

# 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](http://www.wifi-connect.ru/files/BlueBox_AP2100.zip)

#### 2 Hacking xtender range per link lunghi

#### 3 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.

#### 4 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.

#### 5 Jtag e Seriale sul 2100AP

Sulla board del 2100AP sono presenti le connessioni per una seriale e una Jtag.

#### 6 Interfacciarsi al boot-loader del DWL-2100AP

#### 7 VxWorks e file ELF

#### 8 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.

#### 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
- digitare:

alpha

- inserire la password:

sdd21234 (a me questa funziona benissimo)

Password alternative

vec21234 (questa sarà la password dopo l'aggiornamento)

dlk19283

- rm apimg1  
delrcode  
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:
  - telnet 10.0.0.1
  - tftp srvip XXX.XXX.XXX.XXX (XXX.XXX.XXX.XXX è l'IP dell'AP)
  - 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 alzare 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 riloggarci.

```
set super g disable
reboot
set wireless mode 11g
reboot
set rate 0.25
reboot
set super g enable
reboot
```

A questo punto digitando il comando "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. 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](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 recuperare le informazioni di cui avete bisogno

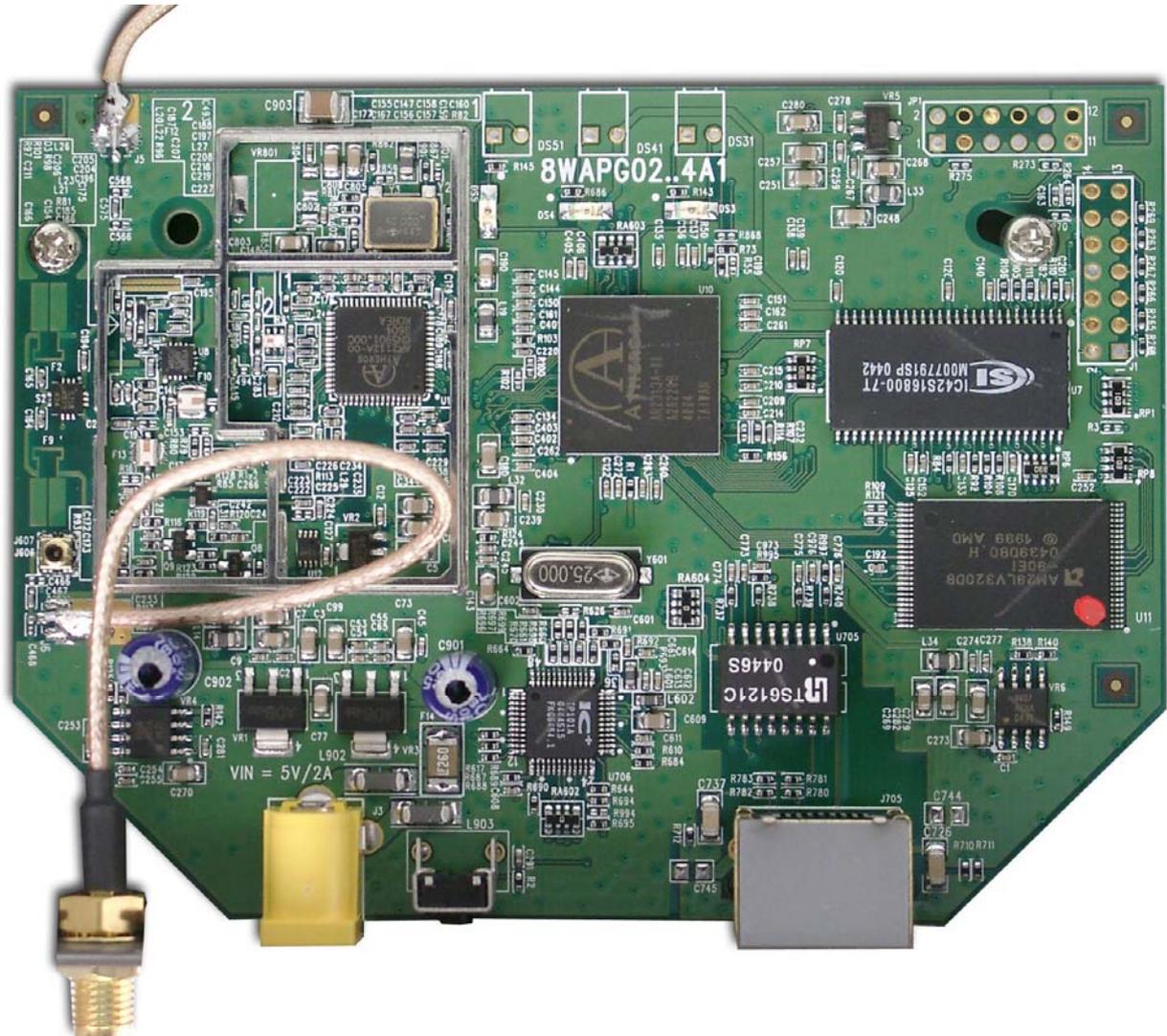
NB ricordo che con tale operazione, qualora sia compiuta su apparati non di nostra proprietà, si compie un reato PENALE.

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

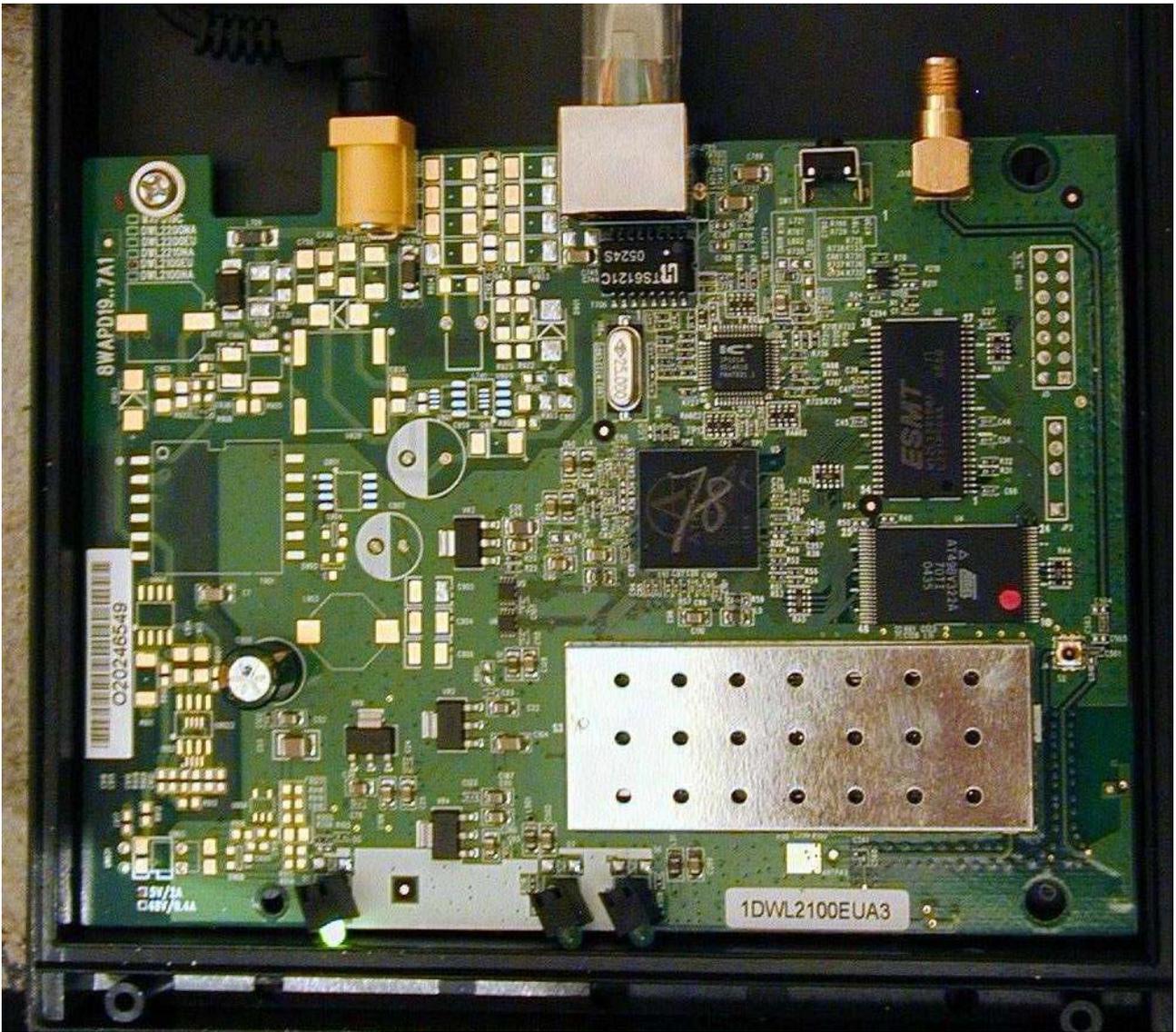
### Premessa:

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

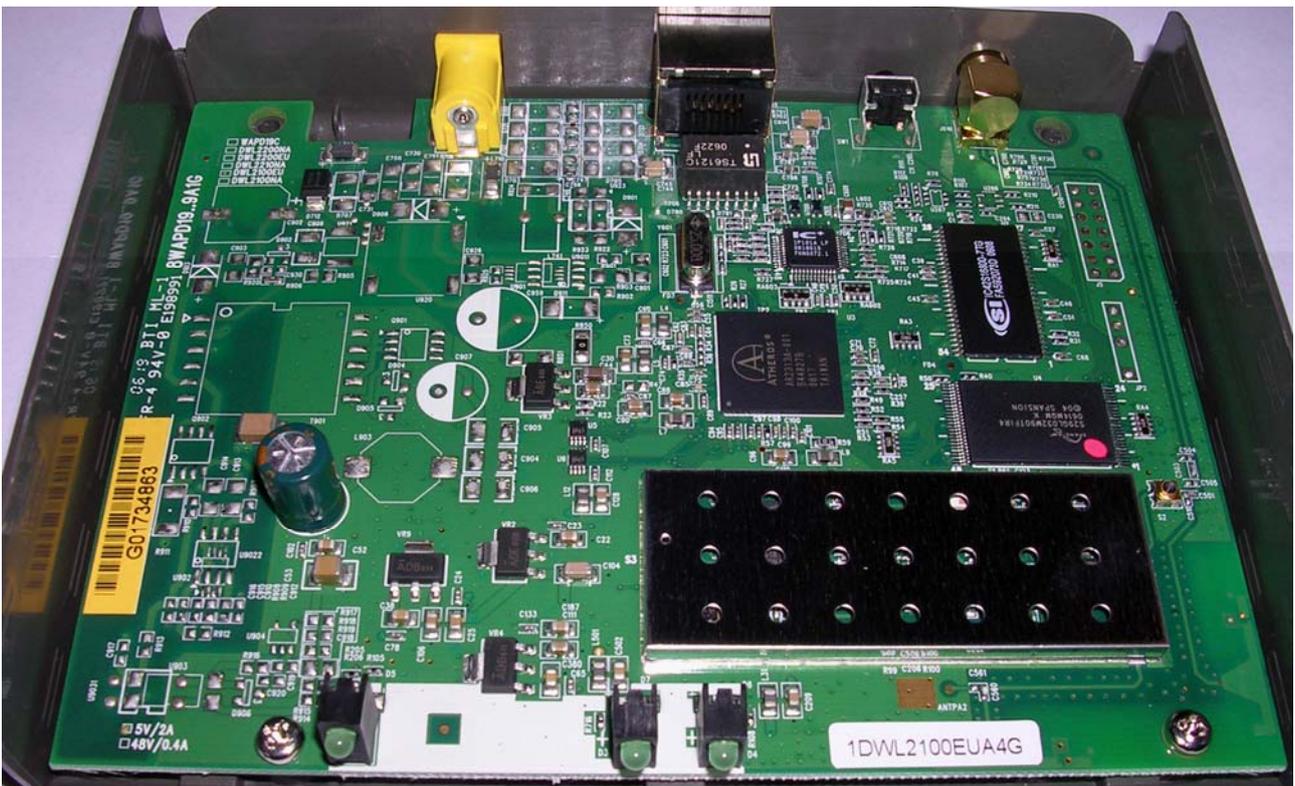
- 8WAPG02..4A1 usata nella Rev. A2
- 8WAPD19..7A1 usata nelle Rev. A3
- 8WAPD19..9A1 usata nelle Rev. A4 (che è quasi identica alla Rev A3)



Rev. A2



Rev. A3



Rev. A4

## Questa modifica riguarda la board 8WAPG02..4A1 della Rev. A2

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

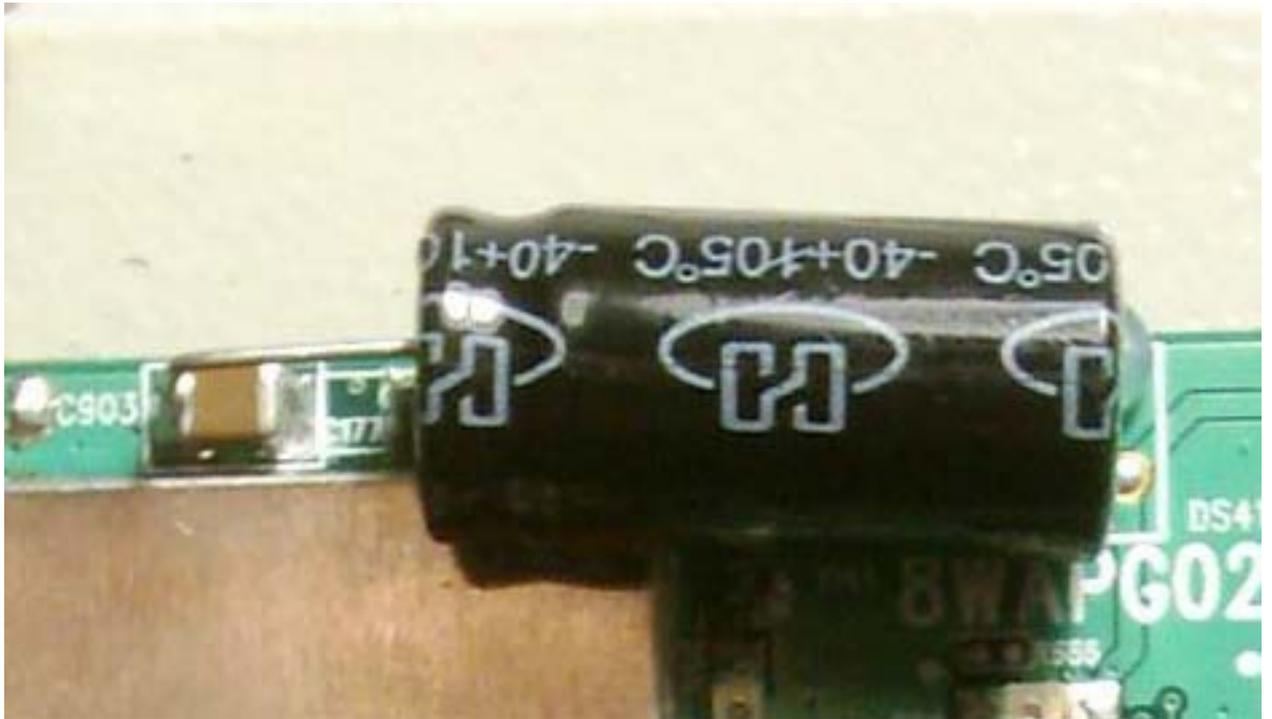
- 2 Condensatori elettrolitici da 10v
- 1 Diodo zener (non sono riuscito a identificarne l'esatta tipologia)
- 1 transistor tipcn42

2 Procediamo con la modifica e le saldature

- Saldiamo un condensatore nel punto C253



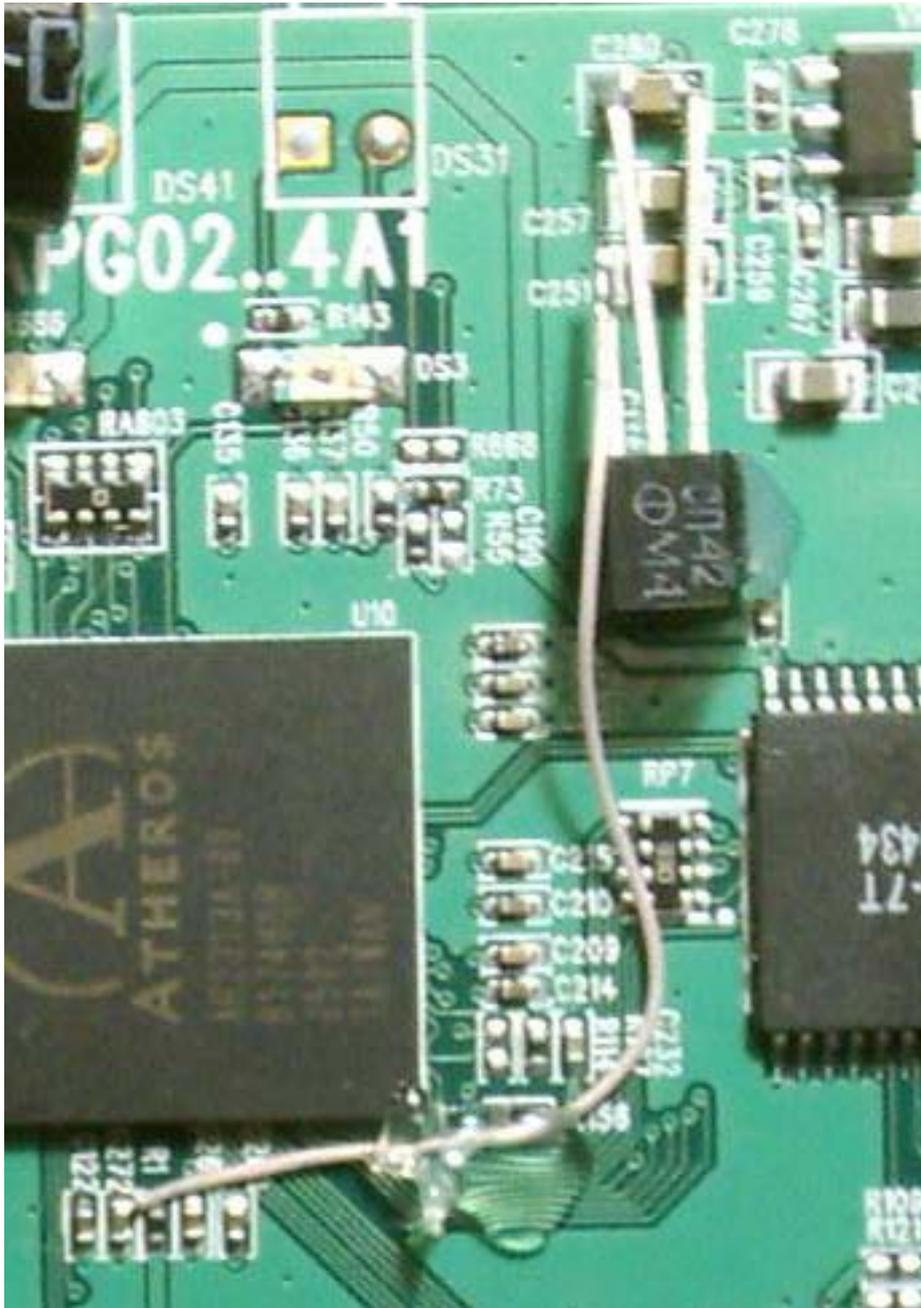
- Saldiamo l'altro condensatore nel punto C903



- 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.





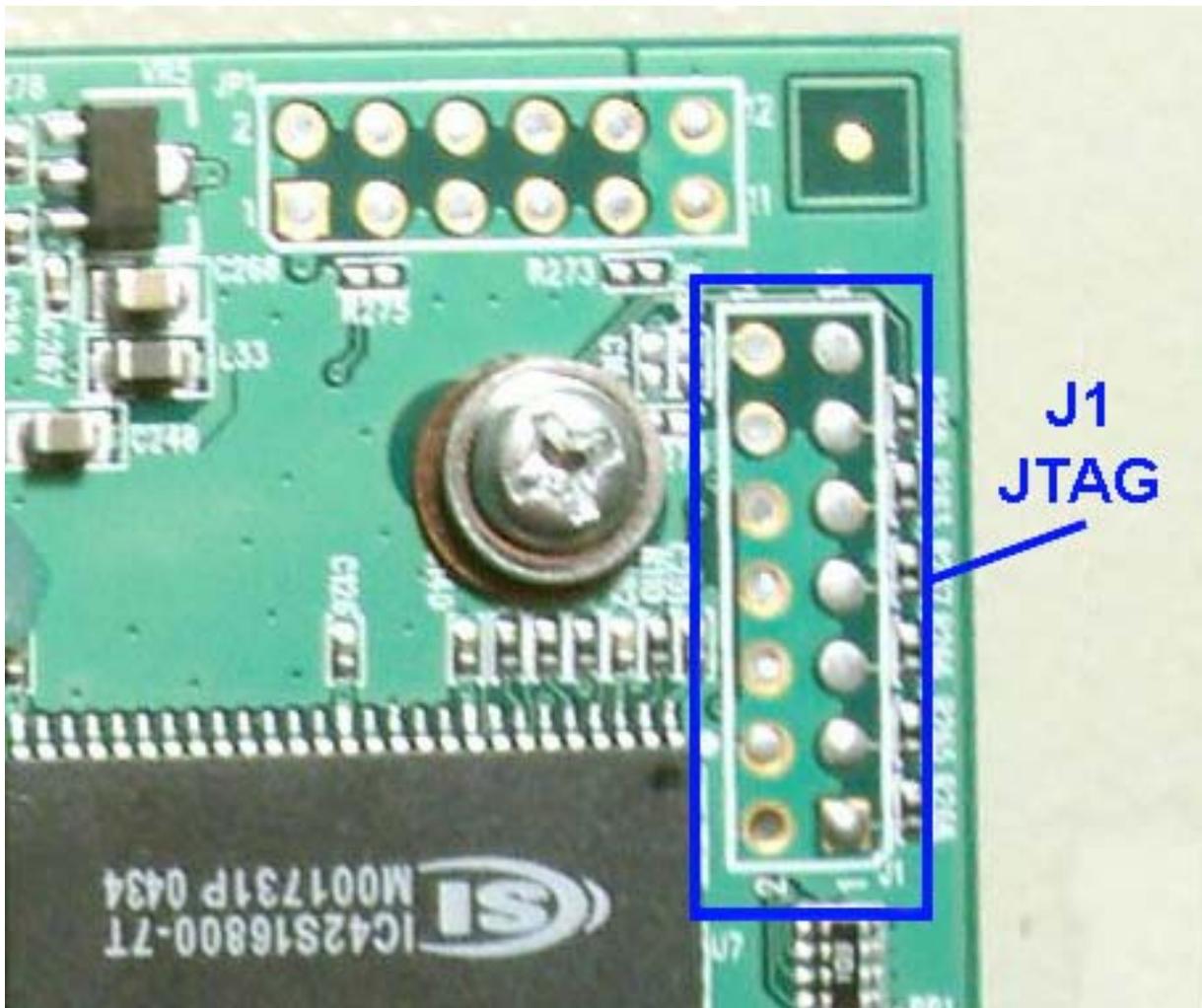
## JTAG e Seriale

In ultimo ho scoperto la presenza, sulla board del 2100AP, della predisposizione per connettori per JTAG e seriale.

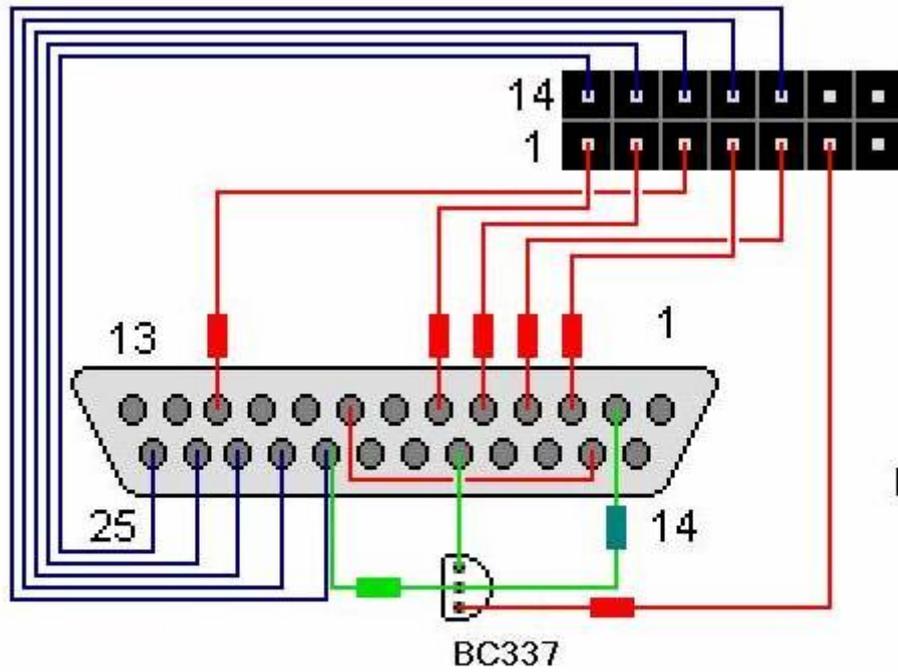
La loro disposizione e forma variano in base alla revision della board del 2100AP

### Rev. A2

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



Per interfacciarsi alla JTAG è sufficiente costruire l'interfaccia "dei poveri"



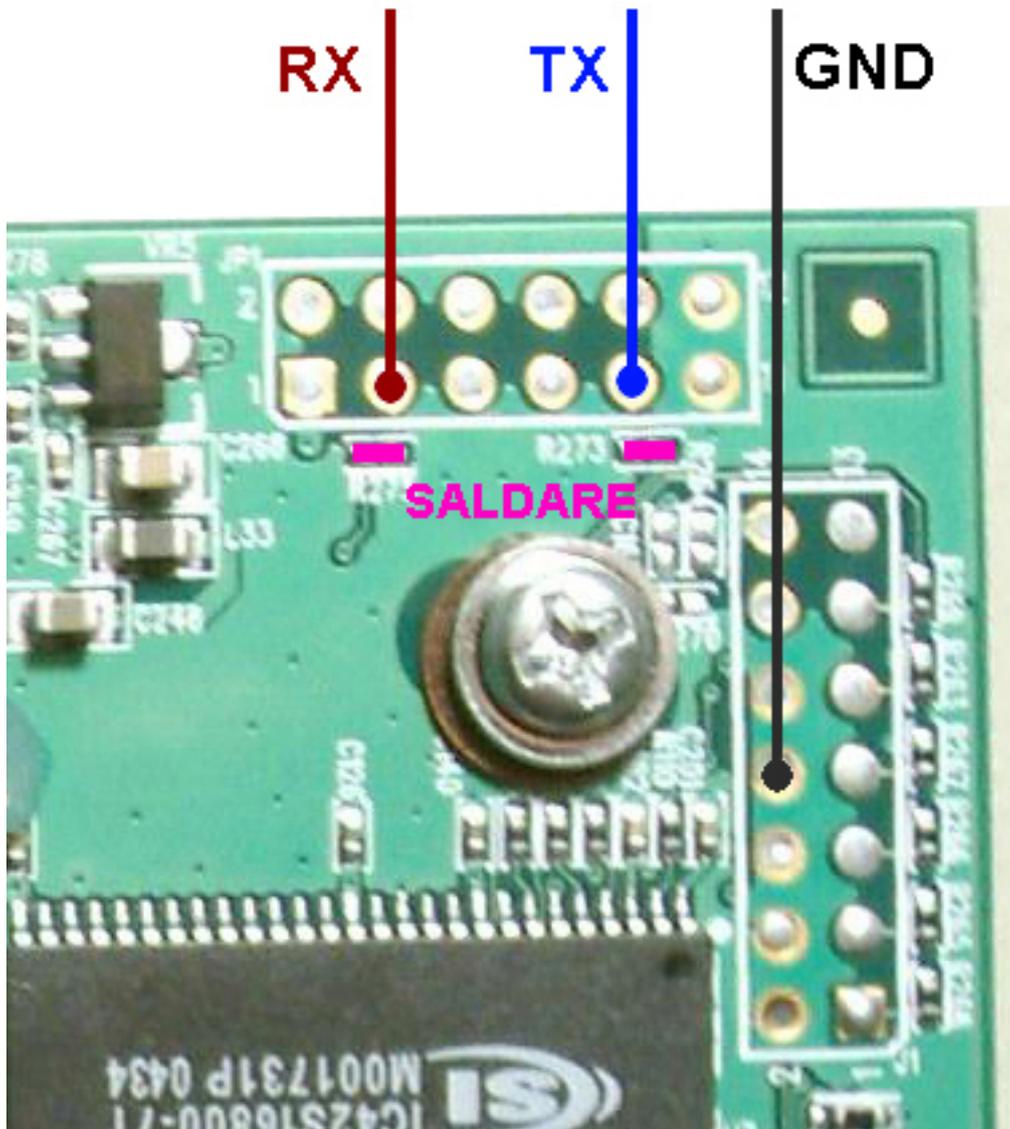
- R=100 ohm - 1/4 w
- R=47 Kohm - 1/4 w
- R=10 Kohm - 1/4 w

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 Kohm 1/4w
- 1 transistor BC337
- 1 connettore parallelo maschio da saldare
- 1 piattina IDE

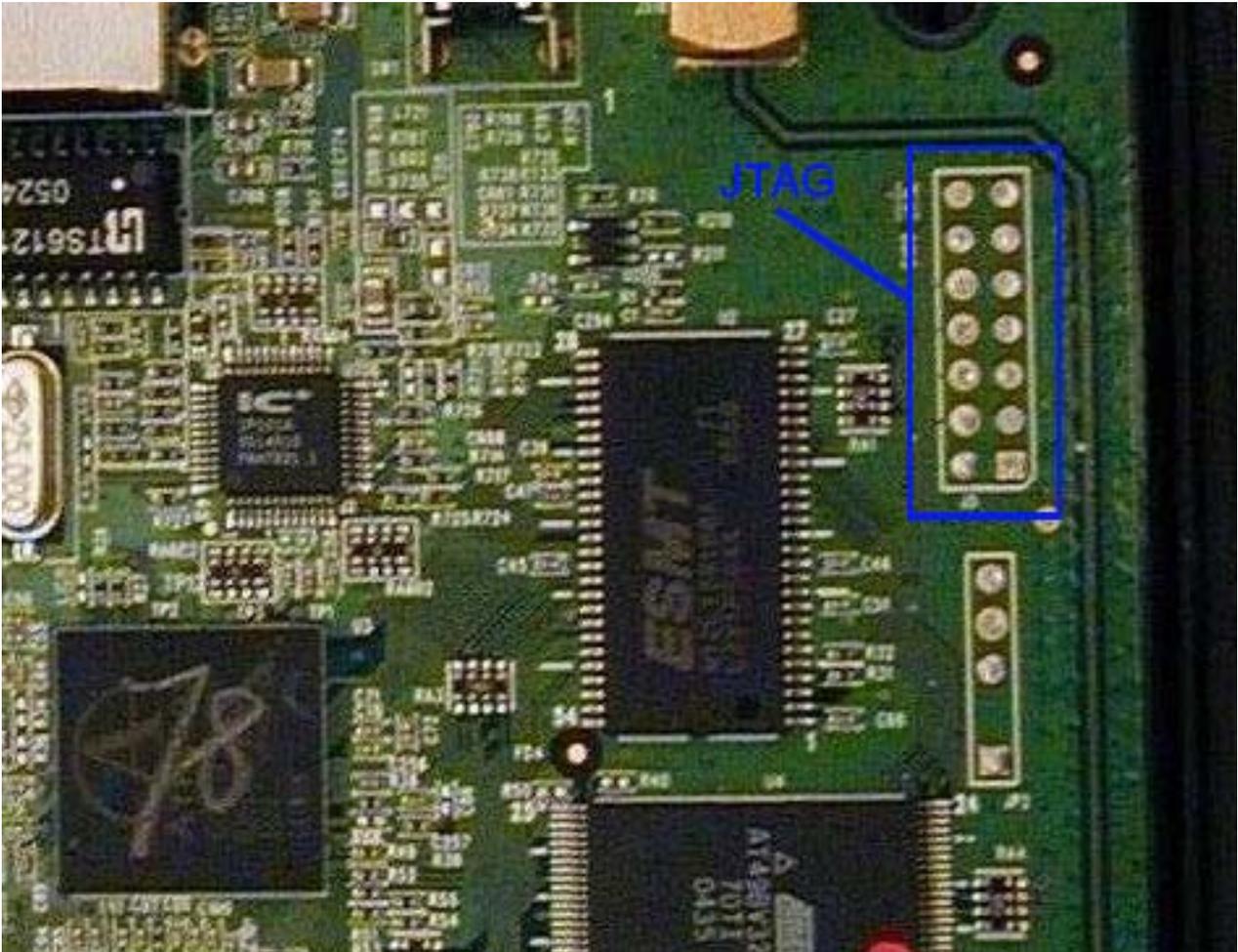
Il connettore JP1 a 12 pin nasconde invece al suo interno un'interfaccia seriale.

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.



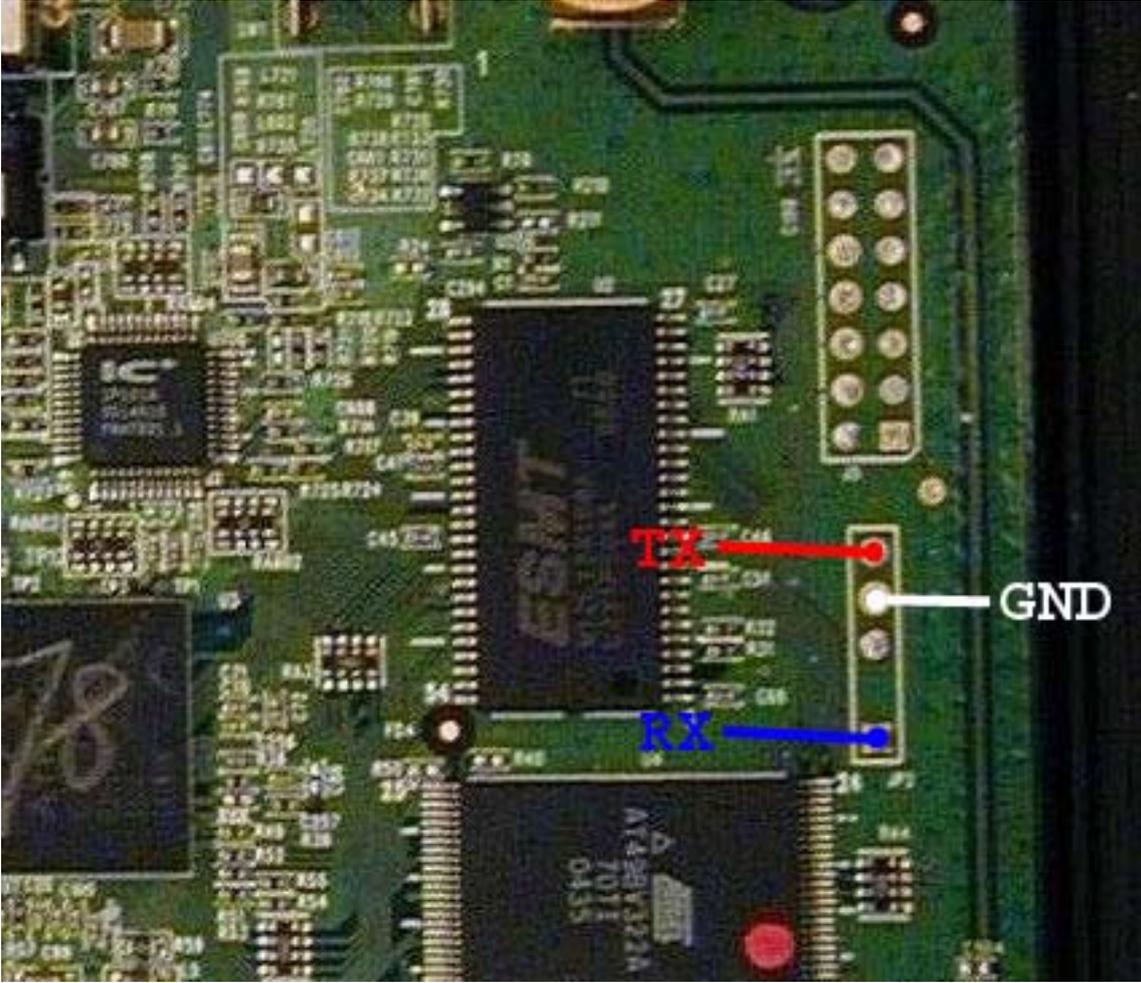
## Rev. A3/A4

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



Devo precisare che purtroppo ad oggi non sono stato in grado di trovare un'utility per utilizzare la Jtag in grado di riconoscere il chipset Atheros. Qual'ora tale ricerca abbia successo verrà implementata questa sezione con tutte le informazioni del caso.

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.



## 6. Interfacciarsi al Boot-loader del Dlink DWL-2100AP

Per interfacciarsi 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.

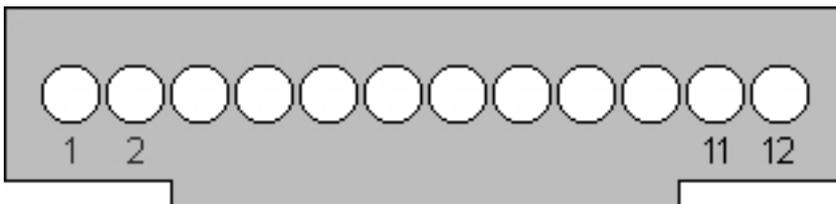


A questo punto è sufficiente aprire il connettore che andrebbe collegato al cellulare ...



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/tterm23.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 interfacciandomi 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=password] 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

ael 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) : 192.168.1.20:0xffffffff00
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) : 192.168.1.20:0xffffffff00
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
```



```

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=192.168.0.5:0xffffffff00 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,e1Pci0"
portsStringParse failed!

```

```

Adding 3836 symbols for standalone.

```

```

]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
(R)  ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
]   ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
0x  ]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]
80fffd0 (]] ]]]]]]]]] ]]]]]]]]] ]]]]]]]]] ]]] ]]] ]]]
]]]]
        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  
iam          "usr" [, "passwd"]  
copy        ["in"] [, "out"]
```

hostname = è il nome del pc sul quale imposteremo un server FTP (filezilla 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 + filename di destinazione

### Esempio

```
hostAdd "pc", "192.168.1.1"  
netDevCreate "net:", "pc", 1  
iam "Anonymous", "Guest"  
copy net:/apimg1, apimg1
```

Ripetete il copy per ogni file e alla fine 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.

## 8 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 installarci 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](http://www.wireless-italia.com)

Ramponi Stefano