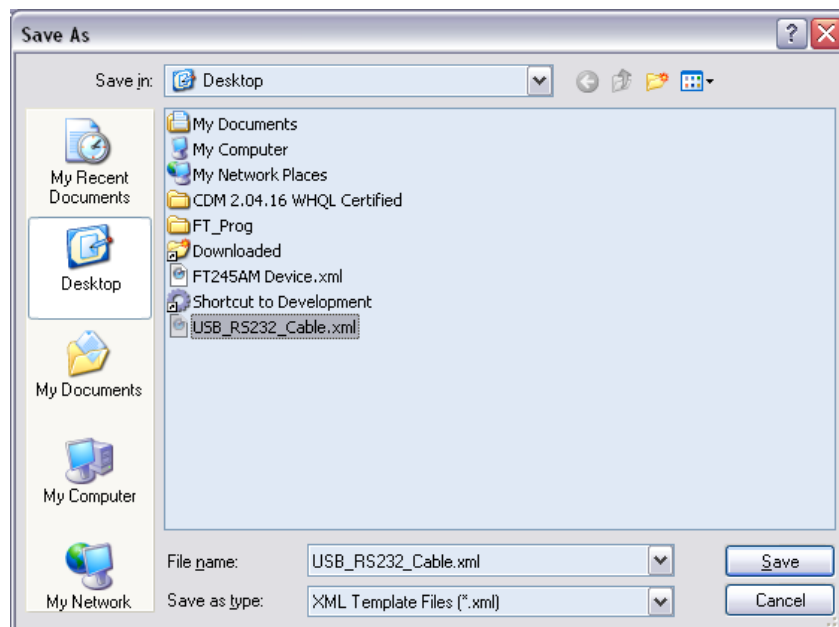
Any changes can be saved, click on the "Save" button to save the changes. Or select **"Save As Template"**  from the **"File"** menu.



Or right click on the required template, and select **"Save As Template"**



Enter a file name for the EEPROM template, perhaps a product name (USB\_RS232\_Cable.xml in this example), and click on **"Save"**.



FT\_Prog will save the EEPROM template.

## 5 Device EEPROM Parameters

### 5.1 Chip\_Details

Select the device type to be programmed from the Chip Details box. When selected, this will enable the menu options appropriate to the device that you have selected.

Property	Value
Type	<div style="border: 1px solid black; padding: 2px;">           FT232/245BM <span style="float: right;">▼</span>            FT232/245AM            FT232/245BM            FT232R            FT245R            FT2232D/C            FT2232H            FT4232H         </div>

### 5.2 USB\_Device\_Descriptor

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

- FTDI Default. If the FTDI default values are used, the VID and PID are fixed and it is not necessary 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 [Contact FTDI](#) 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.
- 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.

Property	Value
Custom VID/PID:	FTDI Default <span style="float: right;">▼</span>
Vendor ID:	0403
Product ID:	6001
USB Version Number:	USB 1.1 <span style="float: right;">▼</span>

**Note:** If non-standard VID and/or PID values are used the driver INF file must be modified to accommodate the new VID and PID combination. FTDI provide a utility to generate custom INF files, this is available from the [Utilities](#) section of the website. Please refer to the application note [AN 107](#).

## 5.3 USB\_Config\_Descriptors

Details of the USB device power requirements/USB Remote wakeup/Pull down IO Pins in USB Suspend can be set in the USB\_Config\_Descriptors.

Property	Value
Bus Powered:	<input checked="" type="radio"/>
Self Powered:	<input type="radio"/>
Max Bus Power:	<input type="text" value="100"/> mAmps
USB Remote Wakeup:	<input checked="" type="checkbox"/>
Pull Down IO Pins in USB Suspend:	<input type="checkbox"/>

Select Bus Powered (the product gets power from USB) or Self Powered (the product has its own power supply) from the power options. The Max Bus Power value defaults are 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.
- 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.

### 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:

- 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 Max 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 Max Bus Power field.

### USB Remote Wakeup:

- FT8U232AM, FT232BM, FT232R and [FT-X](#) 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), as does the FT2232D/H (SI/WUA, SI/WUB), as does the FT4232H (SI/WUA, SI/WUB, SI/WUC, SI/WUD). FT8U245AM does not have support for remote wake up. To use the remote wake-up capability, enable this option by checking the box. On the FT245R RXF# becomes an input and used to wake the host from suspend mode.

### Pull Down IO Pins in USB Suspend:

- All 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.

## 5.4 USB\_String\_Descriptors

Details of the USB device manufacturer /product Description/USB Serial number can be set in the USB\_String\_Descriptors.

Property	Value
Manufacturer:	FTDI
Product Description:	USB <-> Serial Converter
Serial Number Enabled:	<input checked="" type="checkbox"/> Auto Generate Serial No: <input type="checkbox"/>
Serial Number:	FTSASQ2P or Prefix: FT

### Manufacture/product Description:

The Product and Manufacturer Description Strings in the fields allow customization of the description strings the device uses.

- 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. "USB <->Serial Converter"

**USB Serial Number:**

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

Auto generated serial numbers 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 recognize the first instance of such a device connected to a PC.

**Note:** The maximum length of descriptors varies across the range of chips but is typically 40 to 50 characters. The descriptors include the manufacturer, product description and serial number.

## 5.5 FT232R Hardware\_Specific

Additional features available on the FT232R device allow RS232 signals to be inverted and the CBUS pins to be configured through combo-boxes.

Property	Value
Invert TXD	<input type="checkbox"/>
Invert RXD	<input type="checkbox"/>
Invert RTS#	<input type="checkbox"/>
Invert CTS#	<input type="checkbox"/>
Invert DTR#	<input type="checkbox"/>
Invert DSR#	<input type="checkbox"/>
Invert DCD#	<input type="checkbox"/>
Invert RI#	<input type="checkbox"/>

Property	Value
C0	TXDEN <input type="button" value="v"/>
C1	TXDEN <input type="button" value="v"/>
C2	TXDEN <input type="button" value="v"/>
C3	TXDEN <input type="button" value="v"/>
C4	TXDEN <input type="button" value="v"/>

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 FT232R has an internal Oscillator, but it is also possible to select "Use External Oscillator" to use an external oscillator.

**Note:** If "Use External Oscillator" is selected and the hardware design does not have an external oscillator connected, then the FT232R will be rendered useless.

Property	Value
High Current I/O's:	<input checked="" type="checkbox"/>
Load D2XX Driver	<input checked="" type="checkbox"/>
Use External Oscillator	<input type="checkbox"/>

## 5.6 FT245R Hardware\_Specific

Unlike the FT232R, the FT245R does not allow inverting of signals or configuration of the CBUS pins as they are all required for normal FIFO operation.

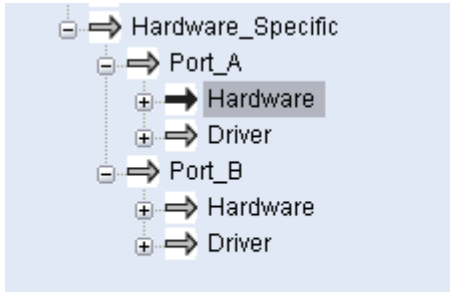
The FT245R has an internal Oscillator, but it is also possible to select "Use External Oscillator" to use an external oscillator.

**Note:** If "Use External Oscillator" is selected and the hardware design does not have an external oscillator connected, then the FT245R will be rendered useless.

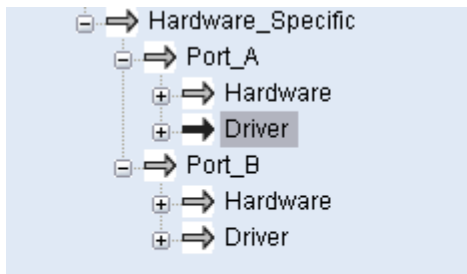
Property	Value
High Current I/O's:	<input checked="" type="checkbox"/>
Load D2XX Driver	<input checked="" type="checkbox"/>
Use External Oscillator	<input type="checkbox"/>

## 5.7 FT2232D Hardware\_Specific

The FT2232D device incorporates all of the BM specific features, but with an additional "CPU FIFO" option made available.



Property	Value
RS232 UART	<input checked="" type="radio"/>
245 FIFO	<input type="radio"/>
CPU FIFO	<input type="radio"/>
OPTO Isolate	<input type="radio"/>
High Current I/O's	<input type="checkbox"/>



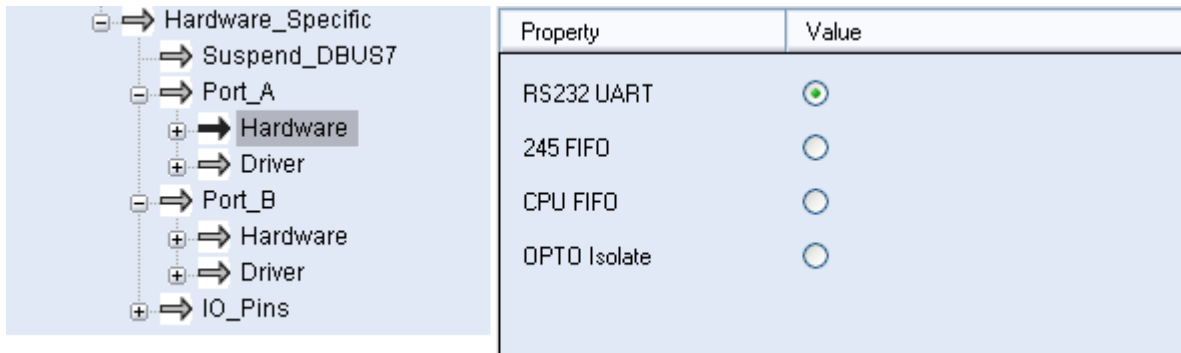
Property	Value
Virtual COM Port	<input checked="" type="radio"/>
D2XX Direct	<input type="radio"/>

- 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 FT2232D 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.



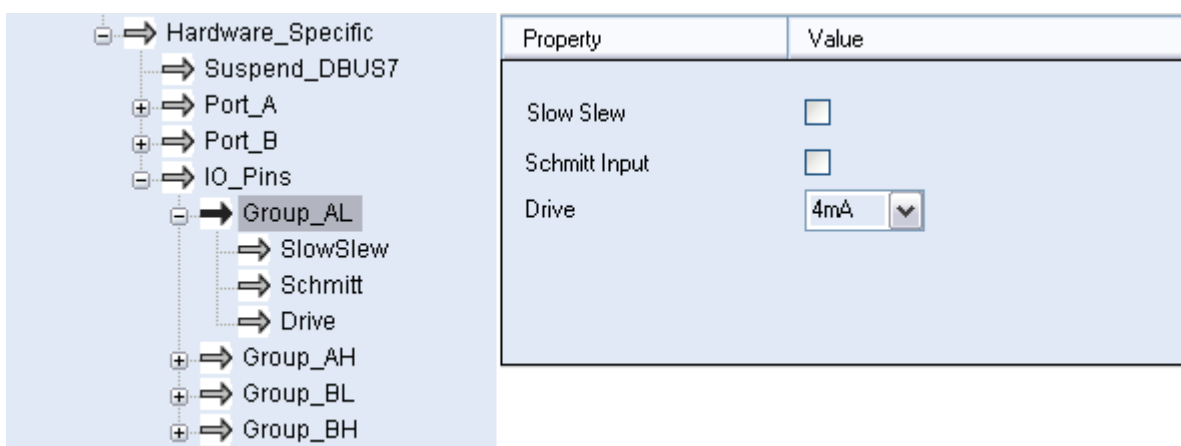
## 5.8 FT2232H Hardware\_Specific

The FT2232H function is similar to the FT2232D. The FT2232H device incorporates all of the BM devices options section and all of the FT2232D Device Options.



Property	Value
RS232 UART	<input checked="" type="radio"/>
245 FIFO	<input type="radio"/>
CPU FIFO	<input type="radio"/>
OPTO Isolate	<input type="radio"/>

- 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 FT2232H device datasheet (available from the [Datasheets](#) page of the [FTDI website](#)) for more information on these modes.
  - The driver that is installed and used by each channel can be set by selecting Virtual COM Port or D2XX Direct option.
- 
- The slow slew function can be selected for slower output driver.
  - The Schmitt Input function can be enabled if the user needs to implement Schmitt trigger function to input pins.
  - The IO pin driving capability can also be modified; the current range is 4mA to 16mA.



Property	Value
Slow Slew	<input type="checkbox"/>
Schmitt Input	<input type="checkbox"/>
Drive	4mA <input type="button" value="v"/>

- The FT2232H's Pin59 (POWERSAV#) is USB Power Save input. This is an EEPROM configurable option used when the FT2232H is used in a self powered mode and is used to prevent forcing current down the USB lines when the host or hub is powered off.

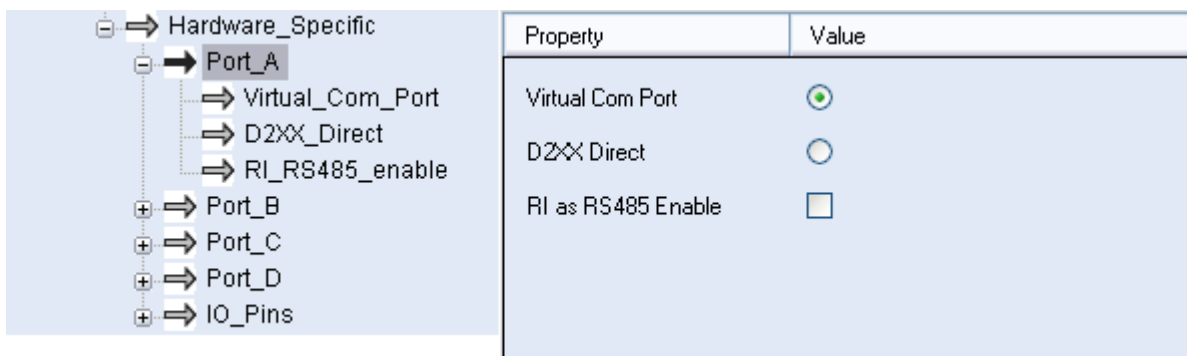
PWRSV# = 1: Normal Operation

PWRSV# = 0: FT2232H forced into SUSPEND mode.

Property	Value
Suspend on DBus7 Low	<input type="checkbox"/>

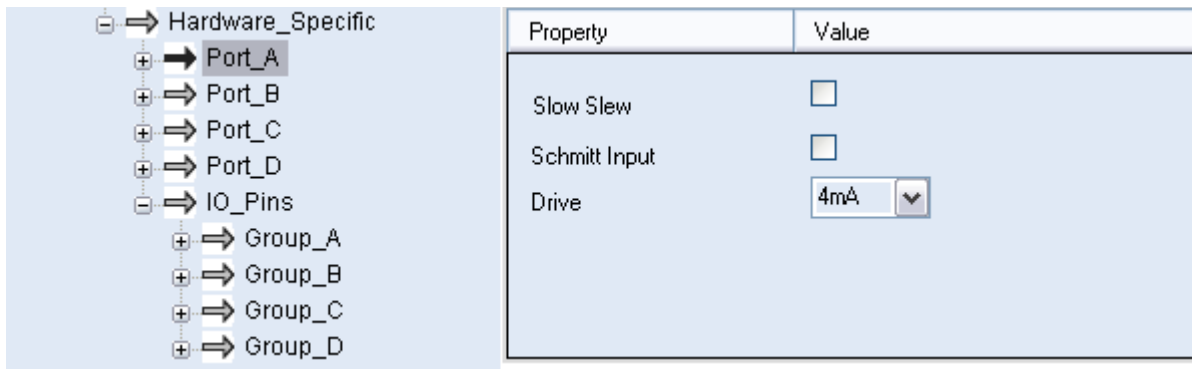
## 5.9 FT4232H Hardware\_Specific

The FT4232H function is similar to the FT2232H. The FT4232H device incorporates all of the BM devices option sections and all of the FT2232H Device Options.



Property	Value
Virtual Com Port	<input checked="" type="radio"/>
D2XX Direct	<input type="radio"/>
RI as RS485 Enable	<input type="checkbox"/>

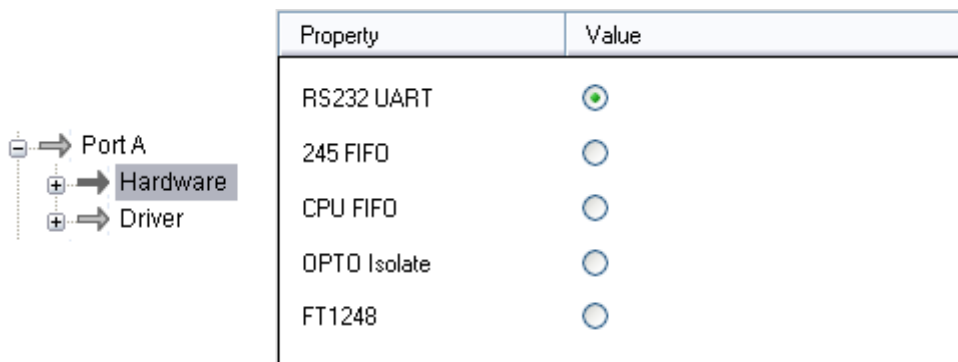
- Each channel can be independently configured as FT232BM-style USB UART.
- Each channel can also be individually configured in synchronous or asynchronous bit-bang mode. See the FT4232H device datasheet (available from the [Datasheets](#) page of the [FTDI website](#)) for more information on these modes.
- The driver that is installed and used by each channel can be set by selecting Virtual COM Port or D2XX Direct option.
- The slow slew function can be selected for slower output driver.
- The Schmitt Input function can be enabled if user needs to implement Schmitt trigger function to input pins.
- The IO pin driving capability can also be set; the current range is 4mA to 16mA.



Property	Value
Slow Slew	<input type="checkbox"/>
Schmitt Input	<input type="checkbox"/>
Drive	4mA <input type="button" value="v"/>

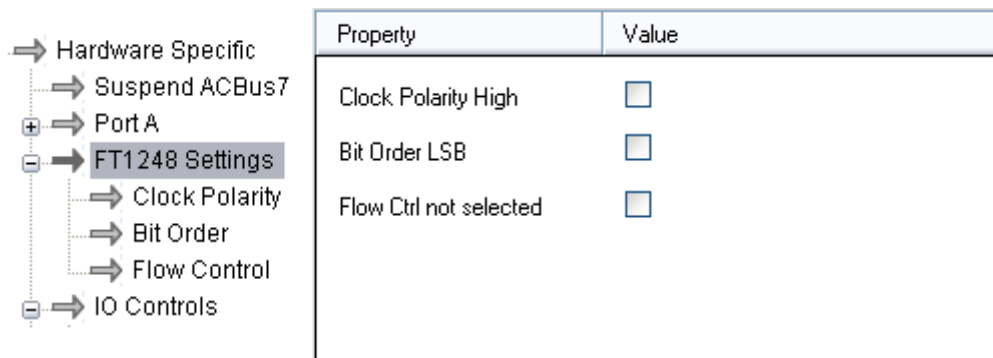
## 5.10 FT232H Hardware\_Specific

The FT232H can be configured in one of 5 modes: as a RS232 UART, 245 FIFO, CPU FIFO, OPTO Isolate or the new FT1248. Please refer to the device datasheet for details of the different configurations.



Property	Value
RS232 UART	<input checked="" type="radio"/>
245 FIFO	<input type="radio"/>
CPU FIFO	<input type="radio"/>
OPTO Isolate	<input type="radio"/>
FT1248	<input type="radio"/>

FT1248 settings can be configured in the EEPROM, the following options are available:



Property	Value
Clock Polarity High	<input type="checkbox"/>
Bit Order LSB	<input type="checkbox"/>
Flow Ctrl not selected	<input type="checkbox"/>

CBus I/O pins can be controlled through the EEPROM, more details are available in the datasheet.

IO Controls

- C0
- C1
- C2
- C3
- C4
- C5
- C6
- **C7**

Property	Value
C5	Tristate <input type="button" value="v"/>
C6	Tristate <input type="button" value="v"/>
C7	Tristate <input type="button" value="v"/>
C8	Tristate <input type="button" value="v"/>
C9	Tristate <input type="button" value="v"/>

As with other FTDI devices the slew rate, Schmitt trigger and drive strength can be configured in the EEPROM.

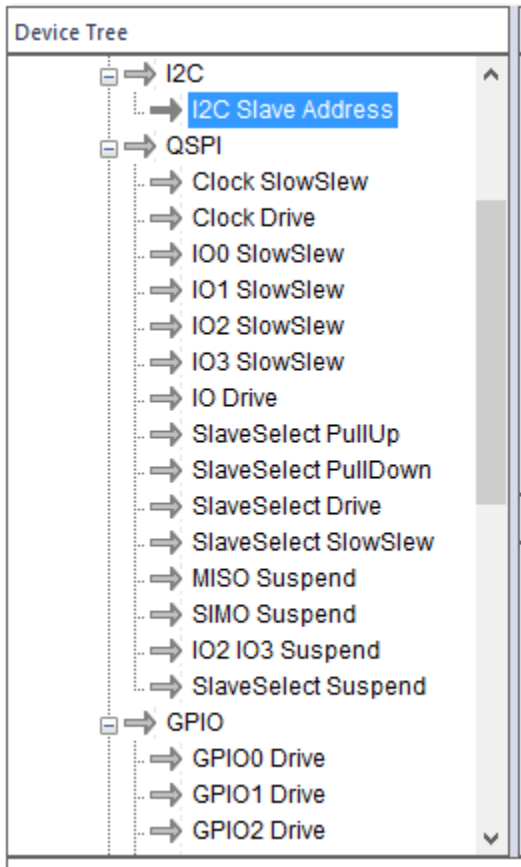
IO Pins

- + → GroupAD
- + → GroupAC

Property	Value
Slow Slew	<input type="checkbox"/>
Schmitt Input	<input type="checkbox"/>
Drive	4mA <input type="button" value="v"/>

## 5.11 FT4222H Hardware Specific

This device contains SPI, I<sup>2</sup>C and GPIO configurable interfaces. The SPI interface can be configured as master mode with single, dual and quad bits data width transfer, or slave mode with single bit data width transfer. The I<sup>2</sup>C interface can be configured in master or slave mode.



Slaves on an I<sup>2</sup>C bus must have unique addresses, and this may be configured here for the FT4222H.

Electrical characteristics such as slew rate, drive current and enabling of pull-up/down resistors may be configured for SPI and GPIO modes.

The FT4222H's output signal to indicate detection of a Battery Charger on the USB port may also be configured here

Property	Value
BCD Disable:	<input type="checkbox"/>
BCD Drive Strength:	4mA
BCD Output Active Low:	<input type="checkbox"/>

## 5.12 FT260 Hardware Specific

### HID over I<sup>2</sup>C

This HID device provides a bridge between USB and either UART or I<sup>2</sup>C peripherals. I<sup>2</sup>C mode supports the HID over I<sup>2</sup>C protocol, the settings of which can be configured here.

Device Tree	Property	Value
<ul style="list-style-type: none"> <li>[-] Device: 0 [Loc ID: HID 0x0]               <ul style="list-style-type: none"> <li>[-] FT EEPROM                   <ul style="list-style-type: none"> <li>[-] Chip Details</li> <li>[-] USB Device Descriptor</li> <li>[-] USB Config Descriptor</li> <li>[-] USB String Descriptors</li> <li>[-] Hardware Specific                       <ul style="list-style-type: none"> <li>[-] EEPROM Address</li> <li>[-] EEPROM Address bits</li> <li>[-] Power Settings</li> <li>[-] GPIO</li> <li style="background-color: #e0f0ff;">[-] <b>HID over I2C</b></li> </ul> </li> </ul> </li> </ul> </li> </ul>	Address:	2C
	EEPROM Descriptor Offset:	80
	External I2C Interrupt:	Rising Edge
	Power Support:	<input type="checkbox"/>
	Idle Support:	<input type="checkbox"/>
	Protocol Support:	<input type="checkbox"/>
	Interface Subclass:	No Subclass
	Interface Protocol:	No Protocol

Power Support – Allows device to be set to preferred power setting i.e. ON, SLEEP or RESERVED.

Idle Support – Allows the device to enter IDLE mode when no data is transferred between USB and I<sup>2</sup>C bus after a period of inactivity.

Protocol Support – Allows setting of protocol mode for the device i.e. Boot, Report or Reserved.

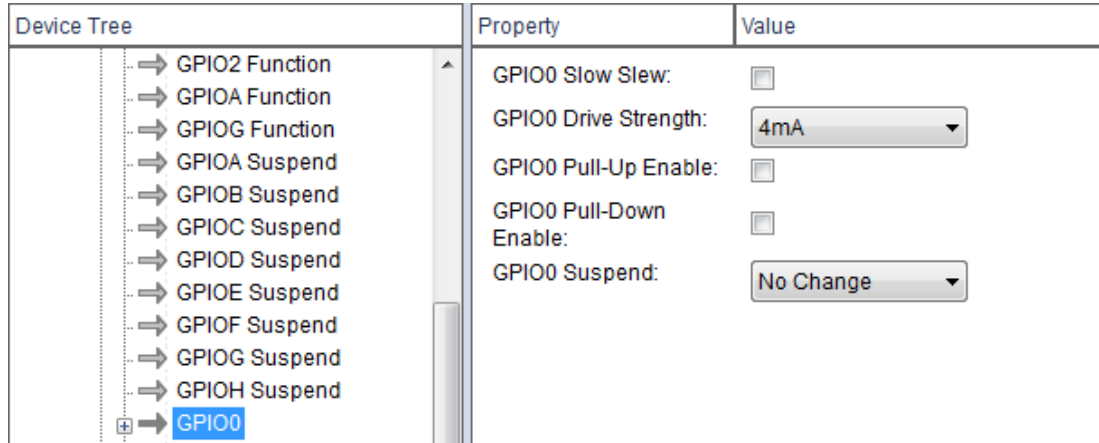
### GPIOs

The FT260 has 14 digital function pins that can be set as I2C/UART related function or GPIO. Some GPIO functions are implemented in the FT260 for various applications like TX\_ACTIVE, TX\_LED, RX\_LED for UART; SUSPOUT\_N, WAKEUP for USB and PWREN and BCD\_DET indicator for power management.

Device Tree	Property	Value
<ul style="list-style-type: none"> <li>[-] Device: 0 [Loc ID: HID 0x0]               <ul style="list-style-type: none"> <li>[-] FT EEPROM                   <ul style="list-style-type: none"> <li>[-] Chip Details</li> <li>[-] USB Device Descriptor</li> <li>[-] USB Config Descriptor</li> <li>[-] USB String Descriptors</li> <li>[-] Hardware Specific                       <ul style="list-style-type: none"> <li>[-] EEPROM Address</li> <li>[-] EEPROM Address bits</li> <li>[-] Power Settings</li> <li style="background-color: #e0f0ff;">[-] <b>GPIO</b></li> <li>[-] HID over I2C</li> </ul> </li> </ul> </li> </ul> </li> </ul>	GPIO2 Function:	SUSPOUT_N
	GPIOA Function:	TX_ACTIVE
	GPIOG Function:	BCD_DET
	GPIOA Suspend:	No Change
	GPIOB Suspend:	No Change
	GPIOC Suspend:	No Change
	GIOD Suspend:	No Change
	GPIOE Suspend:	No Change

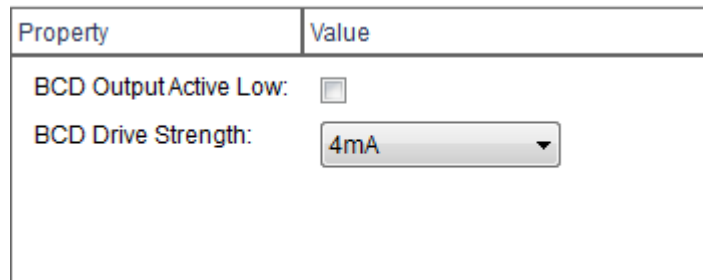
See [Datasheet](#) for further information of GPIO functions for each pin.

The signal drive strength, slew rate control and pull high/low resistors of these Digital I/Os can be configured for different design needs.



### Battery Charge Detection

The FT260 includes USB Battery Charger Detection which allows for USB peripheral devices to detect the presence of a higher power source to enable improved charging.



### 5.13 FT-X Hardware Specific

The FT-X bridges USB ports to UART, I<sup>2</sup>C, enhanced SPI (FT1248) or FIFO interfaces. Slaves on an I<sup>2</sup>C bus must have unique addresses, and this may be configured here.

Property	Value
I2C Slave Address	22
I2C Device ID	0
I2C Disable Schmitt	<input type="checkbox"/>

Battery Charger Detection which allows for USB peripheral devices to detect the presence of a higher power source to enable faster charging.

Property	Value
Battery Charge Enable	<input type="checkbox"/>
Force Power Enable	<input type="checkbox"/>
De-activate Sleep	<input type="checkbox"/>

The device has up to 7 independently configurable CBUS pins that can be set here.

Device Tree	Property	Value
→ USB Suspend VBus	C0	BitBang_RD#
→ Port A	C1	Tristate
+ → Hardware	C2	Tristate
→ Driver	C3	Tristate
→ D2XX	C4	Tristate
→ VCP	C5	Tristate
+ → Battery Charge Detect		
+ → I2C		
→ CBUS Signals		
→ C0		
→ C1		
→ C2		



The USB Suspend VBus option can be enabled here. This can be used for self-powered designs to realise that the host is waking up. Used to prevent forcing current down the USB lines when the host or hub is powered off.

Device Tree	Property	Value
<ul style="list-style-type: none"> <li>[-] [USB] Device: 0 [Loc ID:0x215]               <ul style="list-style-type: none"> <li>[-] FT EEPROM                   <ul style="list-style-type: none"> <li>[+] Chip Details</li> <li>[+] USB Device Descriptor</li> <li>[+] USB Config Descriptor</li> <li>[+] USB String Descriptors</li> <li>[-] Hardware Specific                       <ul style="list-style-type: none"> <li>[-] USB Suspend VBus</li> <li>[+] Port A</li> <li>[+] Battery Charge Detect</li> <li>[+] I2C</li> </ul> </li> </ul> </li> </ul> </li> </ul>	USB Suspend VBus	<input type="checkbox"/>

**Note:** One of the CBUS signals must be set to VBUS\_SENSE before this setting can be enabled.

## 6 Program Mode Function

To enter program mode the user must first scan for FTDI devices connected to the host PC [see section 6.1]. The program mode functions of FT\_Prog can then be accessed through the toolbar or by right clicking on a device within the "Device Tree".



Program mode allows the user to:

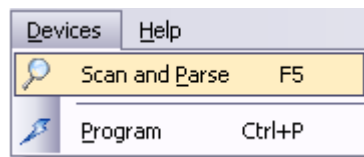
- Erase device EEPROM contents
- Apply templates to connected devices
- Program device EEPROM contents

### 6.1 Scan for Devices

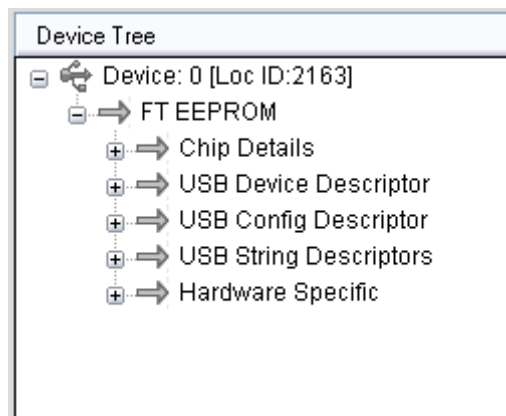
To scan the USB bus for available connected FTDI devices, click on the **"Scan and Parse"** button on the toolbar.



Alternatively, select **"Scan and Parse"** from the **"Devices"** menu.



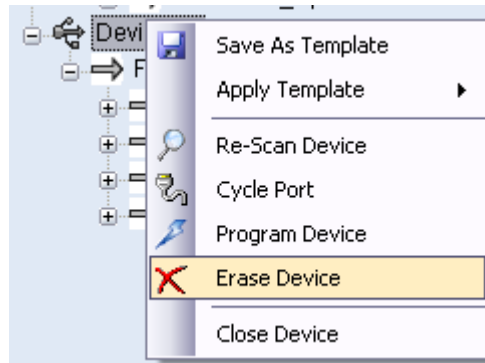
Upon successful completion of the operation any connected devices will be displayed within the "Device Tree" window, as shown below.




## 6.2 Erase Existing Devices

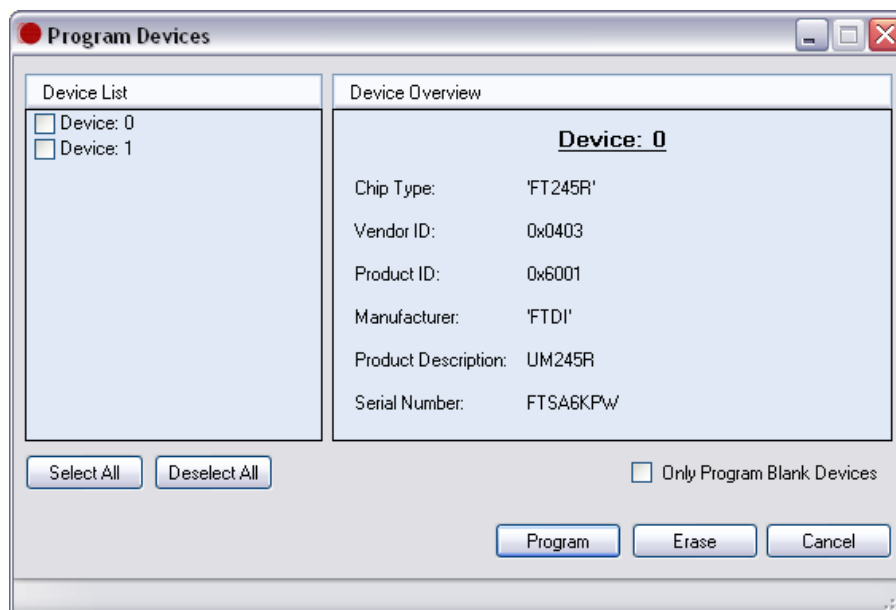
There are two ways to erase the EEPROM of connected devices.

To erase the contents of a single connected device the user can right-click on the desired device within the "Device Tree" and select "Erase Device" from the menu.



Alternatively, to erase the EEPROM of multiple devices, the user can press the program button  within the toolbar or "**Devices**" menu.

The "Program Devices" window [as below] will appear; this will allow the user to select multiple devices from the "Device List" on the left hand side of the window and then erase the EEPROM by pressing the "Erase" button at the bottom right of the window.



After the operation has been carried out successfully the contents of the device EEPROM will be as below:

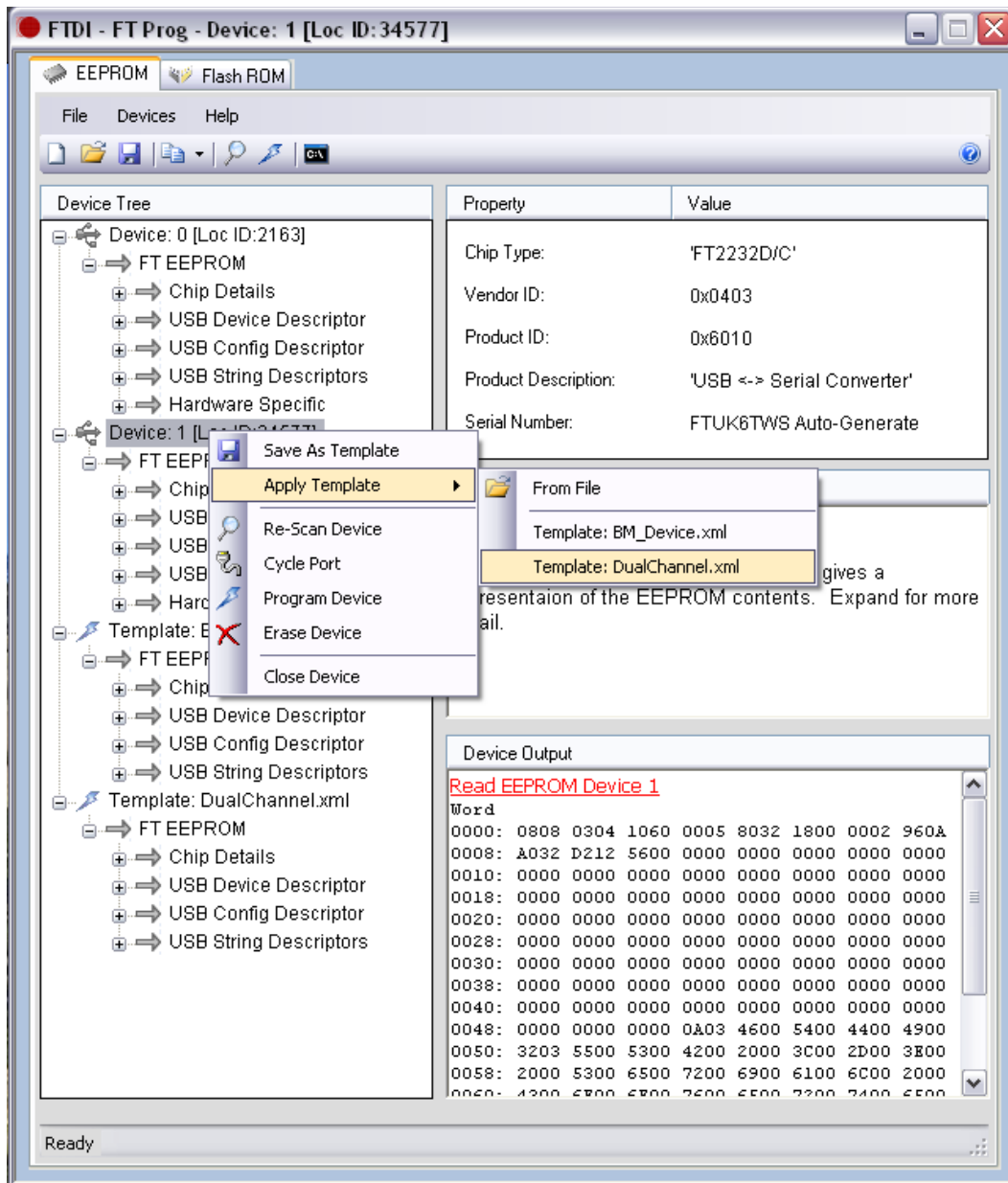
```
Device Output
Read EEPROM Device 1
Word
0000: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
0008: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
0010: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
0018: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
0020: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
0028: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
0030: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
0038: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF
```

**Note:** It is not possible to erase the contents of the FT232R and FT245R chips EEPROM.

### 6.3 Use an Existing EEPROM Template

Templates can be applied to connected devices to modify the contents of their EEPROM.

Any templates that are currently open are displayed within the "Device Tree" window. To apply these to a device: right-click the required **device** within the "Device Tree"; select "Apply Template" from the menu (any templates that are currently open will be displayed within this list). Select the required template and the settings stored within this template will be applied to the selected device.




To apply a template that is not currently open within the application: right-click on the required device and select "Apply Template", "From File". This will open a file browser allowing the user to browse to the required file.

When the EEPROM settings are complete, the next step is to program the device; follow the steps in section 6.4 to complete the process.

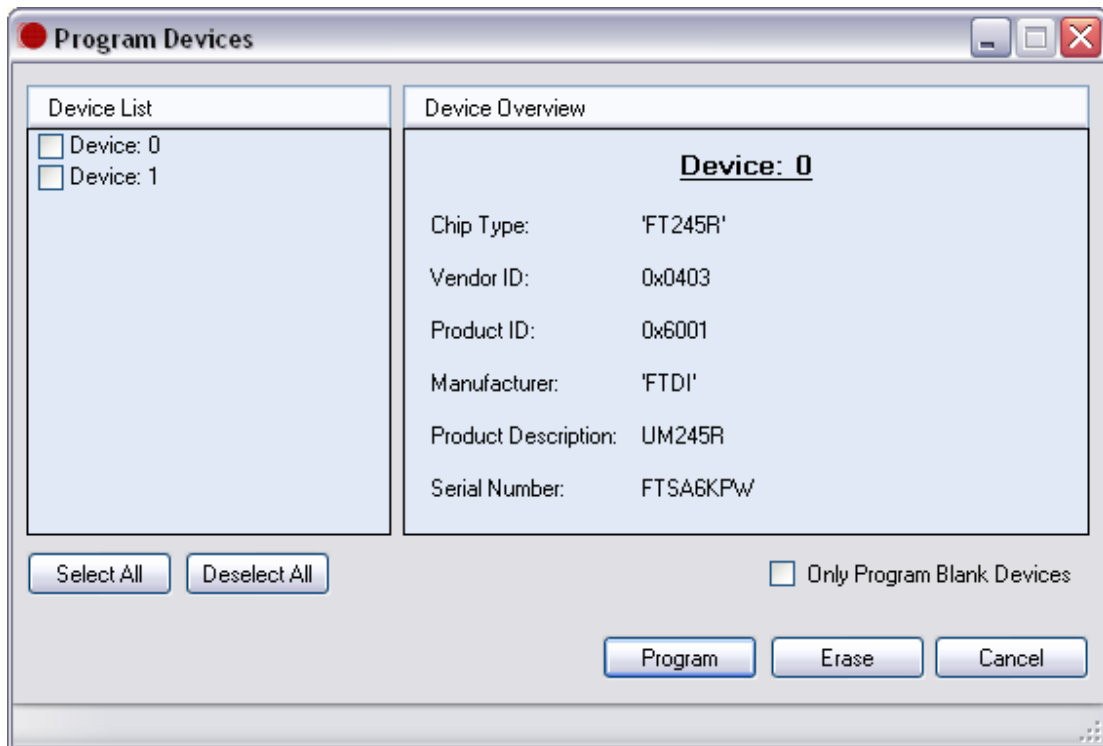
**Note:** It is not possible to apply templates to devices where the chip type of both the device and template do not match. For example, if a template for a FT2232H device is open and the currently connected device is an FT245R the application will not allow the user to apply this template.

## 6.4 Program Existing Devices

Click on the **"Program Devices"** button  within the toolbar.

The **"Program Devices"** GUI will be displayed [as below]:

From this interface, select the required devices from the "Device List" on the left of the window.



If the "Only Program Blank Devices" box is checked, FT\_Prog will select any blank devices within the "Device List" window.

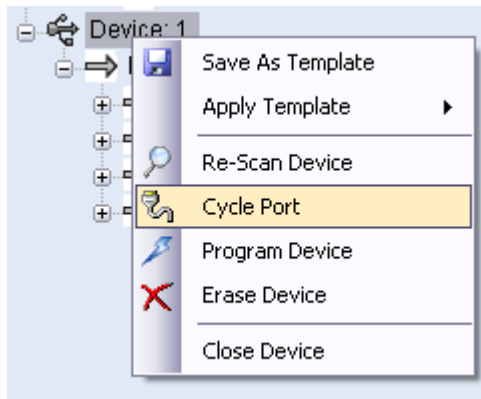
To program the selected devices press "Program" on the bottom right of the window.

Alternatively to program individual devices, right-click on the required device within the **"Device Tree"** and select **"Program Device"** from the context menu.

Device Tree	Property	Value										
<div style="border: 1px solid gray; padding: 5px;">           Device: 0           <ul style="list-style-type: none"> <li>[-] FT_EEPROM               <ul style="list-style-type: none"> <li>[+] Chip_Details</li> <li>[+] USB_Device_Descriptor</li> <li>[+] USB_Config_Descriptor</li> <li>[+] USB_String_Descriptors</li> <li>[+] Hardware_Specific</li> </ul> </li> <li>[-] Device               <ul style="list-style-type: none"> <li>[+] FT                   <ul style="list-style-type: none"> <li>Save As Template</li> <li>Apply Template ▶</li> <li>Re-Scan Device</li> <li>Cycle Port</li> <li style="background-color: #ffffcc;">Program Device</li> <li>Erase Device</li> <li>Close Device</li> </ul> </li> </ul> </li> </ul> </div>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Chip Type:</td> <td style="padding: 5px;">'FT232/245AM'</td> </tr> <tr> <td style="padding: 5px;">Vendor ID:</td> <td style="padding: 5px;">0x0403</td> </tr> <tr> <td style="padding: 5px;">Product ID:</td> <td style="padding: 5px;">0x6001</td> </tr> <tr> <td style="padding: 5px;">Product Description:</td> <td style="padding: 5px;">'USB Serial Cable'</td> </tr> <tr> <td style="padding: 5px;">Serial Number:</td> <td style="padding: 5px;">FTSAKSMA</td> </tr> </table>	Chip Type:	'FT232/245AM'	Vendor ID:	0x0403	Product ID:	0x6001	Product Description:	'USB Serial Cable'	Serial Number:	FTSAKSMA	<div style="border: 1px solid gray; padding: 5px; margin-bottom: 5px;">             Information Box           </div> <div style="border: 1px solid gray; padding: 5px; min-height: 100px;">             (Empty area)           </div> <div style="border: 1px solid gray; padding: 5px;">             Device Output           </div> <pre style="font-family: monospace; font-size: 0.9em; margin: 0;"> <b>Read EEPROM Device 1</b> Word 0000: 0000 0304 0160 0002 802D 0000 0000 940A 0008: 9B22 C012 0A03 4600 5400 4400 4900 2203 0010: 5500 5300 4200 2000 5300 6500 7200 6900 0018: 6100 6C00 2000 4300 6100 6200 6C00 6500 0020: 1203 4600 5400 5300 4100 4B00 5300 4D00 0028: 4100 0000 0000 0000 0000 0000 0000 0000 0030: 0000 0000 0000 0000 0000 0000 0000 0000 0038: 0000 0000 0000 0000 0000 0000 0000 88C1           </pre>
Chip Type:	'FT232/245AM'											
Vendor ID:	0x0403											
Product ID:	0x6001											
Product Description:	'USB Serial Cable'											
Serial Number:	FTSAKSMA											

## 6.5 Cycle Port

The "Cycle Port" button can be used to re-enumerate a device on USB after the EEPROM has been reprogrammed. This is useful because the device only reads the EEPROM configuration when it is enumerated on USB, so it forces the device to use the new EEPROM contents.

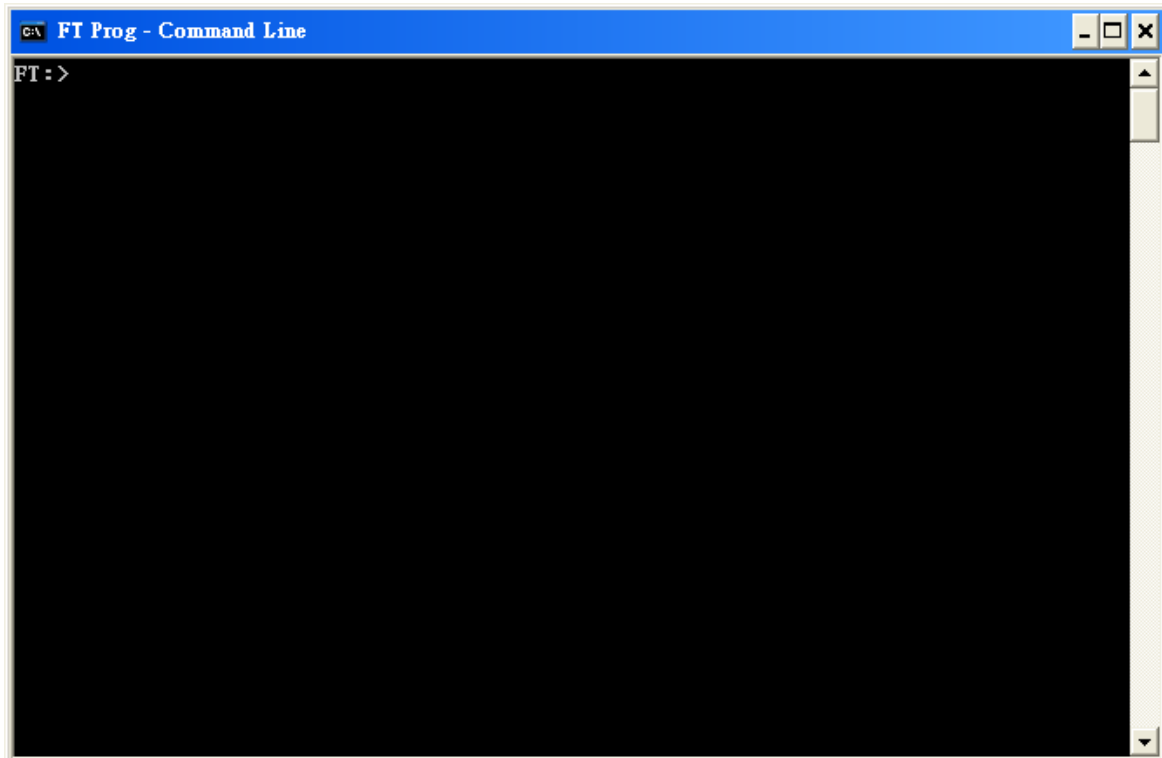




## 7 Command Line Programming

Click on the "Launch Command Line Application"  button on the toolbar.

This will launch the FT\_Prog command line

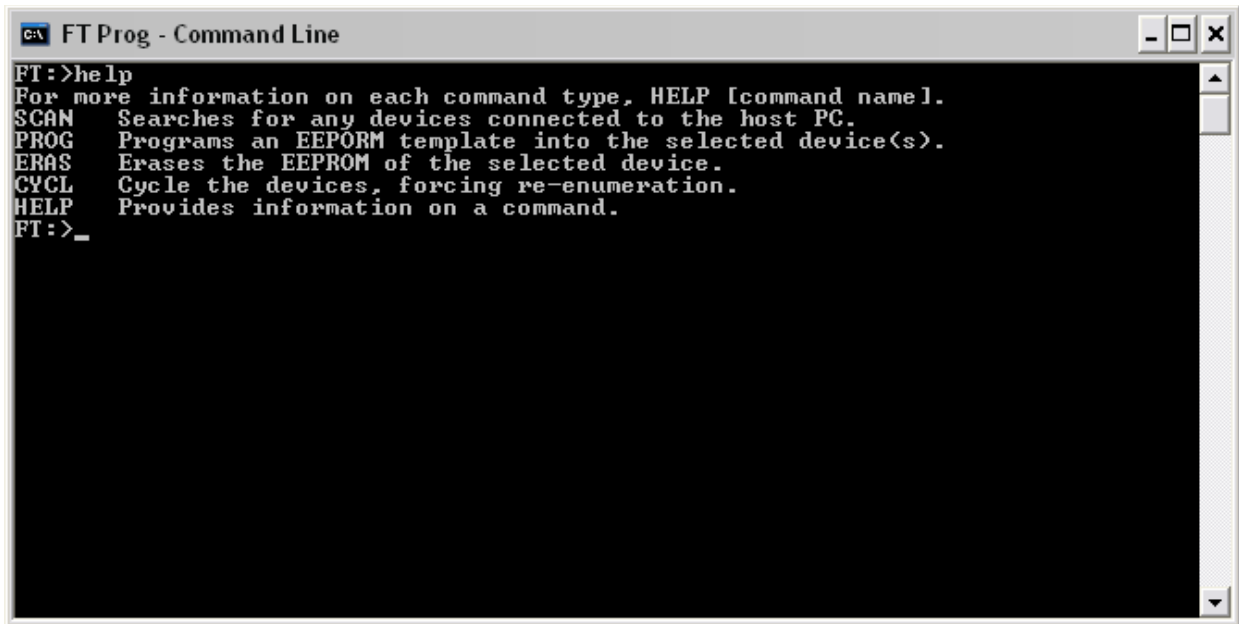


### There are 6 commands:

- SCAN Searches for any devices connected to the host PC.
- PROG Programs an EEPROM template into the selected device(s).
- ERAS Erases the EEPROM of the selected device.
- CYCL Cycles the device port, re-enumerating the device. Equivalent to a un-plug re-plug.
- HELP Provides information on a command.
- EXIT Closes the application.

### Help command:

Type HELP and press "ENTER". This will display the command line list.



```
c:\ FT Prog - Command Line
FT:>help
For more information on each command type, HELP [command name].
SCAN  Searches for any devices connected to the host PC.
PROG  Programs an EEPROM template into the selected device(s).
ERAS  Erases the EEPROM of the selected device.
CYCL  Cycle the devices, forcing re-enumeration.
HELP  Provides information on a command.
FT:>_
```

Type '*HELP SCAN*', '*HELP PROG*', '*HELP ERAS*' or '*HELP CYCL*' and this will display the required format of the associated command.

PROG format:

```
PROG [device number 0],[device number 1], ..... [template file location]
PROG 0,1,3,5 C:\Documents and Settings\template.xml
```

The '\*' symbol can be used to denote all devices connected

```
PROG * C:\Documents and Settings\template.xml
```

ERAS format:

```
ERAS [device number 0], [device number 1], .....
ERAS 0,1,3
```

The '\*' symbol can be used to denote all devices connected

CYCL format:

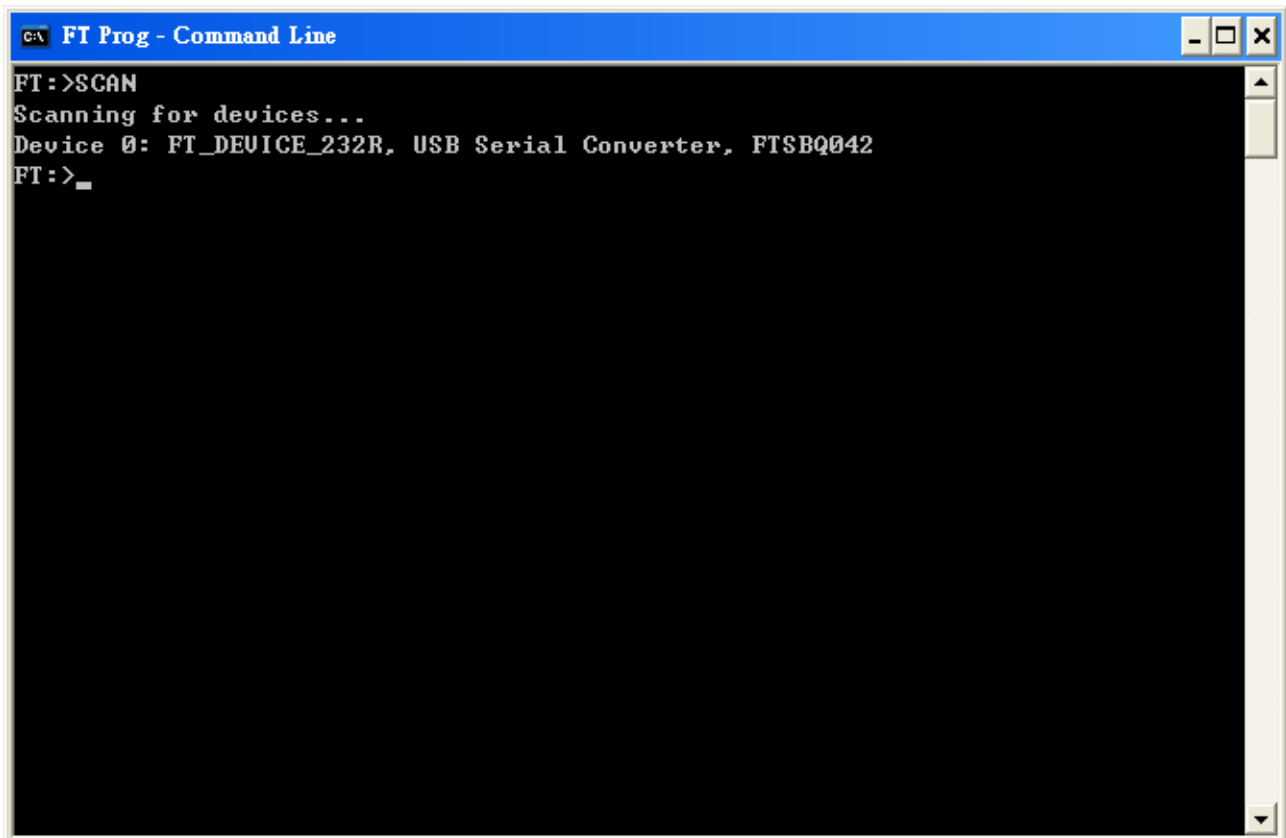
```
CYCL [device number 0], [device number 1] , .....
CYCL 0,1,3
```

The '\*' symbol can be used to denote all devices connected

### Program procedure:

Use the SCAN Command to determine what FTDI devices are connected to the USB interface FTDI Device.

The example below shows the response when a Device 0 '*FT\_DEVICE\_232R*' is connected to USB.



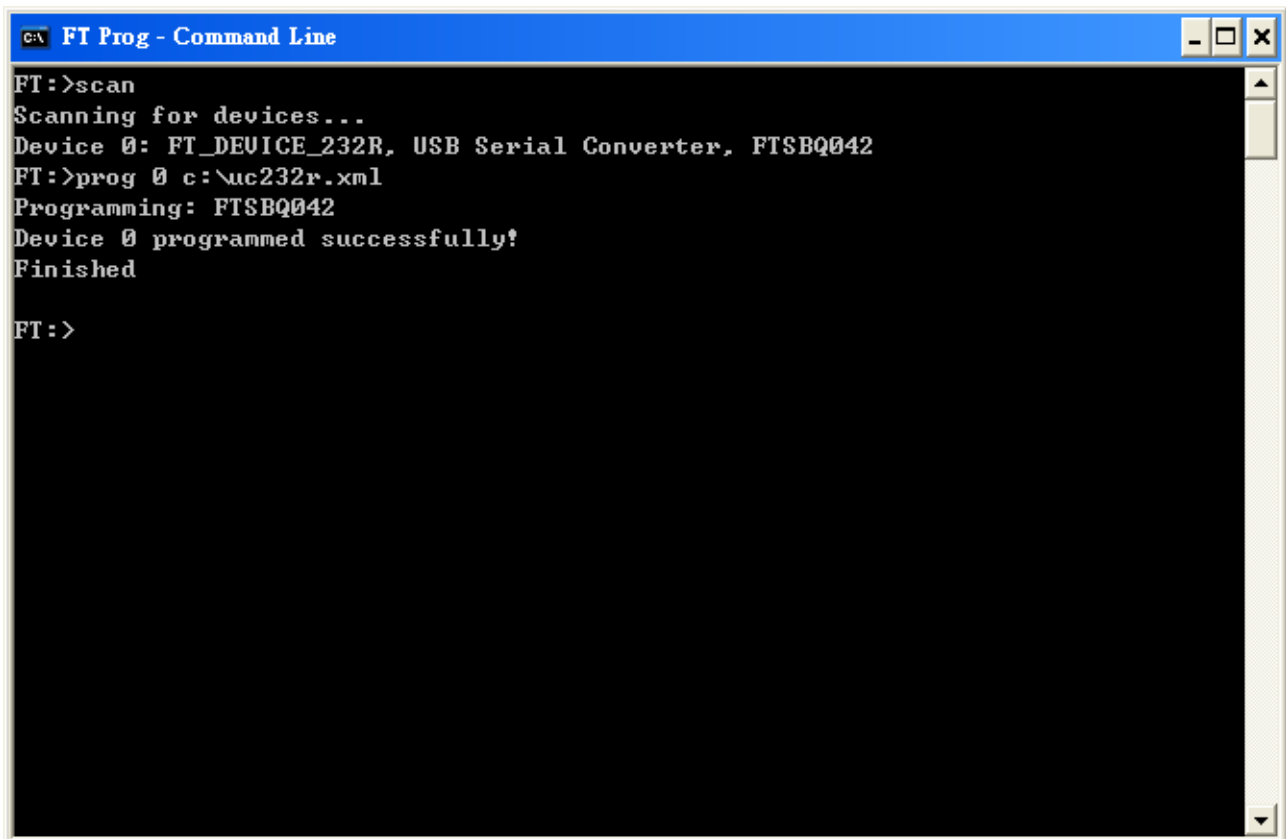
```
c:\> FT Prog - Command Line
FT:>SCAN
Scanning for devices...
Device 0: FT_DEVICE_232R, USB Serial Converter, FTSBQ042
FT:>_
```

Use the *PROG* command to program device 0

The command format is as follows:

```
PROG 0 C:\UC232R.xml
```

The example below shows device 0 programmed successfully using this command. This command uses an .xml file to program the device. The xml files can be created using the FT\_Prog GUI as highlighted in [Section 4.1](#) or by hand using any text editor (an example of this is shown below.) **It is highly recommend that the FT\_Prog application is used to create templates. If the contents of the template are incorrect and this template is used to program a device, then the content of the EEPROM may be corrupted.**

A screenshot of a Windows command prompt window titled "FT Prog - Command Line". The window has a blue title bar and standard window controls. The text inside the window shows the following sequence of commands and output:

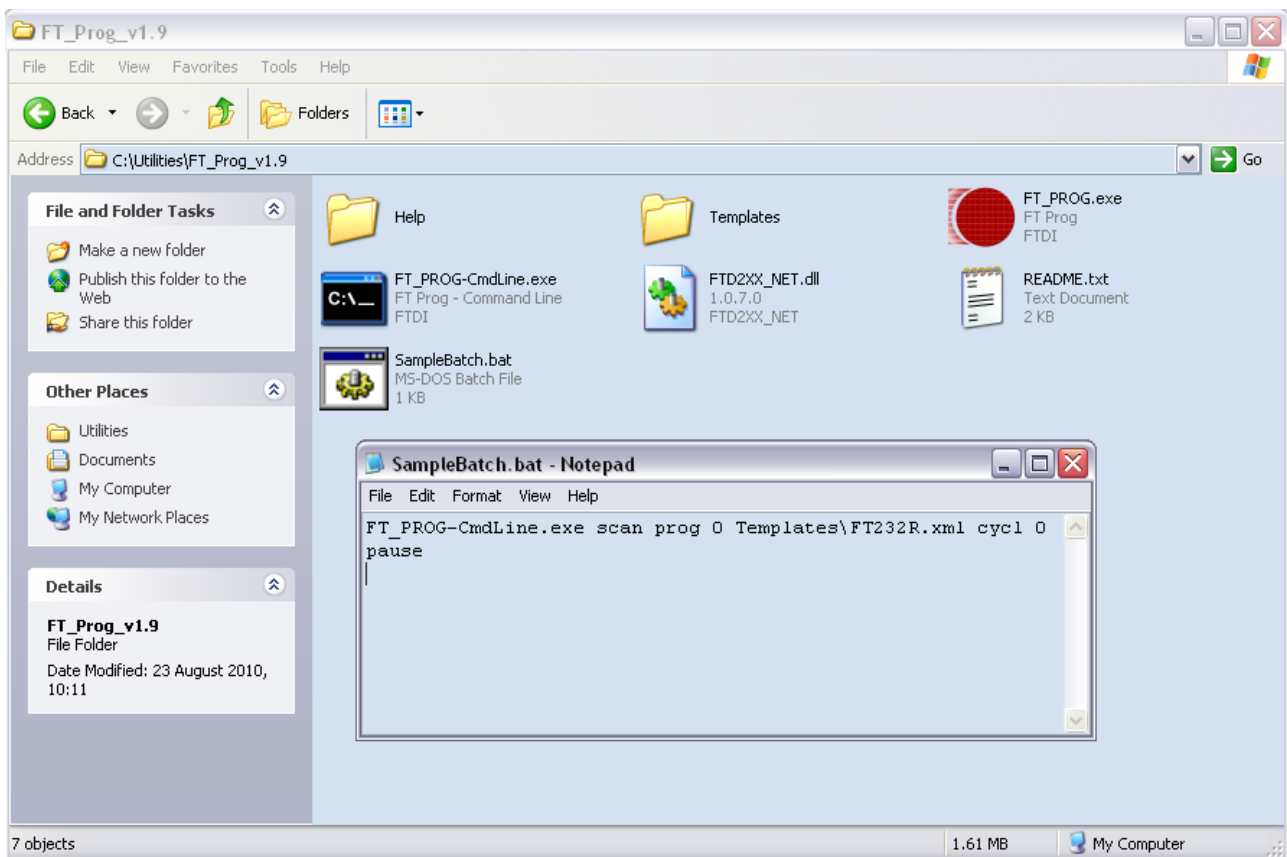
```
FT:>scan
Scanning for devices...
Device 0: FT_DEVICE_232R, USB Serial Converter, FTSBQ042
FT:>prog 0 c:\uc232r.xml
Programming: FTSBQ042
Device 0 programmed successfully!
Finished

FT:>
```

## 7.1 Batch File Programming

It is possible to create Batch (bat) files that can help to automate the process of programming the EEPROM on multiple FTDI devices. Bat files are simple text documents with the extension .bat that contain instructions to be processed by the Windows CLI. The following sample will scan for connected devices, program device 0 and finally cycle all devices to re-enumerate after programming. The bat file must be placed in the same directory as the FT\_Prog-CmdLine utility and the specified xml file must be accessible by the application.

```
FT_Prog-CmdLine.exe scan prog 0 Templates\FT232R.xml cycl 0  
pause
```



## 8 Vinculum Programmer

FT\_Prog can be used to reprogram the Flash memory of the VNC1L-1A and VNC2; it is available as a free download from the Vinculum website. FT\_Prog allows for programming of devices across the UART using the Windows CDM (D2XX) drivers or via a virtual com port. It can be seen as a replacement for the VPROG application.

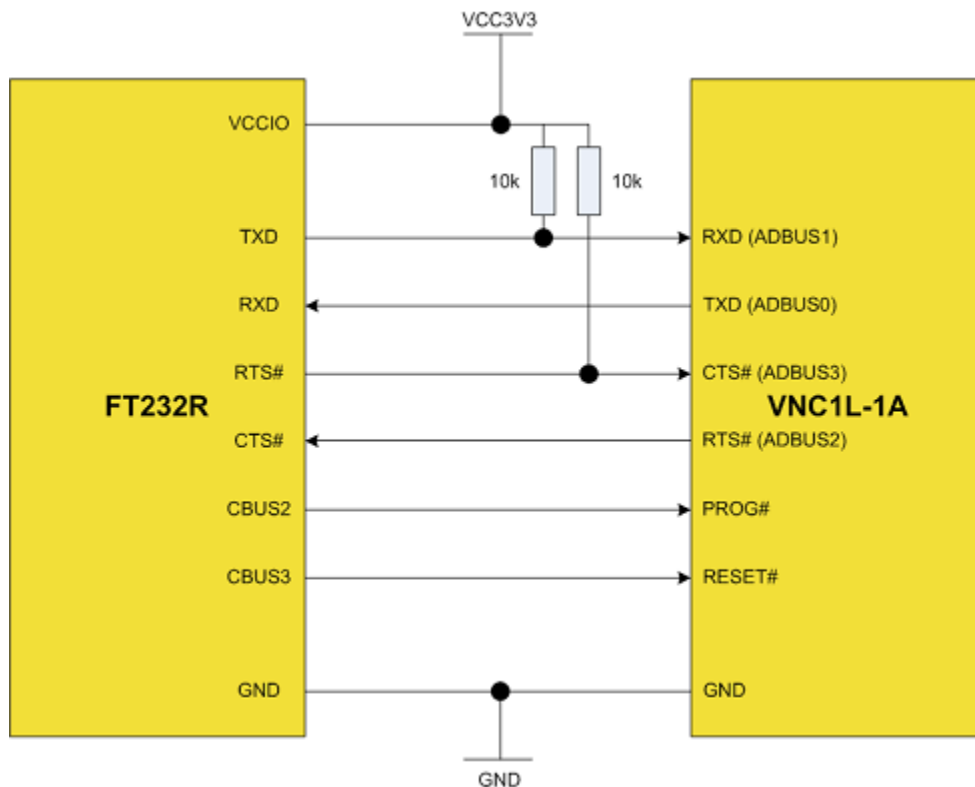
**Note:** VNC2 can also be programmed in FT\_Prog via VNC2 Debug Module as well as UART.

Programming VNC1L example:

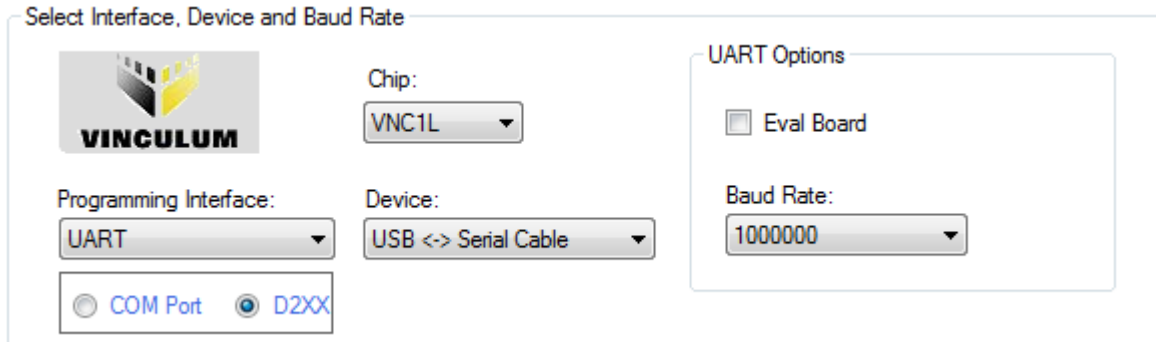
Upgrading [VNC1L firmware](#) requires a USB to UART cable; this is available from FTDI, part number "[TTL-232R-3V3](#)".

Programming requires some hardware connections. These are as follows:

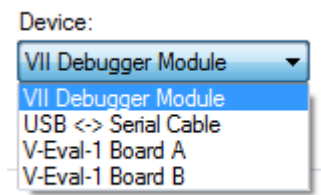
- Connect the VNC1L Pin10 (PROG#) to GND.
- VNC1L's GND connect to TTL-232R Pin1 (GND)
- VNC1L Pin33 (RTS#) connect to TTL-232R Pin2 (CTS#)
- VNC1L's Power connect to TTL-232R Pin3 (Power)
- VNC1L Pin32 (RXD) connect to TTL-232R Pin4 (TXD)
- VNC1L Pin31 (TXD) connect to TTL-232R Pin5 (RXD)
- VNC1L Pin34 (CTS#) connect to TTL-232R Pin6 (RTS#)



Vinculum programming is under the *Flash Rom* heading within FT\_Prog.

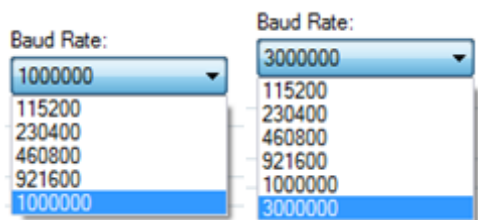


Connect the Vinculum to the host PC and select the required interface. An example of the device box is shown below.




Choose the **"USB<-> Serial Cable"**(TTL-232R).

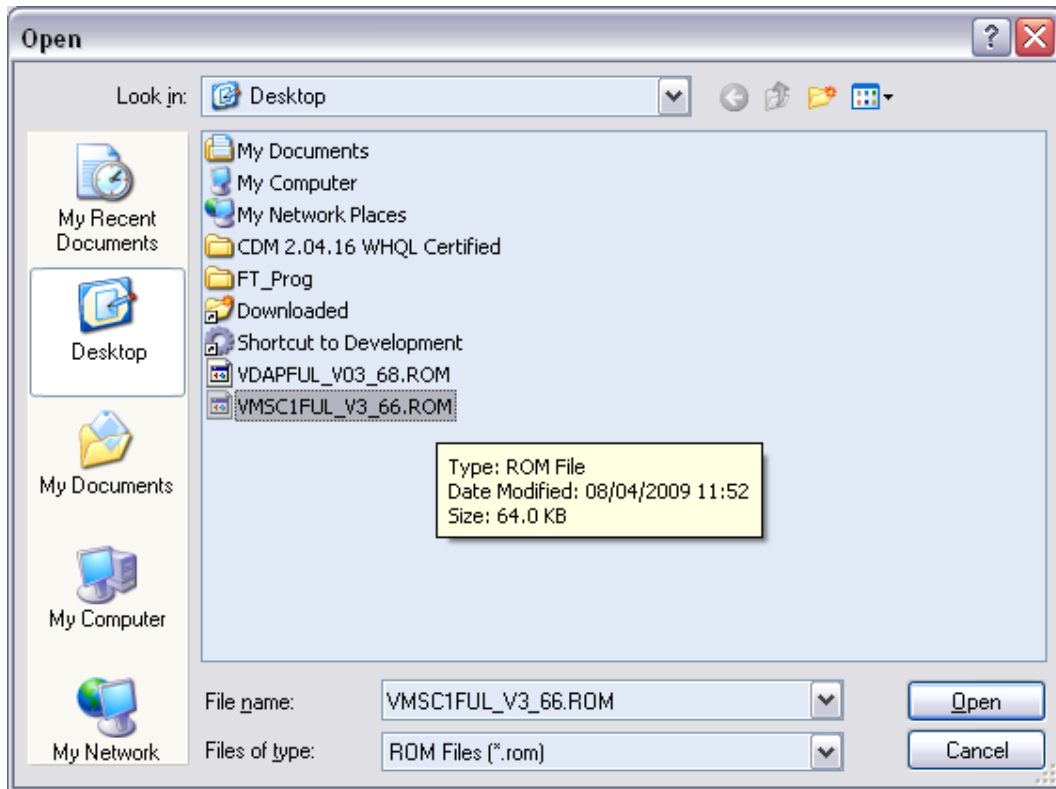
The default Baud Rate for programming a VNC1L device is 1MBaud and 3MBaud for VNC2. These are the recommended baud rates for each device as it allows for faster programming.



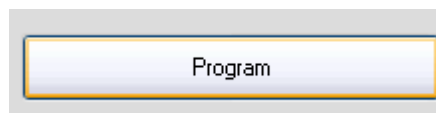
VNC1L firmware is available as a free download from the [Firmware](#) section of the [FTDI Website](#).

VNC2 firmware must first be built using FTDI's [ToolChain](#). The ToolChain is available from the FTDI website and comes with VNC1L equivalent firmware in the shape of V2DAP and V2DPS. After building an application using the ToolChain the resultant ROM file can be programmed into the VNC2. For help with installing and building firmware using the VNC2 ToolChain please refer to the [Getting Started Guide](#).

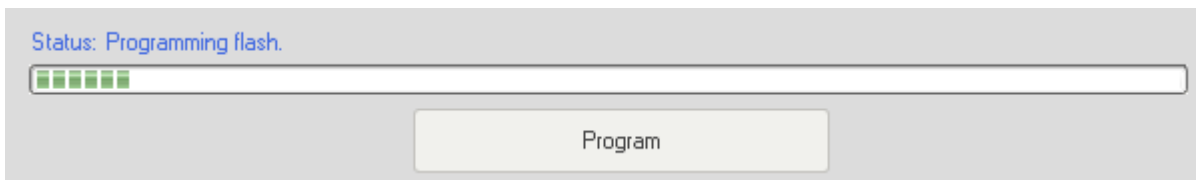
Click the open file button  and browse to the ROM file that has been downloaded.



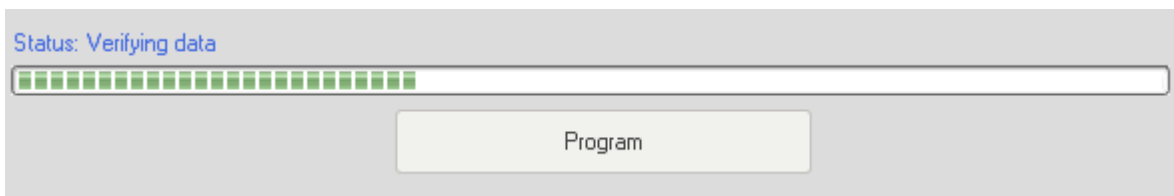
Then press the "Program" button to re-flash the VNC1L firmware.



Programming flash:



Verifying data:





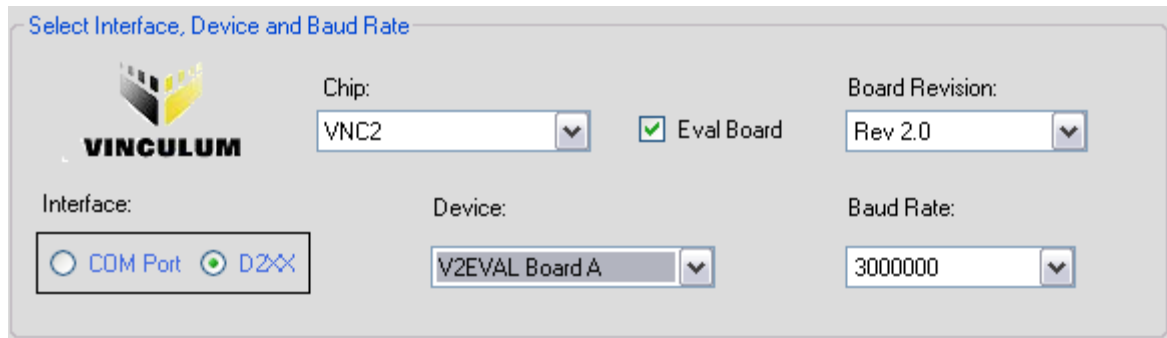
When the programming is complete, the following message is displayed:

Status: Device re-programmed successfully!

Programming Vinculum II example:

Programming the VNC2 is very similar to the VNC1L; in this example a V2Eval board is re-programmed with the V2DAP firmware.

Connect the V2Eval board to the host PC. All connected FTDI devices will appear under the *Device* drop-down menu. V2Eval boards are setup with the UART as channel A of the FT4232H chip. This should be selected from the list as shown below.



Evaluation boards allow the device to be put into prog mode using one of the spare channels of the FT4232H device. The channel used changed between Rev 1 and Rev 2 of the V2Eval board therefore it is required for the user to select the appropriate board revision from the drop-down menu.

The precompiled ROM file for the V2DAP is available as a download from the [FTDI website](#). Browse to this file using the open file browser and press program to flash the ROM image onto the VNC2.

## 9 Definition of Error Messages

### EEPROM mode:

- If when clicking "Scan and Parse" to detect devices, the error message windows as per below is displayed; Check the PC has connected and enumerated the FTDI device correctly.



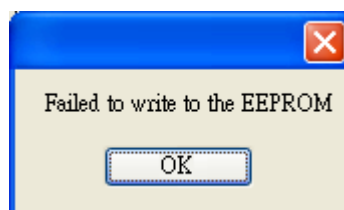
- When programming an EEPROM, a device must be selected. If no device is selected, the error message below is shown.



- When erasing an EEPROM, a device must be selected. If no device is selected, the error message below is shown.



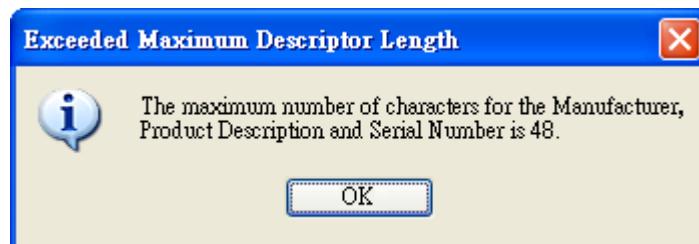
- If the programming doesn't pass, the error message as per the figure below is displayed. This may be a consequence of hardware design or solder issues. Check the schematic and the EEPROM part number fitted. You can refer to [FTDI datasheet](#) to check your schematic



- The FT232R and FT245R have an internal EEPROM which cannot be erased.

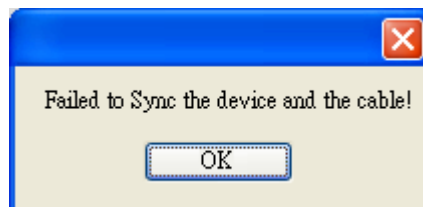


- The Descriptor Length cannot exceed 48 characters.



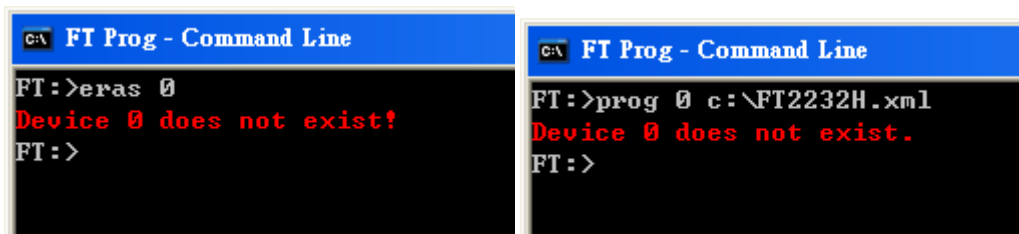
#### Flash ROM mode:

- Failed to sync the VNC1L and the cable, please check the hardware connection.



#### Command line mode:

- System cannot find "Device 0" Use the SCAN command and confirm the device number.



- If the "Failed to write to the EEPROM" error message is displayed, check the cable connection is stable, then try to program this device again. Also, check that the device and the template chip types match.

```
C:\> FT Prog - Command Line
FT:>scan
Scanning for devices...
Device 0: FT_DEVICE_2232H, USB <-> Serial Cable A, FTSDOQ03A
FT:>prog 0 c:\FT2232H.xml
Programming: FTSDOQ03A
Failed to write to the EEPROM
Device 0 programmed successfully!
Finished
```

- Check the device and template chip type match.

```
C:\> FT Prog - Command Line
FT:>scan
Scanning for devices...
Device 0: FT_DEVICE_2232H, USB <-> Serial Cable A, FTSDOQ03A
FT:>prog 0 C:\MC232R.xml
Programming: FTSDOQ03A
Failed to program Device 0!
FT:>
```

- The FT232R and FT245R chip cannot be erased.

```
C:\> FT Prog - Command Line
FT:>scan
Scanning for devices...
Device 0: FT_DEVICE_232R, USB <-> Serial Cable, FTSDJ1Z5
FT:>eras 0
Failed to erase device at location: 66
FT:>_
```

## 10 Contact Information

### Head Office – Glasgow, UK

Future Technology Devices International Limited  
Unit 1, 2 Seaward Place, Centurion Business Park  
Glasgow G41 1HH  
United Kingdom  
Tel: +44 (0) 141 429 2777  
Fax: +44 (0) 141 429 2758

E-mail (Sales) [sales1@ftdichip.com](mailto:sales1@ftdichip.com)  
E-mail (Support) [support1@ftdichip.com](mailto:support1@ftdichip.com)  
E-mail (General Enquiries) [admin1@ftdichip.com](mailto:admin1@ftdichip.com)

### Branch Office – Tigard, Oregon, USA

Future Technology Devices International Limited (USA)  
7130 SW Fir Loop  
Tigard, OR 97223-8160  
USA  
Tel: +1 (503) 547 0988  
Fax: +1 (503) 547 0987

E-mail (Sales) [us.sales@ftdichip.com](mailto:us.sales@ftdichip.com)  
E-mail (Support) [us.support@ftdichip.com](mailto:us.support@ftdichip.com)  
E-mail (General Enquiries) [us.admin@ftdichip.com](mailto:us.admin@ftdichip.com)

### Branch Office – Taipei, Taiwan

Future Technology Devices International Limited (Taiwan)  
2F, No. 516, Sec. 1, NeiHu Road  
Taipei 114  
Taiwan, R.O.C.  
Tel: +886 (0) 2 8791 3570  
Fax: +886 (0) 2 8791 3576

E-mail (Sales) [tw.sales1@ftdichip.com](mailto:tw.sales1@ftdichip.com)  
E-mail (Support) [tw.support1@ftdichip.com](mailto:tw.support1@ftdichip.com)  
E-mail (General Enquiries) [tw.admin1@ftdichip.com](mailto:tw.admin1@ftdichip.com)

### Branch Office – Shanghai, China

Future Technology Devices International Limited (China)  
Room 1103, No. 666 West Huaihai Road,  
Shanghai, 200052  
China  
Tel: +86 21 62351596  
Fax: +86 21 62351595

E-mail (Sales) [cn.sales@ftdichip.com](mailto:cn.sales@ftdichip.com)  
E-mail (Support) [cn.support@ftdichip.com](mailto:cn.support@ftdichip.com)  
E-mail (General Enquiries) [cn.admin@ftdichip.com](mailto:cn.admin@ftdichip.com)

### Web Site

<http://ftdichip.com>

### Distributor and Sales Representatives

Please visit the Sales Network page of the [FTDI Web site](#) for the contact details of our distributor(s) and sales representative(s) in your country.

System and equipment manufacturers and designers are responsible to ensure that their systems, and any Future Technology Devices International Ltd (FTDI) devices incorporated in their systems, meet all applicable safety, regulatory and system-level performance requirements. All application-related information in this document (including application descriptions, suggested FTDI devices and other materials) is provided for reference only. While FTDI has taken care to assure it is accurate, this information is subject to customer confirmation, and FTDI disclaims all liability for system designs and for any applications assistance provided by FTDI. Use of FTDI devices in life support and/or safety applications is entirely at the user's risk, and the user agrees to defend, indemnify and hold harmless FTDI from any and all damages, claims, suits or expense resulting from such use. This document is subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Neither the whole nor any part of the information contained in, or the product described in this document, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH, United Kingdom. Scotland Registered Company Number: SC136640

## Appendix A - References

### Document References

[AN\\_107 Advanced Driver Options](#)

[AN\\_119 FTDI Drivers Installation Guide for Windows 7](#)

[AN\\_142 Vinculum-II Tool Chain Getting Started Guide](#)

[AN\\_234 FTDI Drivers Installation Guide for Windows 8](#)

[AN\\_396 FTDI Drivers Installation Guide for Windows 10](#)

### Acronyms and Abbreviations

Terms	Description
CPU	Central Processing Unit
EEPROM	Electrically Erasable Programmable Read Only Memory
FIFO	First In First Out
I2C	Inter Integrated Circuit
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
Win 7	Windows 7

## Appendix B – Revision History

Document Title : AN\_124 User Guide for FTDI FT\_PROG Utility  
Document Reference No. : FT\_000172  
Clearance No. : FTDI# 106  
Product Page : <http://www.ftdichip.com/FTPProducts.htm>  
Document Feedback : [Send Feedback](#)

Revision	Changes	Date
1.0	Initial Release	2009-09-14
1.1	Added note about .NET requirement	2009-09-21
1.2	Update to Section 7 – Command Line Programming	2010-06-04
1.3	Added reference to the VNC2; Added Batch file programming section to the command line tool	2010-09-09
1.4	Added new section for Programming the V2Eval board; Added reference to the FT232H Device Specific Settings	2011-08-19
1.5	Added reference to FT4222H	2014-11-17
1.6	Added reference to FT260	2016-04-06