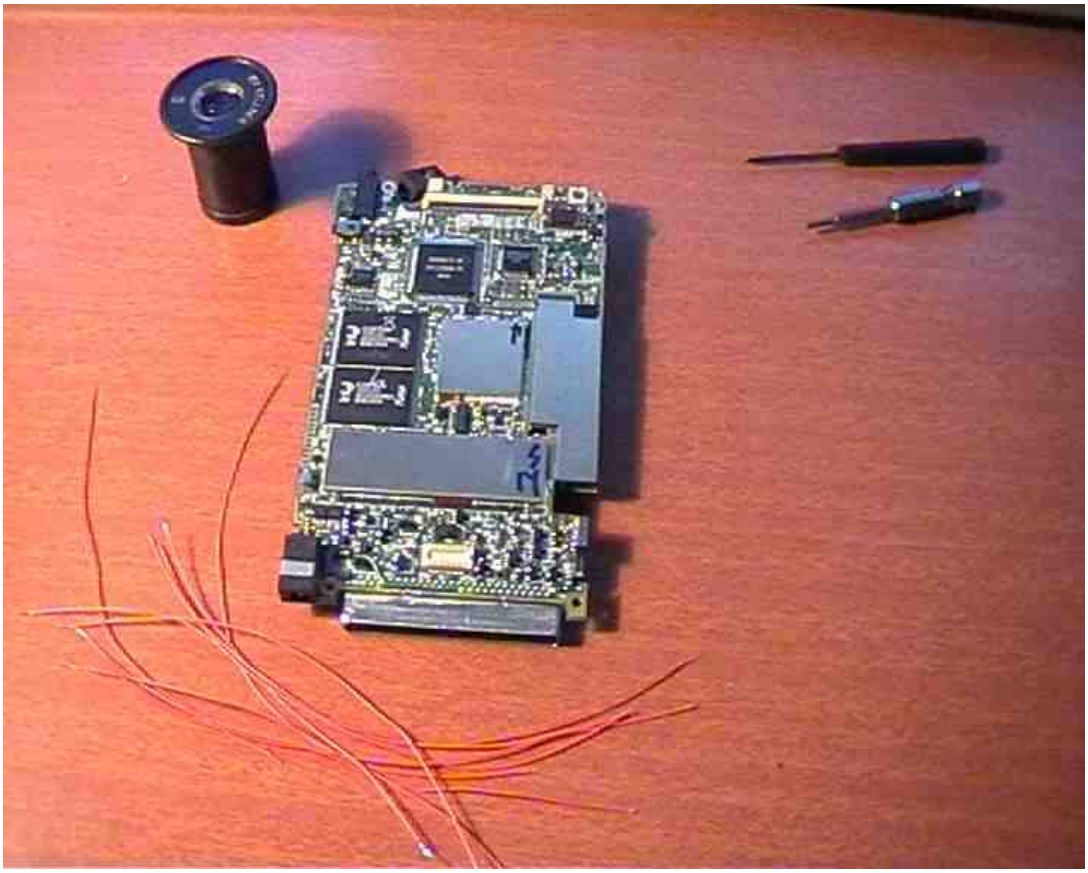
Fitting a JTAG interface to an iPAQ 3600

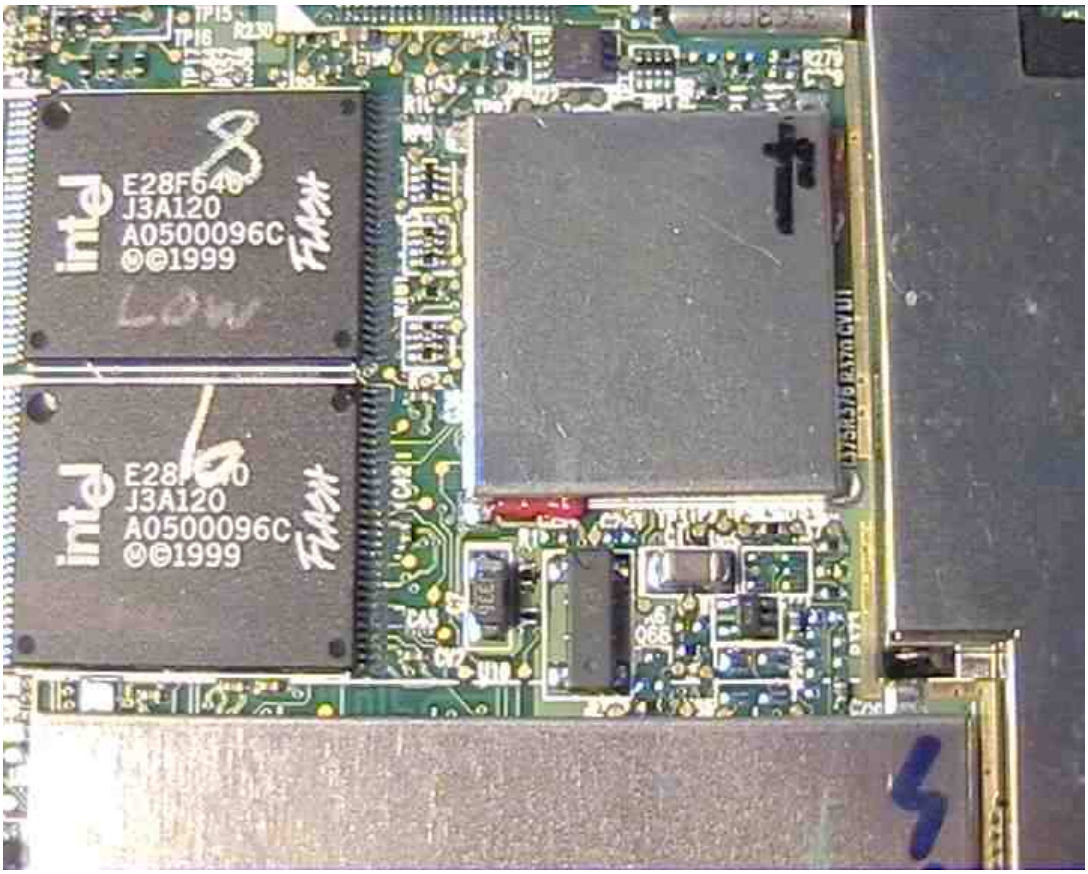
So you couldn't resist messing with your brand new iPAQ, tried to re-flash it and managed to turn it into a brick ?
Now you turn it on and it just sits there, silent, screen blank, just mocking you ?
Do you (think you) have the skills to take delicate, sensitive electronic appliances apart ?
...and solder connections to small contact pads you can barely see with the naked eye ?
...and put them back together ?

If the answer to all of the above it yes, never fear, help is here [cue cheesy music]

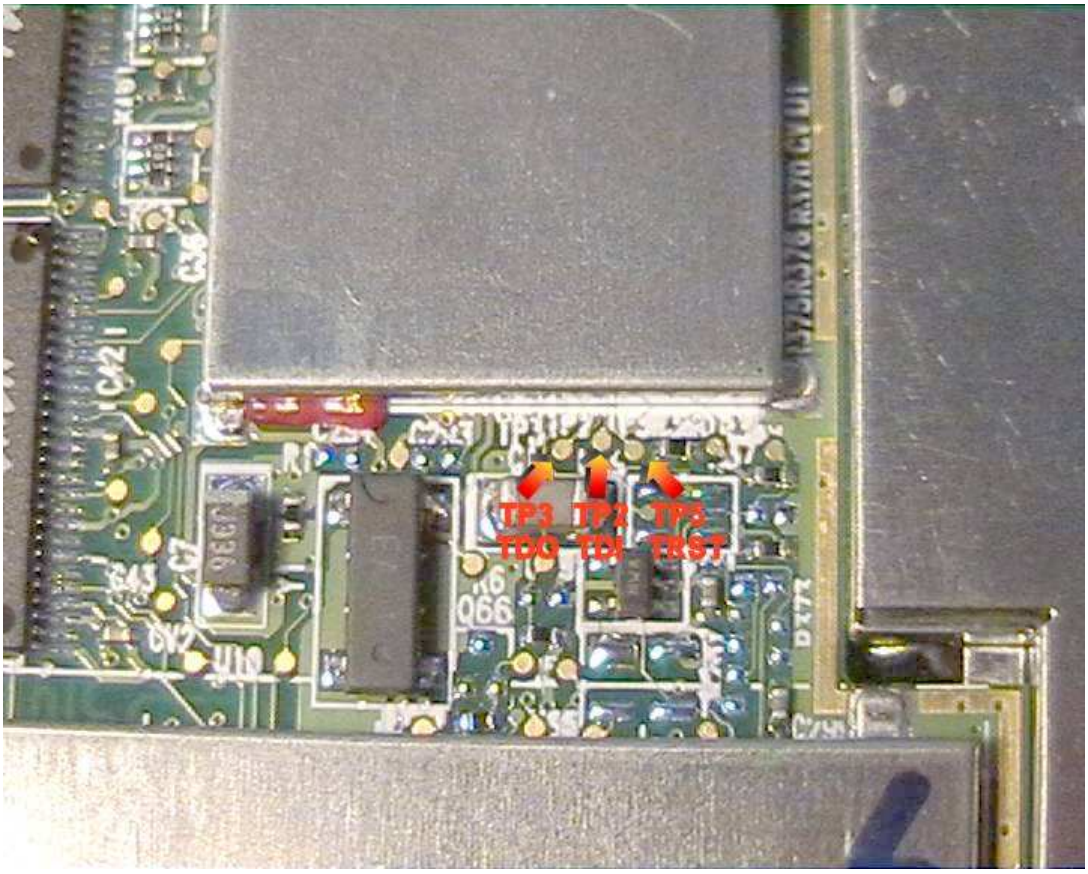
First of all, disassemble the iPAQ and remove the main board. If you wish, desolder the speaker cables.



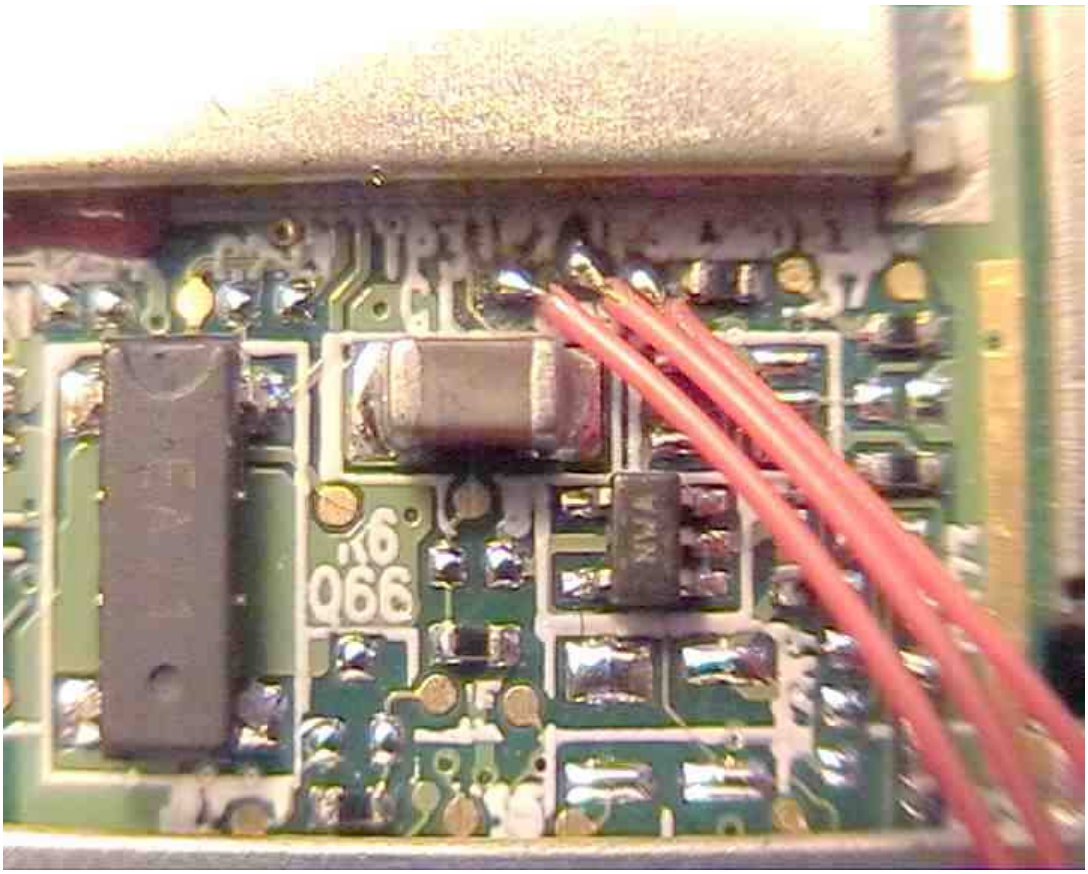
Next turn the board with the CPU facing up. The CPU is the square object located next to the E28F640J3A120 FLASH chips as seen below. It is covered with a metallic shield.



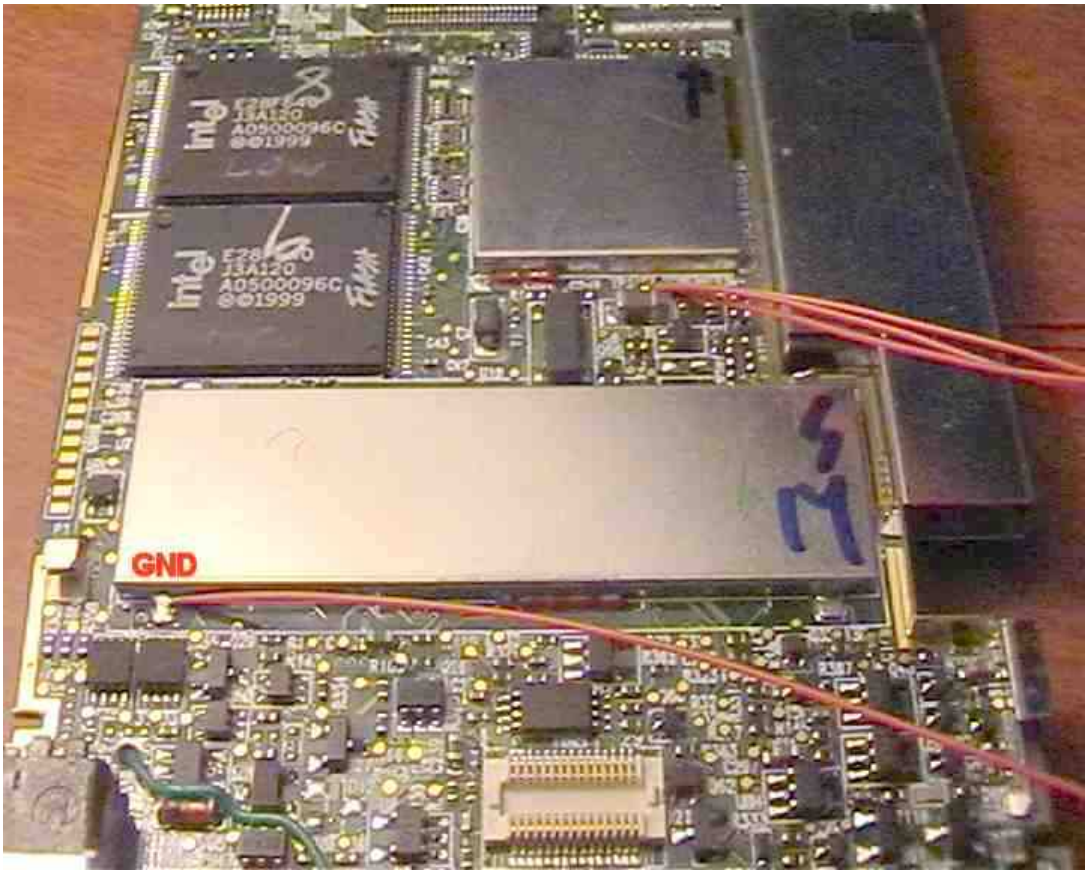
Take a closer look at the bottom edge of the CPU and locate the test points, ordered left to right as TP3 (TDO), TP2 (TDI) and TP5 (TRST)



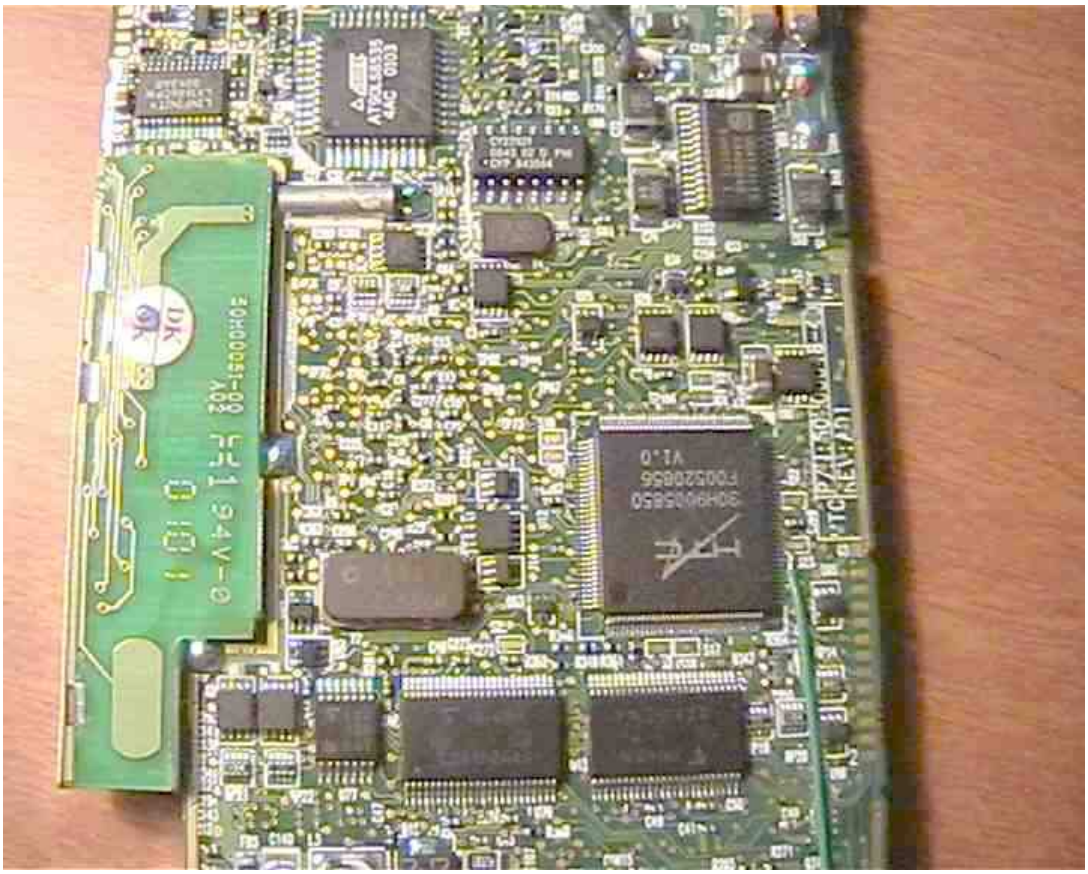
Carefully solder some suitable wires to the test points.



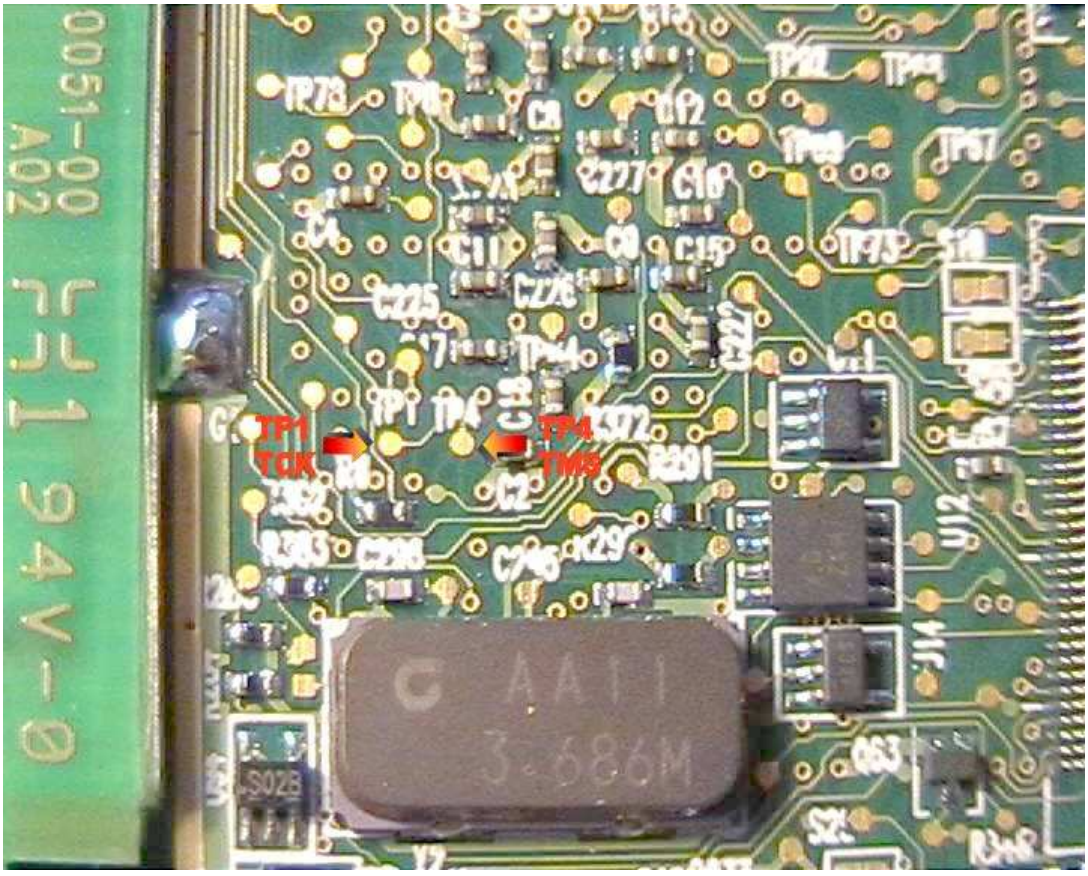
Find a good ground connection and solder a wire to it too.



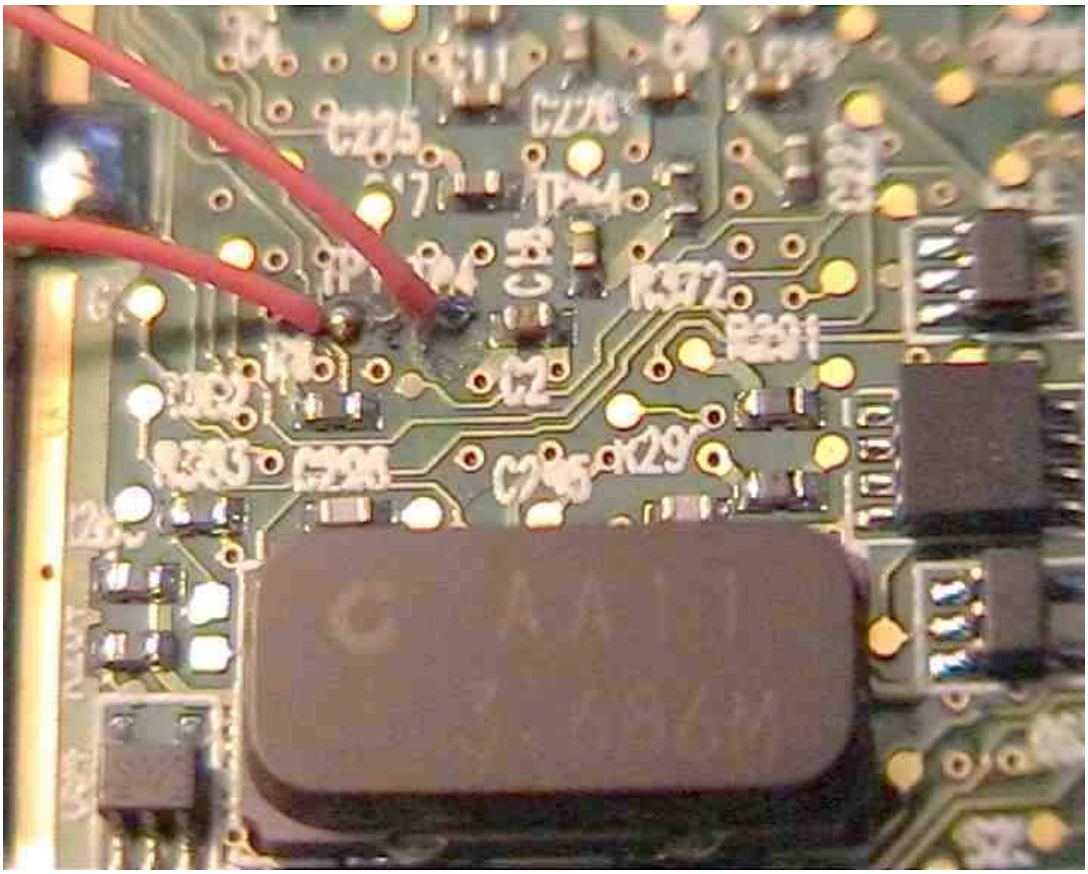
Turn the board over and locate the 3.686Mhz crystal.



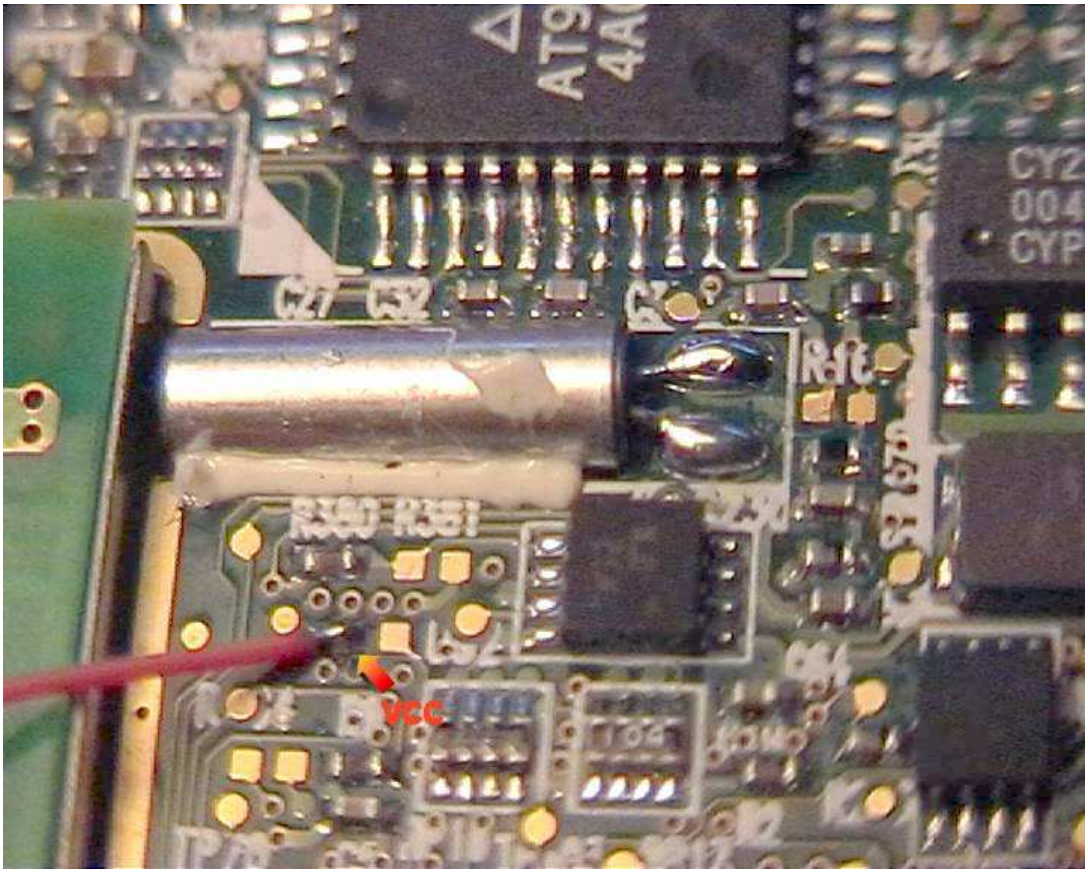
Just above it, you'll find test points TP1 (TCK) and TP4 (TMS)



Solder some suitable wire to these test points too.

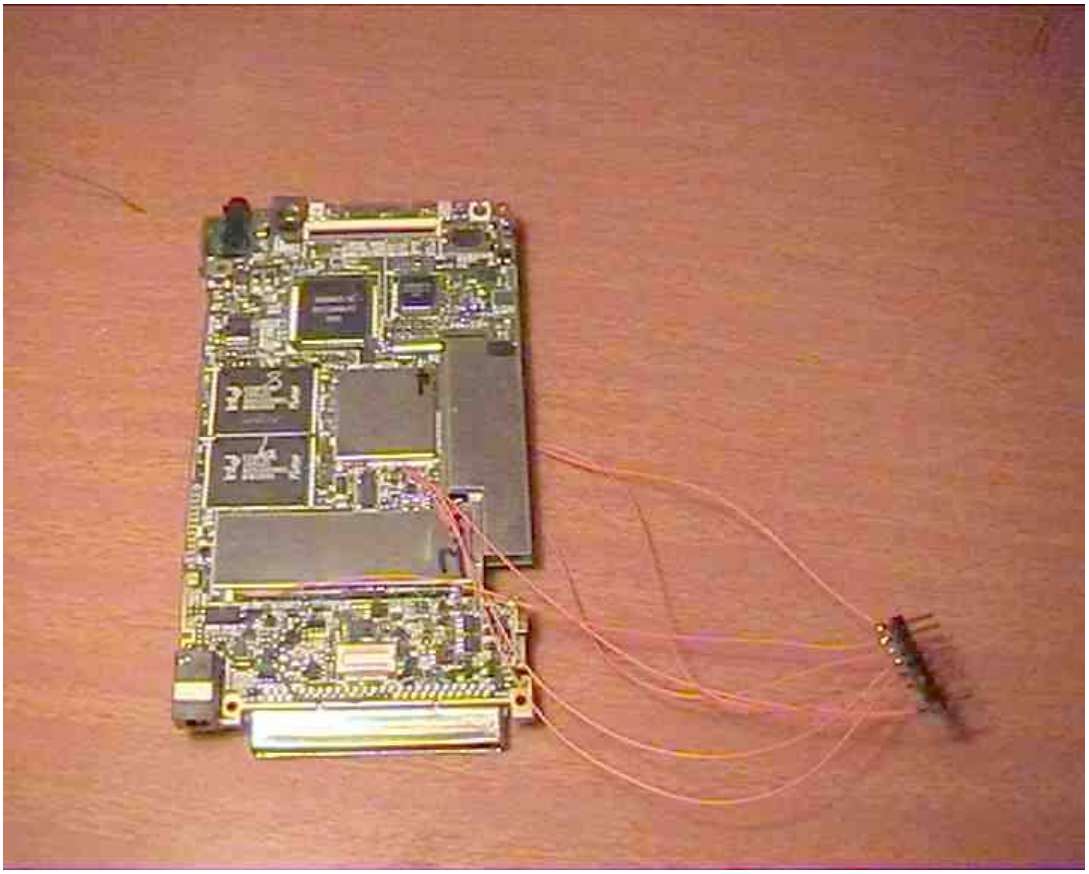


Further up on the same side of the board is a metal can crystal, located just below the Atmel AT90LS8535 microcontroller. This is the VCC connection, so solder a wire to the pad shown below.



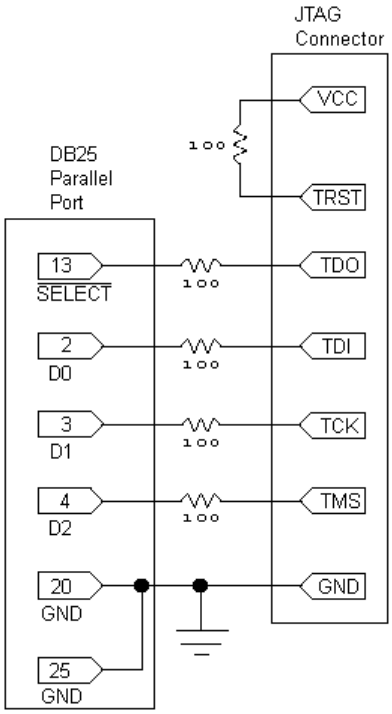
Eventually you end up with a bunch of wires hanging loose of the iPAQ board.

You could, if desired, solder a berg strip or other suitable connector for easy attachment to the JTAG board.



If you don't have a JTAG adapter, Xilinx publishes the schematic of a PC parallel port to JTAG adapter that you can easily build yourself.

I personally ended up using the simplest interface of all, about five resistors (any value 100 - 330 ohm should do) arranged as follows:

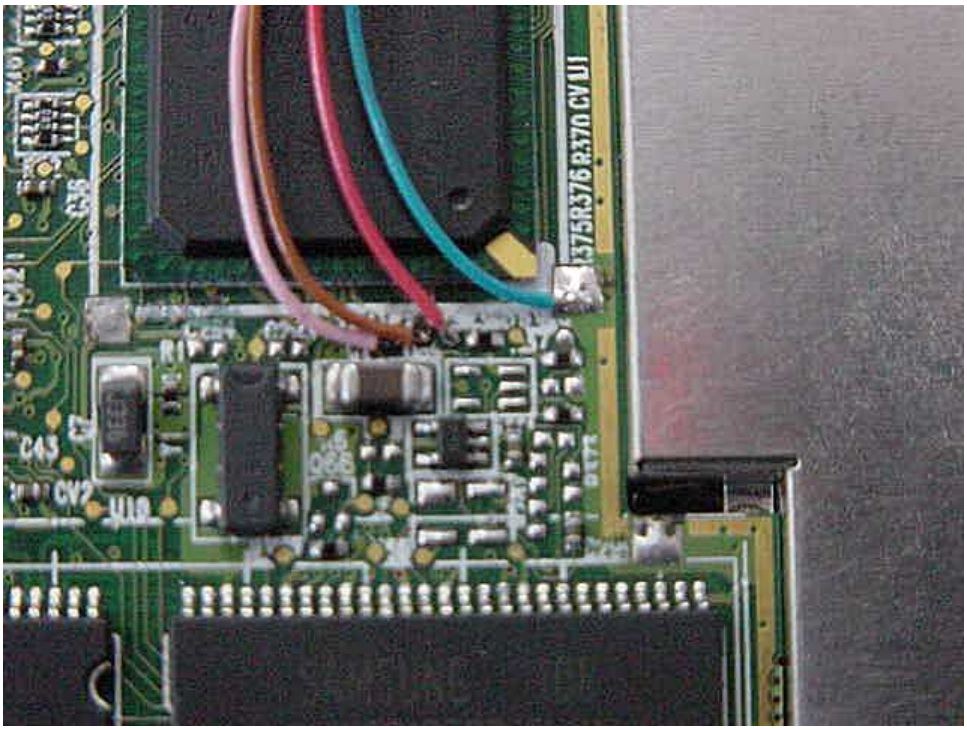


Finally you need some spiffy software to do the programming. I used jtag-0.2.1 which I set up under cygwin on my Windows XP system.
I won't go into the software setup here, however jtag-0.2.1 needs a couple of small mods before it can program the iPAQ.
Download flash.c and sa1110.c and replace the corresponding files in jtag-0.2.1 source tree before compiling.

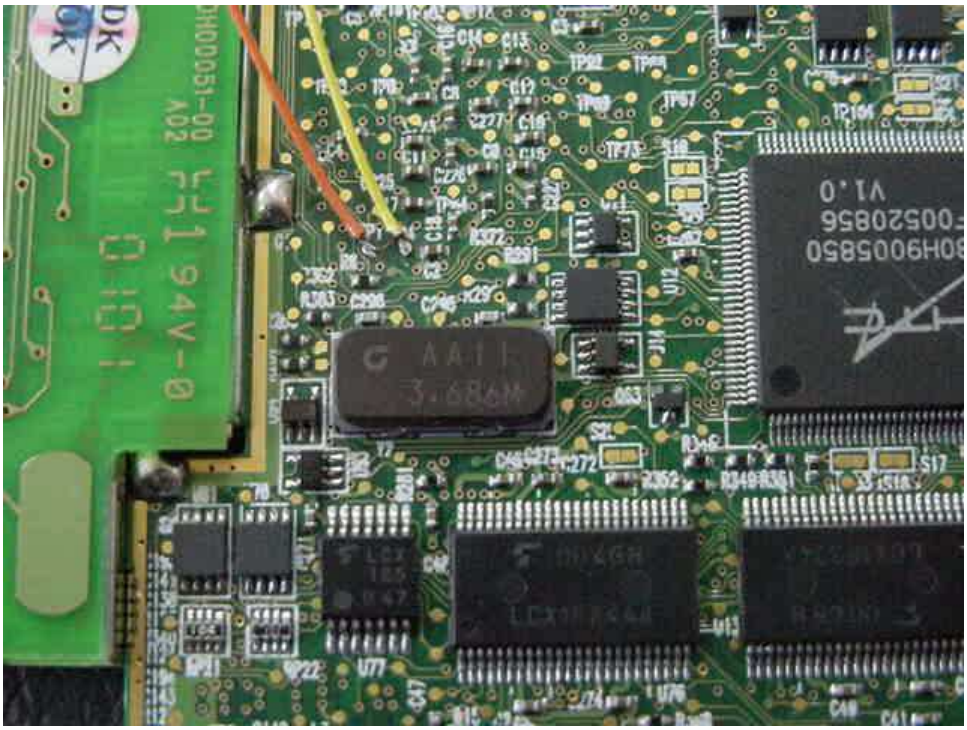
Happy Flashing

Additional pictures of the final JTAG connector permanently fitted to my development iPAQ.
These were taken after desoldering the metal shield covers off the CPU and DRAM.

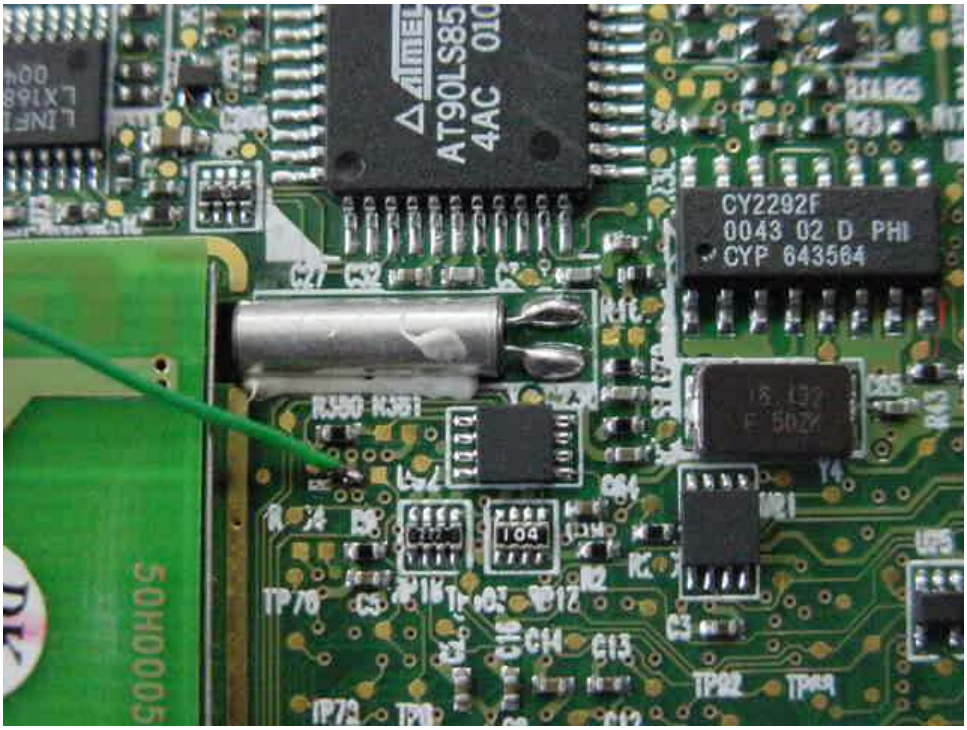
Pink: TDO, Brown: TDI, Red: TRST, Blue: GND



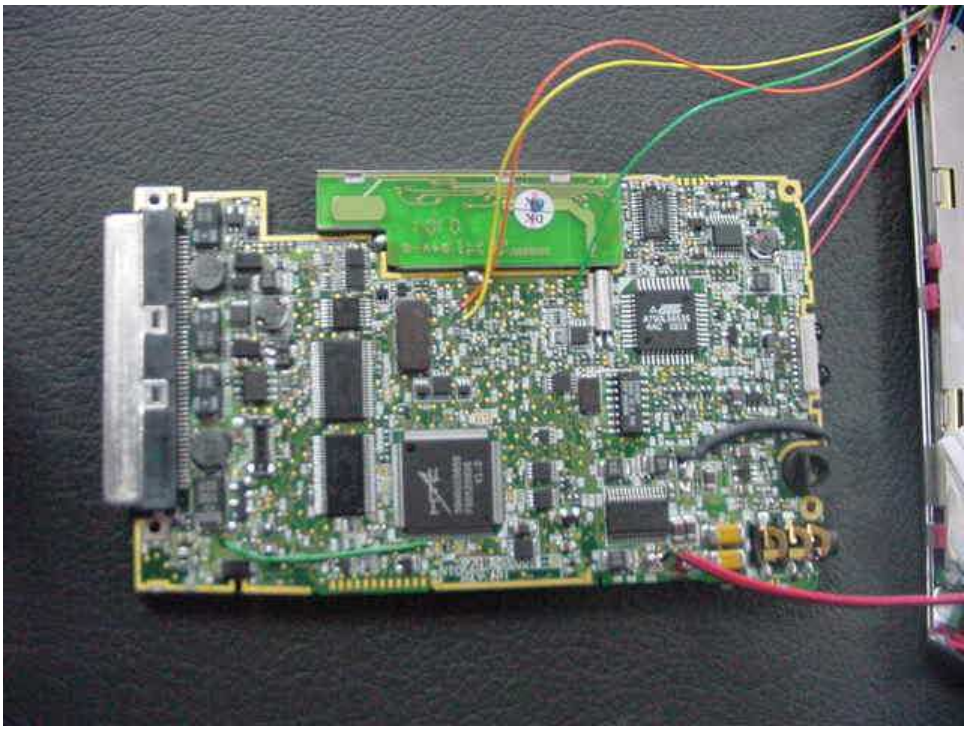
Orange: TCK, Yellow: TMS



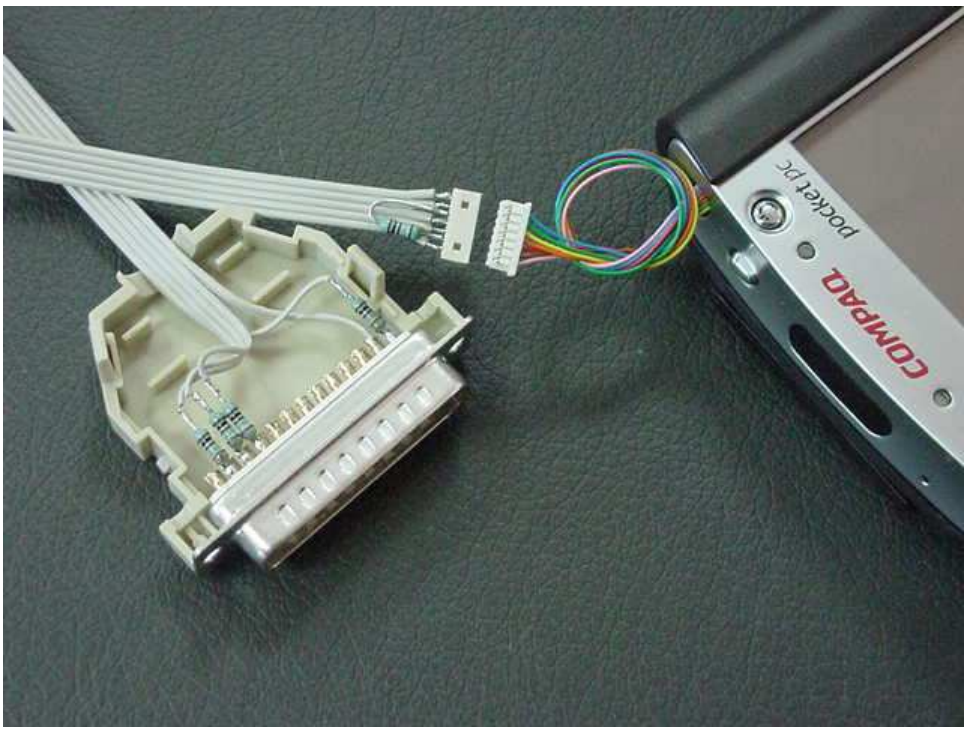
Green: VCC



All cables soldered.



Final assembly. I chose to pull the cables out of the stylus hole as anywhere at the bottom interferes with the docking station seating. There's no other suitable hole and I wasn't about to drill one :)
This means the stylus will not plug in (small sacrifice) but on the plus side, the casing around the stylus hole, clamps the wires in place so you can't accidentally tug on them too hard.



Below is an actual session programming a bootloader into a fully erased iPAQ.

```
$ jtag
jtag 0.2.1
Copyright (C) 2002, 2003 ETC s.r.o.
jtag is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
There is absolutely no warranty for jtag.

Warning: jtag may damage your hardware! Type "quit" for exit!

Type "help" for help.
```

```
jtag> cable parallel 0x378 DLC5
Initializing Xilinx DLC5 JTAG Parallel Cable III on parallel port at 0x378
jtag> detect
Device Id: 10001001001001100001000000010011
  Manufacturer: Intel
  Part:         SA1110
  Stepping:     B4
  Filename:     /usr/local/share/jtag/intel/sa1110/sa1110
jtag> detectflash
Note: Supported configuration is 2 x 16 bit only
ROM_SEL: 32 bits

2 x 16 bit CFI devices detected (QRY ok)!

CFI Query Identification String:
        Primary Vendor Command Set and Control Interface ID Code: 0x0001 (Intel/Sharp Extended Command Set)
        Address of Primary Algorithm extended Query table: P = 0x????
        Alternate Vendor Command Set and Control Interface ID Code: 0x0000 (null)
        Address of Alternate Algorithm extended Query table: A = 0x????
CFI Query System Interface Information:
        Vcc Logic Supply Minimum Write/Erase voltage: 2700 mV
        Vcc Logic Supply Maximum Write/Erase voltage: 3600 mV
        Vpp [Programming] Logic Supply Minimum Write/Erase voltage: 0 mV
        Vpp [Programming] Logic Supply Maximum Write/Erase voltage: 0 mV
        Typical timeout per single byte/word write: 128 us
        Typical timeout for minimum-size buffer write: 128 us
        Typical timeout per individual block erase: 1024 ms
        Typical timeout for full chip erase: 0 ms
        Maximum timeout for byte/word write: 2048 us
        Maximum timeout for buffer write: 2048 us
        Maximum timeout per individual block erase: 16384 ms
        Maximum timeout for chip erase: 0 ms
Device Geometry Definition:
        Device Size: 8388608 B
        Flash Device Interface description: 0x0002 (x8/x16)
        Maximum number of bytes in multi-byte write: 32
        Number of Erase Block Regions within device: 1
        Erase Block Region Information:
                Region 0:
                        Erase Block Size: 131072
                        Number of Erase Blocks: 64
Manufacturer: Intel
Chip: 28F640J3A
jtag> print
 No. Manufacturer         Part                Stepping Instruction          Register
-------------------------------------------------------------------------------------
   0 Intel                SA1110              B4       EXTEST               BSR
jtag> flashmem 0 /home/bootldr-2.18.54.bin
0x00000000
Note: Supported configuration is 2 x 16 bit only
ROM_SEL: 32 bits

2 x 16 bit CFI devices detected (QRY ok)!

program:
block 0 unlocked
erasing block 0: 0
addr: 0x00033500
verify:
addr: 0x00033500
Done.
jtag> quit
$
```