

[English](#) • [Deutsch](#) • [Español](#) • [Français](#) • [Italiano](#) • [???](#) • [Polski](#) • [Português](#) • [??????](#) • [Svenska](#) • [???\(????\)?](#) • [???\(??\)?](#) •

Contents

- [1 Introduction](#)
- [2 Obtaining a Jtag Cable](#)
- [3 Jtag on a Laptop Computer](#)
- [4 Understanding Jtag](#)
- [5 Setting up the Jtag Program](#)
- [6 Using Jtag](#)
- [7 Switches](#)
 - ◆ [7.1 Jtag Commands](#)
- [8 Troubleshooting](#)
- [9 Tricks](#)
- [10 Support TJtag!](#)
- [11 Useful Links](#)

Introduction

Joint Test Action Group (JTAG) was originally known for testing printed circuit boards. This was later standardized as IEEE 1149.1 Standard Test Access Port and Boundary-Scan Architecture. (source [wikipedia:Joint Test Action Group](#))

Today JTAG is commonly known for a **JTAG debug port** in many embedded systems - and (home) routers, wireless access points are examples of embedded systems.

JTAG should only be used as a last resort option, TFTP and Serial should be used first. Also, do not even try JTAG on Atheros it is not supported.

Obtaining a Jtag Cable

A jtag cable can be bought off ebay, or made very inexpensively. Here is a picture of how to make one:[\[1\]](#)

Here is additional information:

- [JTAG-Adapter](#)

Here is a great buffered adapter at a reasonable price:

- [DIYGadget Adapter](#)

JTAG

There is no router requiring a buffered jtag cable but it makes it easier with one since cable length is not so critical and you get less interference. Often this means that the jtag will work without many failures that can come from interference and noise.

A good all around debricking device is a TUMPA (~\$30 US): [TIAO USB Multi-Protocol Adapter \(JTAG, SPI, I2C, Serial\)](#)

Jtag on a Laptop Computer

Using a TUMPA may be the most accessible way to use JTAG. See above for more info.

Laptops don't normally have parallel ports anymore, and if your laptop doesn't you would be hooped. USB Jtag is expensive, and doesn't appear to work consistently well. The best option is to get a ExpressCard Parallel port adapter. Further information is in this thread:

- <http://www.dd-wrt.com/phpBB2/viewtopic.php?t=61256>

Understanding Jtag

Jtag is a program for fixing your router if it is in an otherwise unrecoverable state. Jtag is done with a cable hooked from a computer 25 pin printer port ([USB](#) might also be available) to an electrical connection on your router called a **jtag port**. There are sometimes two similar ports on a router; one is the jtag port and the other is a [serial port](#). These ports do not usually have the pins there to connect to, but are just holes in your router motherboard. You often need to solder a pin connector to your motherboard. This pin connector is called a header.

In order to understand jtag, you need to understand the three parts of the program that runs inside your router (known as the router's [firmware](#)). The firmware is composed of a [bootloader](#), (that starts up the router's operating system), the [NVRAM](#), (where information particular to your router are stored, like it's IP address and your ssid name) and the kernel which is the program that your router uses.

These three parts together are known as the WHOLEFLASH. They are written to, and stored on your routers RAM chip. The bootloader protected and comes with the router. It normally cannot be accessed without a jtag cable or using some dangerous commands.

The bootloader on a DD-wrt is a linux bootloader, known as a [CFE](#). Linksys also used a VXworks bootloader on some routers that has to be replaced with a CFE linux bootloader using a VXKiller program. So, when people talk about the CFE of the router, they are talking about the bootloader. Every router has it's own particular CFE. It has the MAC addresses embedded in it for your router, so each one is a little different. That is why it is so important not to ever delete this without backing it up. If you delete it, you at least have to find another one that is for your make and model of router. This can be tricky in some cases, so **don't delete the bootloader!**

The nvram is the place where [variable](#) information is stored. This is often where things get mucked up and is often the reason why people need to jtag their router. You can erase the nvram by doing a HARD reset of the router but sometimes the router will not respond. Then it is jtag time. If you delete the nvram, and have a proper CFE and kernel on the router, the nvram will rebuild itself. You don't need to jtag the nvram.

JTAG

The kernel is the firmware. This is what you flash when you flash dd-wrt. DD-wrt IS the kernel. Again, if you have a CFE on the router, you don't need to flash the kernel with Jtag. If the CFE is working, you can flash using TFTP.exe or an equivalent program. Although you CAN flash the kernel using JTAG, it takes a LONG time and flashing using a jtag cable is not completely reliable, so you can end up with problems. You should not need to do this.

So if you have followed the bouncing ball, you should now understand that you should use JTAG primarily for two things:

1. Replacing a CFE
2. Erasing the NVRAM or kernel.

More info on Jtag can be found here [\[2\]](#)

With that understanding, we can now turn to the **tjtag** program

Setting up the Jtag Program

NOTE: As of 12/10/2017 this site is offline, it appears to have expired in June of 2017. There are several other Jtag softwares out there:

An archive of the original Jtag program used as the basis of [tjtag original](#)

This is also located on the dd-wrt website here [HairyDairyMaid Debrick Utility](#) - Old, should only be used on WRT54G devices.

[zJtag](#) - Newer, supports a short [list of devices](#)

[UrJTAG](#) - old software, not suggested for dd-wrt recovery.

You will note that there is a version 2.14, and a v.3.0 of tjtag. The 3.0 supports more router chipsets.

You have to, on a Windows system, load giveio.sys. First you have to put it in the c:\windows\system32\drivers\ folder and then you have to load it using the loaddrv.exe program. Make sure you put the full path of the driver in the loaddrv.exe program as well as the file name. (c:\windows\system32\drivers\giveio.sys). Also note the giveio.sys driver needs to be installed only once. Subsequent needs for the driver during additional jtag sessions, or if your computer needs a re-boot, it only needs to be "started" by clicking on the "start" button of the loaddrv.exe driver loader utility.

Note that giveio.sys may NOT be needed for your jtag software. Try running your software first then if it does not work try loading giveio.sys

The giveio.sys included in the original archive is for 32 bit Windows ONLY. Under Windows 7 either install a 32bit version of Windows 7 to load this driver OR there is a 64 bit version of this driver available from an old post on the Internet Archive, here [\[3\]](#)

Latest release is available from Tornado (pm him) after a \$5 donation towards his development. See the link about supporting Jtag, below.

JTAG

Here are the steps:

1. Start your computer and unarchive the contents of 2.14 to your C:
2. Put giveio.sys in the proper directory: c:\windows\system32\drivers\
 - c:\windows\system32\drivers\giveio.sys
4. Then hit start.
5. Then hit OK.
6. Remove the power supply from your router.
7. Hook up your jtag cable. Make sure you have pin one on pin one and the cable is not upside down on your router, and that you have the cable is hooked to your 25 pin parallel port
8. Plug your power supply into your router.
9. You might have to set the parallel port communications settings, but I have always found default settings work. If they don't please note that your rig needs to have a real printer port, not a usb to printer port adapter. The printer port should be set for ecp mode and standard io of 0x378.

Using Jtag

DO NOT POWER CYCLE WITH THE JTAG UTILITY RUNNING! If the jtag utility is running, do a control C to stop it. IF YOU TURN THE POWER OFF WHEN THE JTAG IS RUNNING YOU MIGHT DAMAGE THE FLASH CHIP!

Tjtag.exe is run in a command window in windows. Windows XP works well.

You should check to make sure your cable is working with a probeonly command:

```
tjtag -probeonly
```

If you don't get a response that recognizes your chipset, check your soldering carefully with a multimeter.

If you get a response that recognizes your chipset, the next command should always be to backup your CFE first, even if you think it is FUBAR. Better safe then sorry.

This is done with the command:

```
tjtag -backup:cfe
```

Do this twice and make sure the files match.

JTAG

With most bricked routers, ALL you have to do is erase the nvram and the kernel. You do that with these commands:

```
tjtag -erase:nvram
```

```
tjtag -erase:kernel
```

DO NOT erase:nvram on a Belkin F5D7230-4 router. Doing so will erase important values and require you to have to jtag the kernel back on.

Doing that should put you back to a position where you can tftp the firmware back on. Stop and try that. You must disconnect your jtag cable to flash the firmware. Follow the guidelines for flashing by tftp found at note 11 of the peacock thread announcement, at the top of the broadcom forum.

- <http://www.dd-wrt.com/phpBB2/viewtopic.php?t=51486>

DO NOT REPLACE THE CFE unless it is corrupt. A bad flash should NOT normally corrupt the CFE. However, if you have to replace the CFE, you must rename the CFE file CFE.bin, and then use this command

```
tjtag -flash:cfe
```

It is important to know, that if you do need to replace the CFE, an erase of kernel and nvram should be done prior to flashing the CFE.

```
tjtag -erase:kernel  
tjtag -erase:nvram
```

The reason for this is if the kernel and nvram are left intact and only the CFE (bootloader) is replaced, when the bootloader boots the device, it will load the kernel. If a corrupt kernel or a bad nvram variable caused the bootloader damage in the first place, the offending pieces of the program are still present and may cause bootloader damage again as soon as the router is power cycled after the CFE flash. If you HAVE to replace the CFE, as a last resort, erase the wholeflash twice (tjtag -erase:wholeflash) and then flash the cfe, then tftp the firmware. You should never have to write an entire wholeflash.

If you need a CFE for a Broadcom router, you can find most through this link:

- <http://www.dd-wrt.com/phpBB2/viewtopic.php?t=25971>

However, these CFEs will contain generic Mac addresses, so you will likely have to hexedit your Mac address to the generic CFE prior to flashing. Here is a thread that discusses how to do that:
<http://www.dd-wrt.com/phpBB2/viewtopic.php?t=45826>

Switches

Sometimes, in order to get things to flash correctly, you have to use switches like the /noemw or /noreset. You can get a list of these switches by typeing tjtag /?

JTAG

Redhawk0 has reported using these switches for Linksys units:

54G(S) V1-V6 and GL v1.X

```
tjtagv3 -flash/erase:xxx /noemw /nocwd
```

54G V8.X, GS v7.X and all other 5354,4704 processor based units

```
tjtagv3 -flash/erase:xxx /noreset
```

54G-TM

```
tjtagv3 -flash/erase:xxx /noemw (Note: Only /noemw is required)
```

Redhawk has also stated:

the command line is dependent on the type processor you have.

5352 and earlier.

```
tjtag -erase:kernel /noemw /nocwd  
tjtag -erase:nvram /noemw /nocwd
```

if it is 5354 and later

```
tjtag -erase:kernel /noreset  
tjtag -erase:nvram /noreset
```

Jtag Commands

EJTAG Debrick Utility v3.0.1 Tornado-MOD

ABOUT: This program reads/writes flash memory on the WRT54G/GS and compatible routers via EJTAG using either DMA Access routines or PrAcc routines (slower/more compatible). Processor chips supported in this version include the following chips:

Supported Chips

```
Broadcom BCM4702 Rev 1 CPU  
Broadcom BCM4704 KPBG Rev 9 CPU  
Broadcom BCM4704 Rev 8 CPU  
Broadcom BCM4712 Rev 1 CPU  
Broadcom BCM4712 Rev 2 CPU  
Broadcom BCM4716 Rev 1 CPU  
Broadcom BCM4785 Rev 1 CPU  
Broadcom BCM5350 Rev 1 CPU  
Broadcom BCM5352 Rev 1 CPU  
Broadcom BCM5354 KFBG Rev 1 CPU  
Broadcom BCM5354 KFBG Rev 2 CPU  
Broadcom BCM5354 KFBG Rev 3 CPU  
Broadcom BCM3345 KPB Rev 1 CPU  
Broadcom BCM5365 Rev 1 CPU  
Broadcom BCM5365 Rev 1 CPU  
Broadcom BCM6345 Rev 1 CPU
```

JTAG

```
Broadcom BCM6348 Rev 1 CPU
Broadcom BCM6338 Rev 1 CPU
Broadcom BCM6358 Rev 1 CPU
Broadcom BCM6368 Rev 1 CPU
Broadcom BCM4321 RADIO STOP
Broadcom BCM4321L RADIO STOP
TI AR7WRD TNETD7300GDU Rev 1 CPU
BRECIS MSP2007-CA-A1 CPU
TI TNETV1060GDW CPU
Linkstation 2 with RISC K4C chip
Atheros AR531X/231X CPU
XScale IXP42X 266mhz
XScale IXP42X 400mhz
XScale IXP42X 533mhz
ARM 940T
Marvell Feroceon 88F5181
LX4380
```

```
USAGE: tjtag [parameter] </noreset> </noemw> </nocwd> </nobreak> </noerase>
        </notimestamp> </dma> </nodma>
        <start:XXXXXXXX> </length:XXXXXXXX>
        </silent> </skipdetect> </instrlen:XX> </fc:XX> /bypass /s
```

t5

Required Parameter

```
-backup:cfe
-backup:nvram
-backup:kernel
-backup:wholeflash
-backup:custom
-backup:bsp
-erase:cfe
-erase:nvram
-erase:kernel
-erase:wholeflash
-erase:custom
-erase:bsp
-flash:cfe
-flash:nvram
-flash:kernel
-flash:wholeflash
-flash:custom
-flash:bsp
-probeonly
-probeonly:custom
```

Optional with -backup:, -erase:, -flash: wgrv8bdata, wgrv9bdata, cfe128

Optional Switches

```
/noreset ..... prevent Issuing EJTAG CPU reset
/noemw ..... prevent Enabling Memory Writes
/nocwd ..... prevent Clearing CPU Watchdog Timer
/nobreak ..... prevent Issuing Debug Mode JTAGBRK
/noerase ..... prevent Forced Erase before Flashing
/notimestamp ..... prevent Timestamping of Backups
/dma ..... force use of DMA routines
/nodma ..... force use of PRACC routines (No DMA)
```

JTAG

```
/window:XXXXXXXX ... custom flash window base (in HEX)
/start:XXXXXXXX .... custom start location (in HEX)
/length:XXXXXXXX ... custom length (in HEX)
/silent ..... prevent scrolling display of data
/skipdetect ..... skip auto detection of CPU Chip ID
/instrlen:XX ..... set instruction length manually
/wiggler ..... use wiggler cable
/bypass ..... Unlock Bypass command & disable polling
/st5 ..... Use Speedtouch ST5xx flash routines instead of WRT routines
/reboot..... sets the process and reboots
/swap_endian..... swap endianness during backup - most Atheros based routers
/flash_debug..... flash chip debug messages, show flash MFG and Device ID
```

/fc:XX = Optional (Manual) Flash Chip Selection

```
-----
/fc:01 ..... MX29LV800BTC 512kx16 TopB (1MB)
/fc:02 ..... MX29LV800BTC 512kx16 BotB (1MB)
/fc:03 ..... AMD 29lv160DB 1Mx16 BotB (2MB)
/fc:04 ..... AMD 29lv160DT 1Mx16 TopB (2MB)
/fc:05 ..... EON EN29LV160A 1Mx16 BotB (2MB)
/fc:06 ..... EON EN29LV160A 1Mx16 TopB (2MB)
/fc:07 ..... MBM29LV160B 1Mx16 BotB (2MB)
/fc:08 ..... MBM29LV160T 1Mx16 TopB (2MB)
/fc:09 ..... MX29LV160CB 1Mx16 BotB (2MB)
/fc:10 ..... MX29LV160CT 1Mx16 TopB (2MB)
/fc:11 ..... K8D1716UTC 1Mx16 TopB (2MB)
/fc:12 ..... K8D1716UBC 1Mx16 BotB (2MB)
/fc:13 ..... ST M29W160EB 1Mx16 BotB (2MB)
/fc:14 ..... ST M29W160ET 1Mx16 TopB (2MB)
/fc:15 ..... Macronix MX25L160A (2MB) Serial
/fc:16 ..... Atmel AT45DB161B (2MB) Serial
/fc:17 ..... Atmel AT45DB161B (2MB) Serial
/fc:18 ..... K8D3216UTC 2Mx16 TopB (4MB)
/fc:19 ..... K8D3216UBC 2Mx16 BotB (4MB)
/fc:20 ..... Macronix MX25L1605D (2MB) Serial
/fc:21 ..... Macronix MX25L3205D (4MB) Serial
/fc:22 ..... Macronix MX25L6405D (8MB) Serial
/fc:23 ..... STMicro M25P16 (2MB) Serial
/fc:24 ..... STMicro M25P32 (4MB) Serial
/fc:25 ..... STMicro M25P64 (8MB) Serial
/fc:26 ..... STMicro M25P128 (16MB) Serial
/fc:27 ..... AMD 29lv320MB 2Mx16 BotB (4MB)
/fc:28 ..... AMD 29lv320MT 2Mx16 TopB (4MB)
/fc:29 ..... AMD 29lv320MT 2Mx16 TopB (4MB)
/fc:30 ..... TC58FVB321 2Mx16 BotB (4MB)
/fc:31 ..... TC58FVT321 2Mx16 TopB (4MB)
/fc:32 ..... AT49BV/LV16X 2Mx16 BotB (4MB)
/fc:33 ..... AT49BV/LV16XT 2Mx16 TopB (4MB)
/fc:34 ..... MBM29DL323BE 2Mx16 BotB (4MB)
/fc:35 ..... MBM29DL323TE 2Mx16 TopB (4MB)
/fc:36 ..... AMD 29lv320DB 2Mx16 BotB (4MB)
/fc:37 ..... AMD 29lv320DT 2Mx16 TopB (4MB)
/fc:38 ..... MBM29LV320BE 2Mx16 BotB (4MB)
/fc:39 ..... MBM29LV320TE 2Mx16 TopB (4MB)
/fc:40 ..... MX29LV320B 2Mx16 BotB (4MB)
/fc:41 ..... MX29LV320B 2Mx16 BotB (4MB)
/fc:42 ..... MX29LV320T 2Mx16 TopB (4MB)
/fc:43 ..... MX29LV320T 2Mx16 TopB (4MB)
/fc:44 ..... ST 29w320DB 2Mx16 BotB (4MB)
/fc:45 ..... ST 29w320DT 2Mx16 TopB (4MB)
/fc:46 ..... MX29LV640B 4Mx16 TopB (16MB)
/fc:47 ..... MX29LV640B 4Mx16 BotB (16MB)
```

JTAG

```

/fc:48 ..... W19B(L) 320ST  2Mx16 TopB  (4MB)
/fc:49 ..... W19B(L) 320SB  2Mx16 BotB  (4MB)
/fc:50 ..... W19B(L) 320SB  2Mx16 BotB  (4MB)
/fc:51 ..... M29DW324DT  2Mx16 TopB  (4MB)
/fc:52 ..... M29DW324DB  2Mx16 BotB  (4MB)
/fc:53 ..... TC58FVM6T2A  4Mx16 TopB  (8MB)
/fc:54 ..... TC58FVM6B2A  4Mx16 BopB  (8MB)
/fc:55 ..... K8D6316UTM  4Mx16 TopB  (8MB)
/fc:56 ..... K8D6316UBM  4Mx16 BotB  (8MB)
/fc:57 ..... Intel 28F160B3  1Mx16 BotB  (2MB)
/fc:58 ..... Intel 28F160B3  1Mx16 TopB  (2MB)
/fc:59 ..... Intel 28F160C3  1Mx16 BotB  (2MB)
/fc:60 ..... Intel 28F160C3  1Mx16 TopB  (2MB)
/fc:61 ..... Intel 28F320B3  2Mx16 BotB  (4MB)
/fc:62 ..... Intel 28F320B3  2Mx16 TopB  (4MB)
/fc:63 ..... Intel 28F320C3  2Mx16 BotB  (4MB)
/fc:64 ..... Intel 28F320C3  2Mx16 TopB  (4MB)
/fc:65 ..... Sharp 28F320BJE  2Mx16 BotB  (4MB)
/fc:66 ..... Intel 28F640B3  4Mx16 BotB  (8MB)
/fc:67 ..... Intel 28F640B3  4Mx16 TopB  (8MB)
/fc:68 ..... Intel 28F640C3  4Mx16 BotB  (8MB)
/fc:69 ..... Intel 28F640C3  4Mx16 TopB  (8MB)
/fc:70 ..... Intel 28F160S3/5  1Mx16      (2MB)
/fc:71 ..... Intel 28F320J3  2Mx16      (4MB)
/fc:72 ..... Intel 28F320J5  2Mx16      (4MB)
/fc:73 ..... Intel 28F320S3/5  2Mx16      (4MB)
/fc:74 ..... Intel 28F640J3  4Mx16      (8MB)
/fc:75 ..... Intel 28F640J5  4Mx16      (8MB)
/fc:76 ..... Intel 28F128J3  8Mx16      (16MB)
/fc:77 ..... SST39VF1601  1Mx16 BotB  (2MB)
/fc:78 ..... SST39VF1602  1Mx16 TopB  (2MB)
/fc:79 ..... SST39VF3201  2Mx16 BotB  (4MB)
/fc:80 ..... SST39VF3202  2Mx16 TopB  (4MB)
/fc:81 ..... SST39VF6401  4Mx16 BotB  (8MB)
/fc:82 ..... SST39VF6402  4Mx16 TopB  (8MB)
/fc:83 ..... SST39VF6401B  4Mx16 BotB  (8MB)
/fc:84 ..... SST39VF6402B  4Mx16 TopB  (8MB)
/fc:85 ..... Spansion S29GL032M  BotB  (4MB)
/fc:86 ..... Spansion S29GL032M  TopB  (4MB)
/fc:87 ..... Spansion S29GL064M  BotB  (8MB)
/fc:88 ..... Spansion S29GL064M  TopB  (8MB)
/fc:89 ..... Spansion S29GL128P  U      (16MB)
/fc:90 ..... Spansion S29GL128M  U      (16MB)
/fc:91 ..... Spansion S29GL256P  U      (32MB)
/fc:92 ..... Spansion S29GL512P  U      (64MB)
/fc:93 ..... Spansion S29GL01GP  U      (128MB)
/fc:94 ..... Spansion S25FL016A      (2MB) Serial
/fc:95 ..... Spansion S25FL032A      (4MB) Serial
/fc:96 ..... Spansion S25FL064A      (8MB) Serial
/fc:97 ..... Winbond W19B320AB  BotB  (4MB)
/fc:98 ..... Winbond W19B320AT  TopB  (4MB)
/fc:99 ..... Winbond W25X32      (4MB) Serial
/fc:100 ..... Winbond W25X64      (8MB) Serial
/fc:101 ..... EON EN29LV320  2Mx16 BotB  (4MB)
/fc:102 ..... EON EN29LV320  2Mx16 TopB  (4MB)
/fc:103 ..... EON EN29LV640  4Mx16 TopB  (8MB)
/fc:104 ..... EON EN29LV640  4Mx16 BotB  (8MB)
/fc:105 ..... AT49BV322A  2Mx16 BotB  (4MB)
/fc:106 ..... AT49BV322A(T)  2Mx16 TopB  (4MB)

```

JTAG

- NOTES: 1) If 'flashing' - the source filename must exist as follows:
CFE.BIN, NVRAM.BIN, KERNEL.BIN, WHOLEFLASH.BIN or CUSTOM.BIN
BSP.BIN
- 2) If you have difficulty auto-detecting a particular flash part you can manually specify your exact part using the /fc:XX option.
 - 3) If you have difficulty with the older bcm47xx chips or when no CFE is currently active/operational you may want to try both the /noreset and /nobreak command line options together. Some bcm47xx chips *may* always require both these options to function properly.
 - 4) When using this utility, usually it is best to type the command line out, then plug in the router, and then hit <ENTER> quickly to avoid the CPUs watchdog interfering with the EJTAG operations
 - 5) /bypass - enables Unlock bypass command for some AMD/Spansion type flashes, it also disables polling

```
*****  
* Flashing the KERNEL or WHOLEFLASH will take a very long time using JTAG *  
* via this utility. You are better off flashing the CFE & NVRAM files *  
* & then using the normal TFTP method to flash the KERNEL via ethernet. *  
*****
```

Troubleshooting

1. Bad soldering - One of the most common reasons that your jtag doesn't work is due to bad soldering, especially in making sure the header is soldered in properly. Check your work with a multimeter. Many routers have jtag holes in the pcb filled with solder. Many damage the pcb by trying to clean the holes. Be careful, use lots of flux, and solder wick to remove the solder from the board. Some soldering irons have a pcb tip that will fit right through the holes and can make the job easier.

Here is a post that discusses some soldering techniques [\[4\]](#)

2. Putting the connection on backward - Make sure you have the cable connected to the header properly and not upside down.

3. Interference - Electrical interference can cause a bad flash with tftp.exe. Even having your computer monitor too close can cause bad information and ruin the flash.

4. Cable too long - Similar to electrical interference. You want your cable to be about 6 inches (15,24 cm) in length.

Tricks

1. Sometimes the routers cpu chip gets "stuck". Try using

```
-erase:nvram /nodma
```

a few times followed by the proper command. This will sometimes release the router

2. If you want to run a jtag command continually, use BW's fine script saved as a batch file:

JTAG

```
@echo off
cls
:start
tjtag -backup:wholeflash (or whatever command you want)
goto start
```

This is useful to keep jtag running while you flex the board or just to leave a problem router run overnight to punish it.

Support Tjtag!

If you are reading this page, it is likely because you need HELP! The tjtag program was created by tornado and were it not for him, you would likely be screwed right now. Consider sending him a few dollars as a token of your appreciation. You can do so by clicking on this link:

- [Support Tjtag!](#)

Useful Links

- [JTAG pinouts](#)
- http://www.tiaowiki.comhttp://wiki.dd-wrt.com/w/Debrick_Routers_Using_JTAG_Cable
- [Universal JTAG module](#)
- [Buffered Jtag Adapter](#)