Guida
ai segreti
del
DWL-2100AP

Di Stefano Ramponi.

Oggetto: **questa guida ha lo scopo di approfondire le conoscenze riguarda l'Access Point in oggetto in modo da sfruttarne a pieno tutte le capacità. Ricordo che operazioni di questo tipo vanno affrontate solo da persone esperte e che alcune di queste portano all'invalidazione della garanzia.**

Premessa: Esistono diverse revision hardware del 2100AP che sono giunte sul mercato italiano Rev. A2, A3 e A4. A livello di firmware non ci sono problemi di compatibilità tra queste diverse revision, mentre a livello di hacking hardware (Jtag, Seriale, PoE) potrebbero esserci notevoli differenze.

Argomenti affrontati:

1 Aggiornamento firmware con uno free, svincolando il device dal proprio product code

Tramite un'apposita procedura è possibile rimuovere il product code dal device. In questo modo è possibile caricare sull'Access Point un firmware free compatibile con l'hardware del 2100AP che ne dovrebbe migliorare sensibilmente le prestazioni. Questa procedura può essere utilizzata anche per testa un firmware custom in quanto il boot-loader del device non viene modificato permettendo qualora il firmware non funzionasse di ricaricare quello originale. Nello specifico caricheremo un firmware custom free sviluppato da un team russo, che stando a quanto dichiarato permetterebbe link di 50km con altissimi transfert-rate.
Riporto il link dove è possibile scaricare il firmware V2_180 che ricordo è un firmware FREE

http://www.wifi-connect.ru/files/BlueBox_AP2100.zip

2 Hacking xtender range per link lunghi

3 Hacking regularydomain per l'utilizzo del canale 14

4 Hacking per il recupero di password e cifratura.

Questa una semplice procedura sfrutta una falla di sicurezza del 2100AP permettendo di recuperare senza grossi problemi la password per il web-management e la sessione telnet ed eventualmente la cifratura.

5 Abilitazione di client multipli a valle di un D-Link dwl-2100 in modalità wireless client.

6 Modifica per l'uso del Power Over Ethernet.

Tramite una piccola modifica dell'hardware è possibile rendere il 2100AP Rev. A2 un Access Point con la capacità di alimentarsi tramite il Power Over Ethernet.
N.B. non ho sperimentato direttamente tale mod-hardware ma ne riporto le informazioni così come le ho reperite. Non mi ritengo quindi responsabile di eventuali danneggiamenti agli AP nell'applicazione di tale modifica.

7 Jtag e Seriale sul 2100AP. Sulla board sono presenti le connessioni per una seriale e una Jtag.

8 Interfacciarsi alla JTAG del DWL-2100AP.

9 Interfacciarsi al boot-loader del DWL-2100AP.

10 VxWorks e file ELF.

11 Compilare un firmware custom e Openwrt.

1. Procedura di aggiornamento firmware del Dlink DWL-2100AP

Presentazione del firmware V2_180.

Il firmware V2_180 è un firmware FREE realizzato da un gruppo di programmatori molto probabilmente russi, che si sono adoperati per creare un firmware per AP basati su chip Atheros in grado di stabilire link molto distanti (decine di km) con altissimi transfert rate.
Da tale firmware è stato poi ricavato il firmware della BlueBox, un AP basato sullo stesso hardware del 2100AP Rev. A2 venduto nell'est Europa.

Il firmware ha una gestione unicamente tramite telnet mentre la gestione tramite browser è assente.
Il set dei comandi utilizzabili è molto completo ed è analogo a quello Dlink.

Un tool molto utile per gestire i 2100AP col firmware V2_180 è AP Manager by Acowa, prelevabile a questo indirizzo:

http://www.acowa.narod.ru/

Ho avuto modo di verificare che tale firmware è perfettamente funzionante sull'hardware del DWL-2100AP e che la procedura qui descritta funziona perfettamente, ma purtroppo non ho avuto modo (ne tanto meno il tempo) di testare la veridicità di tale capacità.
Qual'ora vengano effettuati dei test, questi verranno inseriti in questa guida.

La procedura di aggiornamento del firmware tramite un firmware compatibile, ma non firmato dalla propria casa, necessita di un particolare login "Alpha" all'interno di una sessione telnet, che consente l'uso di comandi avanzati nascosti.

Assicurasi di aver disabilitato il firewall di windows o qualsiasi firewall software qual'ora presenti

1 Fase: Aggiornamento firmware originale

- Prima di tutto va aggiornato l'AP con un firmware originale che supporti tale comandi.
  Io ho utilizzato con successo il firmware v210eu-r338

2 Fase: Corruzione volontaria del firmware

- connettersi all'AP via telnet
- loggarsi con il proprio user e password
- digitare:

  alpha

- inserire la password:

  sdd21234 (a me questa funziona benissimo)

  Password alternative
  vec21234 (questa sarà la password dopo l'aggiornamento)
  dlk19283

- rm apimg1
  delpcode
  reboot
- fai un reset hardware dell'AP per 20 secondi

3 Fase: Caricamento del firmware custom

- a questo punto il tuo AP risponderà all'indirizzo IP 10.0.0.1
- apri il tuo browser e digita 10.0.0.1
- l'ap dovrebbe proporti in automatico l'aggiornamento firmware
- seleziona il file v2_180.tfp
- esegui l'aggiornamento
- dopo il reboot fai un reset hardware
- l'ap dovrebbe rispondere all'indirizzo 192.168.1.50

- Alternativamente se il browser non risponde:

- procurasi un server TFTP. Un ottimo TFTP server FREE è scaricabile a questo indirizzo http://tftpd32.jounin.net/
- telnet 10.0.0.1
- tftp srvip XXX.XXX.XXX.XXX (XXX.XXX.XXX.XXX è l'IP del pc con il server TFTP)
- tftp get v2_180.tfp
- tftp update
- reboot
- reset hardware di 20 secondi
- telnet 192.168.1.50

Questa procedura permette di installare un qualsiasi firmware FREE compatibile sui 2100AP.
Il firmware segnalato in questa guida consente di migliorare le capacità del già ottimo 2100AP.
Risulta quindi come l'hardware di questo AP abbia ulteriori possibilità da sfruttare.

2. Hacking xtender range per link lunghi

Questo hacking software dovrebbe inalzare la sensibilità del 2100AP permettendo di stabilire link distanti in maniera più semplice.

Per applicare tale modifica è necessario collegarsi via telnet al management del 2100AP.
Una volta loggati digitando il comando "set rate" si ottiene come risposta la list dei baud rates disponibili. Possiamo notare come il valore più basso sia quello di 1 Mb.

Questa è la sequenza dei comandi da dare per attuare la modifica.
Ovviamente dopo ogni reboot è necessario riconnettersi e riloggarsi.

set superg disable
reboot
set wirelessmode 11g
reboot
set rate 0.25
reboot
set superg enable
reboot

A questo punto digitando il commando “set rate” nella lista dei disponibili baud rate saranno presenti 0,25 e 0,5 indicando che la sezione radio è impostata in extended range mode.

Questa modifica dovrebbe portare a una rilevabile diminuzione dei tempi di risposta medi sul ping e quindi a una diminuzione della latenza media.

3. Hacking regularydomain per l’utilizzo del canale 14

Premessa.

L’utilizzo del canale 14 è vietato in italia. Riporto la procedura di sblocco solo per testing e sperimentazione.

Questo hacking software consente di disattivare le restrizioni a livello di canali utilizzabili dal 2100AP. L’impostazione di un particolare parametro limita o meno l’utilizzo dei canali non consentiti in un determinato paese.

Come per gli altri hack software è necessario loggarsi al 2100AP via telnet e dare i seguenti comandi.

```
set regularydomain none
reboot
```

Al riavvio il 2100AP sarà privo di limitazioni “regionali”.

4. Procedura di recupero della password e della cifratura nel Dlink DWL-2100AP

1 Fase: Collegarsi al DWL-2100AP via ethernet

- Digitare http://(ip del dwl-2100ap)/cgi-bin/Intruders.cfg
- Verrà richiesto dove salvare il file
- Salvare il file su disco

2 Fase: Recupero informazioni

- Aprire il file salvato con un editor
- Scorrerlo e recuperate le informazioni di cui avete bisogno

NB ricordo che con tale operazione, qualora sia compiuta su apparati non di nostra proprietà, si compie un reato PENALE.

5. Abilitazione di client multipli a valle di un D-Link dwl-2100 in modalità wireless client

Tramite questa procedura è possibile collegare (tramite un switch) client multipli a valle di un DWL-2100AP configurato in wireless Client-mode di un altro Access Point.

Per applicare tale modifica è necessario collegarsi via telnet al management del 2100AP.

Questa è la sequenza dei comandi da dare per attuare la modifica.
Ovviamente dopo ogni reboot è necessario riconnettersi e riloggarsi.

set matstate 1
reboot
get matstate (il 2100AP come risposta dovrebbe restituire)
MAT: HomeAddress(AP MAC), MultiEthClient(Yes)!

Grazie a questa modifica, tutti i client collegati al 2100AP vengono mostrati come aventi tutti lo stesso mac address (quello dell'ap in Client-mode). Dopo aver attivato questa procedura, potreste aver bisogno di riavviare gli altri access point e switch della vostra rete, per far si che venga ripulita la ARP-cache.

Per disabilitare la modalità "multi-client" e ripristinare l'impostazione precedente, va ripetuta la procedura descritta sostituendo il comando "set matstate 1" con set matstate 0

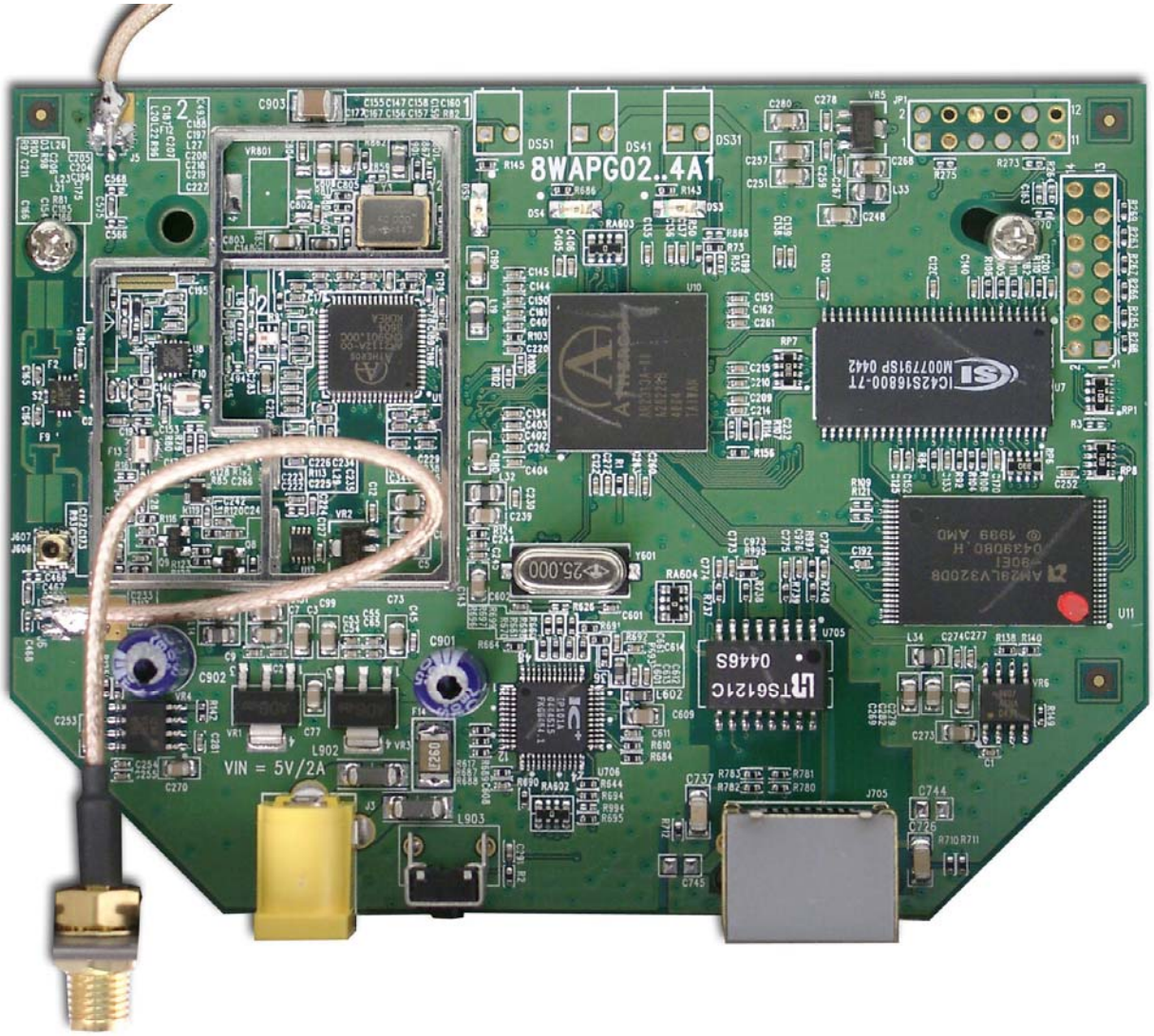
Ringrazio Mirco, alias "Mip", per la segnalazione riguardo questa procedura**.**

6. Modifica del 2100AP Rev. A2 per l'uso del Power Over Ethernet

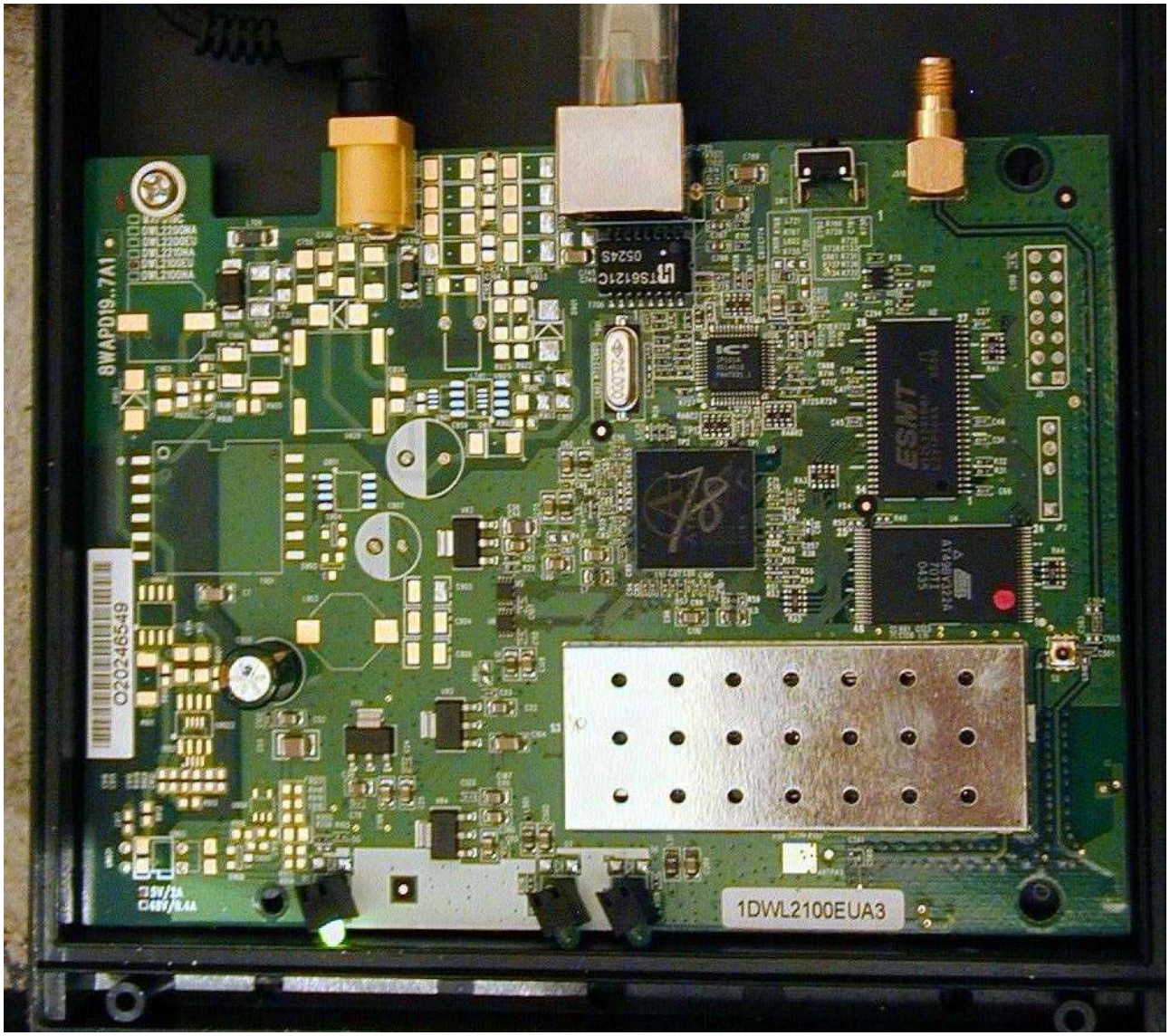
Premessa:

Occorre fare una precisazione, il 2100AP è prodotto in tre differenti board:

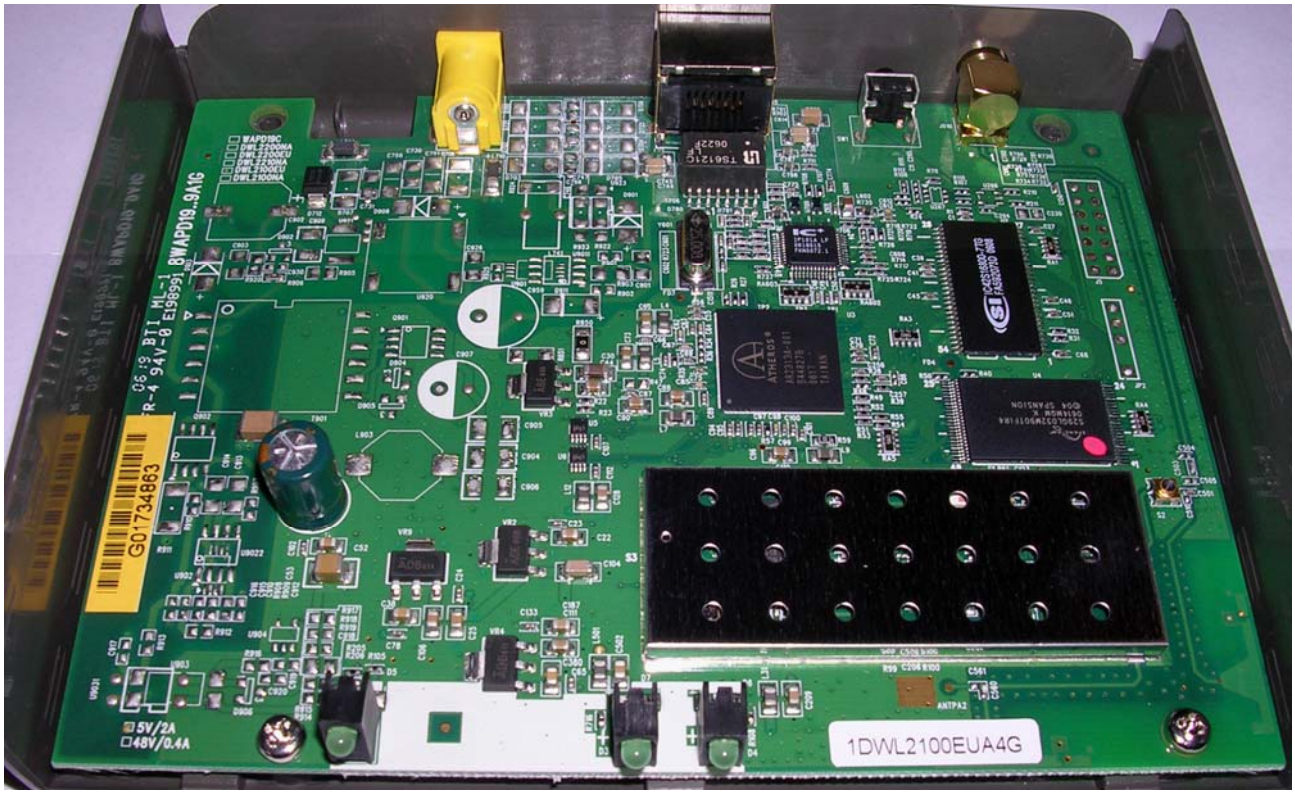
- 8WAPG02..4A1 usata nella Rev. A2
- 8WAPD19..7A1 usata nelle Rev. A3
- 8WAPD19..9A1 usata nelle Rev. A4 (che è quasi identica alla Rev A3)



Rev. A2



Rev. A3



Rev. A4

<u>Questa modifica riguarda la board 8WAPG02..4A1 della Rev. A2</u>

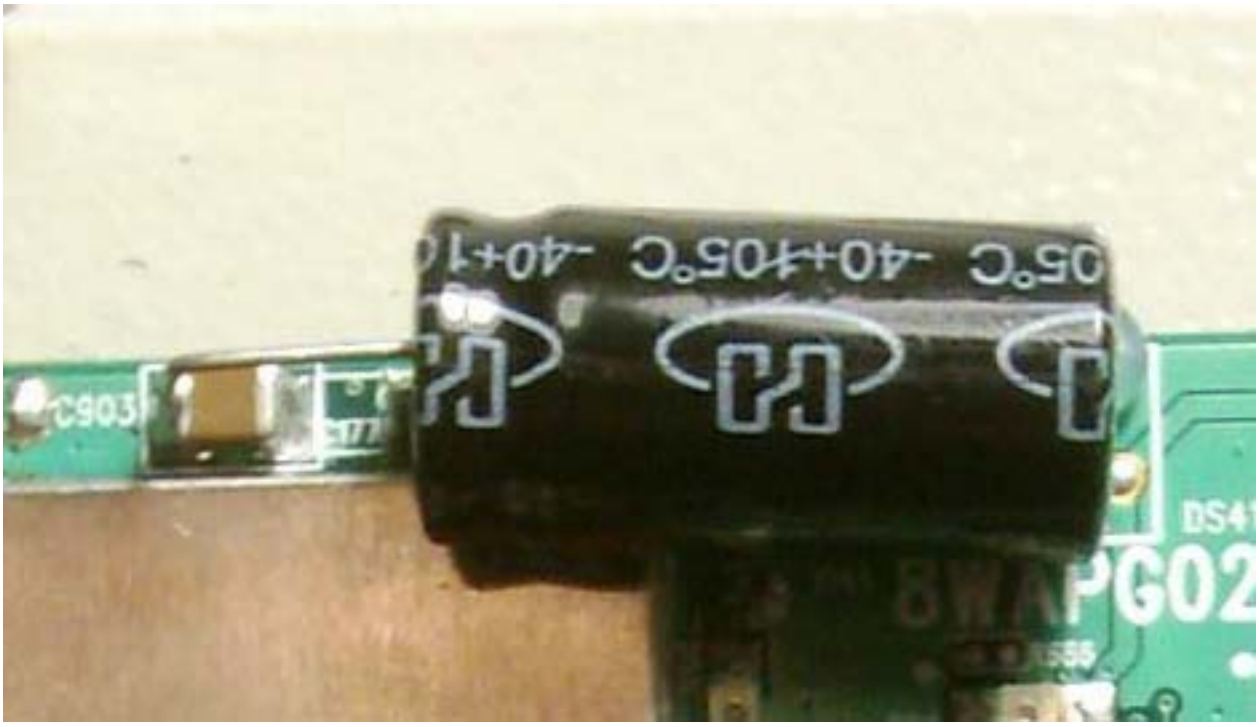
1 Per prima cosa è necessario procurarsi i componenti per la modifica:

- 2 Condensatori elettrolitici da 10v
- 1 Diodo zener (non sono riuscito a identificarne l'esatta tipologia)
- 1 transistor tipcn42

2 Procediamo con la modifica e le saldature

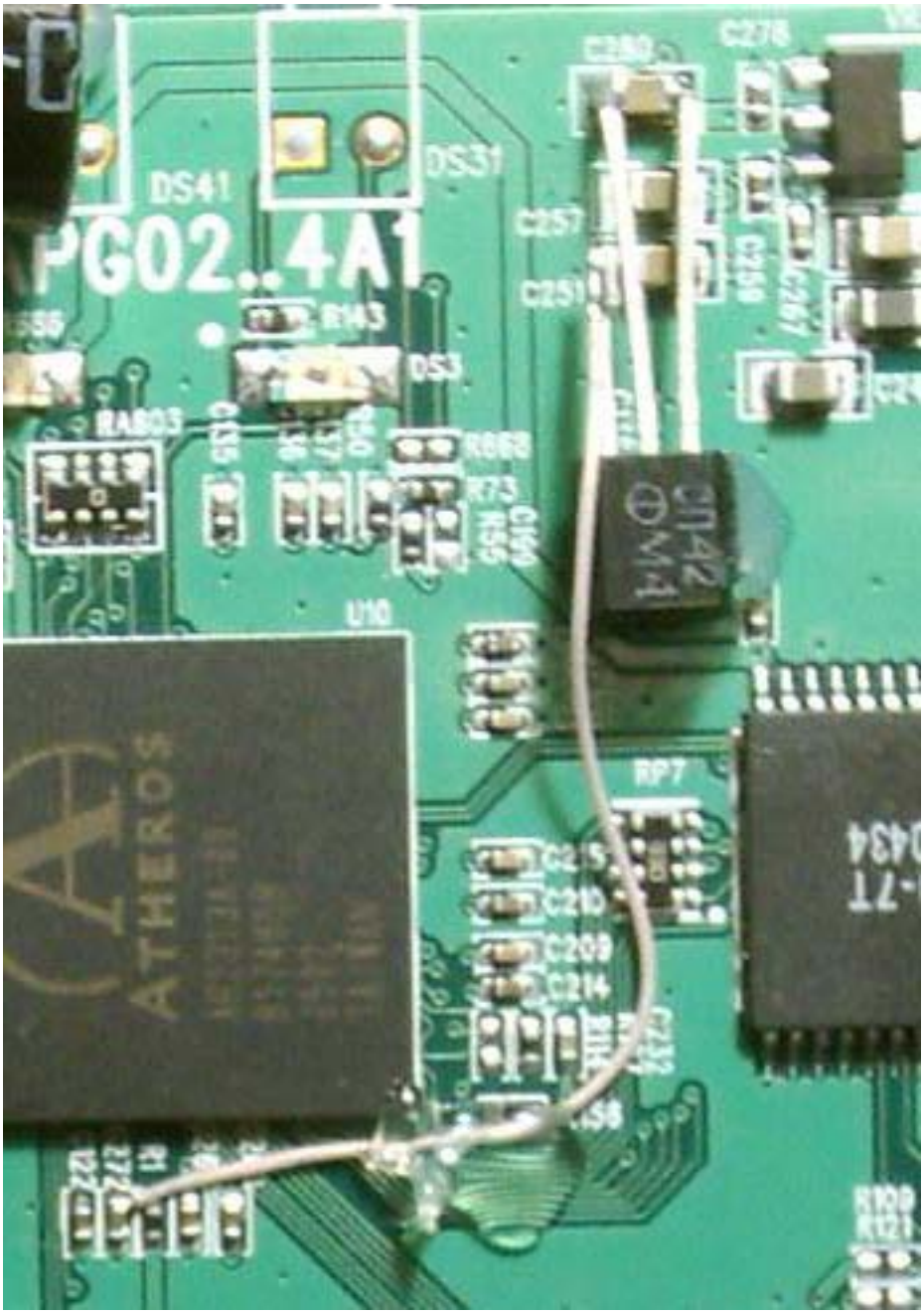
- Saldiamo un condensatore nel punto C253

- Saldiamo l’altro condensatore nel punto C903

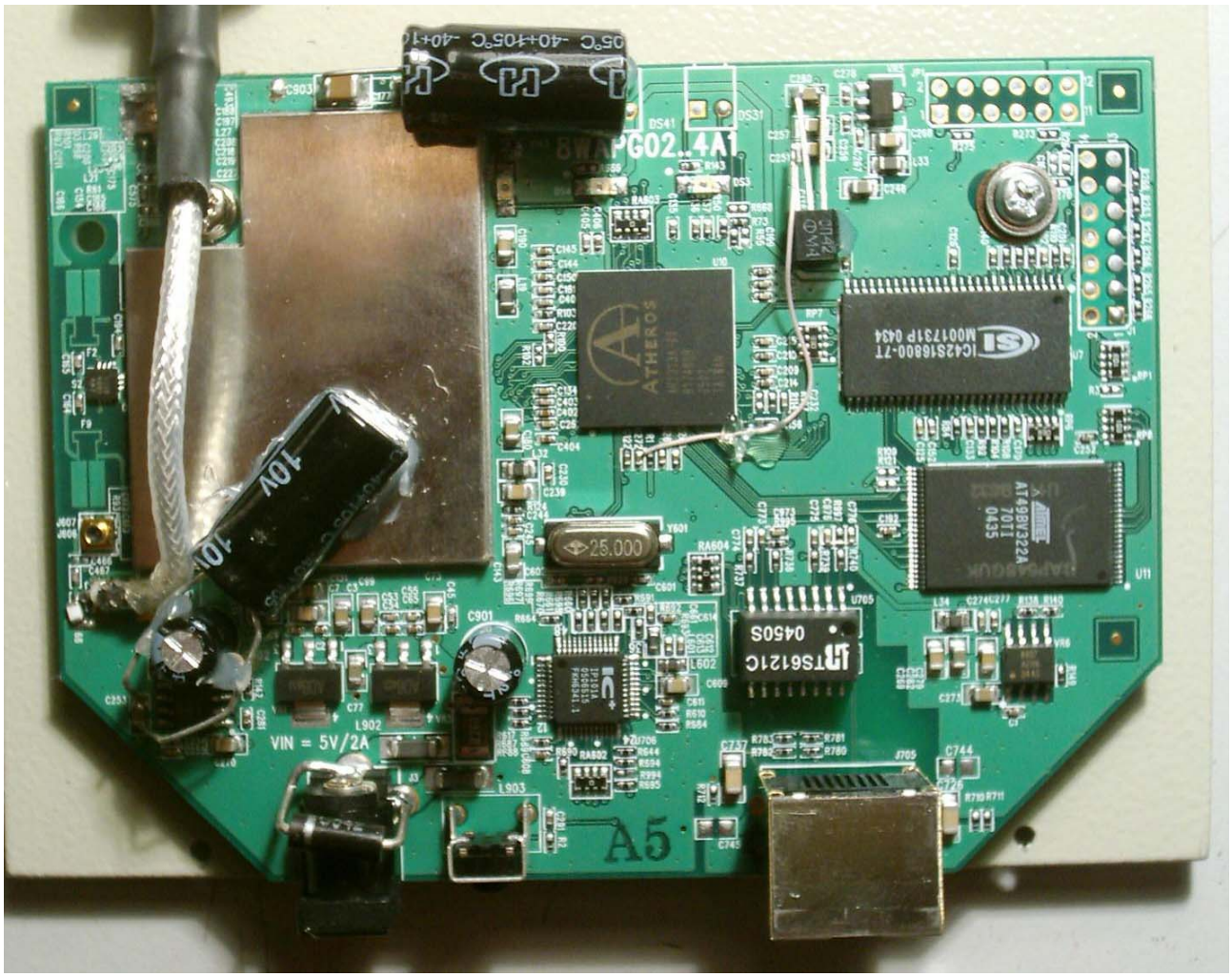


- Saldiamo il diodo al connettore di alimentazione

- Saldiamo infine il tipcn42 nei punti C280, C251 e C272 e lo fissiamo con un po’ di silicone.

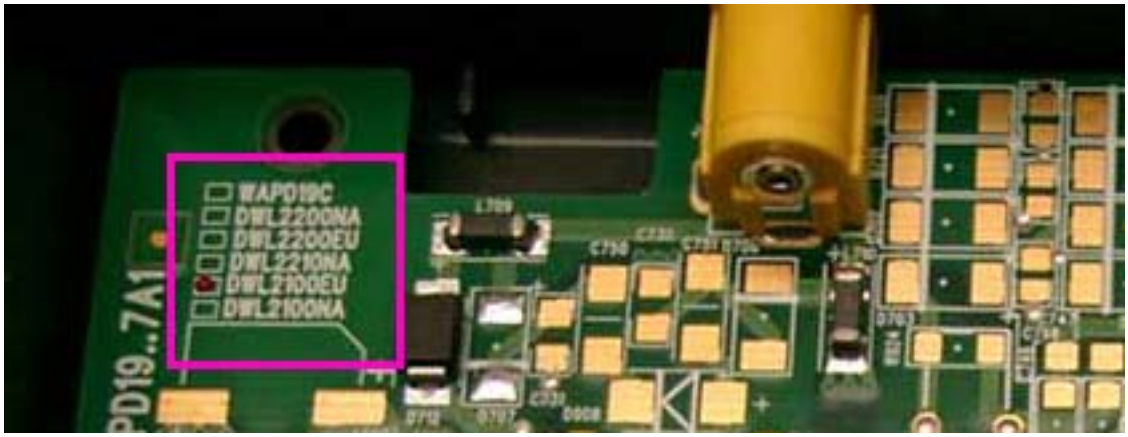


3 Se tutto è stato fatto con cura dovreste ottenere un risultato come quello in figura.



Potete ora alimentare il vostro Access Point tramite PoE con tensioni fino a 40v.
Ripeto che non avendo eseguito in prima persona tale modifica non ho la certezza che le informazioni riportate siano corrette al 100%.

Per ora non sono a conoscenza di un'eventuale modifica analoga per le Rev. A3/A4.
Di sicuro è possibile fare qualche cosa di analogo in quanto la board dal 2100AP Rev. A3/A4 è la stessa utilizzata nell'access point DWL-2200AP che implementa già tale funzionalità

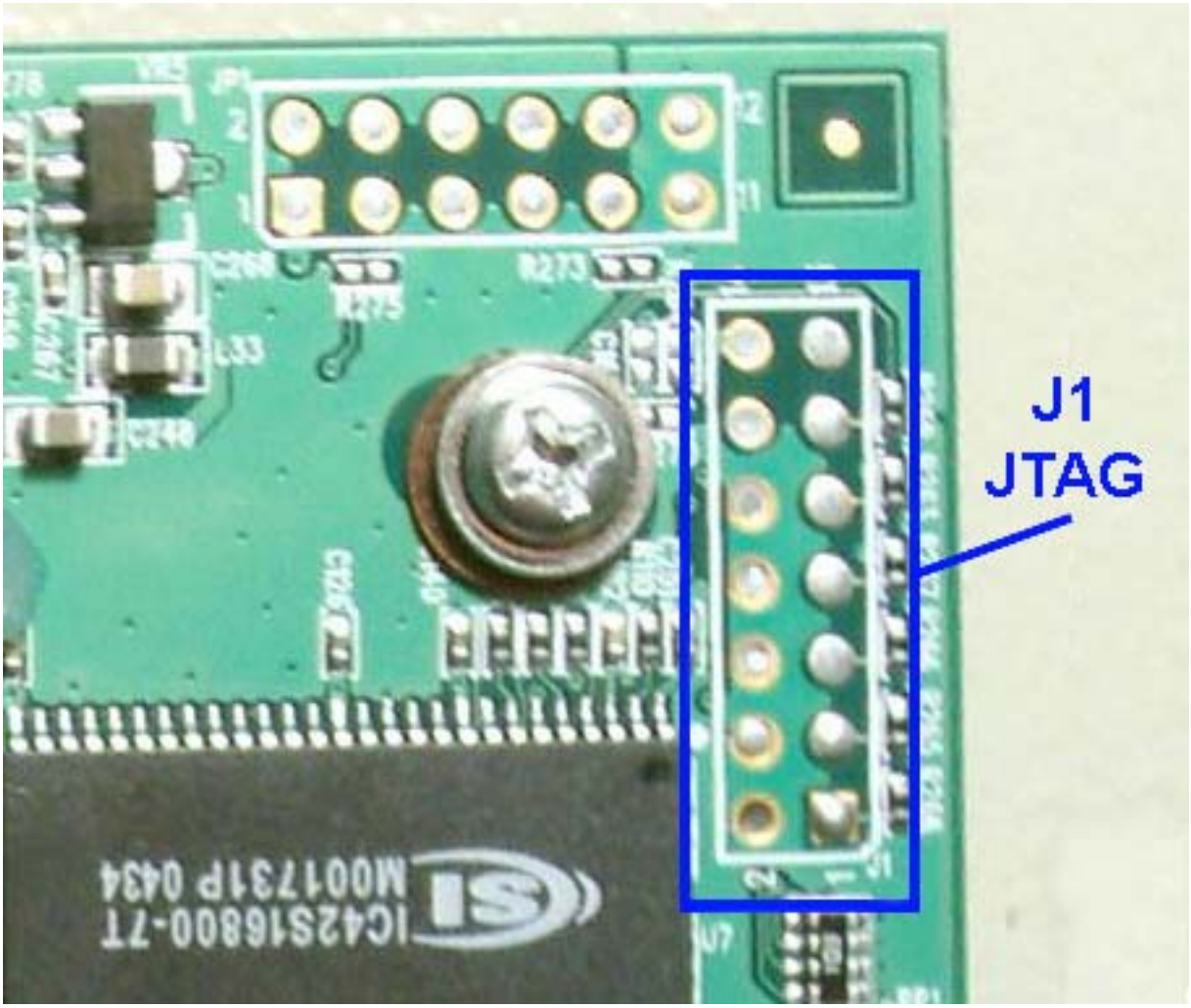


7**. Jtag e Seriale sul 2100AP**

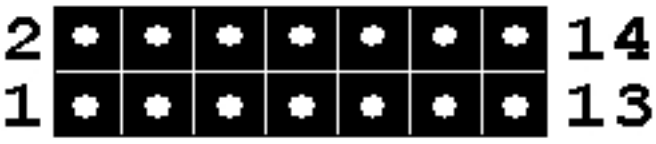
In ultimo ho scoperto la presenza, sulla board del 2100AP, della predisposizione per connettori per JTAG e seriale.
La loro disposizione e forma variano in base alla revision della board del 2100AP

Rev. A2

Il connettore a 14 pin siglato J1 è la predisposizione per la Jtag che è una EJTAG 2.6



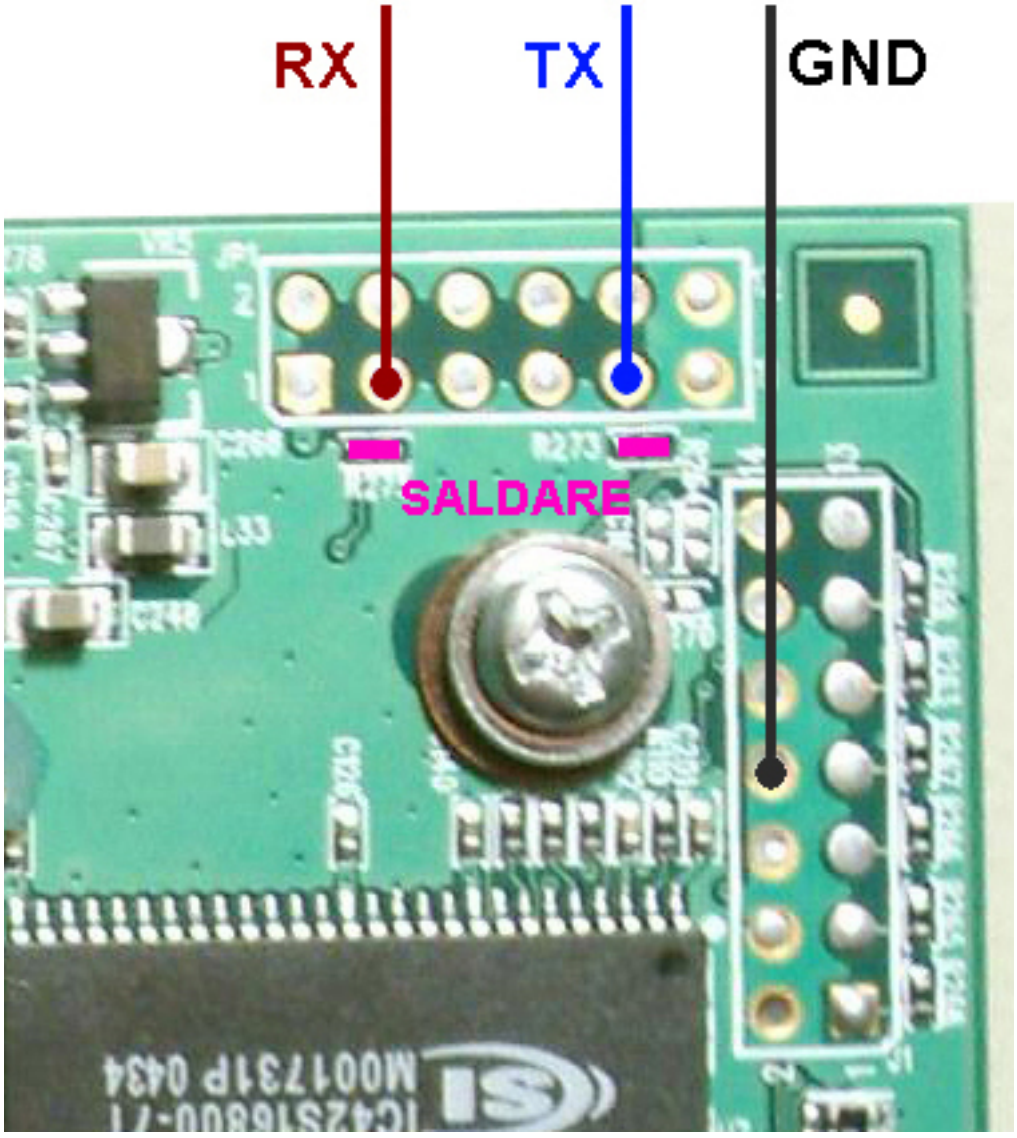
Piedinatura JTAG



	PIN	PIN	
nTRST	1	2	GND
TDI	3	4	GND
TDO	5	6	GND
TMS	7	8	GND
TCK	9	10	GND
RESET	11	12	Key or n/a
n/a	13	14	Vcc +3.3

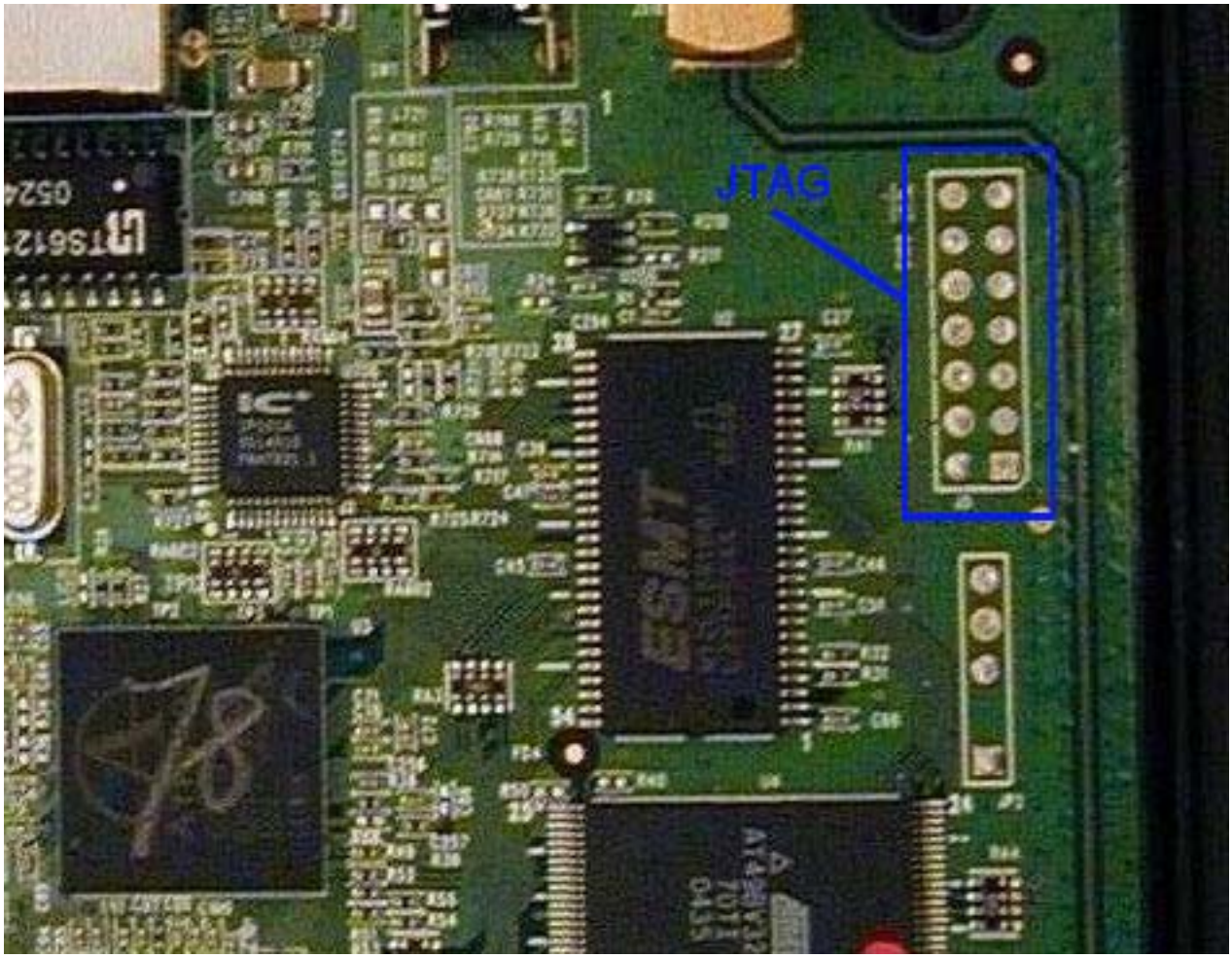
Il connettore JP1 a 12 pin nasconde invece al suo interno un’interfaccia seriale.

Il pin 3 corrisponde al RX, il 9 al TX e per il GND potremo usare il Pin 8 della JTAG.
In alternativa per il GND si potrebbe utilizzare il pin 12 di JP1, ma questo è da verificare.
Sia RX che TX però non funzionano in quanto non sono collegate.
Per risolvere questo è sufficiente ponticellare con un po’ di stagno i punti R273 e R275.

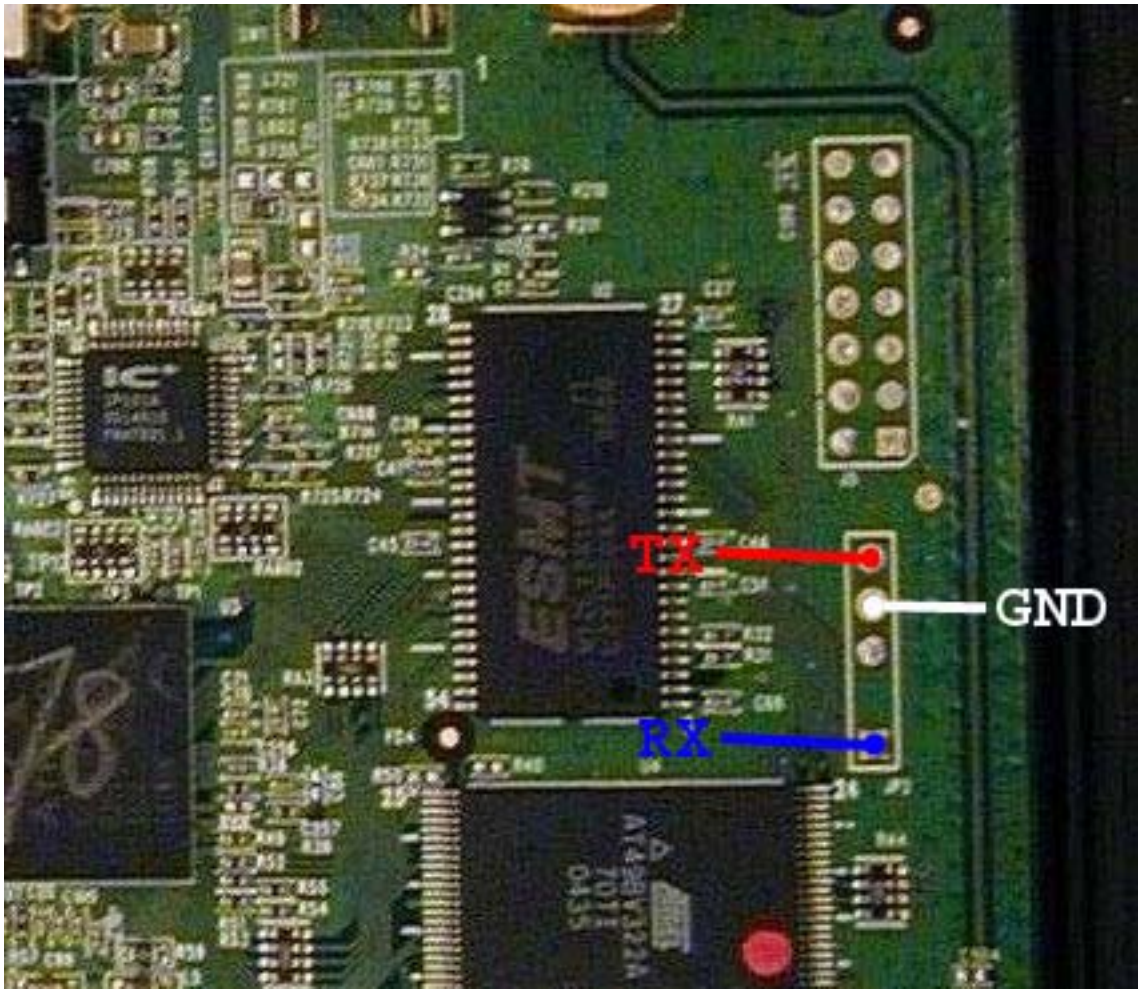


Rev. A3/A4

La predisposizione per la Jtag presente nella board della Rev.A3/A4 è identica a quella della A2. Anche qui abbiano una EJTAG 2.6 e il cavo illustrato precedentemente dovrebbe andar bene anche con questa board.



La seriale presente nella board delle Rev.A3/A4 è molto più semplice da identificare ed è già pronta all’uso senza la necessita di saldature particolari e/o ponticelli.

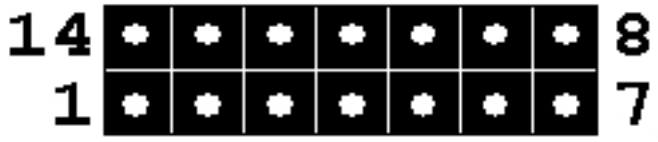


8. Interfacciarsi alla JTAG del Dlink DWL-2100AP

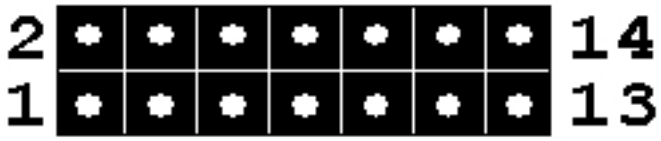
Prima di presentare i vari schemi è necessario fare una precisazione.

La JTAG presente sul 2100AP come ho detto prima è una EJTAG 2.6 e presenta una numerazione in cui la linea inferiore rappresenta i pin “dispari” (1, 3, 5, 7, 9, 11, 13) e quella superiore i pin “pari” (2, 4, 6, 8, 10, 12, 14). Questa la chiameremo per convenzione TIPO A

Altri devi aventi la stessa tipologia di interfaccia hanno però numerazione sequenziale che parte dal primo pin in basso a sinistra per terminare con l’ultimo a destra (1, 2, 3, 4, 5, 6, 7). Riprende poi dal pin in alto a destra per terminare con quello in alto a sinistra (8, 9, 10, 11, 12, 13, 14 da destra verso sinistra). Questa la chiameremo per convenzione TIPO B



TIPO B



TIPO A

Questa premessa è necessaria per evitare dubbi e/o errori nella realizzazione dei cavi di interfaccia JTAG. Invito quindi a tenere come riferimento il pin 1 ed eseguire le saldature/collegamenti seguendo la struttura fisica della piedinatura.

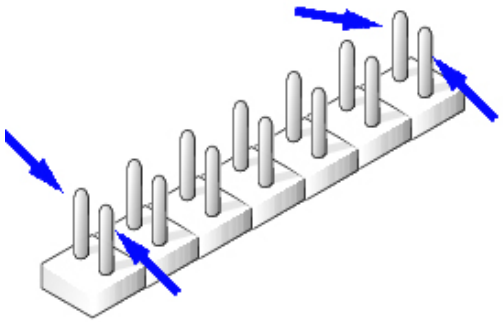
NB non vi sono differenze tra le due interfacce è solo stata adottata una numerazione dei pin differente.

Esempio il Pin 14 del TIPO A fornisce +3.3v così come il Pin 8 del TIPO B.

Come ho detto precedentemente sulla board del 2100AP sono presenti (purtroppo) solo le predisposizioni per JTAG e seriale. Per cui, per interfacciarsi è necessario procurarsi un fila di 7 coppie di pin per saldare il pettine per la JTAG. Mentre per la seriale sono invece sufficienti 3 pin singoli.

Se riuscite a trovarli in un negozio di elettronica è una cosa buona, ma se come me siete un po "occupati" potete seguire una strada alternativa.
Nel mio caso ho recuperato una lunga coppia di pin dissaldando i 50pin di un vecchio controller scsi isa

I fori per i pin, sia dela JTAG che della seriale, sono già stagnati. Per cui, per saldare correttamente il pettine è necessario un saldatore punta fine. Col saldatore scaldate lo stagno del foro del pin dalla parte inferiore della board e non appena questo è fluido spingete il pin nel foro con una leggera pressione.
Suggerisco di iniziare dai pin più esterni (1, 2, 13 e 14) in modo da rendere immediatamente stabile la basetta plastica del pettine. I successivi pin saranno poi più semplici da inserire in quanto già posizionati in direzione del relativo foro.



Se avete "lavorato" bene dovreste ottenere un risultato del genere.

Un po artigianale ma di ottima fattura.

A questo punto è necessario costruirsi l'interfaccia per collegare la parallela del pc alla JTAG del 2100AP.

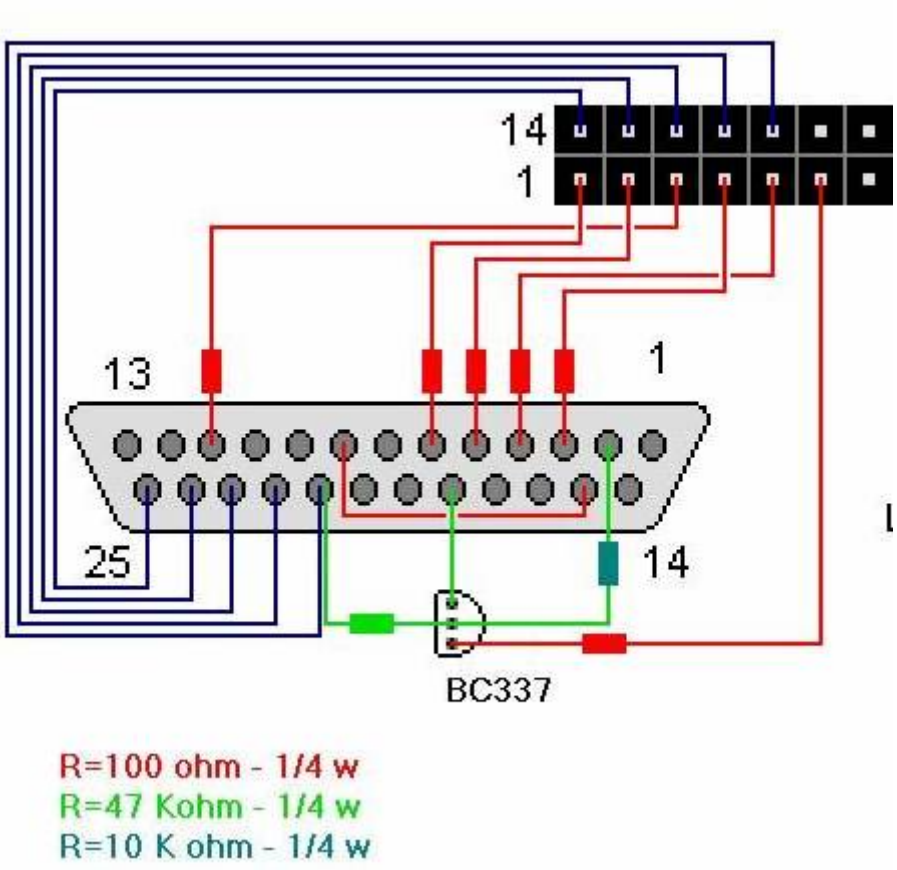
Per interfacciarsi alla JTAG del device è necessario costruirsi un'interfaccia adeguata.

Esistono due tipologie di interfaccie JTAG:

- L'interfaccia "dei poveri" (non bufferizzata)
- L'interfaccia Bufferizzata

Ho trovato diverse versioni sia della prima che della seconda tipologia.

Questa è la prima versione dell'interfaccia "dei poveri"



La sua realizzazione è molto semplice, occorre:

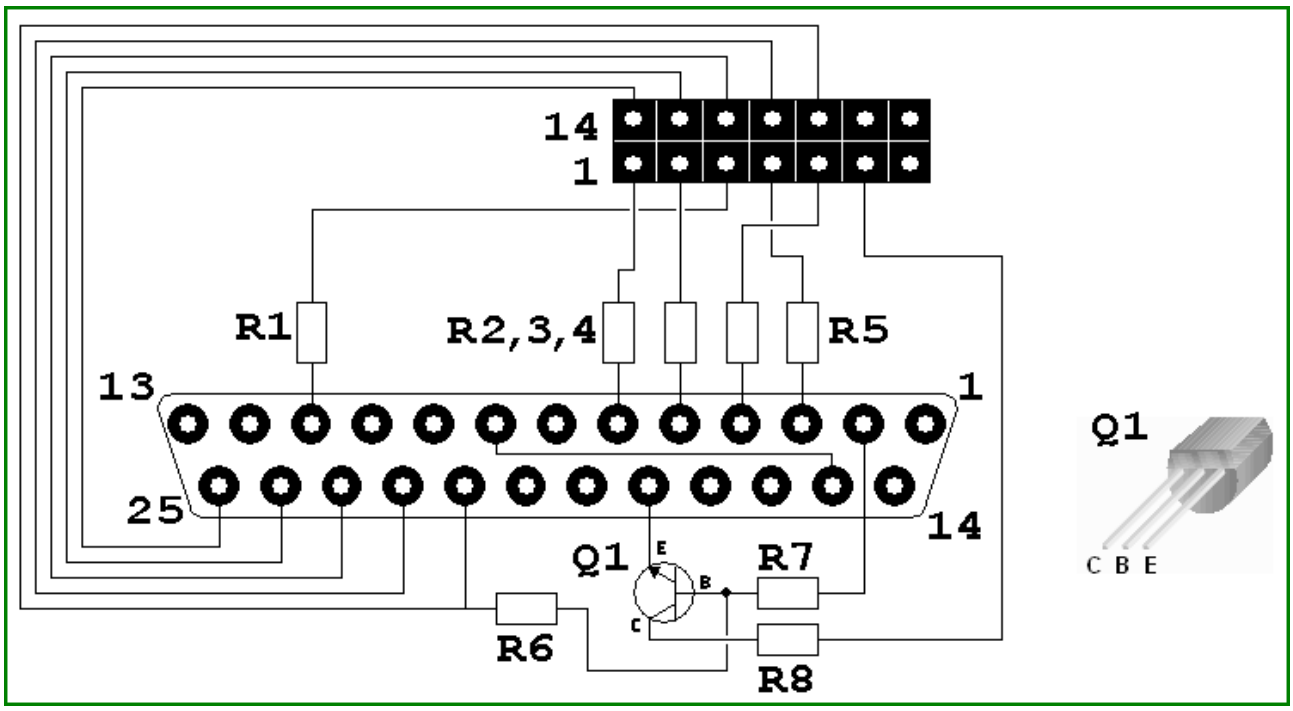
- 6 resistenze da 100 Ohm 1/4w
- 1 resistenza da 47 Kohm 1/4w
- 1 resistenza da 10 Khom 1/4w
- 1 transistor BC337
- 1 connettore parallelo maschio da saldare
- 1 piattina IDE

Ho provato a realizzare l'interfaccia seguendo questo schema e ho fatto alcune prove con OCD Commander, ma non sembra funzionare correttamente.

Da informazioni datemi da "Marven", ho saputo che probabilmente quello schema non è corretto per questo tipo di connettore JTAG. Mi ha poi indirizzato verso degli ulteriori schemi che invece dovrebbero essere corretti.

Ho quindi aggiunto alla guida questi schemi realizzati da "Liquidsky".

Questa è lo schema della "Very Poor Man" WIGGLER JTAG by Liquidsky



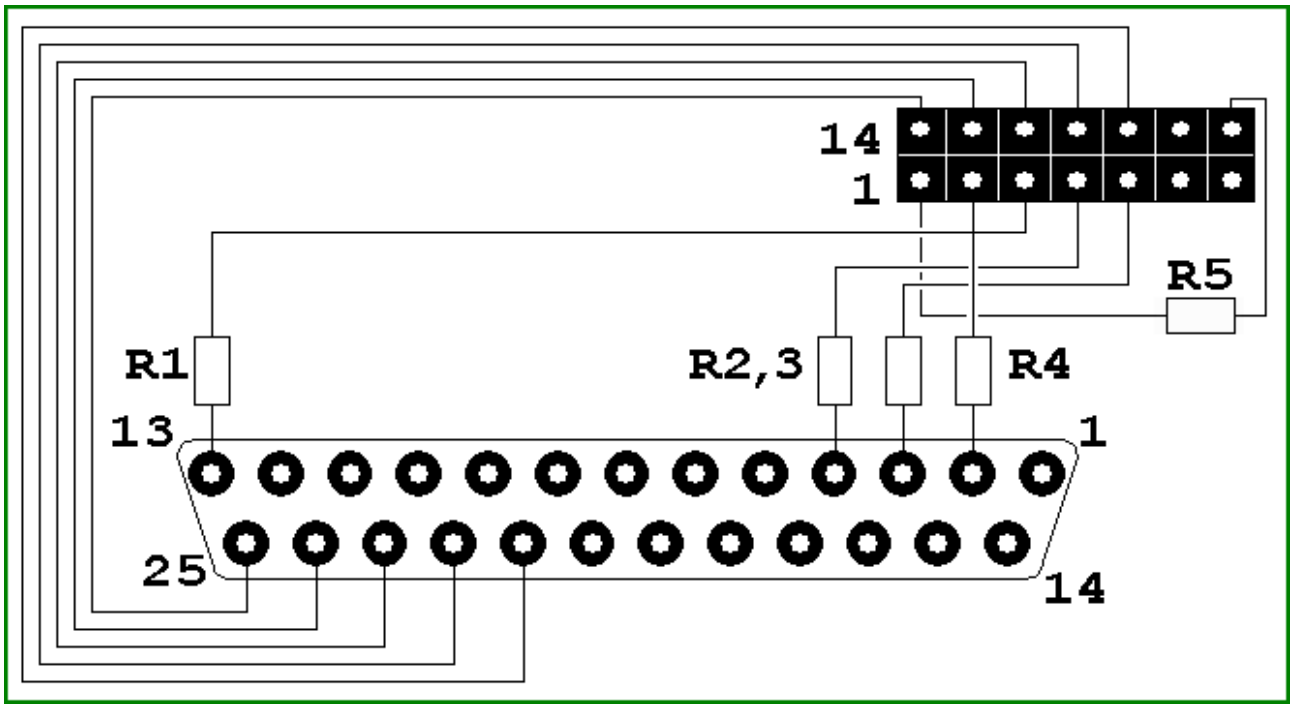
Per la sua realizzazione occorre:

- 6 resistenze da 100 Ohm 1/4w (R1, R2, R3, R4, R5 e R8)
- 1 resistenza da 47 Kohm 1/4w (R6)
- 1 resistenza da 10 Khom 1/4w (R7)
- 1 transistor BC337 (Q1)
- 1 connettore parallelo maschio da saldare
- 1 piattina IDE

Come vedete la componentistica è la stessa dello schema precedente ma le connessioni sono differenti.

Oltre alla versione "Very Poor Man" JTAG, sul sito di “Liquidsky”, vi è anche questa denominata “Xilinx”.

La sua realizzazione è, come potete vedere dallo schema, estremamente semplice

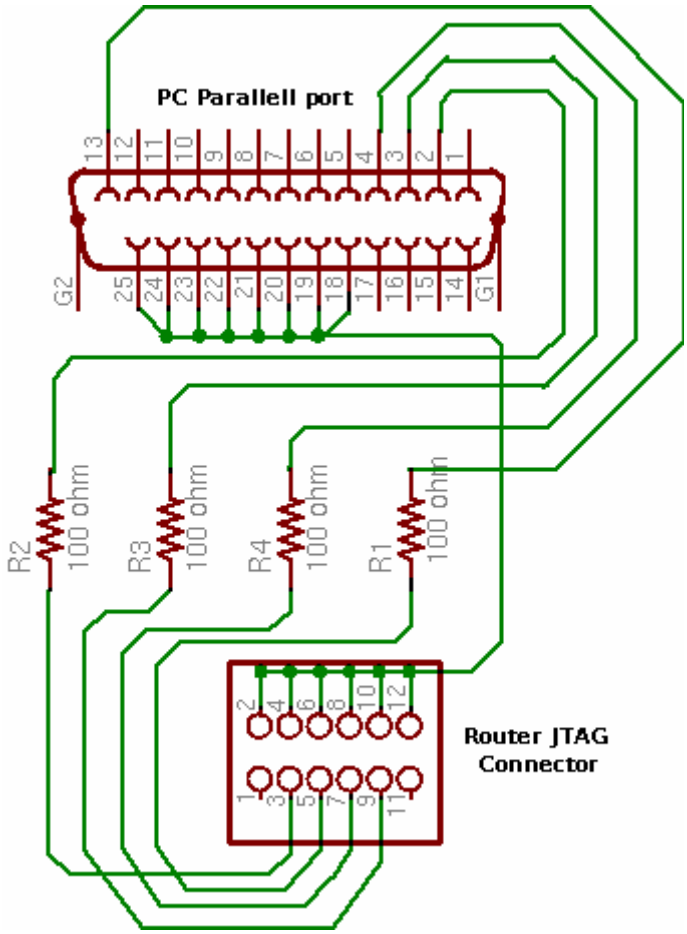


Per la sua realizzazione occorrono soltanto:

- 5 resistenze da 100 Ohm 1/4w (R1, R2, R3, R4, R5)
- 1 connettore parallelo maschio da saldare
- 1 piattina IDE

Consiglio tutti coloro che non dispongono di molta pratica col saldatore di partire con la realizzazione di questa in quanto presenta difficolta davvero minime.

Ho trovato anche uno schema alternativo per l'interfaccia a basso costo.
Si tratta dello schema "Xilinx DLC5 JTAG Parallel Cable III"



Per la sua realizzazione occorrono:

- 4 resistenze da 100 Ohm 1/4w
- 1 connettore parallelo maschio da saldare
- 1 piattina IDE

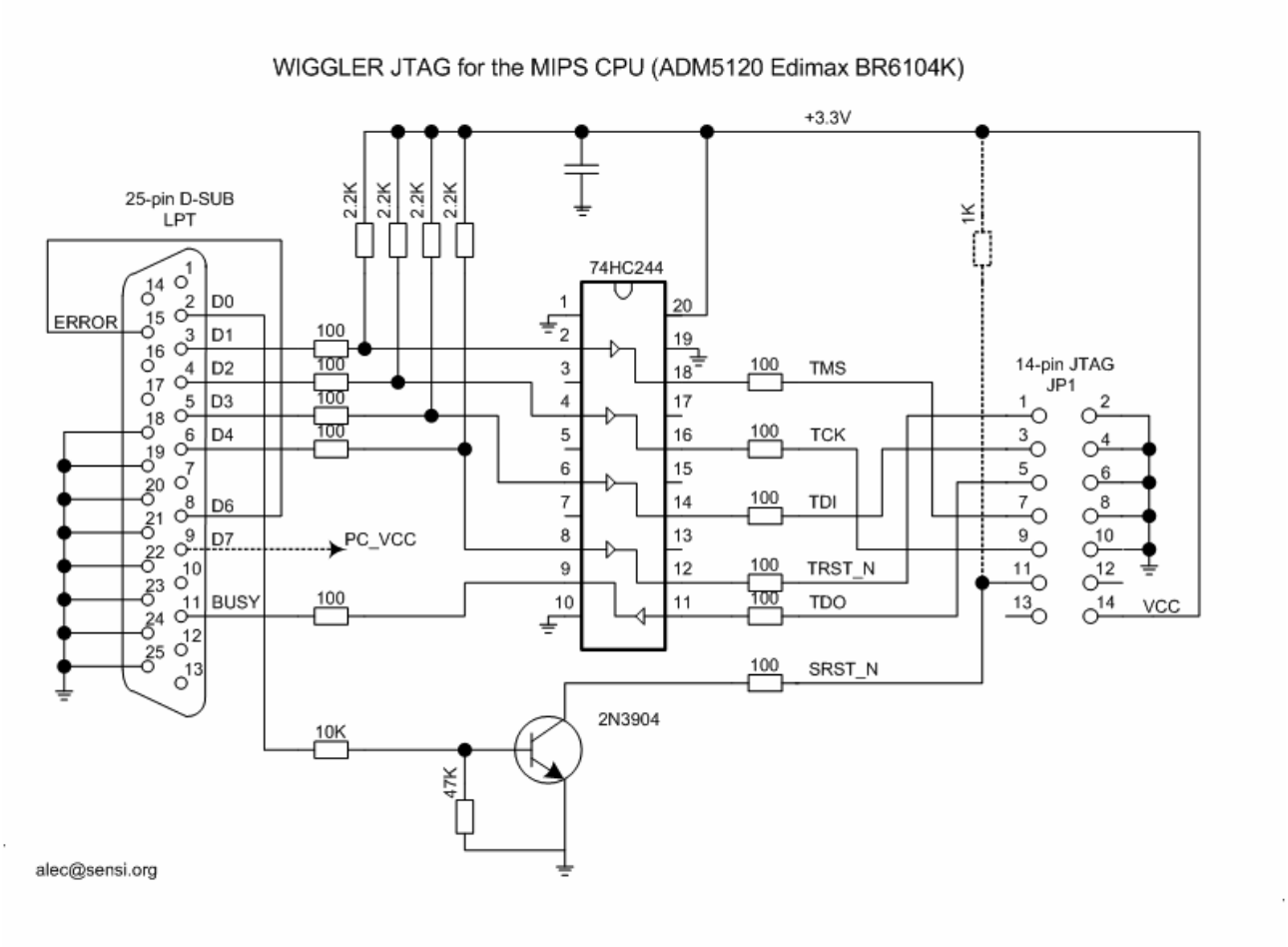
Vanno collegati i pin 2, 3, 4 e 13 sulla parallela ai pin 3, 5, 7 e 9 sul connettore JTAG tramite le resistenze. Vanno poi cortocircuitati i pin dal 17 al 25 sulla parallela coi pin PARI sul connettore JTAG.

È invece possibile realizzare un'interfaccia bufferizzata la cui realizzazione è si più complicata, ma le cui performance sono superiori in termini di trasferimento.

Questa è interfaccia bufferizzata "WIGGLER"

Quest'interfaccia è basata sul progetto originale della Macraigor Systems che vende la sua interfaccia bella e pronta alla non modica cifra di 150$.

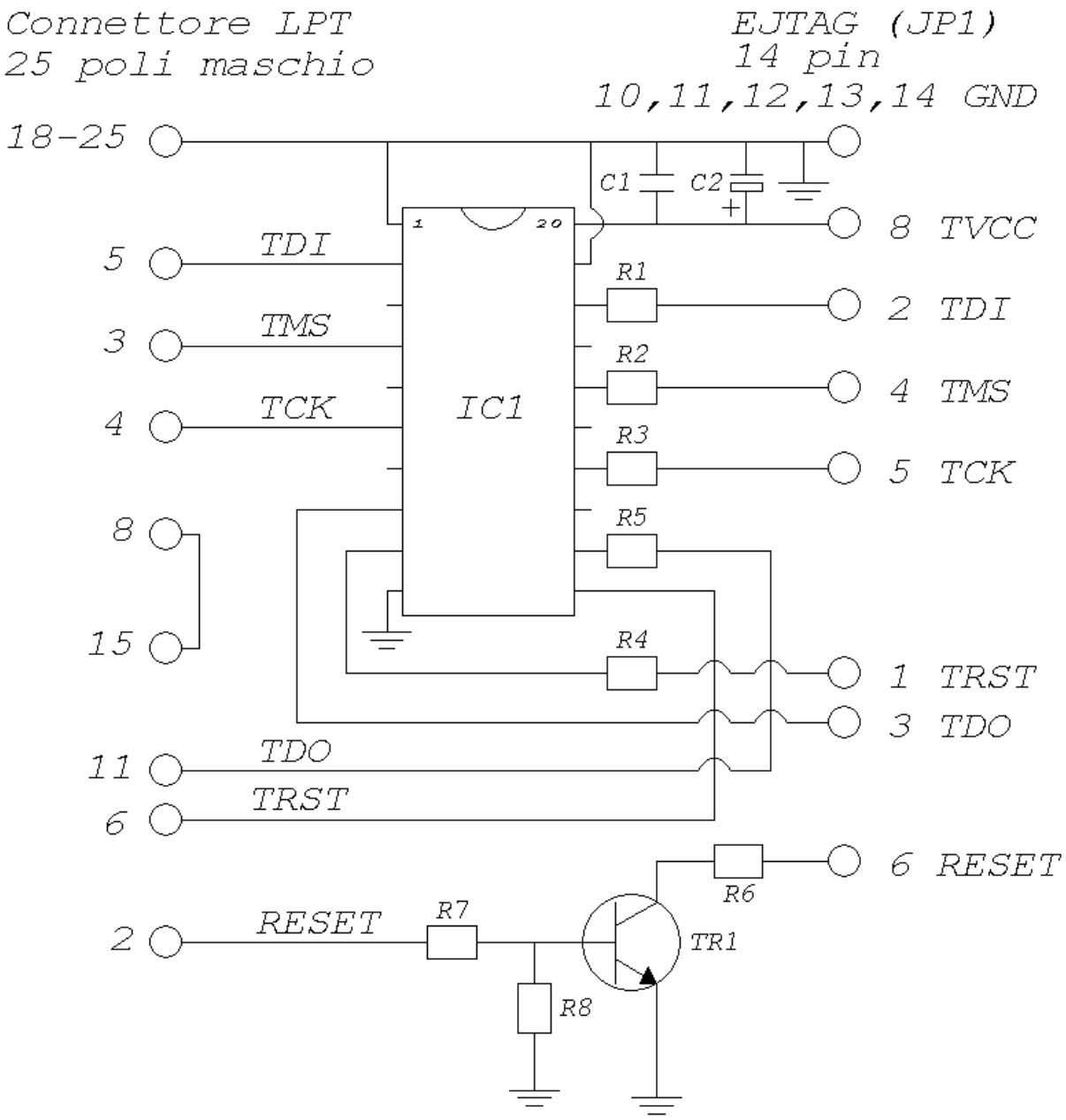
Come si intuisce a prima vista questo tipo di interfaccia è molto più complicata da realizzare, ma disponendo del chip integrato 74HC244 il dialogo tra device a pc è molto meno difficoltoso.



Per la sua realizzazione occorrono:

- 11 resistenze da 100 Ohm 1/4w
- 1 resistenza da 47 Kohm 1/4w
- 1 resistenza da 10 Khom 1/4w
- 1 resistenza da 1 Khom 1/4w
- 4 resistenze da 2.2 Kohm 1/4w
- 1 integrato 74HC244
- 1 transistor BC337
- 1 connettore parallelo maschio da saldare
- 1 piattina IDE

Vi è poi uno schema alternativo per la realizzazione di un'interfaccia bufferizzata.



Per la sua realizzazione occorrono:

- 6 resistenze da 51 Ohm 1/4w (R1, R2... R6)
- 1 condensatore da 4,7 uF elettrolitico (C2)
- 1 condensatore da 200 nF ceramico (C1)
- 1 resistenza da 10 Khom 1/4w (R7)
- 1 resistenza da 47 Kohm 1/4w (R8)
- 1 integrato 74HC244 (IC1)
- 1 transistor BC337 (TR1)
- 1 connettore parallelo maschio da saldare
- 1 piattina IDE

- L'emettitore del BC337 va collegato a massa (GND)
- I piedini 1, 10, 19 dell'integrato vanno collegati a massa
- Le resistenze da 51 ohm se non le trovate mettele del valore che più si avvicina

Tutte le resistenze sono da 1/4W ma, se le trovate, dato che le correnti in gioco sono bassissime, vanno bene anche da 1/8 (potreste mettere il tutto senza basetta dentro il connettore LPT)

Il condensatore da 4,7 uF è un elettrolitico (il - a massa, il + al positivo cioè al piedino 20 dell'IC)

Se l'integrato equivalente che trovate va a 5 volt anzichè a 3.3 niente paura, lasciate inutilizzato il piedino 8 della JTAG e alimentate esternamente il circuito a 5 V (piedino 20 del M74HC244B1).

E' un integrato molto comune, se ne trovate uno equivalente fate riferimento al datasheet per la piedinatura e tenete conto che l'M74HC244B1 accetta un range di alimentazione da 2 a 6 volt quindi si può alimentare con i 3.3 V della JTAG (Pin14).

Ringrazio “Marven” per il suo schema, per l’aiuto datomi nella soluzione dei problemi con la JTAG e per le info attinte su:

http://www.dlinkpedia.net/hardware/interfaccia_JTAG.php

Rigrazio “Liquidsky” per gli schemi e le informazioni prelevate dal suo sito:

http://ciclamab.altervista.org/hard.php?lan=it

Avanzamento dei “Lavori”

Ho provato a realizzare questi schemi:

- L’ interfaccia dei poveri
- “Very Poor Man” WIGGLER by “Liquidsky”
- Xilinx by “Liquidsky”

Dopo una serie di insuccessi con “L’interfaccia dei poveri”, ho provato a realizzare gli schemi by Liquidsky “Very Poor Man” Wiggler e “Xilinx”. Inizialmente la Wiggler non funzionava, mentre la Xilinx mi ha dato le prime risposte.
Purtroppo l’interfaccia Xilinx ha un supporto limitato. Infatti non viene riconosciuta dall’utilissimo OCD Commander, che è un ottimo tool di debug per il controllo via JTAG delle cpu MIPS.
Per cui mi sono rimesso al lavoro sulla Wiggler è ho scoperto che il negoziante mi aveva dato al posto delle resistenze da 100ohm delle resistenze da 100kohm e da li mancato funzionamento dell’interfaccia.
Ho Sostituito le resistenze e ora la mia Wiggler funziona correttamente.

Per leggere e scrivere la flash del 2100AP attraverso l’interfaccia JTAG è necessario un apposito software.

Dopo lungo tempo di ricerca e documentazione ho trovato due software per JTAG in grado di comunicare con il 2100AP:

- Debrick-mod (By “Marven” contenuto nel sofware CICLaMaB per router basati su AR7)
- OPENWINCE Jtag

Entrambi sono software command-line che dispongono di un buon help in linea.

Debrick-mod

Per utilizzare Debrick-mod è sufficiente scaricarsi l'ottimo tool CICLaMaB (sviluppato da LiquidSky) per router basati su AR7.

http://ciclamab.altervista.org/

Una volta installato il tool dovreste trovare nel suo folder il file "debrick-mod.exe"
Tutti i comandi vanno dati a mano dal command prompt.

Comando per il backup

Debrik-mod -backup:custom /skipdetect /instrlen:05 /window:be000000 /start:be000000 /length:4000 /fc:56 /Xilinx

Comando per la scrittura

debrick-mod -flash:custom /skipdetect /instrlen:05 /window:be000000 /start:be000000 /length:20000 /fc:56 /xilinx /f:Bootloader.bin

N.B. nel mio caso ho usato l'opzione /fc:56 visto che il 2100AP in questione usa come flash una Atmel AT49BV322A.

Purtroppo sono riuscito a far funzionare Debrick solo in lettura, mentre in scrittura non ho avuto risultato positivo.

OPENWINCE Jtag

Un altro software col queale interfacciarsi via seriale al 2100AP è OPENWINCE Jtag.

http://openwince.sourceforge.net/jtag/

Che supporta molteplici device, il supporto agli Atheros è possibile trovarlo nel CVS snapshot jtag-0.6-cvs-20051228.

http://www.amelek.gda.pl/rtl8181/jtag/jtag-0.6-cvs-20051228.tar.bz2

Ho in alternativa è possibile utilizzare il pacchetto pronto jtag-brecis-ok

http://star.oai.pp.ru/jtag/jtag-brecis-ok.zip

Io ho ottenuto il risultato voluto con jtag-brecis-ok, modificando alcuni file delle definizioni per “forzare” il riconoscimento della cpu Atheros.

Installazione

Per utilizzare OPENWINCE JTAG sotto Windows XP (scusatemi ma non sono un esperto di linux) è necessario utilizzare Cygwin.

http://www.cygwin.com/

Cygwin è un ambiente simil-linux per Windows. Attraverso la sua interfaccia è possibile eseguire comandi e software dell'ambiente opensource su Windows.
È necessario installare tale pacchetto completo del supporto developer per avere il compilatore C.

Una volta installato è necessario installare l'utility io-perm

http://openwince.sourceforge.net/ioperm/

Ioperm contiene il supporto per l'utilizzo della porta parallela necessaria all'uso della Jtag.

Per installare il pacchetto è sufficiente decomprimere l'archivio, posizionarsi poi nella cartella estratta e dare i seguenti comandi:

./configure (Questo parametro è optionale ma raccomandato --prefix=/usr)
make.
make install.
ioperm -i. (Questo commando attiva il drivers di i/o)

Dopo ioperm è la volta di installare Jtag-brecis-ok.

Come prima scompattate il pacchetto e posizionatevi nella cartella estratta.
A quel punto date questi comandi in sequenza:

./configure (Questo parametro è optionale ma raccomandato --prefix=/usr)
make.
make install.

Dopo aver eseguito questi comandi dovrestre trovarvi nel folder:

{Forder scelto per Cygwin}\usr\local\bin

L'eseguibile jtag.exe

Sostituitelo con quello già presente nel folder decompresso di jtag-brecis-ok.

Se volete automatizzare il riconoscimento del chipset atheros scaricate le definizioni da me modificate.

http://xoomer.alice.it/ramponis/jtag/atheros.rar

Estrete il pacchetto sostituendo le cartelle e i file eventualmente presenti nel folder:

{Forder scelto per Cygwin}\usr\local\share\jtag\atheros

È necessario sostituire anche il file MANUFACTURERS che si trova in

{Forder scelto per Cygwin}\usr\local\share\jtag

Con quello da me modificato

http://xoomer.alice.it/ramponis/jtag/MANUFACTURERS.rar

Se avete eseguito tutto a puntino ora avete la vostra copia di OPENWINCE JTAG funzionante e pronta a dialogare col 2100AP.

N.B. è importante che la parallela sia impostata in EPP o in alternativa in ECP per garantire un corretto funzionamento. Sembra poi (dai miei test) che una CPU di buon livello garantisca maggior stabilità in scrittura di CPU più datate

Con un Pentium III ho rilevato errori sporadici di comunicazione non rilevati su di un pc con cpu Pentium IV.

A questo punto è possibile lanciare il comando JTAG ed entrare nel set dei comandi di Openwince JTAG.

Ho inoltre preparato degli script pronti per configurare in automatico l'interfaccia JTAG:

http://xoomer.alice.it/ramponis/jtag/script.rar

Questi script vanno estratti ec opiati in:

{Forder scelto per Cygwin}\usr\local\share\jtag

Lanciando lo script AthsW si configure la jtag per l'utilizzo dell'interfaccia WIGGLER, mentre lo script AthsX configure la jtag per l'utilizzo dell'interfaccia Xlinx.

Esempio:

Include AthsX

N.B. questi script sono scritti per una parallela il cui indirizzo di input/output sia 0378

Se l’interfaccia è stata realizzata correttamente, dovreste ottenere un’output come il sequente:

```
jtag> include athsx
Initializing Xilinx DLC5 JTAG Parallel Cable III on parallel port
at 0x378
IR length: 5
Chain length: 1
Device Id: 00000000000000000000000000000001
  Manufacturer: Atheros
  Part:         ar2312
  Stepping:     1
  Filename:     /usr/local/share/jtag/atheros/ar2312/ar2312
ImpCode=01000000010000000100000000000000
EJTAG version: 2.6
EJTAG Implementation flags: R4k ASID_8 NoDMA MIPS32
dev ID=00c8   man ID=001f
Found Atmel AT49xV322x flash,  size = 4194304 bytes.
Query identification string:
        Primary Algorithm Command Set and Control Interface ID
Code: 0x0002 (AM
/Fujitsu Standard Command Set)
        Alternate Algorithm Command Set and Control Interface ID
Code: 0x0000 (
ull)
Query system interface information:
        Vcc Logic Supply Minimum Write/Erase or Write voltage: 0
mV
        Vcc Logic Supply Maximum Write/Erase or Write voltage: 0
mV
        Vpp [Programming] Supply Minimum Write/Erase voltage: 0 mV
        Vpp [Programming] Supply Maximum Write/Erase voltage: 0 mV
        Typical timeout per single byte/word program: 0 us
        Typical timeout for maximum-size multi-byte program: 0 us
        Typical timeout per individual block erase: 0 ms
        Typical timeout for full chip erase: 0 ms
        Maximum timeout for byte/word program: 0 us
        Maximum timeout for multi-byte program: 0 us
        Maximum timeout per individual block erase: 0 ms
        Maximum timeout for chip erase: 0 ms
Device geometry definition:
        Device Size: 4194304 B (4096 KiB, 4 MiB)
        Flash Device Interface Code description: 0x0001 (x16)
        Maximum number of bytes in multi-byte program: 0
        Number of Erase Block Regions within device: 2
        Erase Block Region Information:
                Region 0:
                        Erase Block Size: 8192 B (8 KiB)
                        Number of Erase Blocks: 8
                Region 1:
                        Erase Block Size: 65536 B (64 KiB)
                        Number of Erase Blocks: 63
 No. Manufacturer              Part                Stepping
Instruction
  Register
------------------------------------------------------------------
-------------
```

```
-------------
   0 Atheros                    ar2312              1
EJTAG_DATA
  EJDATA

Active bus:
*0: EJTAG compatible bus driver via PrAcc (JTAG part No. 0)
        start: 0x00000000, length: 0x20000000, data width: 8 bit
        start: 0x20000000, length: 0x20000000, data width: 16 bit
        start: 0x40000000, length: 0x20000000, data width: 32 bit
jtag>
```

A questo punto il 2100AP è pronto per la lettura/scrittura della flash

Comandi di lettura/scrittura

Dopo aver lanciato gli script di configurazione (e non aver ottenuto errori) si può procedere alla lettura o alla scrittura.

Sintassi comando di lettura:

readmem [indirizzo iniziale] [lunghezza] [nome del file]

Esempio:

```
readmem 0x1fc00000 0x400000 fullflash.bin
```

Questo comando esegue la lettura dell'intera flash del 2100AP salvandola nel file "fullflash.bin"

Sintassi comando di scrittura:

flashmem [indirizzo iniziale] [nome del file]

Esempio:

```
flashmem 0x1fc00000 fullflash.bin
```

Questo comando scrive il file "fullflash.bin" nella flash partendo dall'indirizzo 0x1fc00000

Tabella degli indirizzi del 2100AP

Descrizione	Inizio	Fine	Lunghezza
RAM	0x80010000		
FLASH	0xbfc00000	0xC0000000	0x400000
BOOTLOADER	0x1fc00000	0x1fc41690	0x41690
FILESYSTEM	0x1fc50000	0x1ff50000	0x300000

Sul mio sito potete trovare alcuni file già pronti per il lavoro con la JTAG:

Bootloader (256K)

http://xoomer.alice.it/ramponis/firmware/Bootloader.rar

Full flash per il recover dei 2100AP “morti” (4Mb)

http://xoomer.alice.it/ramponis/firmware/fullflash.rar

RedBoot (bootloader per linux sul 2100AP)

http://xoomer.alice.it/ramponis/firmware/redboot.z

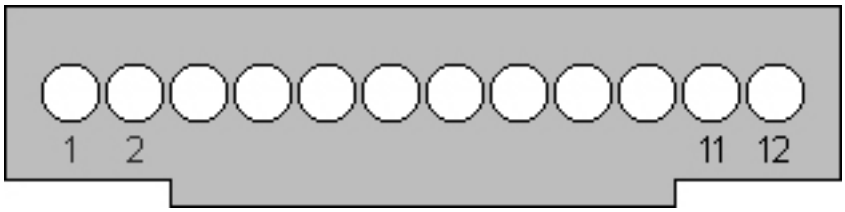
9. Interfacciarsi al Boot-loader del Dlink DWL-2100AP

Per interfacciasi alla seriale ho costruito un cavetto in grado di traslare i segnali da 5V a 3,3V.
Per realizzarlo mi sono procurato un cavo dati per telefoni cellulari Siemens C/M/S 25/35/45.
Il cavo S25, oltre ad essere estremamente economico, racchiude al suo interno un integrato analogo al Maxim max3232 in grado di assolvere a questo compito.

A questo punto è sufficiente aprire il connettore che andrebbe collegato al cellulare …

Al suo interno troviamo tre cavetti colorati (nero, verde e giallo) e un cavetto rosso non collegato.

Pin 1 - cavetto NERO
Pin 2 - NC
Pin 3 - NC
Pin 4 - NC
Pin 5 - cavetto VERDE
Pin 6 - cavetto giallo
Pin 7 - NC
Pin 8 - NC
Pin 9 - NC
Pin 10 - NC
Pin 11 - NC
Pin 12 - NC



Pin 1 - GND
Pin 5 - TX
Pin 6 - RX

Ovviamente nell’effettuare il collegamento vanno invertiti i TX e RX tra seriale e la board del 2100AP

Ho utilizzato l'ottimo terminale seriale FREE tera term pro, scaricabile a questo indirizzo:

http://hp.vector.co.jp/authors/VA002416/ttermp23.zip

Per i miei test ho utilizzato un 2100AP che presentava un problema di reboot continuo.
Nonostante la pressione del reset l'Access Point non ne voleva sapere di funzionare e risultava completamente inaccessibile sia via ethernet che via wi-fi.
Vista la sua situazione "disperata" ho operato a cuor sereno il 2100AP saldandovi in maniera molto rudimentale i pin per la seriale e la jtag. Nonostante i pochi strumenti a disposizione i pin hanno funzionato benissimo e la seriale funziona perfettamente (parametri 9600 8 N 1).

Una volta interfacciatomi al 2100AP ho fatto la "conoscenza" del boot-loader del 2100AP:

```
ar531x rev 0x00005850 firmware startup...
SDRAM TEST...PASSED


  WAP-G02A  Boot Procedure                          V1.0
-------------------------------------------------------
  Start ..Boot.B14..

Atheros AR5001AP default version 3.0.0.43A


 0
auto-booting...

Attaching to TFFS... done.
Loading /fl/APIMG1...

  Please wait, loading  image ...

  image check fail!!!


error loading file: status = 0x3d0001.

Error loading RUNTIME file: errno = 0xd0003.
Loading /fl/backup...

  Please wait, loading  image ...

  image check fail!!!


error loading file: status = 0xd0003.

Error loading BACKUP file: errno = 0xd0003.
Can't load boot file!!
```

Da questo ho potuto verificare che il boot-loader esegue una breve verifica della ram e poi parte con la routine di caricamento del firmware.
Viene verificato il file APIMG1 e qual'ora corretto viene caricato. Nel mio caso tale file doveva essere corrotto, e a questo punto il boot-loader tenta la stessa procedura col firmware di BACKUP, ma anche qui con risultato negativo. A questo punto riavvia l'AP generando un loop continuo.

Tramite seriale è possibile interrompere il boot premendo il tasto ESC

```
ar531x rev 0x00005850 firmware startup...
SDRAM TEST...PASSED


  WAP-G02A  Boot Procedure                          V1.0
-------------------------------------------------------
  Start ..Boot.B14..

Atheros AR5001AP default version 3.0.0.43A


 1 (in questo momento va premuto ESC)

[Boot]:
```

Una volta interrotto il bootloader è possibile accedere sia alla configurazione del bootloader che inviare alcuni comandi.
Digitando il comando **?** il boot-loader restituisce questa lista

```
[Boot]: ?

 ?                      - print this .                       - boot
(load and go)
 p                      - print boot params
 c                      - change boot params
 e                      - print fatal exception
 v                      - print version
 B                      - change board data
 S                      - show board data
 n netif                - print network interface device address
 $dev(0,procnum)host:/file h=# e=# b=# g=# u=usr [pw=passwd] f=#
                            tn=targetname s=script o=other
 boot device: tffs=drive,removable     file name: /tffs0/vxWorks
 Boot flags:
   0x02  - load local system symbols
   0x04  - don't autoboot
   0x08  - quick autoboot (no countdown)
   0x20  - disable login security
   0x40  - use bootp to get boot parameters
   0x80  - use tftp to get boot image
   0x100 - use proxy arp

available boot devices:Enhanced Network Devices
 ae1 tffs
[Boot]:
```

Dando il comando “p” viene visualizzato il riepilogo dei parametri di boot

```
[Boot]: p

boot device           : tffs:
unit number           : 0
processor number      : 0
file name             : /fl/APIMG1
inet on ethernet (e)  : 192.168.1.20:0xffffff00
flags (f)             : 0x0
other (o)             : ae
```

È possibile modificare i parametri di boot indirizzando il 2100AP a eseguire il boot tramite un server TFTP. Un ottimo TFTP server FREE è scaricabile a questo indirizzo
http://tftpd32.jounin.net/
A tale scopo è necessario dare il comando “c”

```
[Boot]: c

'.' = clear field;  '-' = go to previous field;  ^D = quit

boot device           : tffs:0
processor number      : 0
host name             :
file name             : /fl/APIMG1
inet on ethernet (e)  : 192.168.1.20:0xffffff00
inet on backplane (b):
host inet (h)         :
gateway inet (g)      :
user (u)              :
ftp password (pw) (blank = use rsh):
flags (f)             : 0x0
target name (tn)      :
startup script (s)    :
other (o)             : ae

[Boot]:
```

Il boot device può essere tffs (la flash) o ae1 (interfaccia ethernet).
File name specifica il path+nome del file del firmware da caricare.
Inet on ethernet (e) specifica l’ip che il boot loader deve utilizzare per l’ethernet, mentre host inet è l’ip del nostro pc col server TFTP e il flag 0x80 specifica di usare il server TFTP per il boot.

Questo esempio è una configurazione per fare un boot linux tramite un server TFTP

```
boot device : ae1
processor number : 0
host name :
file name : /linux/vmlinux
inet on ethernet (e) : 10.1.1.6:ffffff00
inet on backplane (b):
host inet (h) : 10.1.1.5
gateway inet (g) :
user (u) :
ftp password (pw) (blank = use rsh):
flags (f) : 0x80
target name (tn) :
startup script (s) :
other (o) : ae
```

10. VxWorks e file ELF

Ho scoperto poi che il boot-loader del 2100AP attende un file ELF per il caricamento del firmware. Per approfondire le vostre conoscenze a rigurado vi rimando a questo link:

http://it.wikipedia.org/wiki/Executable_and_linkable_format

Un ELF file è presente nella flash del 2100AP, tale file e nominato ART.
Collegandosi via telnet ad un 2100AP funzionante e dando il comando ls è possibile vedere che tale file viene elencato tra quelli presenti nella flash dell'AP. È quindi possibile recuperare tale file mettendo sul proprio pc un server ftp (es. Filezilla server) e digitando questi comandi:

Telnet (ip del proprio 2100AP)

D-link Corp. Access Point login: admin
Password: ********

Atheros Access Point Rev 3.0.0.43A
D-link Corp. Access Point wlan1 -> alpha
Password: sdd21234
Ok
D-link Corp. Access Point wlan1 -> superftp (ip del proprio pc)
Username: anonymous (lo user definito in filezilla)
Password: ********** (la pwd definita in filezilla)
Remote File: art
Local File: art
download or upload: upload
Putting art -> anonymous@192.168.0.1:art
##
##
##
##
##
##
##
##
##
##
done
639252 bytes
D-link Corp. Access Point wlan1 ->

A questo punto abbiamo sul pc un ELF file "buono" da poter caricare sul 2100AP.
Per far eseguire tale file al boot-loader è sufficiente dare questo comando:

$ae(1,0)[nome pc che fa TFTP]:[nome del file ELF] h=[Ip pc con TFTP] e=[IP da assegnare al 2100AP]:0xffffff00 [questa è la subnet in HEX e corrisponde a 255.255.255.0] f=0x80 [questo flasg specifica il boot tramite TFTP server]

Esempio:

$ae(1,0)PC:art h=192.168.1.1 e=192.168.1.20:0xffffff00 f=0x80

Eseguendolo otteniamo il caricamento del S.O. su cui è basto il 2100AP: VxWorks

```
ar531x rev 0x00005850 firmware startup...
SDRAM TEST...PASSED



  WAP-G02A  Boot Procedure                      V1.0
---------------------------------------------------------
  Start ..Boot.B14..

theros AR5001AP default version 3.0.0.43A


 1
[Boot]: $ae(1,0)PC:art h=192.168.0.3 e=192.168.0.5:0xffffff00 f=0x80
Attached TCP/IP interface to ae1.
Attaching network interface lo0... done.
Loading... 19032 + 618848 + 105600
Starting at 0x80480000...

Failed to attach to device tffs:Attaching interface lo0...done
List of bridge ports not specified! -
Please set the bridge configuration parameter BRIDGE_PORTS.
This must a list of comma-separated device name and unit number pairs.
For example: "fei0,fei1,elPci0"
portsStringParse failed!

Adding 3836 symbols for standalone.


 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
      ]]]]]]]]]]]  ]]]]     ]]]]]]]]]]       ]]              ]]]]
(R)
 ]     ]]]]]]]]]  ]]]]]]     ]]]]]]]]       ]]               ]]]]
0x
80fffdf0  (]]     ]]]]]]]  ]]]]]]]]     ]]]]]] ]     ]]
]]]]
       tRootTask
):  Can't attach unknown device ]]]     ]]]]] ]    ]]]  ]     ]]]] ]]]
]]]]]]]
]]  ]]]]tffs: ]] ]]]]  ]]   ]]]]] (unit
0 ).
]]]]     ]]]  ]]    ]  ]]]     ]] ]]]]] ]]]]]]   ]] ]]]]]]] ]]]] ]]
]]]]
 ]]]]]     ]  ]]]]     ]]]]]      ]]]]]]]] ]]]]   ]] ]]]]    ]]]]]]]
]]]]
 ]]]]]]      ]]]]]     ]]]]]]    ]  ]]]]]  ]]]]   ]] ]]]]    ]]]]]]]]
]]]]
 ]]]]]]]    ]]]]]  ]    ]]]]]]  ]    ]]]   ]]]]   ]] ]]]]    ]]]] ]]]]
]]]]
 ]]]]]]]]  ]]]]]  ]]]    ]]]]]]]      ]     ]]]]]]]  ]]]]    ]]]]  ]]]]
]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]       Development System
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]]
 ]]]]]]]]]]]]]]]]]]]]]]]]]]]       VxWorks version VxWorks5.4.2
 ]]]]]]]]]]]]]]]]]]]]]]]]]]       KERNEL: WIND version 2.5
 ]]]]]]]]]]]]]]]]]]]]]]]]]       Copyright Wind River Systems, Inc.,
1984-2001

                              CPU: Atheros AR5001AP default.  Processor
#0.
                             Memory Size: 0x1000000.  BSP version 1.0.
/fl/  - Volume is OK
-> Starting ART Client...
Waiting for connection from Host
```

A questo punto il S.O. è stato caricato ed è pronto ad eseguire i suoi comandi.

Digitando “help” si ottiene la lista.

```
help

help                            Print this list
ioHelp                          Print I/O utilities help info
dbgHelp                         Print debugger help info
nfsHelp                         Print nfs help info
netHelp                         Print network help info
spyHelp                         Print task histogrammer help info
timexHelp                       Print execution timer help info
h           [n]                 Print (or set) shell history
i           [task]              Summary of tasks' TCBs
ti          task                Complete info on TCB for task
sp          adr,args...         Spawn a task, pri=100, opt=0, stk=20000
taskSpawn name,pri,opt,stk,adr,args... Spawn a task
td          task                Delete a task
ts          task                Suspend a task
tr          task                Resume a task
d           [adr[,nunits[,width]]] Display memory
m           adr[,width]         Modify memory
mRegs       [reg[,task]]        Modify a task's registers interactively
pc          [task]              Return task's program counter

Type <CR> to continue, Q<CR> to stop:

iam         "user"[,"passwd"]     Set user name and passwd
whoami                          Print user name
devs                            List devices
ld          [syms[,noAbort][,"name"]] Load stdin, or file, into memory
                                (syms = add symbols to table:
                                -1 = none, 0 = globals, 1 = all)
lkup        ["substr"]          List symbols in system symbol table
lkAddr      address             List symbol table entries near address
checkStack  [task]              List task stack sizes and usage
printErrno  value               Print the name of a status value
period      secs,adr,args... Spawn task to call function periodically
repeat      n,adr,args...    Spawn task to call function n times (0=forever)
version                         Print VxWorks version info, and boot line

NOTE:  Arguments specifying 'task' can be either task ID or name.

value = 1 = 0x1
->
```

Digitando “netHelp” si ottiene un’ulteriore lista dei comandi di rete.

```
-> netHelp


hostAdd   "hostname","inetaddr"    - add a host to remote host table;
                                     "inetaddr" must be in standard
                                     Internet address format e.g. "90.0.0.4"
hostShow                           - print current remote host table
netDevCreate "devname","hostname",protocol
                                   - create an I/O device to access files
                                     on the specified host
                                     (protocol 0=rsh, 1=ftp)
routeAdd     "destaddr","gateaddr" - add route to route table
routeDelete  "destaddr","gateaddr" - delete route from route table

routeShow                          - print current route table
iam          "usr"[,"passwd"]      - specify the user name by which you
                                     will be known to remote hosts
                                     (and optional password)
whoami                             - print the current remote ID
rlogin       "host"                - log in to a remote host;
                                     "host" can be inet address or
                                     host name in remote host table

Type <CR> to continue, Q<CR> to stop:



ifShow       ["ifname"]            - show info about network interfaces
inetstatShow                       - show all Internet protocol sockets
tcpstatShow                        - show statistics for TCP
udpstatShow                        - show statistics for UDP
ipstatShow                         - show statistics for IP
icmpstatShow                       - show statistics for ICMP
arptabShow                         - show a list of known ARP entries
mbufShow                           - show mbuf statistics

EXAMPLE:   -> hostAdd "wrs", "90.0.0.2"
           -> netDevCreate "wrs:", "wrs", 0
           -> iam "fred"
           -> copy <wrs:/etc/passwd   /* copy file from host "wrs" */
           -> rlogin "wrs"            /* rlogin to host "wrs" */

value = 1 = 0x1
->
```

Ripristino dell’access point

Per ripristinare i file corretti sulla flash del 2100AP, ho fatto prima un backup della flash di un 2100AP funzionante tramite telnet. Dando il comando ls si ottiene questa lista:

apcfg
apcfg.bak
apimg1
art
backup
pcode

La procedura per recuperare I file è la stessa descritta sopra utilizzata per recuperare il file ELF chiamato art.

Una volta recuperati tutti i file si può procedere:

- si interrompe il boot-loader
- si fa caricare il file art tramite TFTP server
- si sostituiscono i file della flash con quelli corretti tramite questi comandi:

```
hostAdd   "hostname","inetaddr"
netDevCreate "devname","hostname",protocol
cd "devname"
iam          "usr"[,"passwd"]
copy      ["in"][,"out"]
```

hostname = è il nome del pc sul quale imposteremo un server FTP (filezilla server consigliato)
inetaddr = è l'ip del pc con il server FTP
devname = è il nome che viene assegnato al device di rete mappato sul server FTP
protocol = è il protocollo scelto per il trasferimento (0=rsh 1=ftp)
usr = è il nome utente definito nel server FTP
passwd = e la password relativa all'utente definito nel server FTP
in = path + filename di origine
out = path + finename di destinazione

Esempio

```
hostAdd "pc", "192.168.1.1"
netDevCreate "pc:", "pc", 1
cd "pc:"
iam "Anonymous", ”Guest”
```

Attenzione!!!: I comandi sono case sensitive per cui va rispettato i minuscolo/maiuscolo

Dopo l'invio di ogni comando il 2100AP dovrebbe rispondere con il seguente codice

```
value = 0 = 0x0
```

Se dovesse visualizzarvi un codice diverso (es. `undefined symbol`) significa che avete commesso qualche errore.

Una volta impostato l'host, creato il device di rete e inserita l'autenticazione... suggerisco di dare il comando "ls" in modo da verificare che il 2100AP e il server FTP sul pc dialoghino.

```
ls
```

Se il comando "ls" restituisce la lista corretta dei file presenti nella cartella del server FTP vuol dire che la configurazione è corretta e si può procedere al caricamento dei file nella memoria flash.
Se invece ottenete un codice di errore ricontrollate le operazioni precedenti.

```
copy “pc:/apimg1”, “/fl/apimg1”
copy “pc:/apcfg”, “/fl/apcfg”
copy “pc:/apcfg.bak”, “/fl/apcfg.bak”
copy “pc:/art”, “/fl/art”
copy “pc:/backup”, “/fl/backup”
```

Sintassi corretta: copy "nomepc:/file da copiare", "/fl/nomefile copiato"

Attenzione!!!:
Dopo la virgola va uno spazio
Le virgolette sono importanti

Una volta terminato il ripristino dei file corretti nella flash date il comando "reboot" per far riavviare il 2100AP.

Se tutto è stato fatto correttamente dovreste vedere questo:

```
ar531x rev 0x00005850 firmware startup...
SDRAM TEST...PASSED


  WAP-G02A  Boot Procedure                          V1.0
-------------------------------------------------------------
  Start ..Boot.B14..

Atheros AR5001AP default version 3.0.0.43A


 0
auto-booting...

Attaching to TFFS... done.
Loading /fl/APIMG1...

  Please wait, loading  image ...

  image check ok!!!

/fl/  - Volume is OK
Reading Configuration File "/fl/apcfg".
Configuration file checksum: 5f3a7c is good
Attaching interface lo0...done
wireless access point starting...
wlan1 Ready
vxWorksTftpPackageInit: init. finish & success!
Ready
```

Bene il 2100AP è tornato al “lavoro”.
Visto che ora è possibile ripristinarlo è possibile sperimentare dei firmware open source.

11. Compilare un firmware custom e Openwrt

Nella sezione relativa a Jtag e seriale, nell'immagine relativa alla serigrafia del modello abbiamo visto che la board del 2100AP Rev. A3/A4 è il cuore di molti device tra cui il DWL-2210AP il cui firmware è GPL. Quindi partendo da tali sorgenti dovrebbe essere possibile compilare un firmware perfettamente compatibile con il 2100AP.
I sorgenti sono prelevabili a questo indirizzo.

ftp://ftp.dlink.com/GPL/DWL-2210AP/

È inoltre possibile lavorare sull'AP per istallarci un firmware OPENWRT .

Nello specifico sembra che la versione Kamikaze di OPENWRT supporti pienamente l'hardware del DWL-2100AP.

Per ulteriori informazioni vi rimando a questo link:

http://wiki.openwrt.org/OpenWrtDocs/Hardware/D-Link/DWL-2100AP

Allo stato attuale non ho ancora avuto modo/tempo di compilare un firmware open-source per il 2100AP. Non appena avrò effettuato tale operazione implementerò la guida con gli ulteriori passi avanti.

Questa guida non ha la presunzione di considerarsi esaustiva ed è da considerarsi ancora in "evoluzione". Ogni ulteriore scoperta sarà seguita da una nuova edizione di tale guida.

Desidero ringraziare coloro che mi hanno dato e mi stanno dando il loro supporto in tale progetto:

- Luca alias "Lepro" per il 2100AP offerto come cavia per i test e senza il quale gran parte delle informazioni di questa guida non sarebbero state scoperte.
- Fabio alias "fabiocroc" per l'aiuto nel completare i contenuti della guida e per la condivisione della passione per i 2100AP.
- Mirco alias "mip" per il supporto linuxiano.
- Matteo per il suo importantissimo portale www.wireless-italia.com
- "Marven" per il suo supporto nel lavoro sulla Jtag

Ramponi Stefano