

Cómo extraer y grabar el binario de los FTDI (para cualquier fabricante de FPGAs-SoCs)

Creador: David Rubio G.

Entrada: <https://soceame.wordpress.com/2025/02/03/como-extraer-y-grabar-el-binario-de-los-ftdi-para-cualquier-fabricante-de-fpgas-socs/>

Blog: <https://soceame.wordpress.com/>

GitHub: <https://github.com/DRubioG>

Fecha última modificación: 24/02/2025

<https://soceame.wordpress.com/2025/02/03/como-extraer-y-grabar-el-binario-de-los-ftdi-para-cualquier-fabricante-de-fpgas-socs/>

Como ya he comentado en dos entradas anteriores para Xilinx.

<https://soceame.wordpress.com/2025/02/01/como-utilizar-un-chip-de-ftdi-como-cable-depurador-de-xilinx/>

<https://soceame.wordpress.com/2025/02/02/como-anadir-la-uart-a-un-ft232-con-jtag/>

Pues ahora aplicado a cualquier FTDI de cualquier fabricante de FPGAs.

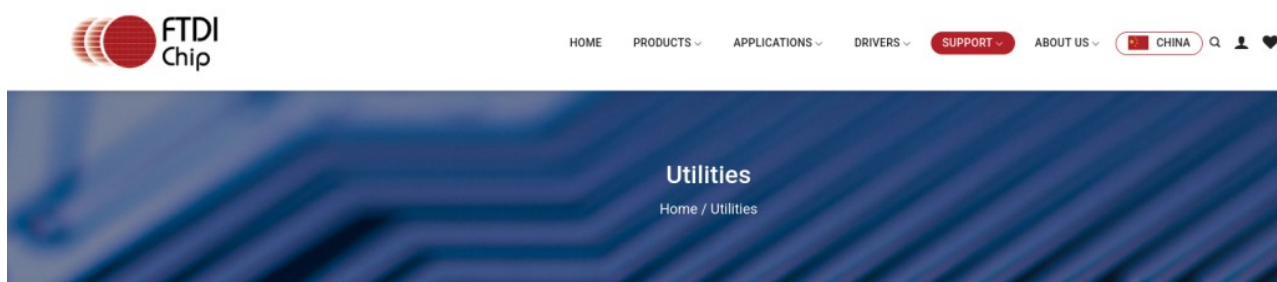
Extraer el binario

Para poder realizar la tarea se requiere de dos sistemas operativos, uno en Windows y otro en Linux.

El sistema operativo de Windows es para ejecutar la aplicación FT Prog de FTDI para detectar el número de vendedor y el ID del producto, y el sistema operativo Linux para extraer el binario.

- **Windows**

Lo primero que necesitamos es en el sistema operativo Windows instalarnos el programa FT Prog.



Utilities

The utilities page contains useful programs that provide extended capabilities for application development and manufacturing support. Engineers should take care to validate output from these utilities. Extreme vigilance should be exercised when dealing with programming/download utilities so that device contents are not erroneously deleted or corrupted.

- [FT60x Configuration](#)
- [FT_PROG](#)
- [CDM Uninstaller](#)
- [Microsoft USBView](#)
- [Linux USBView](#)
- [FT312D Configuration](#)
- [CDM PORT Assignment](#)
- [FT311D Configuration](#)
- [FTDIPort Monitor](#)
- [Mini Module EEPROM Recovery](#)
- [FT232R Clock utility](#)
- [VEVAL](#)
- [VNC_PROG](#)
- [V2 PROG](#)
- [Vinculum Customiser](#)
- [Ignore HW Serial Number Reg Edit](#)
- [ST232R v1.4](#)
- [ST232R](#)
- [FT2232ST](#)
- [ST232-245](#)
- [FTD2XXST](#)
- [E2PROG](#)
- [EEPROM Prog for RSD](#)
- [VPLAY](#)
- [Win CE VCP Test](#)
- [FT2232 UART Test](#)
- [FT4232H UART Test](#)

FT_PROG 3.12.55.667 - EEPROM Programming Utility

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 Vinculum firmware.

PLEASE NOTE - The use of some of these utilities by an end user may result in a device being rendered useless.

FT_PROG is available for download by clicking [here](#).

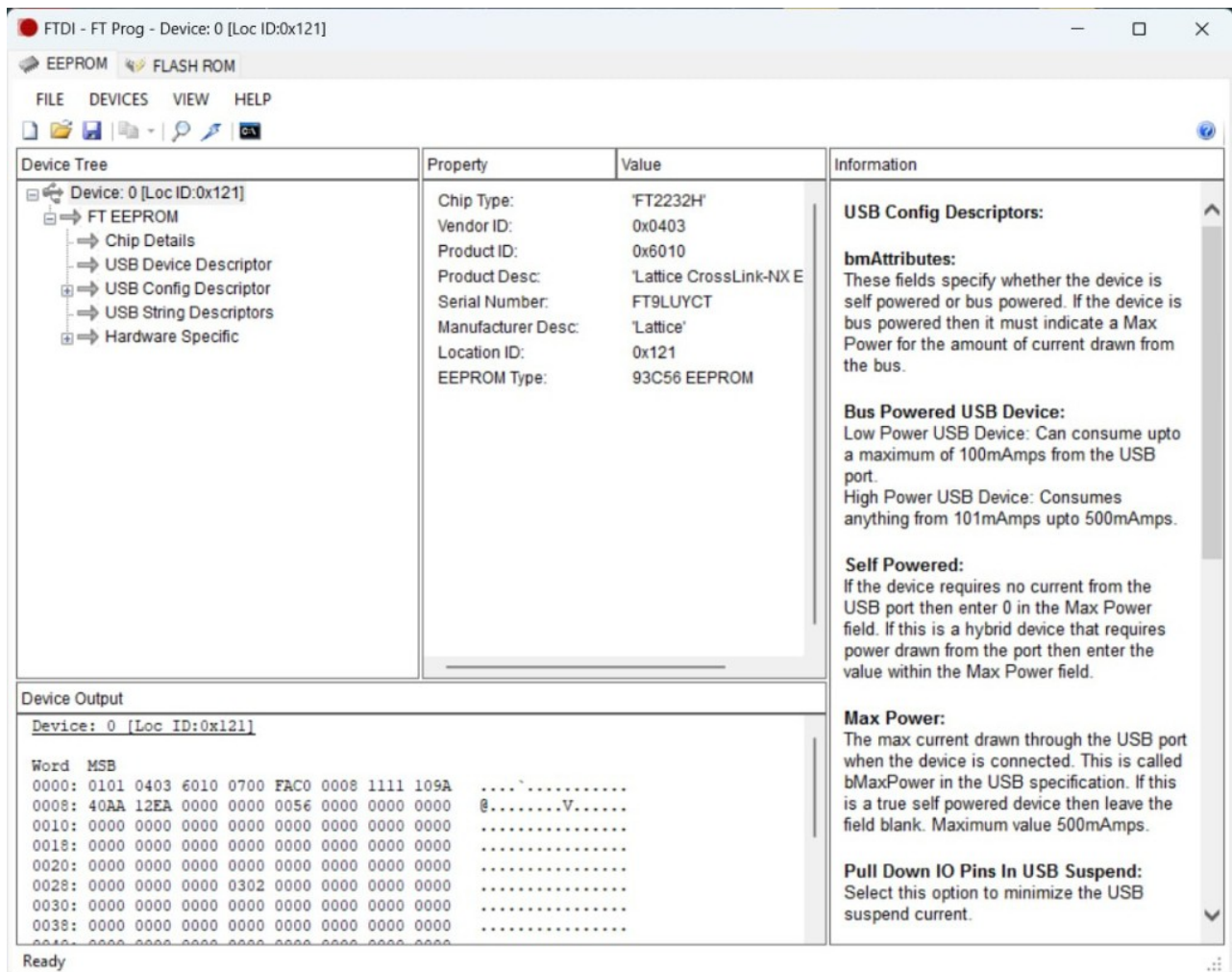
The full FT_PROG User Guide can be downloaded [here](#).

Please Note: FT_PROG requires the Microsoft .NET Framework 4.0 installed on your system to run the application. This can be obtained from the Microsoft Website https://www.microsoft.com/download/en/details.aspx?id=17851&WT.mc_id=MSCOM_EN_US_DLG_DETAILS_121LSUS007996

If your system does not have .NET 4.0 installed please download the file from the above link. To install, double click on the dotnetfx.exe and follow the instructions in the wizard.

Este programa evita que vayamos a ciegas con los chips de FTDI, gracias a que nos dice el dispositivo que tenemos acoplado al ordenador.

Una vez instalado, enganchamos el cable que tiene el chip y le damos a Scan and Parse. Esto hace que busque los dispositivos de FTDI que haya acoplados.



Una vez encuentra el dispositivo FTDI que tenemos en el extremo del cable, también nos da el **Vendor ID** y el **Product ID** que son los números que necesitaremos en el sistema operativo Linux.

- **Linux**

Ahora en Linux necesitamos tener instalado los siguientes paquetes.

```
sudo apt-get install libftdi1 ftdi-eepprom
```

Con estos dos paquetes ahora lo único que necesitamos es crear un fichero .conf que contenga tenga lo siguiente dentro.

```
vendor_id=<vendor ID>
product_id=<product ID>
```

<https://soceame.wordpress.com/2025/02/03/como-extraer-y-grabar-el-binario-de-los-ftdi-para-cualquier-fabricante-de-fpgas-socs/>

```
flash_raw=true
```

```
filename="<nombre del binario>.bin"
```

Y ahora lo siguiente es ejecutar el comando de lectura del binario.

```
sudo ftdi_eeeprom --read-eeeprom <nombre del .CONF>.conf
```

Esto lo que hace es que aparezca un fichero binario (.bin) con el binario que tenía dentro el FTDI.

```
00000000 01 08 03 04 10 60 00 07 80 fa 00 00 11 11 9a 10 |.....`.....|
00000010 aa 3c 00 00 00 00 00 00 56 00 00 00 00 00 00 ||.<.....V.....|
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ||.....|
*
00000090 00 00 00 00 00 00 00 00 00 00 10 03 4c 00 61 00 ||.....L.a.|
000000a0 74 00 74 00 69 00 63 00 65 00 3c 03 4c 00 61 00 ||t.t.i.c.e.<.L.a.|
000000b0 74 00 74 00 69 00 63 00 65 00 20 00 46 00 54 00 ||t.t.i.c.e. .F.T.|
000000c0 55 00 53 00 42 00 20 00 49 00 6e 00 74 00 65 00 ||U.S.B. .I.n.t.e.|
000000d0 72 00 66 00 61 00 63 00 65 00 20 00 43 00 61 00 ||r.f.a.c.e. .C.a.|
000000e0 62 00 6c 00 65 00 00 00 00 00 00 00 00 00 00 ||b.l.e.....|
000000f0 00 00 00 00 00 00 00 00 00 00 00 00 00 5b 12 ||.....[.|
00000100
```

Y desde este instante el cable es totalmente funcional.

Grabar el binario

Para grabar el binario lo único es que hay que tener el FTDI conectado. Y solo hace falta llamar al comando que graba el binario que tenemos.

```
sudo ftdi_eeeprom --flash-eeeprom <binario_depurador>.conf
```

Repositorio con binarios

Aquí dejo los enlaces con los diferentes binarios para los distintos tipos de FPGAs/SoCs:

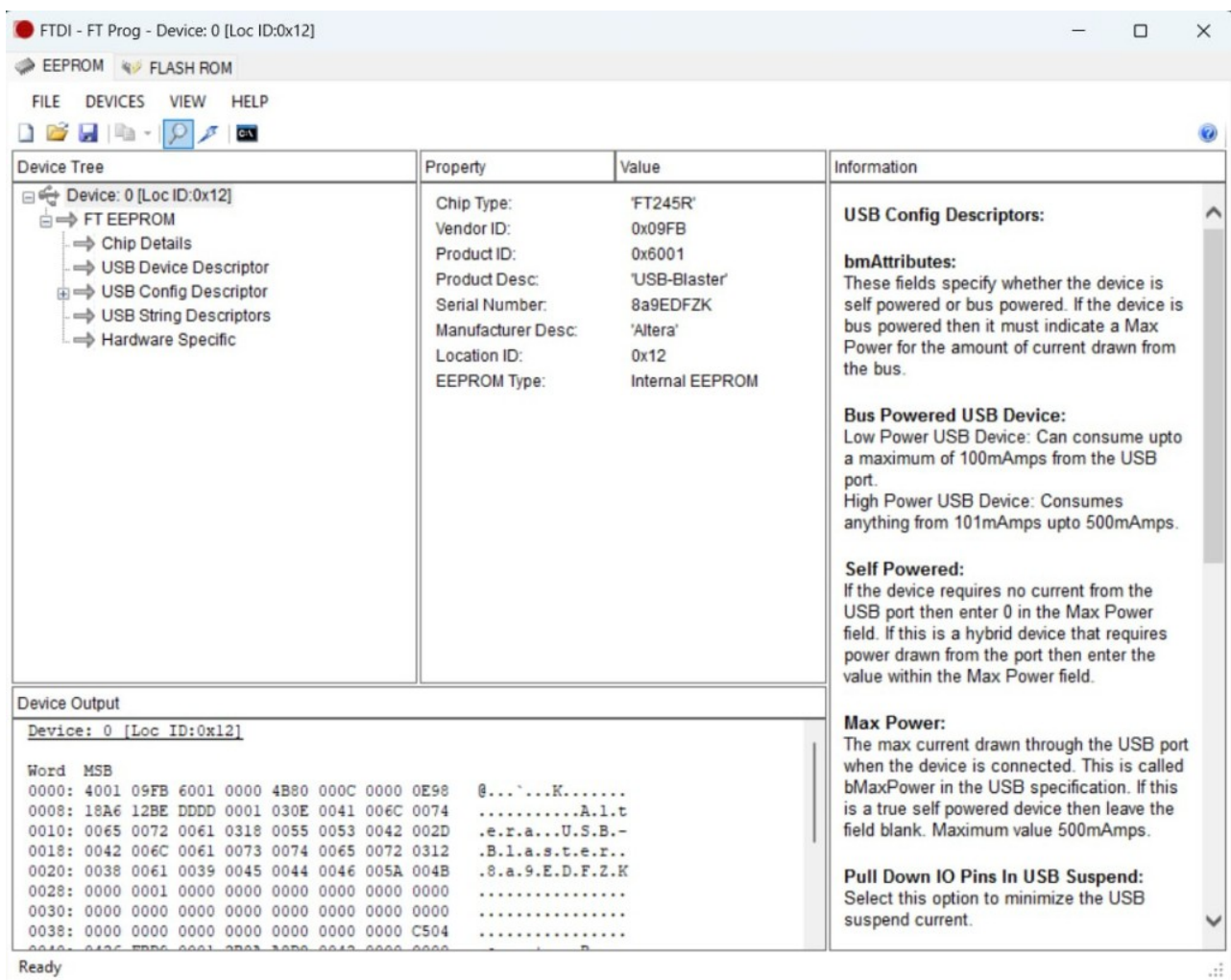
- Xilinx: https://github.com/DRubioG/FTDI_flash_binaries_Xilinx
- Altera/Intel: https://github.com/DRubioG/FTDI_flash_binaries_Altera_Intel
- Microchip: https://github.com/DRubioG/FTDI_flash_binaries_Microchip
- Lattice: https://github.com/DRubioG/FTDI_flash_binaries_Lattice

Ejemplo Quartus

Ahora queremos sacar el código binario de un cable depurador de Quartus como este.



Lo primero que hacemos es conectarlo a un Windows y ejecutar la aplicación FT Prog.



Entonces, al darle a detectar, podemos ver que dentro tiene un FT245R (este chip sólo tiene una interfaz, por lo que sólo puede utilizarse como JTAG). Ahora lo único que tenemos que hacer es capturar los siguiente parámetros:

- Vendor ID: 0x09FB
- Product ID: 0x6001

Ahora en Linux creamos el siguiente fichero .conf

```
vendor_id=0x09FB  
product_id=0x6001
```

```
flash_raw=true
```

```
filename="jatg_quartus.bin"
```

Ahora lo único que tenemos que hacer es acoplar el cable al Linux y ejecutar el comando de lectura.

```
FTDI eeprom generator v0.17
(c) Intra2net AG and the libftdi developers <opensource@intra2net.com>
FTDI read eeprom: 0
EEPROM size: 128
FTDI close: 0
```

Ahora solo tenemos que abrir el binario para ver lo que tiene guardado.

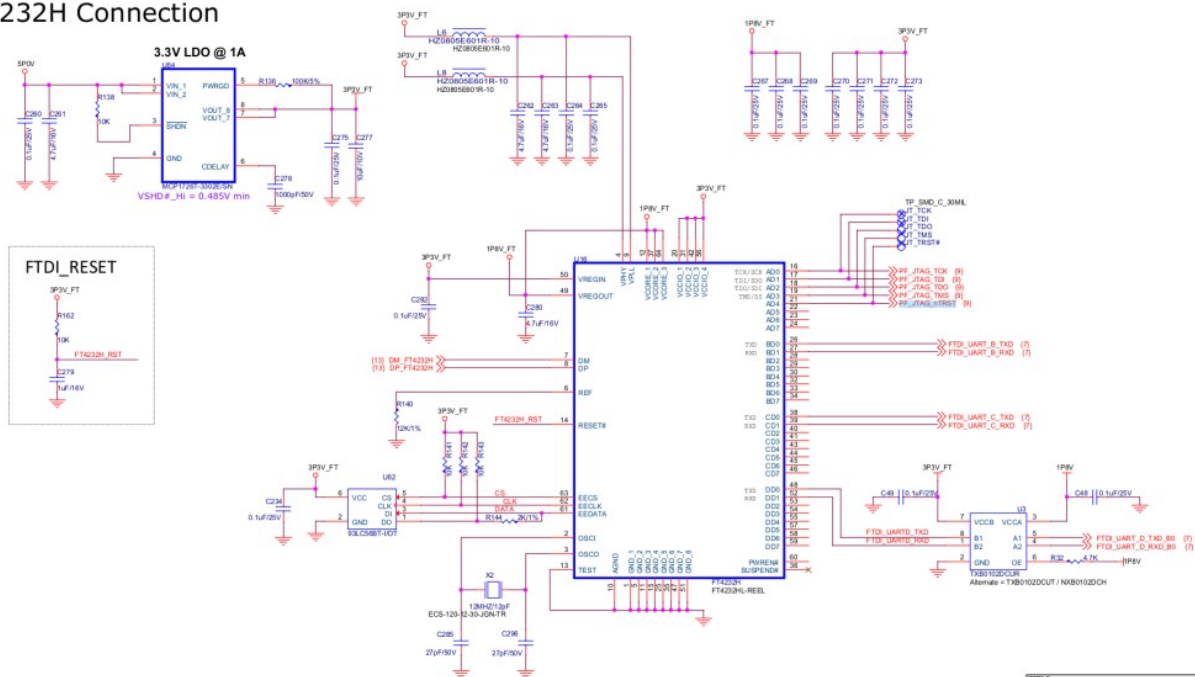
00000000	01 40 fb 09 01 60 00 00	80 4b 0c 00 00 00 98 0e	.@...`...K.....
00000010	a6 18 be 12 dd dd 01 00	0e 03 41 00 6c 00 74 00A.l.t.
00000020	65 00 72 00 61 00 18 03	55 00 53 00 42 00 2d 00	e.r.a...U.S.B.-
00000030	42 00 6c 00 61 00 73 00	74 00 65 00 72 00 12 03	B.l.a.s.t.e.r...
00000040	38 00 61 00 39 00 45 00	44 00 46 00 5a 00 4b 00	8.a.9.E.D.F.Z.K.
00000050	00 00 01 00 00 00 00 00	00 00 00 00 00 00 00 00
00000060	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
00000070	00 00 00 00 00 00 00 00	00 00 00 00 00 00 04 c5
00000080			

Y ahora para grabarlo solo tenemos que ejecutar el comando de grabación sobre el chip que queremos.

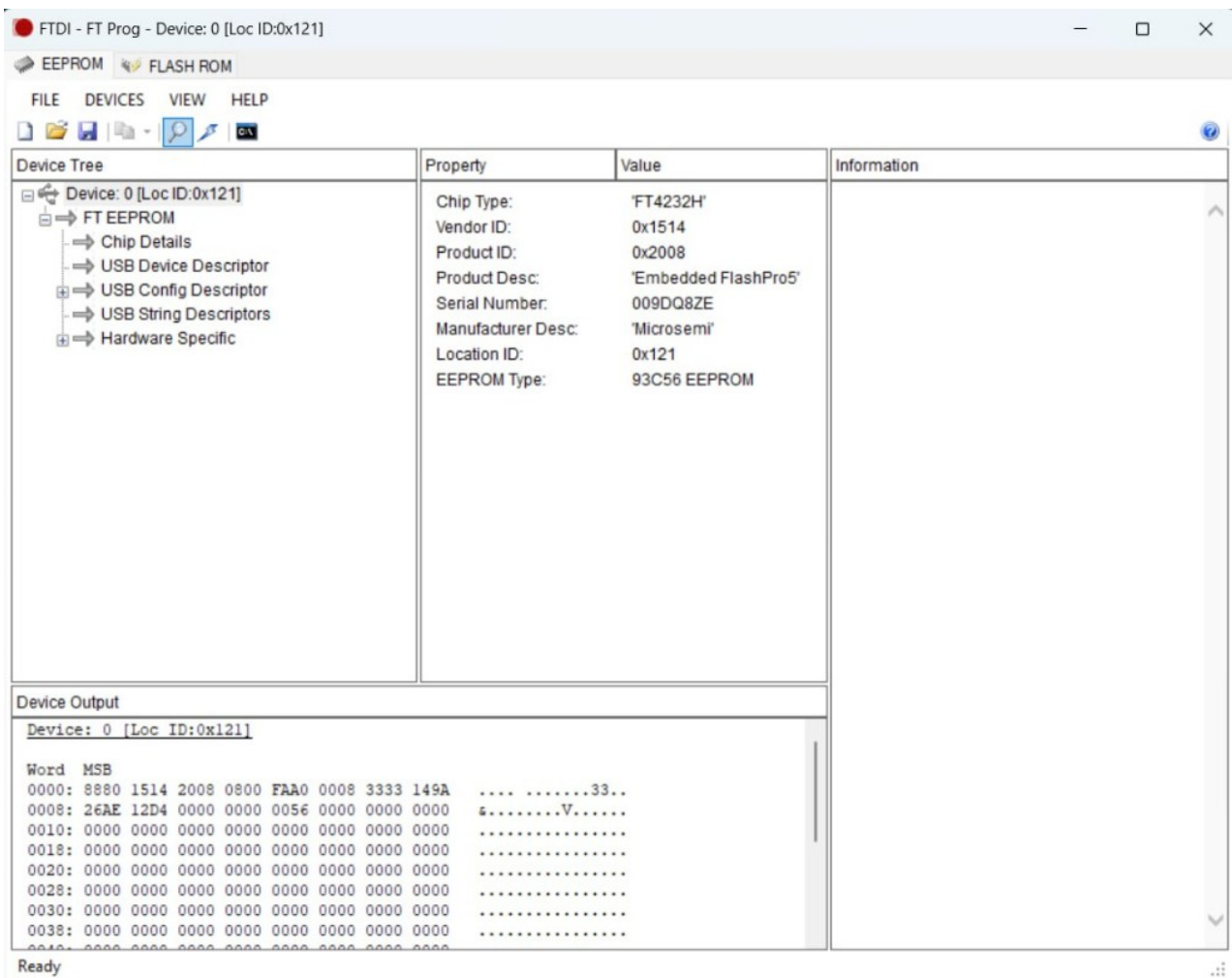
Ejemplo Microchip

Para este ejemplo tenemos un FTDI de tipo FT4232 según nos dice el esquemático de la placa (PolarFire SoC Discovery Kit).

FT4232H Connection



Ahora como en el ejemplo anterior abrimos el FT Prog, y comprobamos el dispositivo.



Podemos ver que tenemos un FT4232, que tiene hasta 4 interfaces. Ahora lo que hacemos es extraer los siguientes datos.

- Vendor ID: 0x1514
- Product ID: 0x2008

Ahora con estos parámetros vamos a Linux y creamos un fichero llamado *jtag_microchip.conf*, este fichero contendrá el siguiente código.

```
vendor_id=0x1514
product_id=0x2008
```

```
flash_raw=true
```

```
filename="jtag_microchip.bin"
```

Ahora lo único que tenemos que hacer es lanzar el comando de lectura con el dispositivo acoplado.

```
sudo ftdi_eeprom --read-eeprom jtag_microchip.conf
```

Cuando termine nos devuelve que está correcto.


```
FTDI eeprom generator v0.17
(c) Intra2net AG and the libftdi developers <opensource@intra2net.com>
FTDI read eeprom: 0
EEPROM size: 256
FTDI close: 0
```

Ahora se lo podemos grabar sobre cualquier dispositivo.

Nota final

Y así se puede hacer con cualquier fabricante al que se le quiera extraer o grabar el binario de un FTDI.

Y por último, el FT Prog también nos devuelve el tipo de memoria que tiene acoplado el FTDI en el campo EEPROM Type.