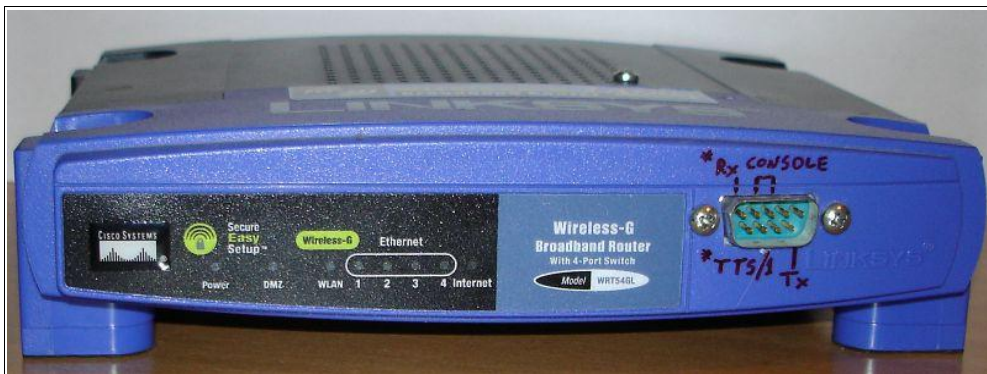


Προσθήκη δύο σειριακών θυρών RS-232C στον ασύρματο router WRT54G

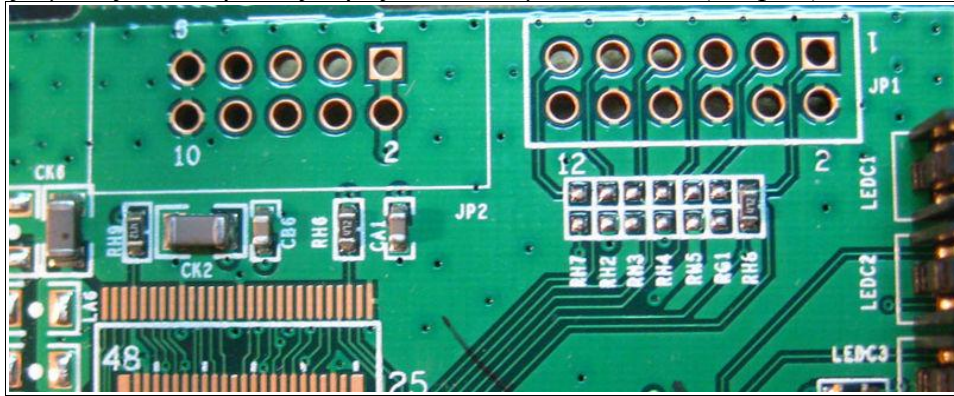


Σταύρος Σ. Φώτογλου (SV6GMP)

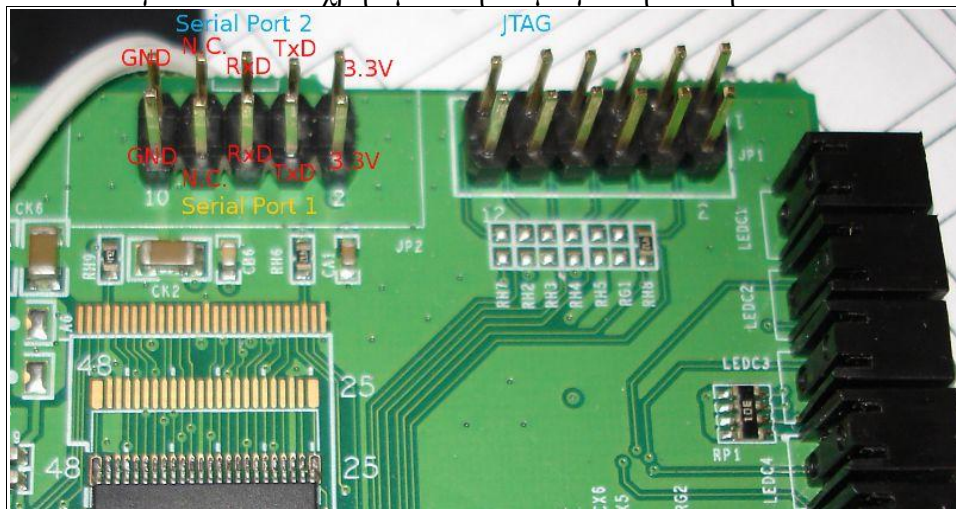
Προσθήκη σειριακών θυρών RS-232 στον WRT54GL

Ο WRT54GL όπως και όλα τα άλλα μοντέλα της ίδιας σειράς έχουν την δυνατότητα προσθήκης δύο σειριακών θυρών που είναι λειτουργικές 100%. Για λόγους κόστους δεν υπάρχουν συνδετήρες στο router αλλά ούτε και level converters για την μετατροπή από 0V / 3,3V σε +12V / -12V που θέλει η RS-232C.

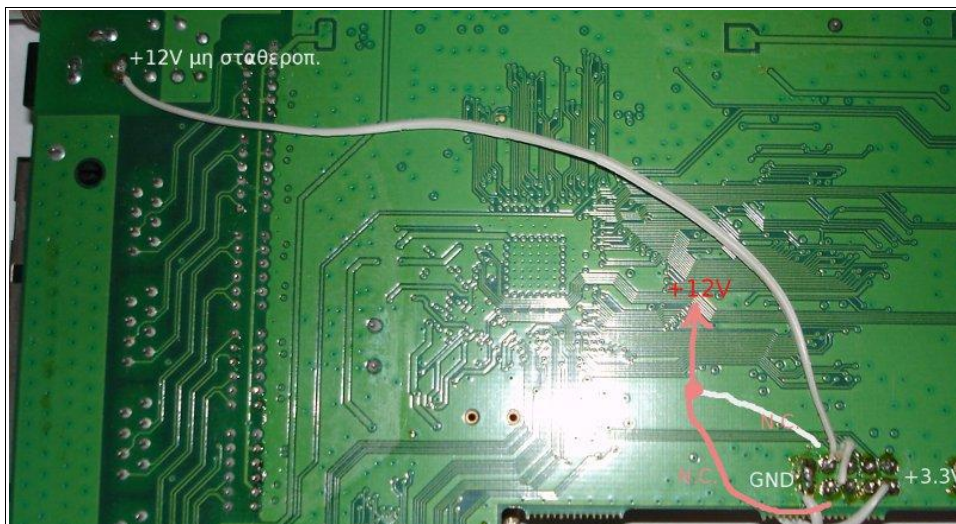
Αυτά λοιπόν καλούμαστε να τα κατασκευάσουμε εμείς. Ας ξεκινήσουμε λοιπόν. Αρχικά εντοπίζουμε τις δύο σειριακές θύρες και κολλάμε το header. (2x5pins)



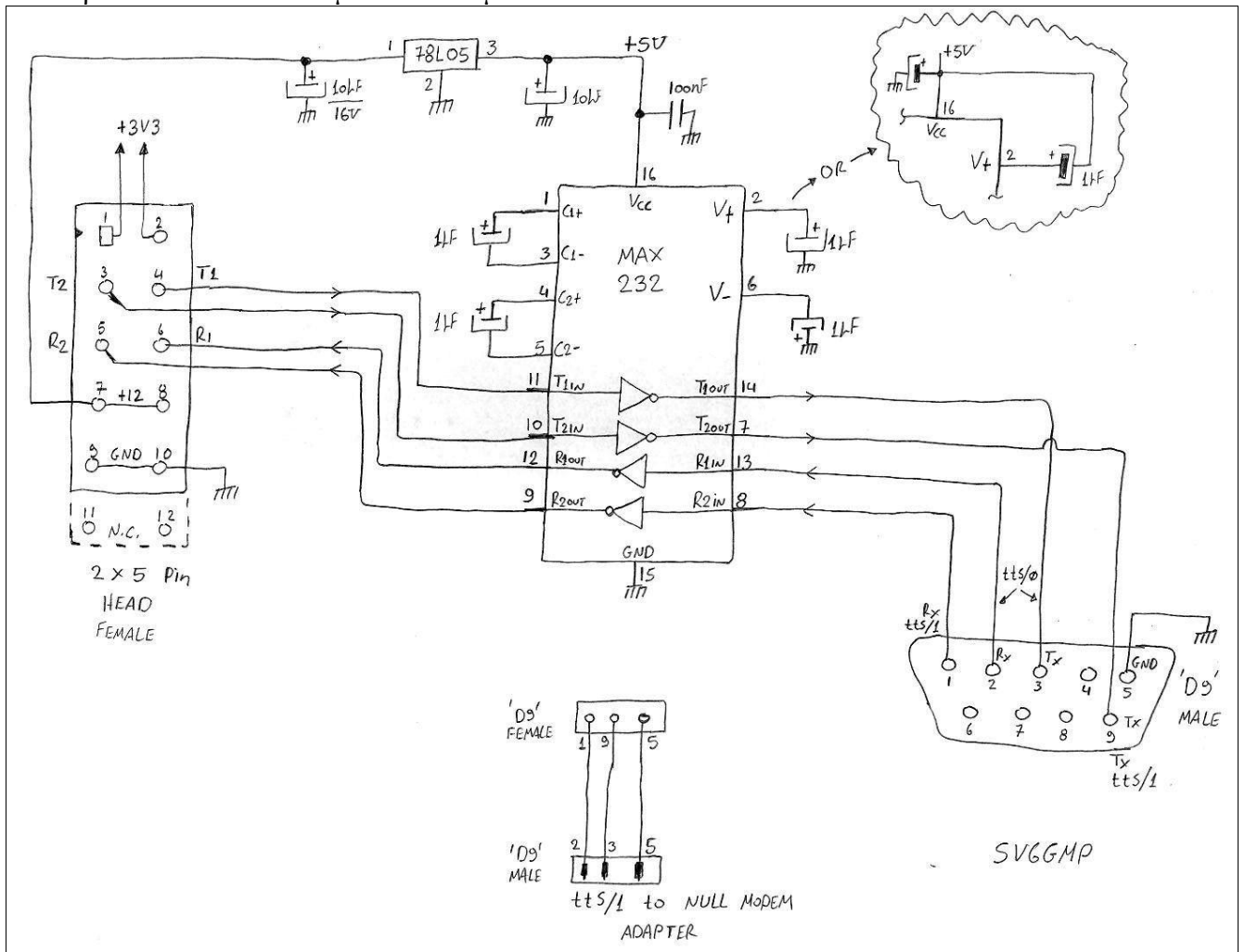
Στην παρακάτω εικόνα βλέπουμε τα σήματα – τάσεις που φέρει το κάθε pin. Παρατηρούμε ότι τα 7 & 8 δεν συνδέονται κάπου γι' αυτό θα τα χρησιμοποιήσουμε για την τάση των 12V.



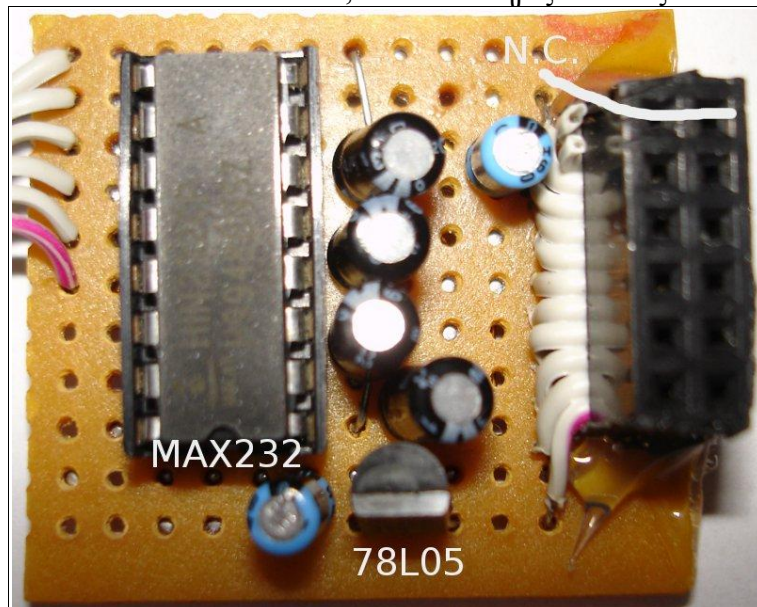
Στην εικόνα που ακολουθεί βλέπουμε ότι πρέπει να συνδέσουμε την ασταθεροποιητή τάση από το ομοαξονικό βύσμα εισόδου του εξωτερικού τροφοδοτικού 12V που βρίσκεται στη πίσω πλευρά του router.

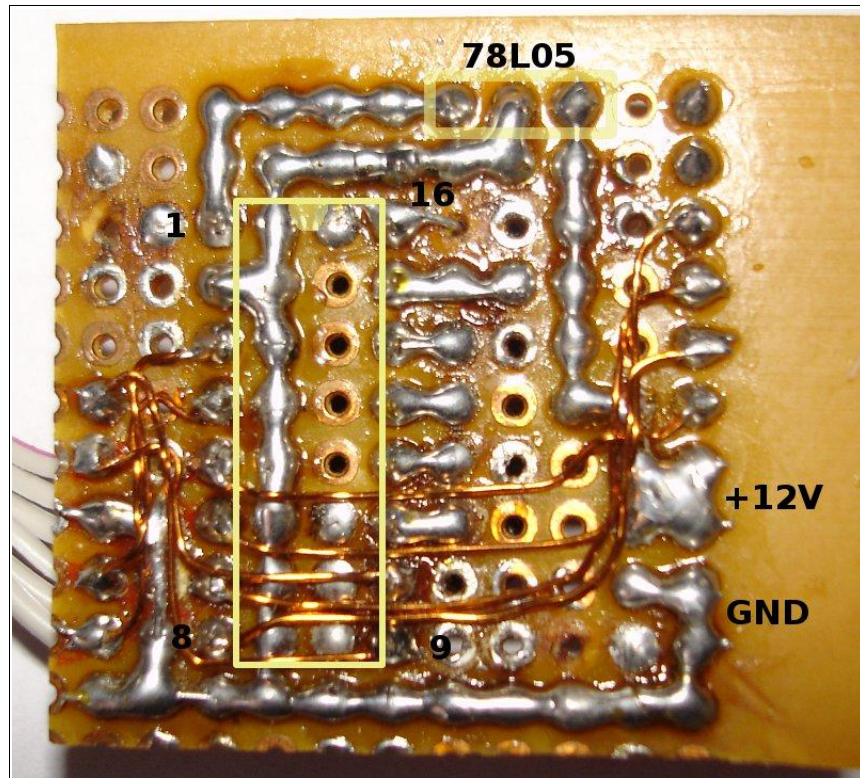


Στο σημείο αυτό έχουμε τελειώσει με τις τροποποιήσεις που έπρεπε να γίνουν στην πλακέτα του router και τώρα θα κατασκευάσουμε το κύκλωμα.

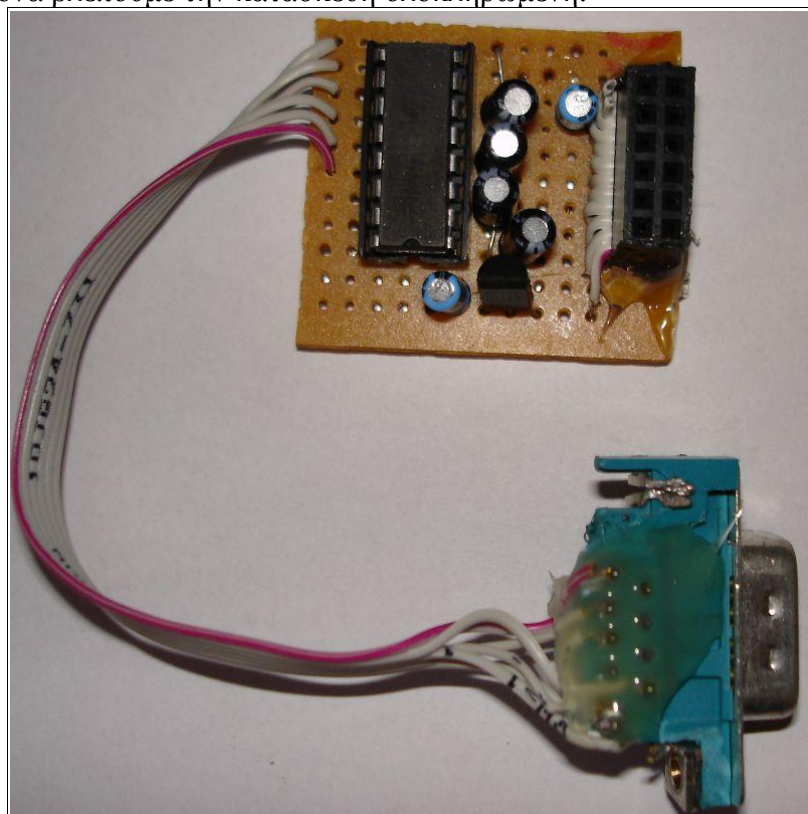


Όπως βλέπουμε στο παραπάνω θεωρητικό κύκλωμα θα χρειαστούμε ένα MAX232, ένα 78L05 και μερικούς ηλεκτρολυτικούς πυκνωτές, ένα βύσμα 'D9' αρσενικό και ένα θηλυκό pin header 2x5. Το ολοκληρωμένο κύκλωμα 78L05 είναι σταθεροποιητής τάσης ο οποίος σταθεροποιεί τα ασταθεροποίητα 12V σε 5V ώστε να τροφοδοτήσει το MAX232. Το MAX232 είναι ένας μετατροπέας επιπέδων σήματος από TTL σε RS232. Επίσης έχει και γεννήτριες τάσης -12 και +12V από την μοναδική τάση των 5V. Το MAX232 είναι διπλό, έτσι υποστηρίζει και τις δύο θύρες.





Στην παρακάτω εικόνα βλέπουμε την κατασκευή ολοκληρωμένη.



Εμείς για λόγους εξοικονόμησης χώρου βάλαμε μόνο ένα 'D9' στην πρόσοψη του router. Στα κανονικά pins της RS232 (2, 3) συνδέεται η θύρα 1 (tts/0 ή console) και στα (1, 9) η θύρα 2 (tts/1). Αν θέλουμε να συνδέσουμε την θύρα 2 σε κάποια συσκευή βάζουμε ένα extension (adaptor) που μεταφέρει τα pins 1 & 9 σε 2 & 3 ώστε να είναι δυνατή η σύνδεση ενός καλωδίου NULL modem.



Η χειροποίητη πλακέτα τοποθετημένη στον router λίγο πριν κλείσουμε τα καπάκια.



Ο router κλεισμένος με το καλώδιο μετατροπής (adaptor) για σύνδεση της 2ης θύρας στον Η/Υ.



Αν συνδέσω τον router στην 1η σειριακή που είναι και η κονσόλα θα πρέπει στο putty να βάλω com1 και ταχύτητα 115200. Τότε θα δω ότι βλέπω και από telnet ή ssh.

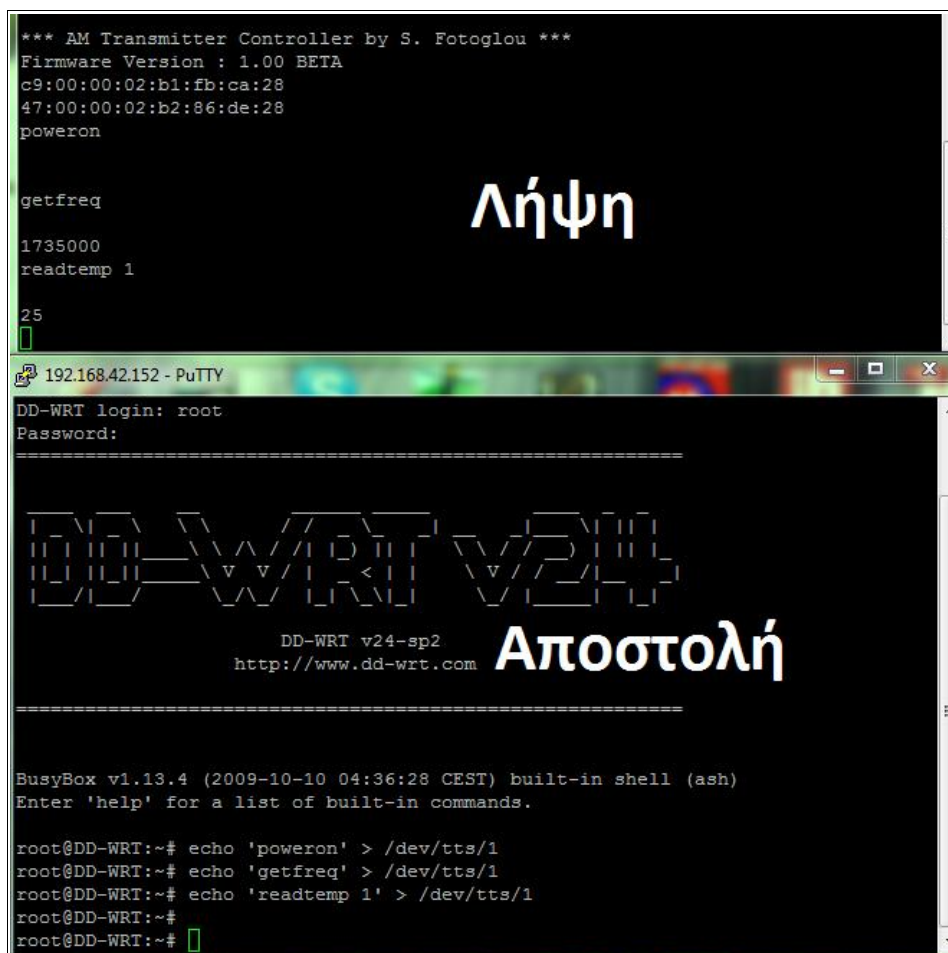
```
COM1 - PuTTY
DD-WRT v24-sp2 std (c) 2010 NewMedia-NET GmbH
Release: 08/12/10 (SVN revision: 14929)
□
DD-WRT login: root
Password:
=====
DD-WRT v24
http://www.dd-wrt.com
=====
Jun  4 22:29:09 login[2764]: root login on 'console' ✓
BusyBox v1.13.4 (2010-08-12 03:24:38 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
root@DD-WRT:~#
```

Στη δεύτερη θύρα η προκαθορισμένη ταχύτητα είναι 9600. Μπορούμε να δούμε την ταχύτητα και τις άλλες ρυθμίσεις με την εντολή :

```
root@DD-WRT:~# stty -F /dev/tts/1 -a
speed 19200 baud; rows 24; columns 80;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 hupcl -cstopb cread clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echopr
echoctl echoke
root@DD-WRT:~#
```

Αν θέλουμε να αλλάξουμε ταχύτητα γράφουμε :

```
stty -F /dev/tts/1 9600
```

```
*** AM Transmitter Controller by S. Fotoglou ***
Firmware Version : 1.00 BETA
c9:00:00:02:b1:fb:ca:28
47:00:00:02:b2:86:de:28
poweron

getfreq
1735000
readtemp 1
25

Λήψη

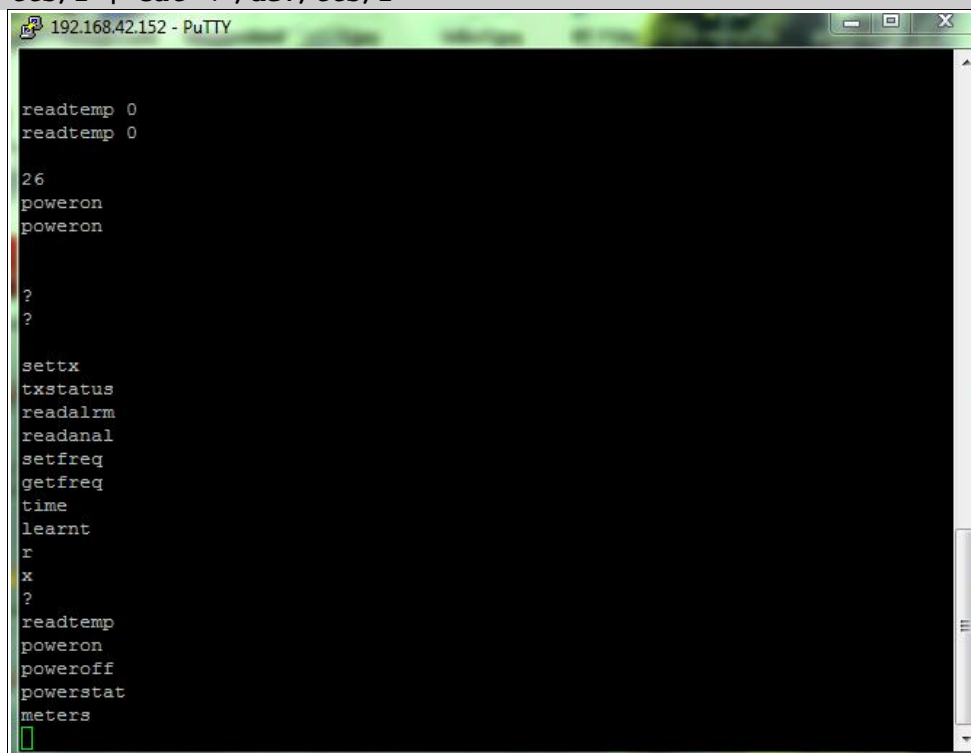
DD-WRT v24-sp2
http://www.dd-wrt.com
Αποστολή

BusyBox v1.13.4 (2009-10-10 04:36:28 CEST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

root@DD-WRT:~# echo 'poweron' > /dev/tts/1
root@DD-WRT:~# echo 'getfreq' > /dev/tts/1
root@DD-WRT:~# echo 'readtemp 1' > /dev/tts/1
root@DD-WRT:~#
root@DD-WRT:~#
```

Βέβαια μπορούμε να χρησιμοποιήσουμε και ένα τερματικό με την εντολή:

```
cat > /dev/tts/1 | cat < /dev/tts/1
```



```
Αποστολή

readtemp 0
readtemp 0

26
poweron
poweron

?
?

settx
txstatus
readalarm
readanal
setfreq
getfreq
time
learnt
r
x
?
readtemp
poweron
poweroff
powerstat
meters
```

Σε openwrt backfire βάζω αρχικά το πακέτο coreutils-stty για να έχω την εντολή stty. Δηλώνω την ταχύτητα επικοινωνίας της σειριακής να είναι ίδια με αυτή του AVR.

```
stty -F /dev/ttyS1 38400. Προσοχή εδώ οι σειριακές είναι ttyS0 και ttyS1.
```

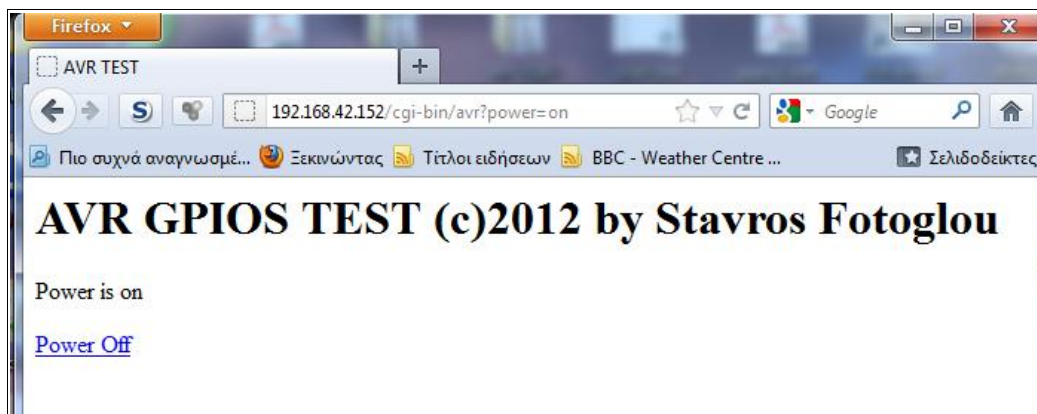

Στο /www/cgi-bin φτιάχνω ένα script όπως παρακάτω και το κάνω εκτελέσιμο 755.

```
root@OpenWrt:/www/cgi-bin# cat avr
#!/bin/sh

echo Content-type: text/html
echo ""
POWER=`echo "$QUERY_STRING" | sed -n 's/^. *power=\([^&]*\) .*$/\1/p'`
if [ "$POWER" = "" ]
then
    POWER="off"
fi
/bin/cat << EOM
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>AVR TEST</title>
<script language = "JavaScript">
function poweron()
{
    url="avr?power=on";
    window.location=url;
}
function poweroff()
{
    url="avr?power=off";
    window.location=url;
}
</script>
</head>
<body>
<h1>AVR GPIO TEST (c)2012 by Stavros Fotoglou</h1>
<p>
EOM

echo "<p>Power is $POWER</p>"
if [ "$POWER" != "on" ]
then
    echo '<font color = "#FF0000"><b>'
    echo '<a href = "javascript:poweron()">Power On</a>'
    echo '</b></font>'
    echo 'poweroff' > /dev/ttyS1
else
    echo '<a href="javascript:poweroff()">Power Off</a>'
    echo 'poweron' > /dev/ttyS1
fi
/bin/cat << EOM
</body>
</html>
EOM
```

Με την υπογραμμισμένη εντολή, επιλέγουμε από τις μεταβλητές περιβάλλοντος αυτή που έχει όνομα power και την στέλνω με μέθοδο get και url encoding. Με τις εντολές που είναι με πλάγια γραφή, στέλνω την διαταγή στο shell του avr μέσω σειριακής θύρας.



Script για διάβασμα κατάστασης από AVR

Αρχικά ρυθμίζω την σειριακή να μην έχει επιπλέον echo και να μην κάνει αλλαγή γραμμής αν δεχθεί CR.

```
stty -F /dev/ttyS1 -echo -icrnl
```

Μετά γράφω script και το κάνω εκτελέσιμο:

```
root@OpenWrt:/www/cgi-bin# cat xx
#!/bin/sh

exec 3<>/dev/ttyS1

echo "powerstat" >&3
read response <&3
read response <&3
read response <&3
echo "$response" >>rlog

exec 3>&-

STATE=`echo "$response" | sed -n 's/^Is\([^&]*\) .*$/\1/p'`
echo "$STATE"
```

Όταν δώσω την εντολή powerstat με δύο συνόδους ssh (μία λήψη και μία αποστολή), όπως παραπάνω στη σελίδα 7, βλέπουμε ότι απαντάει 'Is ON' ή 'Is OFF' ανάλογα. Στην υπογραμμισμένη εντολή κάνουμε χρήση του sed και κανονικών εκφράσεων (regular expressions) ώστε η μεταβλητή STATE να περιέχει το string 'ON' ή 'OFF' ανάλογα. Οι τρεις εντολές read διαβάζουν μία γραμμή εισόδου η κάθε μία. Στη 1η γραμμή είναι η ίδια εντολή που έγινε echo από avr. Η 2η είναι κενή. Τέλος στη 3η γραμμή υπάρχει η απάντηση που περιμένω.

Εκτελώ το script με ./xx .

```
root@OpenWrt:/www/cgi-bin# ./xx
OFF
root@OpenWrt:/www/cgi-bin# ./xx
ON
root@OpenWrt:/www/cgi-bin#
```

Στη συνάρτηση getcommand() από την πλευρά του AVR πρόσθεσα το σκιασμένο τμήμα γιατί το shell του openwrt και dd-wrt στέλνει cr και lf. Το επόμενο τμήμα (else if ...) λειτουργεί κανονικά σε putty serial terminal από windows.

```

/***** Getcommand moves characters from Ring Buffer to Command Buffer *****/
/***** Return 1 if new command exist in buffer, else return 0 *****/
u08 getcommand(void)
{
    static u08 cmdlen = 0;
    if (rxFlag) //If new character received in Ring Buffer
    {
        do { //loop
            if (rx_i >= BUFFER_SIZE) //If end of ring buffer
                rx_i = 0; //Set to start of ring buffer
            CmdBuffer[cmdlen++] = UART_RxBuffer[rx_i++]; //Copy from RxBuffer to CmdBuffer
        } while (rx_i != rx_n); //Until no new charcters in ring buffer
        rxFlag = 0;
        if (CmdBuffer[cmdlen-1] == '\n') //If last character is CR
        {
            CmdBuffer[cmdlen-2] = '\0';
            CmdBuffer[cmdlen-1] = '\0'; //trim '\r' and finish string with 0
            cmdlen = 0; //For next use
            //rx_i = rx_n = 0;
            return(1); //Enter pressed and new command in command buffer
        }
        else if (CmdBuffer[cmdlen-1] == '\r')
        {
            CmdBuffer[cmdlen-1] = '\0';
            cmdlen = 0;
            return(1);
        }
        else
            return(0); //No enter pressed
    }
    return(0); //No new character received
}

```