

# USB port JTAG programmer

Author : Andras Tantos



## Introduction

This little interface module allows the programming and debugging of many JTAG enabled devices using a PC with a USB port. Traditional JTAG programmer modules, like the [CPLD-based programmer](#) presented on this site attach to the parallel port of the PC. While this method allows for easy connectivity, it has many drawbacks. Many current PCs, especially laptops don't even have a parallel printer port any more. In most cases programming speeds are rather slow as well, which is problematic in interactive environments and with large devices.

This module solves both issues. It attaches to the USB port, which is available on almost every PC in use today. It works well with both laptop and desktop computer. The USB interface chip used on the board allows for JTAG data rates up to 6Mbps, which is significantly higher than most parallel port based solutions can offer.

The module is compatible with the [OpenOCD](#) project, which enables great integration with tools such as [GDB](#) and [Eclipse CDT](#).

## Features

- Standard USB interface
- Integrated 3.3V power supply, running from the USB port
- Can provide (3.3V) power for the target device
- Supports programming port voltages down to 1.2V
- Supported by GDB and Eclipse through the OpenOCD project

## License

This document and all the accompanying design documentation (for example schematic and PCB files) are covered by the H-Storm Non-Commercial License (HSNCL).

## H-Storm Non-Commercial License (HSNCL)

Copyright 2004-2007 Andras Tantos and Modular Circuits. All rights reserved.

Redistribution and use in source or binary forms, or incorporated into a physical (hardware) product, with or without modification, are permitted **for non-commercial use only**, provided that the following conditions are met:

- **The redistribution doesn't result in financial gain.**
- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in any other form must contain in printed or electronical format the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- All advertising materials mentioning features or use of this technology must display the following acknowledgment:  
This product includes H-Storm technology developed by Andras Tantos and Modular Circuits.
- Neither the name of Andras Tantos or Modular Circuits may be used to endorse or promote products derived from or using this technology without specific prior written permission.

ALL THE INFORMATION, TECHNOLOGY, AND SOFTWARE IS PROVIDED BY THE AUTHORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANDRAS TANTOS, MODULAR CIRCUITS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE OR TECHNOLOGY, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## Design description

### USB interface

The heart of the circuit is the [FT2232](#) dual-port USB-to-serial bridge from FTDI. While this chip was originally designed to provide standard RS-232 ports for PCs with only USB interfaces, it also has many additional features. One of these features is the support for the JTAG serial interface. This operating mode is used in this design to provide a USB to JTAG bridge.

### JTAG interface

The outputs of the FT2232 chip are buffered by level-shifter gates to provide wide-range logic-level support. These devices, the [SN74AVC4T245](#) and the [SN74AVC2T45](#) can manage logical level-shifting between two power domains in the 1.2V to 3.3V range. One of the power domains is fixed 3.3V, and is connected to the power supply running the FT2232. The other domain can run from the same power supply or potentially run from an external power supply provided by the target board. In this latter operating mode arbitrary logic-levels can be used from 1.2V to 3.3V. A jumper can select between the internal or external power supply for the second power domain.

The outputs of the level-shifters can be disabled through software control, using some GPIO pins on the FT2232 device. The pins used make the device compatible with the [JTagKey](#) from Amontec.

### Power supply

While the FT2232 chip contains an internal 3.3V regulator to allow for bus-powered operation, an external LDO regulator is used. This device, the IRU1207-33, can easily provide a couple of hundred mA-s, much more current than the built-in regulator of the USB bridge chip can deal with. This additional current-delivery capability allows the circuit to power the target circuit as well in many situations.

### Other features

Two feedback LEDs are provided on the board. One shows the presence of the 3.3V power supply of the FTDI chip. Since this power is generated from the USB power, this LED pretty much shows if the device is connected to the PC.

Another LED shows the presence of the target power supply. If this supply is provided by the programmer, than this LED will always be turned on. However if the target provides its own power and powers the second power domain of the level shifters, than the LED monitors the presence of that power and can be used to visually verify that a target device is hooked up to the programmer. The LED is driven by a low turn-on level MOSFET, so it can accurately detect the presence of power supplies down to 1.2V. The power detection line is also hooked up to a GPIO pin of the FT2232 device so the presence of the target power can be verified programmatically.

### Design files

[Schematic and PCB in PDF format \(HSNCL\)](#)