# Cómo utilizar un chip de FTDI como cable depurador de Xilinx



Creador: David Rubio G.

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

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

GitHub: https://github.com/DRubioG

Fecha última modificación: 24/02/2025

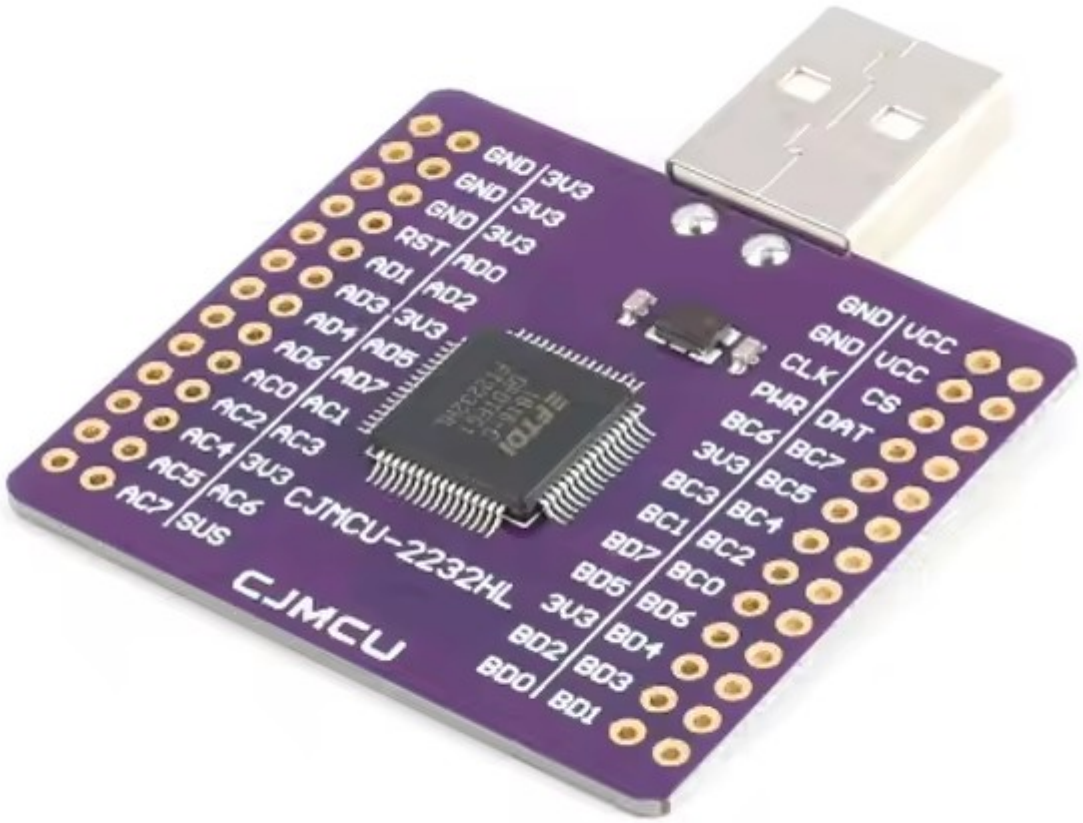
*(Esta entrada está más enfocada a Linux. Para Windows tienes la siguiente entrada. Aún así te recomiendo leer esta entrada para entender los conceptos).*

https://soceame.wordpress.com/2025/02/19/como-programar-un-ftdi-desde-vivado/

Las nuevas FPGAs de Xilinx (y las de otras fabricantes) ya están prescindiendo de los cables depuradores, para ello están haciendo uso de los dispositivos de FTDI (FT2232, FT4232 y FT232H **[Importante: tiene que ser la versión H del FT232,** no las otras]). Estos dispositivos poseen la conversión de USB a JTAG, esta conversión hace que muchos cables depuradores y clones de estos estén basados en este chip.
Lo que ocurre es que este chip no es un chip de «plug-and-play», si no que necesita tener cargado un binario dentro para que Vivado lo reconozca. Entonces, lo que se puede hacer es extraer este binario necesario de las diferentes placas y clones del mercado mediante un software. Además, este mismo software también permite cargar el binario dentro del dispositivo de la memoria donde se guarda.
Este chip además posee diferentes interfaces, por lo que el USB que hace que de interfaz JTAG también puede hacer de puerto serie, SPI, I2C, etc.

- El FT232H, esta versión solo tiene una interfaz, por lo que solo se puede utilizar para JTAG.
- El FT2232 permite hasta dos interfaces, una para el JTAG y otra para otros protocolos, como por ejemplo la UART.
- El FT4232 permite hasta cuatro interfaces, una para el JTAG y tres para otros protocolos, como por ejemplo 3 UARTs.

**NOTA**: para poder probar todo lo que se comenta en esta entrada se recomienda hacerse con una placa de pruebas, para comprobar que la configuración es la correcta. Las placas de pruebas son como la siguiente, que por menos de 20€ puedes comprobar que todo funciona correctamente. Estas placas tienen que tener los dispositivos FT2232, FT4232 y FT232H.




Ahora comentaré el proceso para hacerse con el código binario de los diferentes dispositivos que llevan incorporado este chip (*también puedes ahorrarte este paso cogiendo los binarios de este repositorio de GitHub*: https://github.com/DRubioG/FTDI_flash_binaries_Xilinx). En el siguiente apartado explico cómo se tiene que grabar el binario en el FTDI.

# Descargar el binario de un FTDI

*(Esta operación te la puedes saltar con el repositorio de arriba)*

Para hacer la descarga necesitamos operar desde un sistema operativo Linux *(desconozco si en Windows también se puede hacer este paso)*.

Primero nos descargamos estos paquetes, que son los que se encargarán de descargar los binarios.

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

Después, creamos un fichero .conf con el nombre de nuestro proyecto, por ejemplo, ***binario_depurador.conf***. Dentro de ese fichero ponemos este código *(esto se aplica a placas de*

*Digilent que tienen el dispositivo de FTDI integrado)*, este código tiene que incluir un fichero .bin dentro:

```
vendor_id=0x0403
product_id=0x6010

flash_raw=true

filename="binario_depurador.bin"        # Filename, leave empty to skip file
writing
```

**NOTA**: para conocer los parámetros vendor_id y product_id necesitas utilizar el programa FT Prog que solo existe para Windows. En esta entrada dejo cómo se hace.

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

Ahora lo que hacemos es en el mismo directorio en que hemos creado el fichero abrimos la terminal y ejecutamos el comando que descarga el binario del chip en nuestro ordenador.

```
sudo ftdi_eeprom --read-eeprom <binario_depurador>.conf
```

Entonces, al ejecutarlo tenemos que tener dos ficheros el .conf que hemos creado y un fichero .bin que es el binario extraído del chip.

Ahora si ejecutamos el comando podemos ver el binario que se ha bajado del FTDI:

```
hexdump -C <binario_depurador>.bin
```

Podemos ver el código que tiene dentro del binario que nos hemos bajado del FT2232.

```
00000000  01 08 03 04 10 60 00 07  80 00 08 00 00 00 9a 12  |.....`..........|
00000010  ac 34 e0 1a 00 00 00 00  56 00 01 00 c7 92 6a 35  |.4......V.....j5|
00000020  57 01 50 f0 50 59 4e 51  2d 5a 31 00 00 00 00 00  |W.P.PYNQ-Z1.....|
00000030  00 00 00 00 00 58 69 6c  69 6e 78 20 50 59 4e 51  |.....Xilinx PYNQ|
00000040  2d 5a 31 00 00 00 00 00  00 00 00 00 00 00 00 00  |-Z1.............|
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  |................|
*
00000090  00 00 00 00 00 00 00 00  00 00 12 03 44 00 69 00  |............D.i.|
000000a0  67 00 69 00 6c 00 65 00  6e 00 74 00 34 03 44 00  |g.i.l.e.n.t.4.D.|
000000b0  69 00 67 00 69 00 6c 00  65 00 6e 00 74 00 20 00  |i.g.i.l.e.n.t. .|
000000c0  41 00 64 00 65 00 70 00  74 00 20 00 55 00 53 00  |A.d.e.p.t. .U.S.|
000000d0  42 00 20 00 44 00 65 00  76 00 69 00 63 00 65 00  |B. .D.e.v.i.c.e.|
000000e0  1a 03 30 00 30 00 33 00  30 00 31 00 37 00 42 00  |..0.0.3.0.1.7.B.|
000000f0  33 00 45 00 44 00 46 00  44 00 02 03 00 00 02 a7  |3.E.D.F.D.......|
00000100
```

# Cargar el binario en un FTDI

*(Si has llegado a este paso sin el apartado anterior, es necesario que descargues los paquetes del paso anterior para poder grabar el dispositivo)*

Ahora para cargar el binario en el FTDI tenemos que ejecutar el comando:

```
sudo ftdi_eeprom --flash-eeprom <binario_depurador>.conf
```

Para que el comando anterior funcione tiene que tener además del fichero .conf un fichero .bin.

Si todo ha ido correctamente saldría un mensaje como el siguiente:

```
FTDI eeprom generator v0.17
(c) Intra2net AG and the libftdi developers <opensource@intra2net.com>
FTDI read eeprom: 0
EEPROM size: 256
FIXME: Build FT232H specific EEPROM settings
Used eeprom space: 244 bytes
Flashing raw eeprom from file jtag-alinx-ft232h.bin (256 bytes)
FTDI write eeprom: 0
FTDI close: 0
```

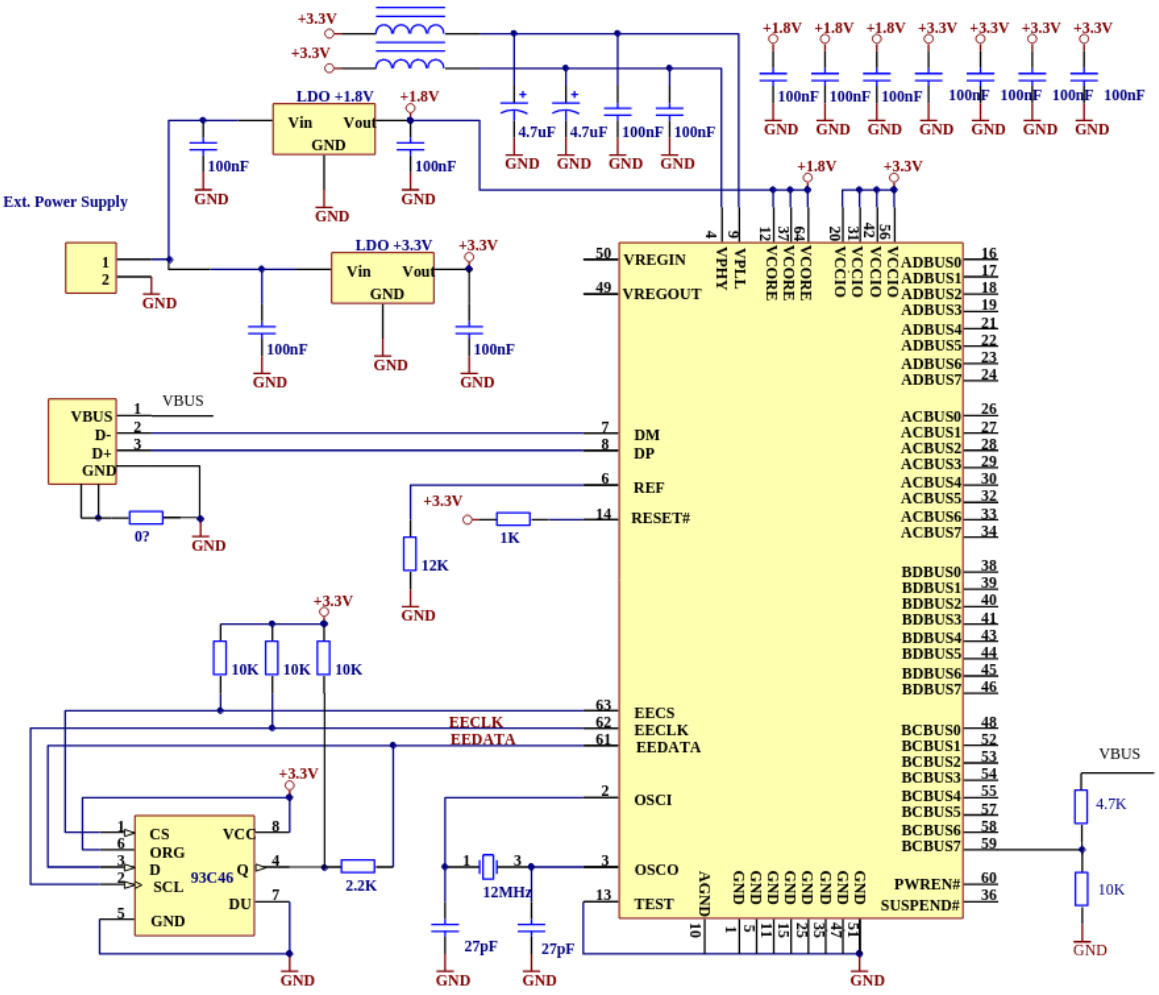
Si ha habido un error en la grabación dará un mensaje como el siguiente:

```
FTDI eeprom generator v0.17
(c) Intra2net AG and the libftdi developers <opensource@intra2net.com>
Unable to find FTDI devices under given vendor/product id: 0x403/0x6010
Error code: -3 (device not found)
Retrying with default FTDI pid=0x6001.
Error: device not found
```

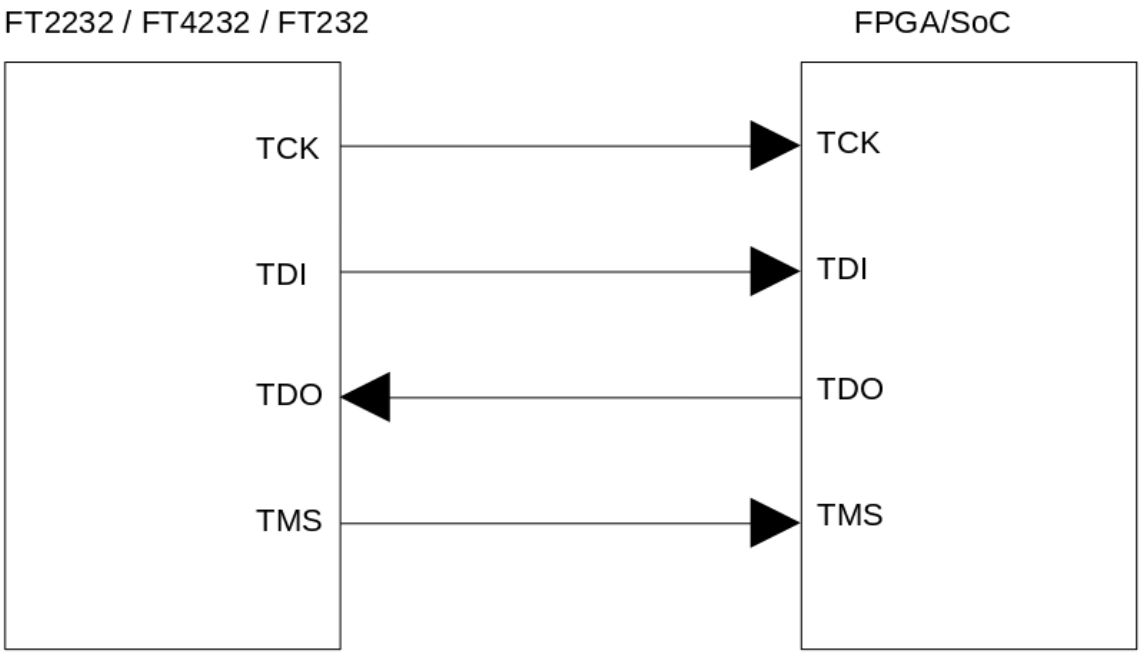
Esto es debido a que existen diferentes chip y es necesario que el binario se pueda asociar a cada chip de forma correcta, eso significa que un binario para un FT232 no se puede cargar en un FT2232 y viceversa.

# Conexión

Es conveniente saber que el FTDI no tiene una memoria interna dónde guardar el binario, por lo que, según es el esquemático de conexión que da fabricante, es necesario añadir una memoria EEPROM (el esquema pone una 93C46).




La conexión entre el chip programador y la FPGA se tiene que hacer siguiendo el siguiente esquema.




**NOTA**: en caso de no hacerse así no habrá reconocimiento de la FPGA.

**NOTA 2**: puede parecer que el esquema está mal, pero no es así, esto es debido a que el dispositivo FTDI tiene los pines nombrados de cara al JTAG que se le va a conectar.

Ahora cada dispositivo en su datasheet pone los pines que se pueden utilizar para JTAG.

- **FT232H**, este dispositivo solo tiene un JTAG asociado a unos pines determinados.

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

| FT232H | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Pin | | Pin functions (depends on configuration) | | | | | | | | |
| Pin # | Pin Name | ASYNC Serial (RS232) | SYNC 245 FIFO | STYLE ASYNC 245 FIFO | ASYNC Bit-bang | SYNC Bit-bang | MPSSE | Fast Serial interface | CPU Style FIFO | FT1248 |
| 13 | ADBUS 0 | TXD | D0 | D0 | D0 | D0 | TCK/SK | FSDI | D0 | MIOSI0 |
| 14 | ADBUS 1 | RXD | D1 | D1 | D1 | D1 | TDI/DO | FSCLK | D1 | MIOSI1 |
| 15 | ADBUS 2 | RTS# | D2 | D2 | D2 | D2 | TDO/DI | FSDO | D2 | MIOSI2 |
| 16 | ADBUS 3 | CTS# | D3 | D3 | D3 | D3 | TMS/CS | FSCTS | D3 | MIOSI3 |
| 17 | ADBUS 4 | DTR# | D4 | D4 | D4 | D4 | GPIOL0 | ** TriSt-UP | D4 | MIOSI4 |
| 18 | ADBUS 5 | DSR# | D5 | D5 | D5 | D5 | GPIOL1 | ** TriSt-UP | D5 | MIOSI5 |
| 19 | ADBUS 6 | DCD# | D6 | D6 | D6 | D6 | GPIOL2 | ** TriSt-UP | D6 | MIOSI6 |
| 20 | ADBUS 7 | RI# | D7 | D7 | D7 | D7 | GPIOL3 | ** TriSt-UP | D7 | MIOSI7 |
| 21 | ACBUS 0 | * TXDEN | RXF# | RXF# | ACBUS0 | ACBUS0 | GPIOH0 | ** ACBUS0 | CS# | SCLK |
| 25 | ACBUS 1 | ** ACBUS1 | TXE# | TXE# | WRSTB# | WRSTB# | GPIOH1 | ** ACBUS1 | A0 | SS_n |
| 26 | ACBUS 2 | ** ACBUS2 | RD# | RD# | RDSTB# | RDSTB# | GPIOH2 | ** ACBUS2 | RD# | MISO |
| 27 | ACBUS 3 | * RXLED# | WR# | WR# | ACBUS3 | ACBUS3 | GPIOH3 | ** ACBUS3 | WR# | ACBUS3 |
| 28 | ACBUS 4 | * TXLED# | SIWU# | SIWU# | SIWU# | SIWU# | GPIOH4 | SIWU# | Note1 | ACBUS4 |
| 29 | ACBUS 5 | ** ACBUS5 | CLKOUT | ACBUS5 | ** ACBUS5 | ** ACBUS5 | GPIOH5 | ** ACBUS5 | ** ACBUS 5 | ACBUS5 |
| 30 | ACBUS 6 | ** ACBUS6 | OE# | ACBUS6 | ACBUS6 | ACBUS6 | GPIOH6 | ** ACBUS6 | ** ACBUS 6 | ACBUS6 |
| 31 | ACBUS 7 | WRSAV# | PWRSAV # | PWRSAV # | PWRSAV # | PWRSAV # | *** GPIOH7 | PWRSAV# | PWRSA V# | PWRSAV # |
| 32 | ACBUS 8 | ** ACBUS8 | ** ACBUS8 | ** ACBUS8 | ** ACBUS8 | ** ACBUS8 | ** ACBUS 8 | ** ACBUS8 | ** ACBUS 8 | ACBUS8 |
| 33 | ACBUS 9 | ** ACBUS9 | ** ACBUS9 | ** ACBUS9 | ** ACBUS9 | ** ACBUS9 | ** ACBUS 9 | ** ACBUS9 | ** ACBUS 9 | ACBUS9 |

- **FT2232**, este dispositivo tiene dos canales en los que ambos pueden hacer de JTAG. Por lo que se puede tener 1 UART y 1 JTAG.

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

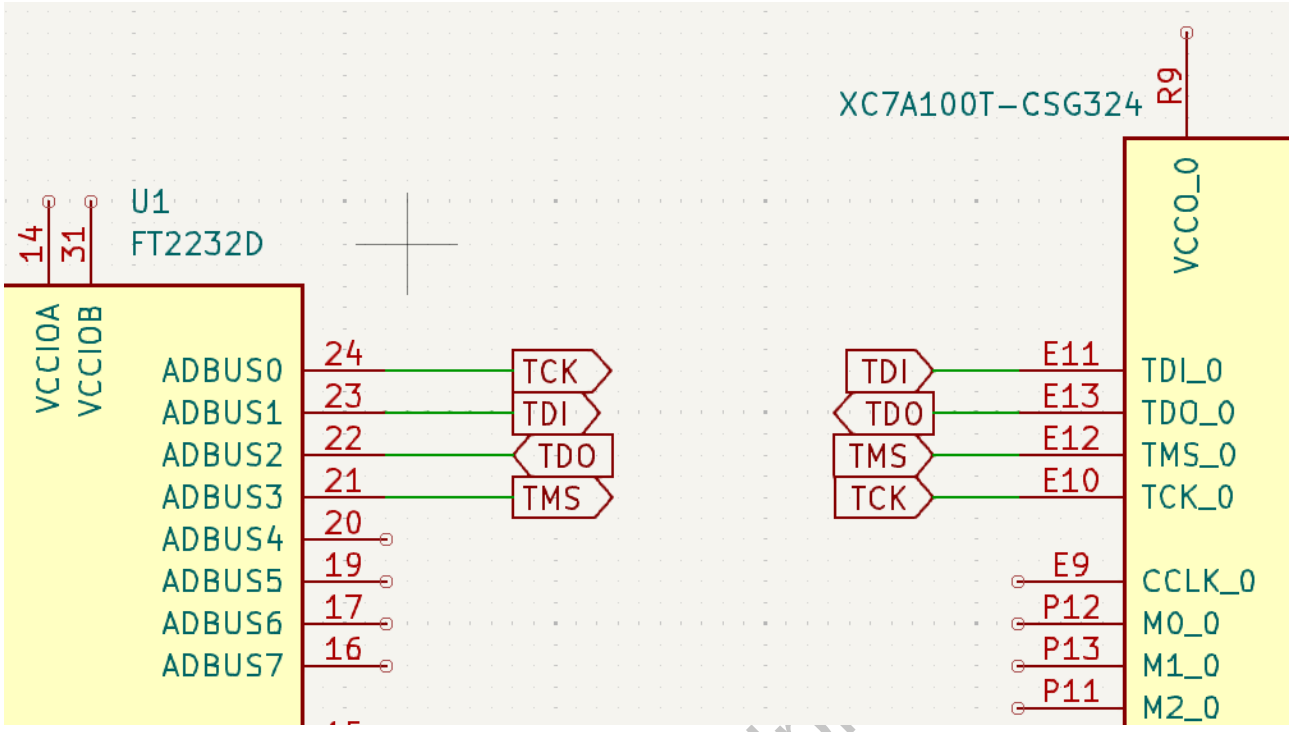
| Pin # | Pin Name | ASYNC Serial (RS232) | 245 FIFO SYNC | 245 FIFO | ASYNC Bit-bang | SYNC Bit-bang | MPSSE | Fast Serial interface | CPU Style FIFO | Host Bus Emulation |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **Channel A** | | | | | | | | |
| 16 | ADBUS0 | TXD | D0 | D0 | D0 | D0 | TCK/SK | USES | D0 | AD0 |
| 17 | ADBUS1 | RXD | D1 | D1 | D1 | D1 | TDI/DO | | D1 | AD1 |
| 18 | ADBUS2 | RTS# | D2 | D2 | D2 | D2 | TDO/DI | | D2 | AD2 |
| 19 | ADBUS3 | CTS# | D3 | D3 | D3 | D3 | TMS/CS | | D3 | AD3 |
| 21 | ADBUS4 | DTR# | D4 | D4 | D4 | D4 | GPIOL0 | | D4 | AD4 |
| 22 | ADBUS5 | DSR# | D5 | D5 | D5 | D5 | GPIOL1 | | D5 | AD5 |
| 23 | ADBUS6 | DCD# | D6 | D6 | D6 | D6 | GPIOL2 | | D6 | AD6 |
| 24 | ADBUS7 | RI# | D7 | D7 | D7 | D7 | GPIOL3 | | D7 | AD7 |
| 26 | ACBUS0 | TXDEN | RXF# | RXF# | ** | ** | GPIOH0 | | CS# | A8 |
| 27 | ACBUS1 | ** | TXE# | TXE# | WRSTB | WRSTB | GPIOH1 | | A0 | A9 |
| 28 | ACBUS2 | ** | RD# | RD# | RDSTB# | RDSTB# | GPIOH2 | | RD# | A10 |
| 29 | ACBUS3 | RXLED# | WR# | WR# | ** | ** | GPIOH3 | | WR# | A11 |
| 30 | ACBUS4 | TXLED# | SIWUA | SIWUA | SIWUA | SIWUA | GPIOH4 | | SIWUA | A12 |
| 32 | ACBUS5 | ** | CLKOUT | ** | ** | ** | GPIOH5 | | ** | A13 |
| 33 | ACBUS6 | ** | OE# | ** | ** | ** | GPIOH6 | | ** | A14 |
| 34 | ACBUS7 | ** | ** | ** | ** | ** | GPIOH7 | | ** | A15 |
| | | **Channel B** | | | | | | | | |
| 38 | BDBUS0 | TXD | | D0 | D0 | D0 | TCK/SK | FSDI | D0 | CS# |
| 39 | BDBUS1 | RXD | | D1 | D1 | D1 | TDI/DO | FSCLK | D1 | ALE |
| 40 | BDBUS2 | RTS# | | D2 | D2 | D2 | TDO/DI | FSDO | D2 | RD# |
| 41 | BDBUS3 | CTS# | | D3 | D3 | D3 | TMS/CS | FSCTS | D3 | WR# |
| 43 | BDBUS4 | DTR# | | D4 | D4 | D4 | GPIOL0 | | D4 | IORDY |
| 44 | BDBUS5 | DSR# | | D5 | D5 | D5 | GPIOL1 | | D5 | CLKOUT |
| 45 | BDBUS6 | DCD# | | D6 | D6 | D6 | GPIOL2 | | D6 | I/O0 |
| 46 | BDBUS7 | RI# | | D7 | D7 | D7 | GPIOL3 | | D7 | I/O1 |
| 48 | BCBUS0 | TXDEN | | RXF# | ** | ** | GPIOH0 | | CS# | ** |
| 52 | BCBUS1 | ** | | TXE# | WRSTB | WRSTB | GPIOH1 | | A0 | ** |
| 53 | BCBUS2 | ** | | RD# | RDSTB# | RDSTB# | GPIOH2 | | RD# | ** |
| 54 | BCBUS3 | RXLED# | | WR# | ** | ** | GPIOH3 | | WR# | ** |
| 55 | BCBUS4 | TXLED# | | SIWUB | SIWUB | SIWUB | GPIOH4 | SIWUB | SIWUB | ** |
| 57 | BCBUS5 | ** | | ** | ** | ** | GPIOH5 | | ** | ** |
| 58 | BCBUS6 | ** | | ** | ** | ** | GPIOH6 | | ** | ** |
| 59 | BCBUS7 | PWRSAV | PWRSAV | PWRSAV | PWRSAV | PWRSAV | GPIOH7 | PWRSAV | PWRS | PWRSAV |
| 60 | PWREN | PWREN | PWREN | PWREN | PWREN | PWREN | PWREN | PWREN | PWRE | PWREN# |
| 36 | SUSPEND# | SUSPEND# | SUSPEND# | SUSPEND# | SUSPEND# | SUSPEND# | SUSPEND# | SUSPEND# | SUSPEND# | SUSPEND# |
| | | **Configuration memory interface** | | | | | | | | |
| 63 | EECS | | | | | | | | | |
| 62 | EECLK | | | | | | | | | |
| 61 | EEDATA | | | | | | | | | |

- **FT4232**, este dispositivo tiene dos canales para utilizar el JTAG. Además, tiene otros dos canales que no tienen JTAG, por lo que puede llegar a tener 3 UARTs y un JTAG con este chip.

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

| FT4232HL and FT4232HQ (64-pin) | | | | | |
|---|---|---|---|---|---|
| Pins | | Pin functions (depend on configuration) | | | |
| Pin # | Pin Name | ASYNC Serial (RS232) | ASYNC Bit-bang | SYNC Bit-bang | MPSSE |
| Channel A | | | | | |
| 16 | ADBUS0 | TXD | D0 | D0 | TCK/SK |
| 17 | ADBUS1 | RXD | D1 | D1 | TDI/DO |
| 18 | ADBUS2 | RTS# | D2 | D2 | TDO/DI |
| 19 | ADBUS3 | CTS# | D3 | D3 | TMS/CS |
| 21 | ADBUS4 | DTR# | D4 | D4 | GPIOL0 |
| 22 | ADBUS5 | DSR# | D5 | D5 | GPIOL1 |
| 23 | ADBUS6 | DCD# | D6 | D6 | GPIOL2 |
| 24 | ADBUS7 | RI#/ TXDEN* | D7 | D7 | GPIOL3 |
| Channel B | | | | | |
| 26 | BDBUS0 | TXD | D0 | D0 | TCK/SK |
| 27 | BDBUS1 | RXD | D1 | D1 | TDI/DO |
| 28 | BDBUS2 | RTS# | D2 | D2 | TDO/DI |
| 29 | BDBUS3 | CTS# | D3 | D3 | TMS/CS |
| 30 | BDBUS4 | DTR# | D4 | D4 | GPIOL0 |
| 32 | BDBUS5 | DSR# | D5 | D5 | GPIOL1 |
| 33 | BDBUS6 | DCD# | D6 | D6 | GPIOL2 |
| 34 | BDBUS7 | RI#/ TXDEN* | D7 | D7 | GPIOL3 |
| Channel C | | | | | |
| 38 | CDBUS0 | TXD | D0 | D0 | RS232 or Bit-Bang interface |
| 39 | CDBUS1 | RXD | D1 | D1 | RS232 or Bit-Bang interface |
| 40 | CDBUS2 | RTS# | D2 | D2 | RS232 or Bit-Bang interface |
| 41 | CDBUS3 | CTS# | D3 | D3 | RS232 or Bit-Bang interface |
| 43 | CDBUS4 | DTR# | D4 | D4 | RS232 or Bit-Bang interface |
| 44 | CDBUS5 | DSR# | D5 | D5 | RS232 or Bit-Bang interface |
| 45 | CDBUS6 | DCD# | D6 | D6 | RS232 or Bit-Bang interface |
| 46 | CDBUS7 | RI#/ TXDEN* | D7 | D7 | RS232 or Bit-Bang interface |
| Channel D | | | | | |
| 48 | DDBUS0 | TXD | D0 | D0 | RS232 or Bit-Bang interface |
| 52 | DDBUS1 | RXD | D1 | D1 | RS232 or Bit-Bang interface |
| 53 | DDBUS2 | RTS# | D2 | D2 | RS232 or Bit-Bang interface |
| 54 | DDBUS3 | CTS# | D3 | D3 | RS232 or Bit-Bang interface |
| 55 | DDBUS4 | DTR# | D4 | D4 | RS232 or Bit-Bang interface |
| 57 | DDBUS5 | DSR# | D5 | D5 | RS232 or Bit-Bang interface |
| 58 | DDBUS6 | DCD# | D6 | D6 | RS232 or Bit-Bang interface |
| 59 | DDBUS7 | RI#/ TXDEN* | D7 | D7 | RS232 or Bit-Bang interface |
| 60 | PWREN# | PWREN# | PWREN# | PWREN# | PWREN# |
| 36 | SUSPEND# | SUSPEND# | SUSPEND# | SUSPEND# | SUSPEND# |
| | | Configuration memory interface | | | |

Entonces, la conexión con un FT2232 y una FPGA se tiene que hacer de la siguiente forma.

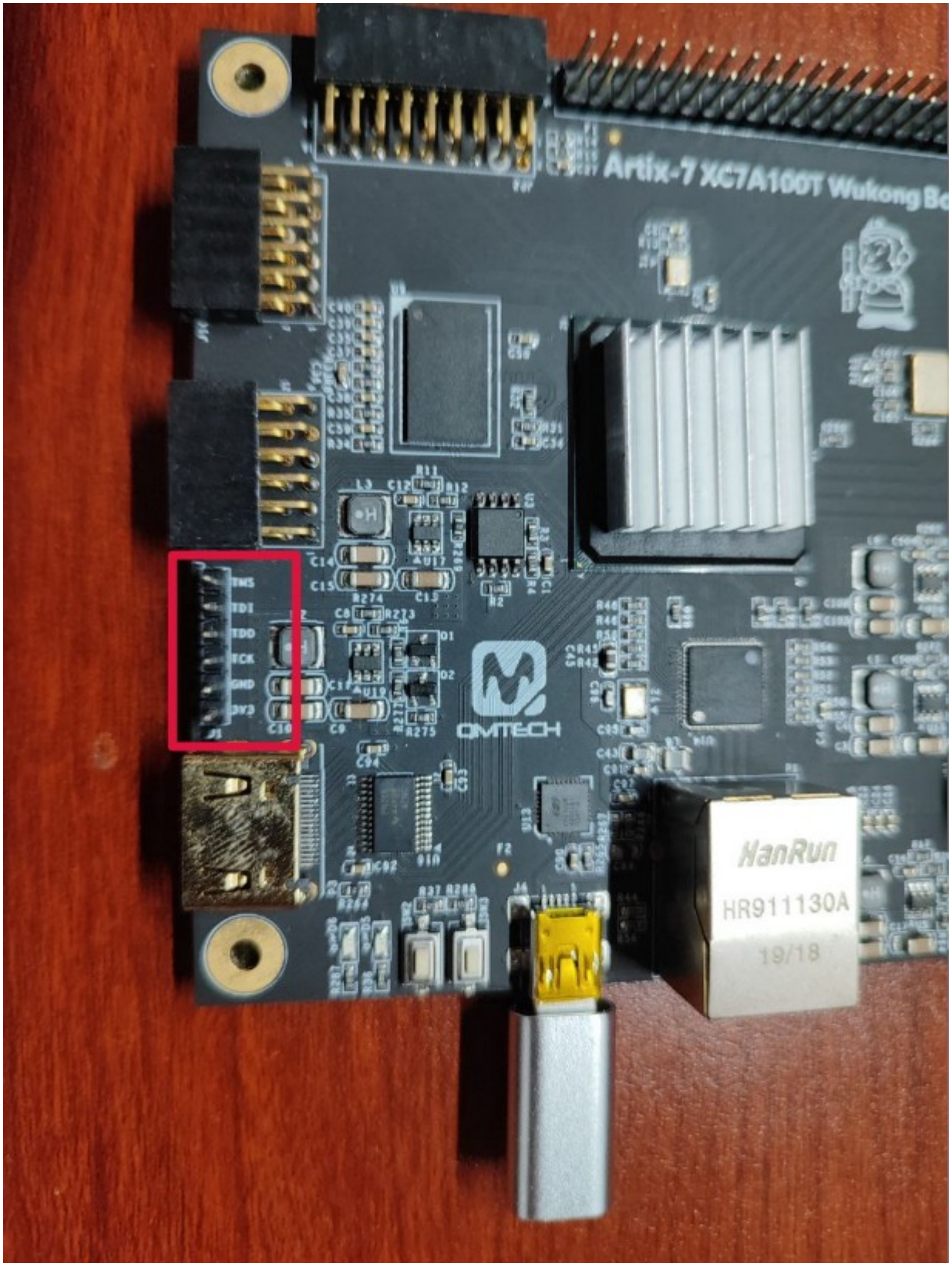



**NOTA**: para los otros dispositivos también se haría así.

## Ejemplo FT232H

Para este ejemplo vamos a hacer una demostración en real. Tenemos una FPGA de la serie 7 de Xilinx y queremos grabar un bitstream dentro de la FPGA con un dispositivo de FTDI.

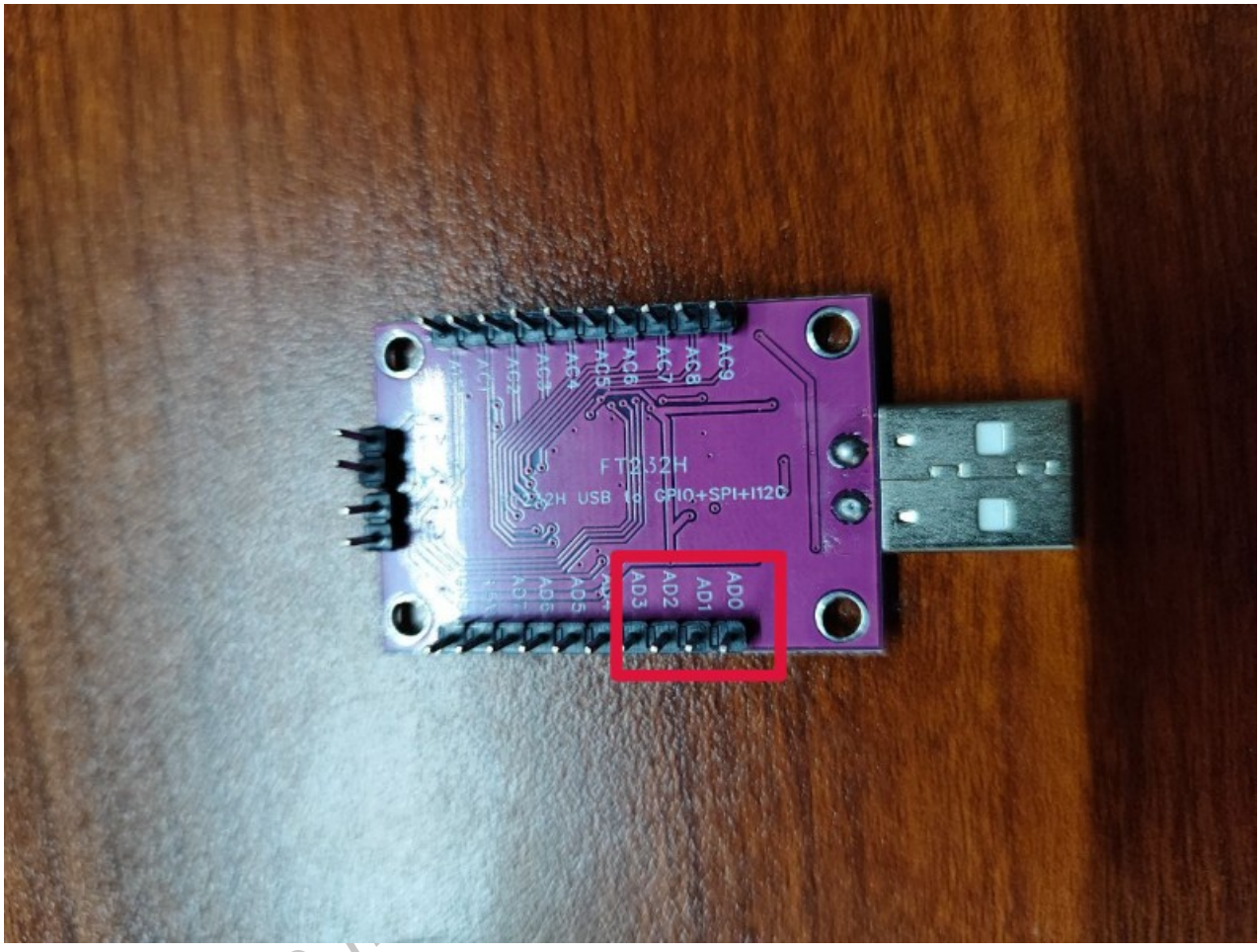
https://soceame.wordpress.com/2025/02/01/como-utilizar-un-chip-de-ftdi-como-cable-depurador-de-xilinx/




El dispositivo que se va a utilizar es un FT232H. Este dispositivo tiene los pines de programación del JTAG en los pines:

- Pin 13 (ADBUS0): AD0 -> TCK
- Pin 14 (ADBUS1): AD1 -> TDI
- Pin 15 (ADBUS2): AD2 -> TDO
- Pin 16 (ADBUS3): AD3 -> TMS

También sería necesario conectar una masa del FTDI.




Ahora lo que vamos a hacer es grabar el FT232H. Para ello tenemos que utilizar el binario *flash_alinx_ft232h.conf*. Este binario pertenece a los cables depuradores que distribuye Alinx para sus placas que incluyen desde la serie 7 a la serie UltraScale de las FPGAs/SoCs de Xilinx. También tiene que estar junto a un .bin para poder grabar el dispositivo.


Ahora lo único que tenemos que hacer es conectar el dispositivo al ordenador. Ejecutando el comando *sudo dmesg* podemos comprobarlo.

```
usb 1-3.1.3: new high-speed USB device number 14 using xhci_hcd
usb 1-3.1.3: New USB device found, idVendor=0403, idProduct=6014, bcdDevice= 9.00
usb 1-3.1.3: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-3.1.3: Product: Digilent USB Device
usb 1-3.1.3: Manufacturer: Digilent
usb 1-3.1.3: SerialNumber: 201706300081
ftdi_sio 1-3.1.3:1.0: FTDI USB Serial Device converter detected
usb 1-3.1.3: Detected FT232H
usb 1-3.1.3: FTDI USB Serial Device converter now attached to ttyUSB0
```

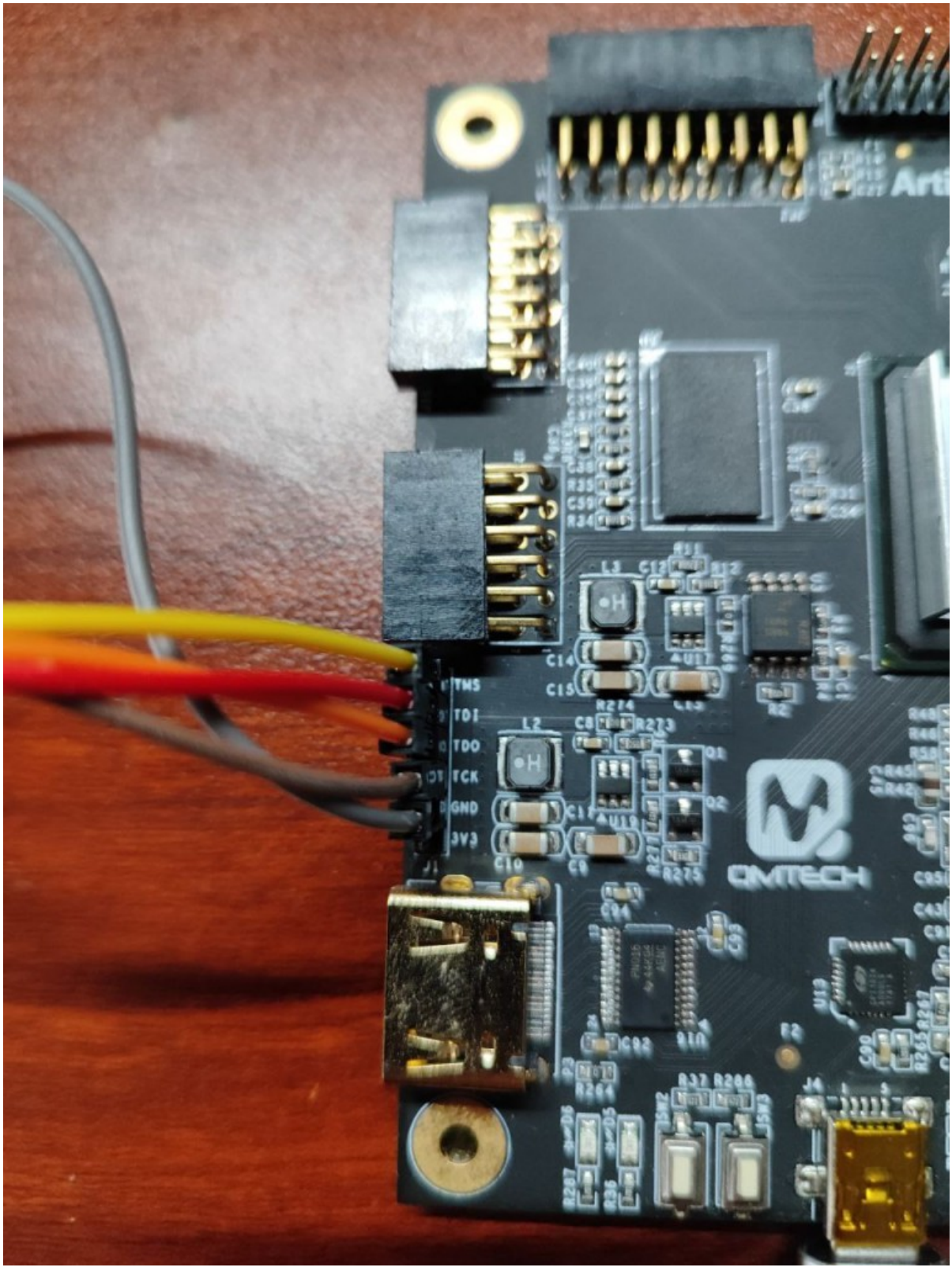
Ahora ejecutamos el comando de grabación de la Flash del dispositivo
*( sudo ftdi_eeprom –flash-eeprom flash_alinx_ft232h.conf )*.

```
FTDI eeprom generator v0.17
(c) Intra2net AG and the libftdi developers <opensource@intra2net.com>
FTDI read eeprom: 0
EEPROM size: 256
FIXME: Build FT232H specific EEPROM settings
Used eeprom space: 244 bytes
Flashing raw eeprom from file jtag-alinx-ft232h.bin (256 bytes)
FTDI write eeprom: 0
FTDI close: 0
```

Una vez grabado el FTDI podemos conectarlo a la FPGA.

Para la conexión con el JTAG para la FPGA se ha hecho de la siguiente forma:

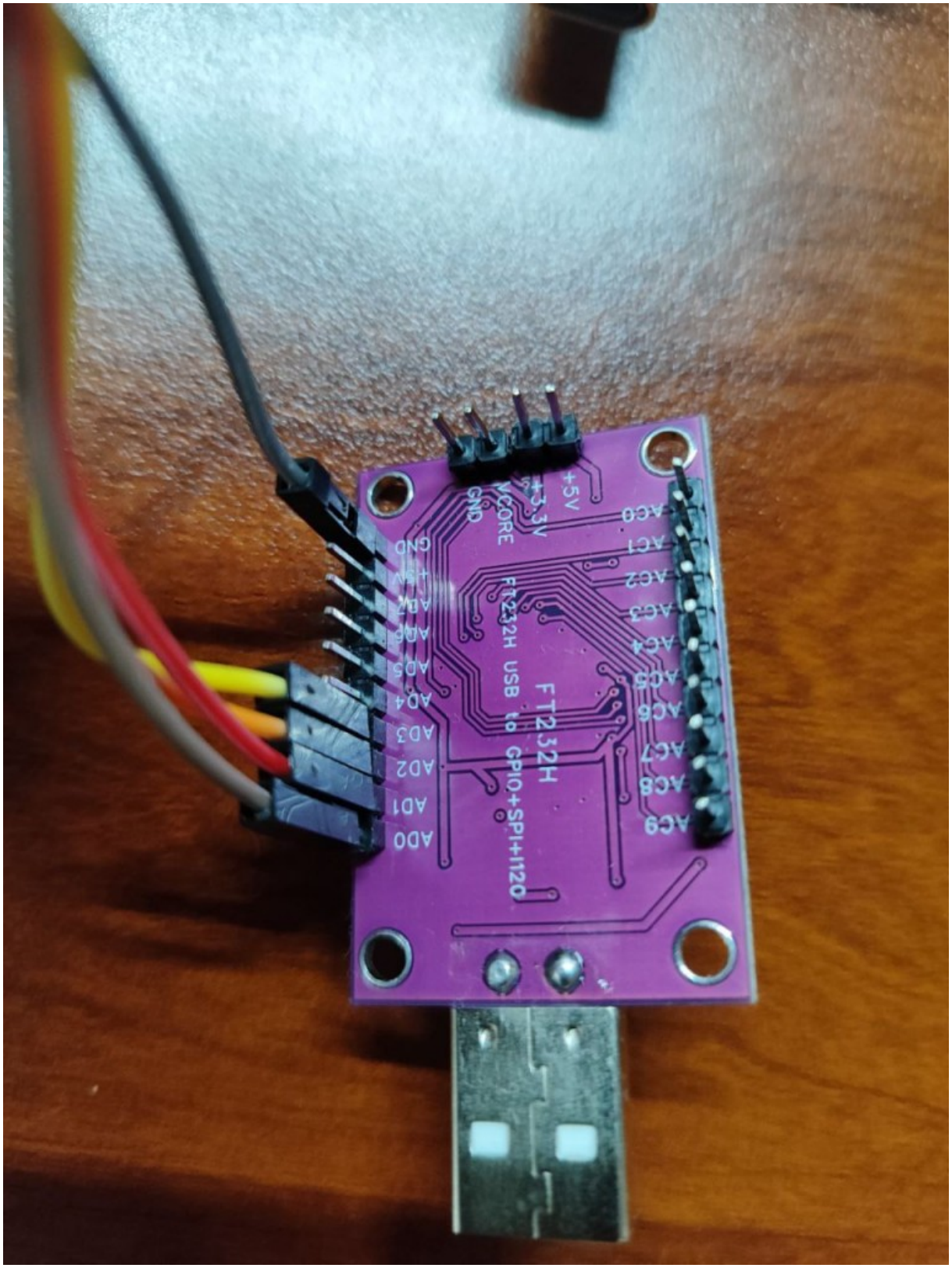
- TMS: cable amarillo
- TDI: cable rojo
- TDO: cable naranja
- TCK: cable marrón
- GND: cable gris



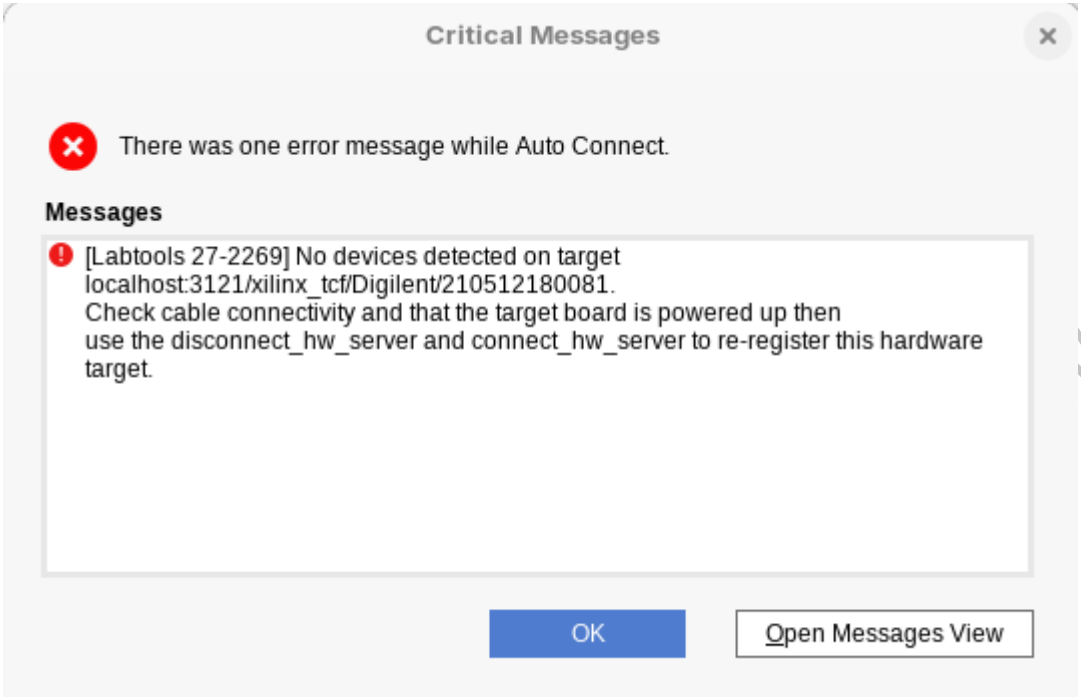

Y desde el FT232H:

- AD0 (TCK): cable marrón
- AD1 (TDI): cable rojo

- AD2 (TDO): cable naranja
- AD3 (TMS): cable amarillo
- GND: cable gris



Si hemos conectado mal los cables nos saldrá un mensaje como el siguiente en Vivado. Esto seguramente sea debido a que tenemos cambiados TDI y TDO en la conexión.




Si los cambiamos podemos ver como Vivado ahora reconoce la placa.


Y ya solo quedaría grabar un bitstream en la placa.

# Ejemplo FT2232

Ahora vamos a replicar el ejemplo anterior para un FT2232. Recordando que este chip tiene dos bancos para programar por JTAG.

**Banco A:**

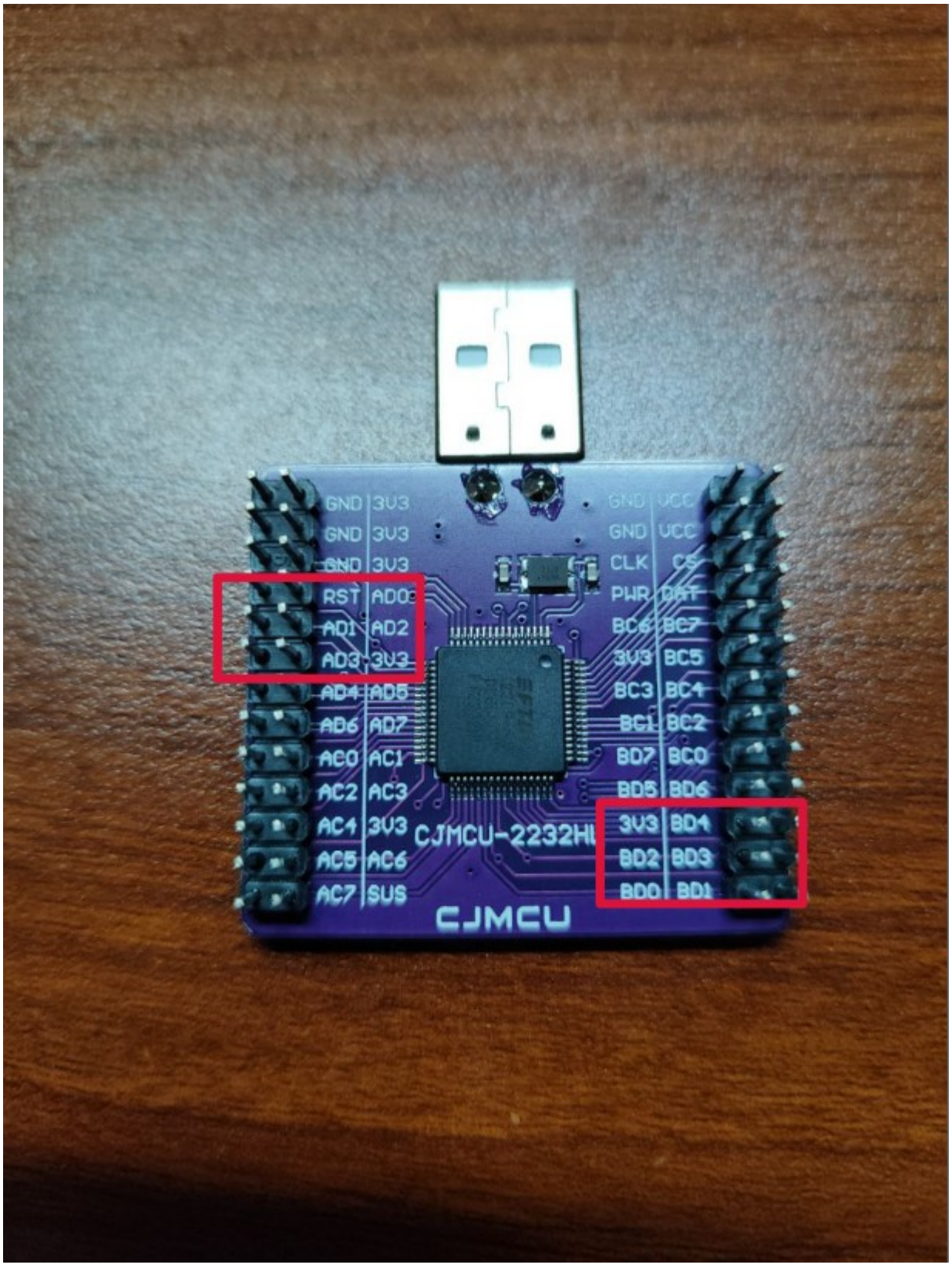
- Pin 16 (ADBUS0): AD0 -> TCK
- Pin 17 (ADBUS1): AD1 -> TDI
- Pin 18 (ADBUS2): AD2 -> TDO
- Pin 19 (ADBUS3): AD3 -> TMS

**Banco B:**

- Pin 38 (BDBUS0): BD0 -> TCK
- Pin 39 (BDBUS1): BD1 -> TDI

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

- Pin 40 (BDBUS2): BD2 -> TDO
- Pin 41 (BDBUS3): BD3 -> TMS



Lo siguiente es programar el FT2232, para ello se utiliza el fichero *flash_digilent_smt1.conf*, con su respectivo fichero .bin.


Ahora se graba de forma correcta le FT2232.

```
FTDI eeprom generator v0.17
(c) Intra2net AG and the libftdi developers <opensource@intra2net.com>
FTDI read eeprom: 0
EEPROM size: 256
Used eeprom space: 238 bytes
Flashing raw eeprom from file jtag-smt1.bin (256 bytes)
FTDI write eeprom: 0
FTDI close: 0
```

Ahora solo tenemos que conectar la FPGA con el FT2232. Para ello lo vamos a hacerlo solo por el canal A (con el canal B no he conseguido que Vivado lo reconozca, esto puede ser debido a que el dispositivo que utilizo tiene problemas o debido a que el binario que he grabado solo deja activado el canal A para el JTAG, sea como fuere, solo se hace la prueba con el canal A).

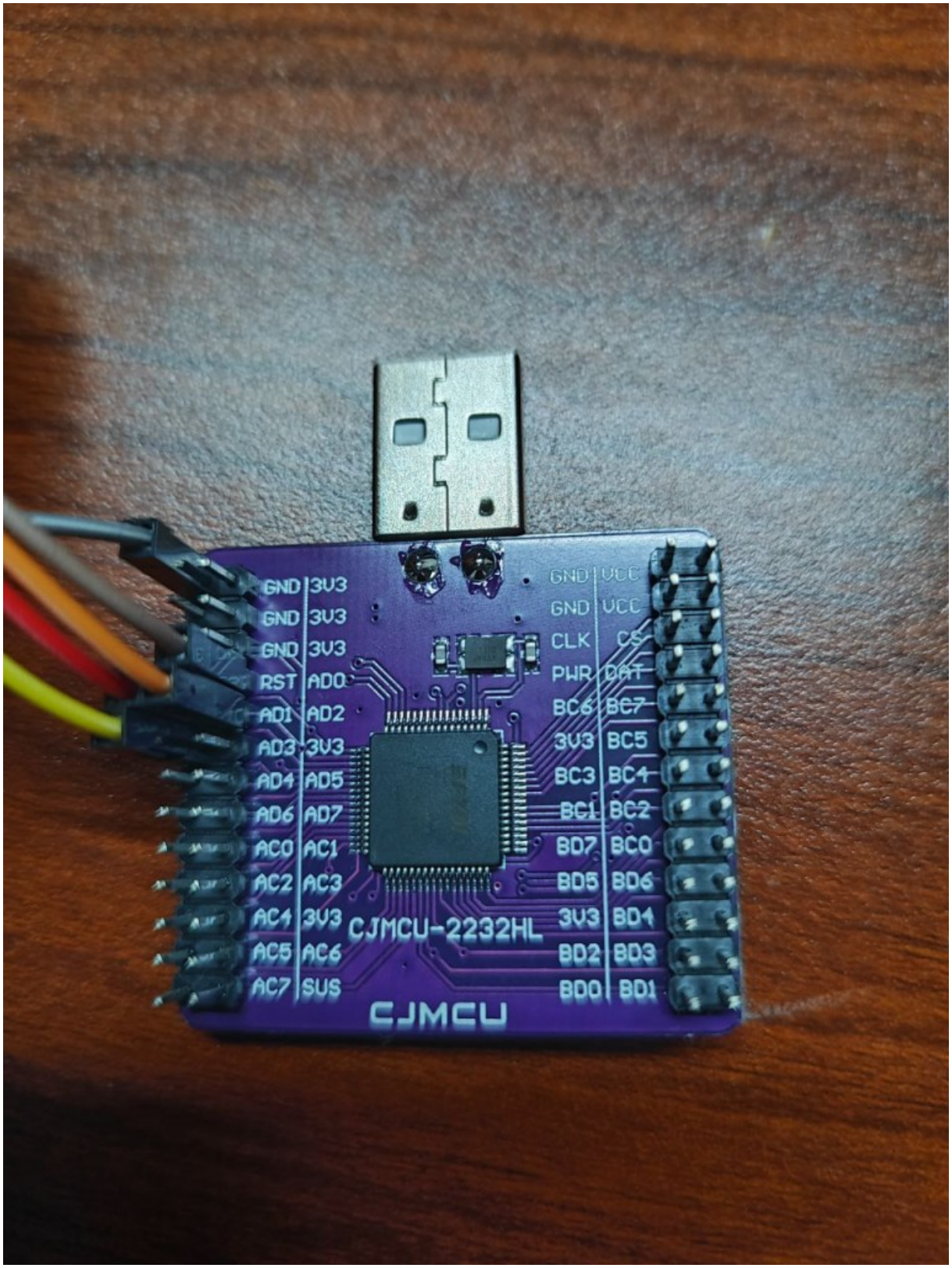
- **Canal A**

Para esto seguimos el mismo esquema de colores de cables del ejemplo anterior para la FPGA.

El FT2232 lo conectamos de la siguiente forma.

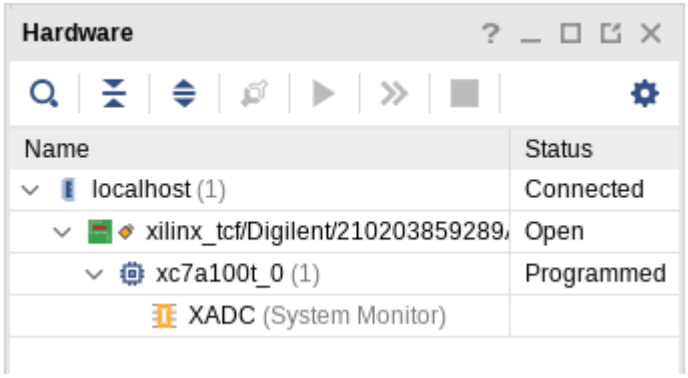
- AD0 (TCK): cable marrón
- AD1 (TDI): cable rojo
- AD2 (TDO): cable naranja
- AD3 (TMS): cable amarillo
- GND: cable gris

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




Para comprobar que está conectado de forma correcta solo es necesario ver si Vivado detecta la FPGA. Ahora solo hace falta grabar la FPGA.

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




# Nota final

En base a todas las pruebas anteriores recomiendo que si quieres utilizar un chip de FTDI, que revises el funcionamiento con un dispositivo como los anteriores, y que si quieres utilizar uno que utilices un FT232.

El FTDI como se ha comentado anteriormente también puede incorporar diferentes protocolos como varias UARTs, etc. Para eso se necesita investigar un poco más para que estén habilitadas varias interfaces del FTDI, para ello se recomienda ver el enlace de Referencia.

Por último, también puede surgir la duda si con este chip se puede depurar una FPGA de Xilinx. Entonces, se puede ver en la siguiente imagen como este chip también permite depurar la FPGA como si fuese una placa que lleva este chip de serie.


*(al final de la siguiente entrada tienes como conectar los leds indicativos de la UART para los chips FT2232 y FT232, el FT4232 no tiene esos leds)*

https://soceame.wordpress.com/2025/02/16/como-configurar-un-ftdi-para-que-funcione-solo-mediante-uart/

# Ampliaciones

https://soceame.wordpress.com/2025/02/02/como-anadir-la-uart-a-un-ft2232-con-jtag/

La siguiente entrada se puede utilizar para cualquier familia de FPGAs además de la de Xilinx.

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

# Referencia

- Pasos para la descarga/carga: https://gist.github.com/rikka0w0/24b58b54473227502fa0334bbe75c3c1