TACTICAL
NETWORK SOLUTIONS

Debugging Embedded Systems With JTAG

Craig Heffner

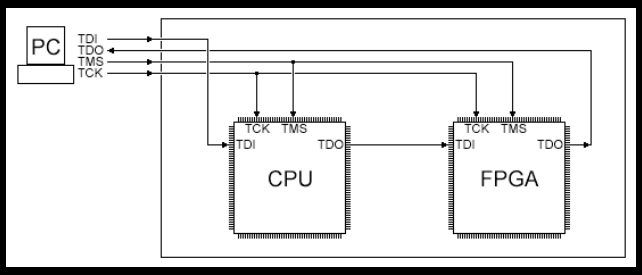
What is JTAG?

- Joint Test Action Group
- Facilitates:
  - Testing chip connectivity
  - Reading / writing memory
  - CPU leve debugging

When to Use JTAG

- Reading / writing memory
- Bootloader / Kernel / RTOS debugging

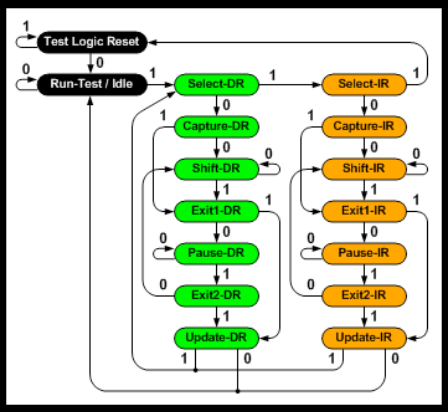
JTAG Bus



JTAG Registers

- Instruction Register
  - Fixed width
  - Control the TAP
- Data Register
  - Variable width
  - Read / write data to the TAP

JTAG State Machine



www.tacnetsol.com

TACTICAL NETWORK SOLUTIONS

JTAG IDCODE

- Put TAP in Shift-IR state
- Clock in the IDCODE instruction
- Put TAP in the Shift-DR state
- Clock out the 32 bit ID code

www.tacnetsol.com

TACTICAL NETWORK SOLUTIONS

Identifying JTAG

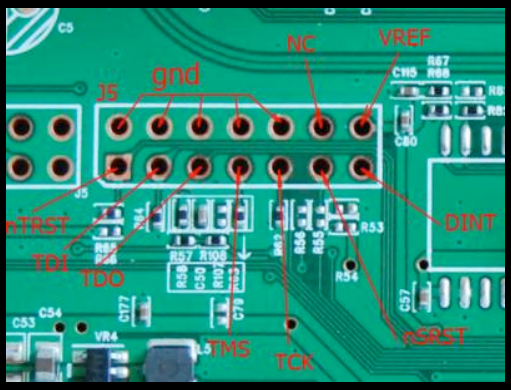
VTref	1	2	Vsupply		
nTRST			GND		
TDI			GND		
TMS			GND		
TCK			GND		
RTCK			GND		
TDO			GND		
nSRST			GND		
DBGRQ			GND		
DBGACK			GND		

TRST_N	i	1	2	p	GND
TDI	i	3	4	p	GND
TDO	o	5	6	p	GND
TMS	i	7	8	p	GND
TCK	i	9	10	p	GND
SRST_N	od	11	12	k	KEY
DINT	i	13	14	p	VREF

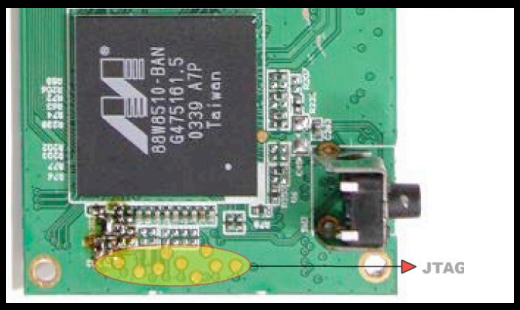
www.tacnetsol.com

TACTICAL NETWORK SOLUTIONS

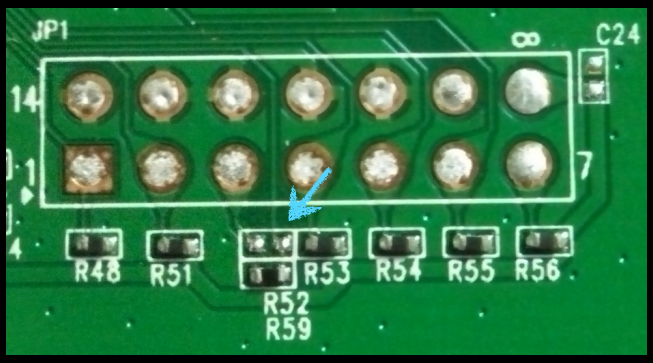
Typical JTAG Connector



JTAG Test Points



Broken JTAG Traces



JTAG Adapters

www.tacnetsol.com

TACTICAL
NETWORK SOLUTIONS

TinCanTools Flyswatter2

www.tacnetsol.com

TACTICAL
NETWORK SOLUTIONS

OpenOCD

- Recognizes many JTAG adapters
- Read/write memory, debugging
- Supports ARM and MIPS

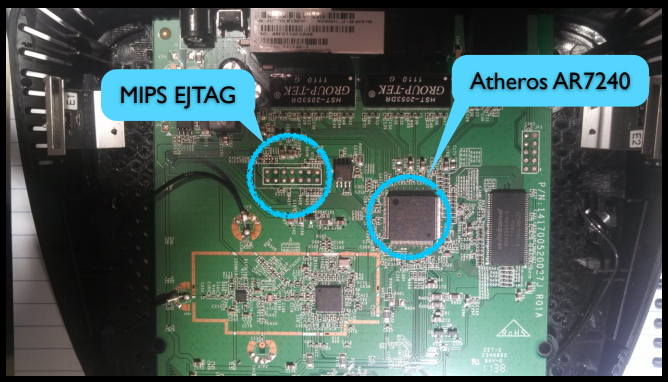
www.tacnetsol.com

TACTICAL
NETWORK SOLUTIONS

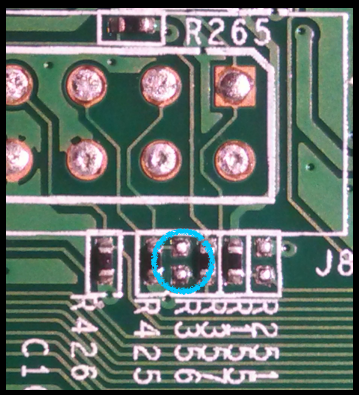
Building OpenOCD

- apt-get install libftdi-dev
- ./configure --enable-ftdi \
  --enable-legacy-ft2232_libftdi

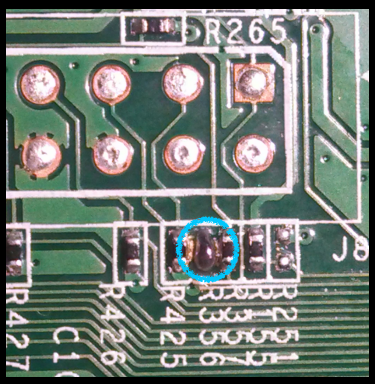
Example Target



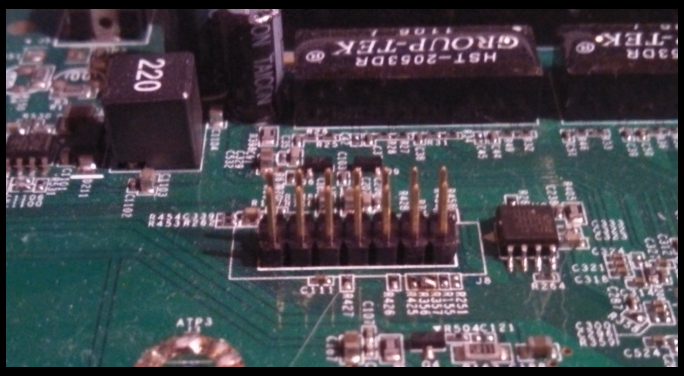
Missing R356



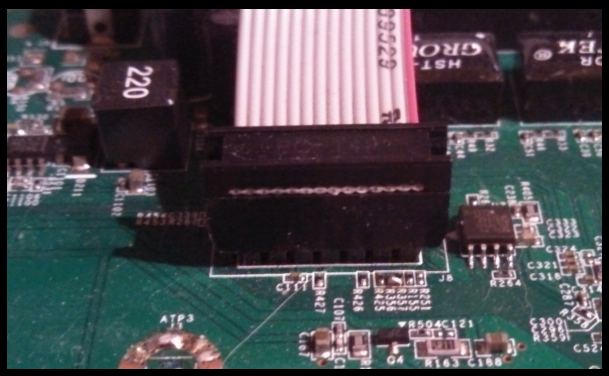
Replacing R356



Adding Headers



Physical Connection



Configuring OpenOCD

- Declare an adapter
- Define a TAP
- Declare a TAP as a target

Declaring an Adapter

interface ft2232

ft2232_device_desc "Flyswatter2"

ft2232_layout "flyswatter2"

ft2232_vid_pid 0x0403 0x6010

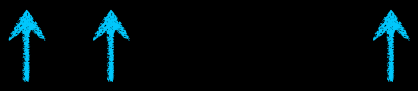
adapter_khz 6000 

How Many TAPs?

```
Warn : There are no enabled taps.  AUTO PROBING MIGHT NOT WORK!!
Warn : AUTO auto0.tap - use "jtag newtap auto0 tap -expected-id 0x00000001 ..."
Warn : AUTO auto0.tap - use "... -irlen 5"
```

Defining a TAP

jtag newtap mips cpu -expected-id 0 -irlen 5



Declaring a Target

target create mips.cpu mips_m4k \
-endian big -chain-position mips.cpu

Enabling TSRT / SRST

reset_config trst_and_srst

Running OpenOCD

$ openocd -f flyswatter2.cfg -f wrt120n.cfg

Controlling OpenOCD

$ telnet localhost 4444

> **halt**

target state: halted

target halted in MIPS32 mode due to debug-request, pc: 0x80008114

System Reset

> **reset init**

JTAG tap: mips.cpu tap/device found: 0x00000001 (mfg: 0x000, part: 0x0000, ver: 0x0)

target state: halted

target halted in MIPS32 mode due to debug-request, pc: 0xbfc00000

Single Stepping

> **step**

target state: halted

target halted in MIPS32 mode due to single-step, pc: 0xbfc00400

Setting Breakpoints

> **bp 0xbfc00404 4 hw**

breakpoint set at 0xbfc00404

> **resume**

target state: halted

target halted in MIPS32 mode due to breakpoint, pc: 0xbfc00404

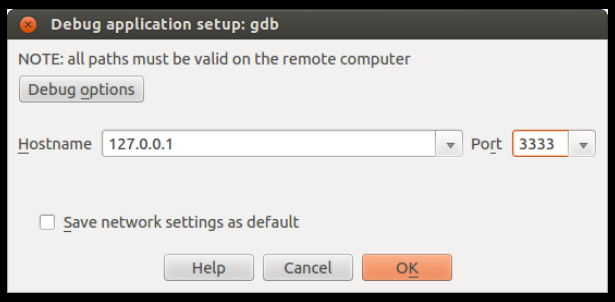
Dumping Memory

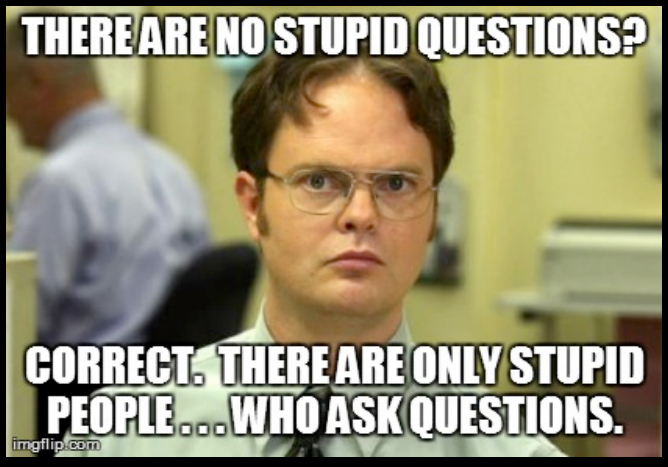
> **mdb 0x81544af0 11**

0x81544af0: 70 61 73 73 77 6f 72 64 31 32 33

"password123"

Connecting IDA / GDB





Resources

- http://github.com/devttys0/wrt120n
- http://www.tincantools.com/wiki/Flyswatter2
- http://openocd.sourceforge.net/
- http://www.edetraining.com/