

You are here: [DD-WRT wiki mainpage](#) / [Scripting](#) / [LED Scripts](#)

This page contains scripts which can modify the behavior of your router's LED's.

To find out how to load scripts onto the router, see [Startup Scripts](#).

## Contents

- [1 GPIO Information](#)
  - ◆ [1.1 Linksys WRT54G](#)
  - ◆ [1.2 Linksys WRT150N v1.1](#)
  - ◆ [1.3 Linksys WRT610N v2.0](#)
  - ◆ [1.4 Linksys WRTSL54GS](#)
  - ◆ [1.5 Linksys WRT1900AC](#)
  - ◆ [1.6 Buffalo WHR ???](#)
  - ◆ [1.7 LaFonera 2100](#)
  - ◆ [1.8 La Fonera 2200](#)
  - ◆ [1.9 La Fonera 2201](#)
  - ◆ [1.10 D-Link DIR-320](#)
  - ◆ [1.11 TP-Link TL-WR1043ND](#)
  - ◆ [1.12 Netgear WNDR4000](#)
  - ◆ [1.13 Buffalo WZR-1166DHP](#)
- [2 LED Scripts](#)
  - ◆ [2.1 Wifi Toggle D-Link DIR-300](#)
  - ◆ [2.2 Wifi Toggle D-Link DIR-615 Rev C2](#)
  - ◆ [2.3 Display Load via LED \(load.sh\)](#)
  - ◆ [2.4 Display number of users connected \(WRT54G\)](#)
  - ◆ [2.5 WDS link and/or associated wireless clients](#)
  - ◆ [2.6 WLAN Status SESbutton WRT54GL](#)
  - ◆ [2.7 WLAN Status \(wlan.sh\)](#)
  - ◆ [2.8 WLAN Status \(wlan.sh\) - Buffalo Routers](#)
  - ◆ [2.9 WLAN Client Mode Status](#)
  - ◆ [2.10 WLAN Disable Radio If No Clients Connected](#)
  - ◆ [2.11 Show VPN status/activity with SES/Cisco LED](#)
  - ◆ [2.12 USB Disc Mount Status and Umount Button DIR-320](#)
  - ◆ [2.13 Display signal strength over the SES light - Linksys routers](#)
    - ◇ [2.13.1 Script Version 2](#)
    - ◇ [2.13.2 Script Version 1](#)
  - ◆ [2.14 Display signal strength over the Diagnostic light - Buffalo routers](#)
    - ◇ [2.14.1 Broadcom based Hardware](#)
    - ◇ [2.14.2 Atheros based Hardware](#)
  - ◆ [2.15 USB Disc Mount Status and Umount Button TP-Link TL-WR1043ND](#)
- [3 See Also](#)

## GPIO Information

The CPU controls LED's (and some other things) using General Purpose In/Out interfaces. GPIO's are wired to different things on different models of routers so you will likely need to change the scripts to suit your model. To detect which GPIO's control which LED's on Broadcom devices see [this page](#).

**The example LED scripts below are usually written using the GPIO info for the Linksys WRT54G. Remember to change them for whatever router you have.**

You can't use the GPIO's on Atheros chips (Fonera, Meraki, etc.) without PROC\_GPIO or some other utility. PROC\_GPIO is a driver, and thus it must be compiled for your specific kernel. Broadcom routers do not need an external driver to drive GPIO's.

### Linksys WRT54G

Pin	Direction	Name	Use
GPIO 0	Output	WLAN LED	(LED - wireless)
GPIO 1	Output	POWER LED	(LED - power)
GPIO 2	Output	ADM_EECS	(LED - white, Cisco button v. 3.0+)
GPIO 3	Output	ADM_EESK	(LED - amber, Cisco button v. 3.0+)
GPIO 4	Input	ADM_EEDO	(Button - Cisco Button v. 3.0+)
GPIO 5	Output	ADM_EEDI	(Unknown) Seems to cycle all LED colors disabled.
GPIO 6	Input	RESET	(Button - reset button)
GPIO 7	Output	DMZ LED	(LED - DMZ)

### Linksys WRT150N v1.1

Here is a short summary of my experiments with WRT150N v1.1 on dd-wrt.v24\_mini\_generic firmware. WRT150N has a SECURITY LED (the most right LED)

```
root@DD-WRT:~# gpio enable 5 #(SECURITY LED - off - green)
root@DD-WRT:~# gpio disable 5 #(SECURITY LED - on - green)
root@DD-WRT:~# gpio disable 3 #(SECURITY LED - on - amber)
root@DD-WRT:~# gpio enable 3 #(SECURITY LED - off - amber)
```

When you switched to gpio disable 3 (SECURITY LED - on - amber)

```
root@DD-WRT:~# gpio enable 5 #(SECURITY LED - will give you amber)
root@DD-WRT:~# gpio disable 5 #(SECURITY LED - will give you amber bright)
```

**WARNING:** "gpio disable 6" will completely erase your NVRAM (i.e. do a complete reset on your router)

### Linksys WRT610N v2.0

It should also work for the v1.0. These commands concerns the little LED near the button.

Pin	Direction	Name	Use
GPIO 0	Output	SES LED	(Security LED - Amber)
GPIO 3	Output	SES LED	(Security LED - Blue)

## Linksys WRTSL54GS

This is for the WRTSL54GS model only.

Pin	Direction	Name	Use
GPIO 5	Output	SES LED	(Cisco white LED)
GPIO 7	Output	SES LED	(Cisco amber LED)

## Linksys WRT1900AC

This should also work for WRT1200AC, WRT1900ACS, and WRT3200ACM

Pin	Direction	Name	Use
GPIO 9	Output	WPS LED	(WPS white LED)
GPIO 10	Output	WPS LED	(WPS amber LED)

see the [WRT1900 page](#) for more information.

## Buffalo WHR ???

Pin	Direction	Use
GPIO 0	Input	AOSS button
GPIO 1	Output	Bridge LED
GPIO 2	Output	WLAN LED
GPIO 3	Output	Extra LED between bridge and WLAN
GPIO 4	Input	Reset button
GPIO 5	Input	Bridge/auto switch
GPIO 6	Output	AOSS LED
GPIO 7	Output	DIAG LED
GPIO 8	n/a	Unkown/none
GPIO 9	Output	Power LED

## LaFonera 2100

- [According to this link](#)

Pin	Use
0	TP3
1	pin 5 of SW1
2	WLAN LED
3	pin 1 of SW1
4	pin 2 of SW1
5	Reset (!)
6	Reset button
7	pin 6 of SW1

## La Fonera 2200

2 WIFI LED contact at bottom of local resistor  
 5 Reset! (Can be used as a GPIO, but you lose reset functionality) This is the line closest to the  
 Cut the trace and use the end closest to the CPU as a GPIO.  
 6 Reset button (other end of button is VDD, (3.3 V)) remove nearby capacitor  
 7 Power LED contact at bottom of local resistor

## La Fonera 2201

1 WLAN Orange LED  
 2 WLAN Green LED  
 4 Power Green LED  
 7 Power Orange LED

## D-Link DIR-320

Pin	Direction	Use
GPIO 0	Output	(LED - WIRELESS)
GPIO 1	Output	(LED - STATUS)
GPIO 3	Output	(LED - RED)
GPIO 4	Output	(LED - BLUE)
GPIO 5	Output	(LED - USB)
GPIO 6	Input	(Button on the right)

## TP-Link TL-WR1043ND

Pin	Direction	Name
GPIO 1	Output	USB LED (gpio disable 1 -> ON, gpio enable 1 -> OFF)
GPIO 2	Output	SYS LED (gpio disable 2 -> ON, gpio enable 2 -> OFF)
GPIO 3	Input	RESET
GPIO 5	Output	QSS LED (gpio enable 5 -> ON, gpio disable 5 -> OFF)
GPIO 7	Input	QSS Button

## Netgear WNDR4000

GPIO #	LED	Notes
0	Power	Enable/Disable LED
1	Power	Toggle LED Color (Green/Amber)
2	Internet	Toggle LED Color (Green/Amber)
3	2.4 GHz	
4	5 GHz	
5	USB	
6	WPS	
7	Wireless	
8	TBD	

## Buffalo WZR-1166DHP

GPIO#	LED/SW	Notes
11	Reset Button	00 Pressed
12	AOSS Button	00 Pressed
13	Mode Button	00 Pressed
14	AP/WB Switch	00 AP, 01 WB
15	USB Eject Button	00 Pressed
40	Amber WIFI LED	
41	Blue WIFI LED	
42	Blue INET LED	
43	Amber Router LED	
44	Blue Router LED	
45	White Logo LED	
46	Red Logo LED	

## LED Scripts

### Wifi Toggle D-Link DIR-300

This script will enable the button on the right side to act as a wifi toggle. On press the button-led will light up

- **RED**, when wifi is going to be **disabled**. wifi-led on the front will go out.
- **BLUE**, when wifi is going to be **enabled** again. wifi-led on the front will light up.

```
#!/bin/sh
/sbin/gpio enable 4
echo "0" > /proc/gpio/4_out
echo "0" > /proc/gpio/4_in
while : ; do
WIFI=`nvram get ath0_net_mode`
sleep 1;
  if [ "$(cat /proc/gpio/4_in )" == "1" ]; then
    if [ "$WIFI" == "disabled" ]; then
      /sbin/gpio enable 1
      nvram set ath0_net_mode=mixed
      /sbin/ifconfig ath0 up
      /sbin/gpio enable 2
      echo "0" > /proc/gpio/4_in
      sleep 3
      /sbin/gpio disable 1
    else
      /sbin/gpio enable 3
      nvram set ath0_net_mode=disabled
      /sbin/ifconfig ath0 down
      /sbin/gpio disable 2
      echo "0" > /proc/gpio/4_in
      sleep 3
      /sbin/gpio disable 3
    fi
  fi
done &
```

## LED\_Scripts

Alternate script to the one above: This script will enable the button on the right side to act as a wifi toggle. On press the button-led will light up

- **RED**, when wifi is going to be **disabled**. wifi-led on the front will go out.
- **BLUE**, when wifi is going to be **enabled** again. wifi-led on the front will light up.

```
# --- COPY THE TEXT BELOW TO DD-WRT / ADMINISTRATION / COMMANDS and click SAVE STARTUP ---
SES_BUTTON_BLUE_LED=1
WIFI_LED=2
SES_BUTTON_RED_LED=3
SES_BUTTON=4
gpio enable $SES_BUTTON
echo "0" > /proc/gpio/${SES_BUTTON}_out
echo "0" > /proc/gpio/${SES_BUTTON}_in
# Function: set_leds [blue|red]
set_leds() {
  case "$1" in
    blue)
      # Light SES LED up to blue, enable WIFI LED.
      gpio disable $SES_BUTTON_RED_LED
      gpio enable $SES_BUTTON_BLUE_LED
      gpio enable $WIFI_LED
      ;;
    red)
      # Light SES LED up to red, disable WIFI LED.
      gpio disable $SES_BUTTON_BLUE_LED
      gpio enable $SES_BUTTON_RED_LED
      gpio disable $WIFI_LED
      ;;
  esac
}
# Function: check_wlan_status
check_wlan_status() {
  WIFI=`nvrans get ath0_net_mode`
  if [ "$WIFI" == "disabled" ]; then
    set_leds red
  else
    set_leds blue
  fi
}
# Check WLAN status.
check_wlan_status
# Process #1: Watch SES button.
while ;; do
  check_wlan_status
  WIFI=`nvrans get ath0_net_mode`
  SES_BUTTON_IN=`cat /proc/gpio/${SES_BUTTON}_in`
  [ "$SES_BUTTON_IN" = "1" ] && {
    if [ "$WIFI" == "disabled" ]; then
      # Enable WIFI.
      set_leds blue
      nvrans set ath0_net_mode=mixed
      ifconfig ath0 up
    else
      # Disable WIFI.
      set_leds red
      nvrans set ath0_net_mode=disabled
      ifconfig ath0 down
    fi
  }
  echo "0" > /proc/gpio/${SES_BUTTON}_in
  sleep 2
}
```

```

}
sleep 1
done &

```

## Wifi Toggle D-Link DIR-615 Rev C2

This script will enable the button on the right side to toggle the wireless status between on and off and to give an indication using the right blue LED on the button.

- **LED ON**, when wireless is turned on.
- **LED OFF**, when wireless is turned off.

LED assignment can be changed by settings GPIO variable with one of the LED constants.

Wireless start state can be configured using disableWifiOnStart.

```

#!/bin/sh
# LED constants
SIDE_LED=3
POWER_GREEN_LED=16
POWER_ORANGE_LED=1
WAN_GREEN_LED=4
WAN_ORANGE_LED=15

# Set the LED to turn on/off
GPIO=$SIDE_LED
# Set whether to enable/disable wireless on start
disableWifiOnStart=1

# Disable wireless on start accorindg to disableWifiOnStart setting
if [ $disableWifiOnStart == 1 ];then
    logger -s -p local0.notice -t WirelessToggle "Turning off wireless on start"
    /sbin/ifconfig ath0 down
    /sbin/gpio enable $GPIO
fi

# Generate SESButton script and make it executable
logger -s -p local0.notice -t WirelessToggle "Generating SESButton script"
mkdir /tmp/etc/config/
echo "if [ `ifconfig | grep ath0` == "" ]; then
    logger -s -p local0.notice -t WirelessToggle \"Current wireless status: Off, Turning wirel
    /sbin/ifconfig ath0 up
    /sbin/gpio disable $GPIO
else
    logger -s -p local0.notice -t WirelessToggle \"Current wireless status: On, Turning wirel
    /sbin/ifconfig ath0 down
    /sbin/gpio enable $GPIO
fi
" > /tmp/etc/config/wifi_toggle.sesbutton
chmod +x /tmp/etc/config/wifi_toggle.sesbutton
logger -s -p local0.notice -t WirelessToggle "SESButton script generated and ready for use"

```

For GPIO information, send a private message to DD-WRT user "meltyblood"

## Display Load via LED (load.sh)

- Uses front button LED to display current load on router.
- For WRT54G/GL/GS

```
#!/bin/sh
gpio="gpio"

amber=3
white=2

delay=3

meltdown=400
overload=100
highload=70
medload=30

while sleep $delay;do
  set -- $(cat /proc/loadavg)
  load="${1%.*}${1#*}"

  if [ $load -gt $meltdown ];then
    $gpio disable $amber
    usleep 50000
    $gpio disable $white
    usleep 50000
    reboot
  elif [ $load -gt $overload ];then
    $gpio disable $amber
    usleep 50000
  elif [ $load -gt $highload ];then
    $gpio disable $amber
    usleep 12500
    $gpio enable $amber
    usleep 12500
    $gpio disable $amber
    usleep 12500
    $gpio enable $amber
    usleep 12500
    $gpio disable $amber
    usleep 12500
    $gpio enable $amber
    usleep 12500
  elif [ $load -gt $medload ];then
    $gpio enable $amber
    $gpio disable $white
    usleep 25000
    $gpio enable $white
    usleep 25000
    $gpio disable $white
    usleep 25000
    $gpio enable $white
    usleep 25000
  else
    $gpio disable $white
    usleep 50000
    $gpio enable $white
    usleep 50000
  fi
done
```



## LED\_Scripts

done

Modified Script for simple display of cpu usage/load -- I use the power LED Low activity/idle (led off), medium (short flashes), high (long flashes), max (led on solid)

---

```
#!/bin/sh
gpio="gpio"

#This is the LED to flash
led=0

extreme=110
high=75
med=35

while [ 0 ];do
  set -- $(cat /proc/loadavg)
  load="${1%.*}${1#*}"

  counter=10

  if [ $load -gt $extreme ];then
    $gpio disable $led
    sleep 8
  elif [ $load -gt $high ];then
    while [ $counter -gt 0 ];do
      usleep 200000
      $gpio disable $led
      usleep 800000
      $gpio enable $led
      let counter-=1
    done
  elif [ $load -gt $med ];then
    while [ $counter -gt 0 ];do
      usleep 800000
      $gpio disable $led
      usleep 200000
      $gpio enable $led
      let counter-=1
    done
  else
    $gpio enable $led
    sleep 8
  fi
done
```

## Display number of users connected (WRT54G)

This is useful if you are running a guest network or an OpenWifi hotspot. You can just look at the Cisco illuminated button to see if anyone is connected.

- 0 users = black
- 1 user = amber
- 2 users = white
- 3 users = amber flashing
- 4 users = white flashing

## LED\_Scripts

- 5 or more = amber/white alternating flashing

When users disconnect, the lights stay on for a few minutes, because the router takes a while to close the TCP connections when an active client disconnects. See "Active Clients" on the Status/LAN DD-WRT GUI webpage.

This is based on the command ``cat /proc/net/arp``. A better command might be: ``iw dev wlan0 station dump | grep ath0 | wc -l``

I made a variable NC for number of connections, not thinking that it could be confused with NoCat nvram variables. The "trap lightsoff..." turns off the light if the script stops running for some reason.

*How to load the script into a DD-WRT router ...(more later).*

```
#!/bin/sh
AMBER='gpio disable 3 ; gpio enable 2'
WHITE='gpio enable 3 ; gpio disable 2'
BLACK='gpio enable 3 ; gpio enable 2'
#possible to use usleep 10000 instead of sleep 1

trap lightsoff 1 2 3 6 14 15
lightsoff()
{
gpio enable 3 ; gpio enable 2 ; exit 1
}

while true; do
NC=`grep 0x /proc/net/arp | wc -l`
if [ "$NC" -eq 0 ]; then
    eval $BLACK
fi
if [ "$NC" -eq 1 ]; then
    eval $AMBER
fi
if [ "$NC" -eq 2 ]; then
    eval $WHITE
fi
if [ "$NC" -eq 3 ]; then
    eval $AMBER
    sleep 1
    eval $BLACK
fi
if [ "$NC" -eq 4 ]; then
    eval $WHITE
    sleep 1
    eval $BLACK
fi
if [ "$NC" -ge 5 ]; then
    eval $AMBER
    sleep 1
    eval $WHITE
fi
sleep 1
done
```

## WDS link and/or associated wireless clients

- illuminates front button LED white when able to ping the gateway
- illuminates front button LED amber when wireless clients are connected
- illuminates both colors on the button when both conditions are true
- no light when neither condition is true
- developed / tested on a WRT54G-TM

```
GATEWAY=`ip route | awk '/default via/ {print }'`
WDS_LINK_ACTIVE=false
ASSOC_CLIENTS=false
LOOP=0

[ `sh -c "ps | grep | grep -v grep -c"` -gt 1 ] && echo Already running && exit

while sleep 1
do

    [ $LOOP -lt 300 ] && LOOP=$(( $LOOP + 1 )) || LOOP=1

    while ! $WDS_LINK_ACTIVE
    do
        sleep 5
        ping -c 1 $GATEWAY >/dev/null && WDS_LINK_ACTIVE=true
    done

    [ $LOOP -eq 1 ] && {
        ping -c 1 $GATEWAY >/dev/null || WDS_LINK_ACTIVE=false
    }

    [ -z $(wl -i `nvram get wl0_ifname` assoclist) ] && ASSOC_CLIENTS=false || ASSOC_CLIENTS=

    $WDS_LINK_ACTIVE && gpio disable 2 || gpio enable 2
    $ASSOC_CLIENTS && gpio disable 3 || gpio enable 3

done
```

## WLAN Status SESbutton WRT54GL

Here's a simple script that will use the SES LED to display the status of the Wifi when using the SESbutton to enable or disable the Wifi. Whenever the button is pressed to enable the Wifi, the LED will light up in bright white. When disabling Wifi, it will extinguish.

The script could for example be used by saving it as "action.sesbutton" in the /jffs/etc/config directory.

```
#!/bin/sh
sleep 2
/sbin/gpio enable 3
if test "$(wl radio)" == "0x0000" ; # check for wifi state
then
/sbin/gpio disable 2 # switch LED on
else
/sbin/gpio enable 2 # switch LED off
fi ;
```

## WLAN Status (wlan.sh)

Works on k2.4 and k2.6

- Uses front button LED to show WLAN state. Amber LED indicates one or more associated clients, white LED flashes when data is sent over WLAN.

```
#!/bin/sh
I=`nvram get wl0_ifname`

while sleep 1; do
  if [ -z `wl -i $I assoclist` ]; then
    XFER=`ifconfig $I|grep bytes`
    if [ "$XFER" != "$PXFER" ]; then
      LED='gpio disable 3 ; gpio disable 2'
      PXFER=$XFER
    else
      LED='gpio disable 3 ; gpio enable 2'
    fi
  else
    LED='gpio enable 3 ; gpio enable 2'
  fi

  if [ "$LED" != "$PLED" ]; then
    eval $LED
    PLED=$LED
  fi
done
```

## WLAN Status (wlan.sh) - Buffalo Routers

- AOSS LED to show WLAN is associated with clients.
- Bridge LED flashes when data is transmitted over WLAN.

```
#!/bin/sh
I=`nvram get wl0_ifname`

while sleep 1; do
  if [ -z `wl -i $I assoclist` ]; then
    XFER=`ifconfig $I|grep bytes`
    if [ "$XFER" != "$PXFER" ]; then
      LED='gpio disable 1 ; gpio enable 1 ; gpio disable 6'
      PXFER=$XFER
    else
      LED='gpio disable 6'
    fi
  else
    LED='gpio enable 6'
  fi

  if [ "$LED" != "$PLED" ]; then
    eval $LED
    PLED=$LED
  fi
done
```

## WLAN Client Mode Status

Works on k2.4 and k2.6

Works on Repeater Bridge as well

- White LED if we can ping the gateway
- Orange LED if associated to an AP, but pinging fails.

```
#!/bin/sh
AMBER='gpio disable 3 ; gpio enable 2'
WHITE='gpio enable 3 ; gpio disable 2'
BLACK='gpio enable 3 ; gpio enable 2'
PACKETS='1'
INTERVAL='10'

trap lightsoff 1 2 3 6 14 15
lightsoff()
{
gpio enable 3 ; gpio enable 2 ; exit 1
}

while true ; do
if [ -z $(wl -i `nvram get wl0_ifname` assoclist) ]; then
TARGET=`ip route | awk '/default via/ {print }'`
RET=`ping -c $PACKETS $TARGET 2> /dev/null | awk '/packets received/ {print }'`
if [ "$RET" -eq "$PACKETS" ]; then
LED=$WHITE
else
LED=$AMBER
fi
else
LED=$BLACK
fi
if [ "$LED" != "$PLED" ]; then
eval $LED
PLED=$LED
fi
sleep $INTERVAL
done
```

## WLAN Disable Radio If No Clients Connected

This is a cron job that will run every hour and turn off the radio if no clients are connected as requested [here](#).

```
0 * * * * root I=`nvram get wl0_ifname`;[ -z `wl -i $I assoclist` ] && wl -i $I radio off
```

## Show VPN status/activity with SES/Cisco LED

I worked out this little script out for my WRT54G-TM and WRT54GL and I thought I would share here since its based from examples shown on this forum and the wiki.

The script requires that you set 2 options in your vpn config file:

## LED\_Scripts

```
status /path/to/file (example: /tmp/vpn-status)
status-version 2
```

For activity it parses the info in /proc/net/dev

Be sure to edit the first 2 lines (VPN\_STATUS= and VPN\_DEV=) of this script to match your settings.. then Paste the following into your Startup script. Code:

```
vpn_leds() {
VPN_STATUS=/tmp/vpn-status
VPN_DEV=tap0

# make sure status file exists
[ ! -f "$VPN_STATUS" ] && return 1

# Set initial Cisco/SES Led mode
AMBER='gpio disable 3 ; gpio enable 2'
WHITE='gpio enable 3 ; gpio disable 2'
BLACK='gpio enable 3 ; gpio enable 2'
eval $BLACK

LAST_ACT=0
while [ 1 ]; do
    if [ $(cat "$VPN_STATUS" |grep -c "") -gt 6 ]; then
        # tunnel up. White LED
        LED=$WHITE
        ACT_STRING="$(cat /proc/net/dev |grep "$VPN_DEV")"
        if [ "$ACT_STRING" != "$LAST_ACT" ]; then
            LED=$AMBER
            LAST_ACT=$ACT_STRING
        else
            LED=$WHITE
            LAST_ACT=$ACT_STRING
        fi
    else
        # tunnel down. All leds OFF
        LED=$BLACK
    fi
    if [ "$LED" != "$PLED" ]; then
        eval $LED
        LAST_LED=$LED
    fi
    sleep 1
done
}
vpn_leds &
```

Reboot your router and viola!

Light off = Tunnel down

White = Tunnel up

Amber = Activity

You can change the behavior of this script by changing each LED=\$COLOR line.

Response time for tunnel up/down indication for my setup is 60sec for client connect and 120sec for disconnect. I believe the disconnect response could be improved by changing your keepalive setting in your

## LED\_Scripts

VPN config.

Mine is set to: keepalive 10 120

This means OpenVPN will ping the client every 10sec and assume the client has disconnect if no reply is received within 120sec. lowering the last value should improve the response time but be careful not to lower it too much as not to cause connection drops due to timeout

Hope this helps,

-onemyndseye

[Original Post](#)

Update: Fun with egrep!

Here is an updated version that will monitor for activity on ANY tap or tun device that exists. Still only showing up/down status for 1 tunnel

```
vpn_leds() {
VPN_STATUS=/tmp/vpn-status

# make sure status file exists
[ ! -f "$VPN_STATUS" ] && return 1

# Set initial Cisco/SES Led mode
AMBER='gpio disable 3 ; gpio enable 2'
WHITE='gpio enable 3 ; gpio disable 2'
BLACK='gpio enable 3 ; gpio enable 2'
eval $BLACK

LAST_ACT="$(cat /proc/net/dev |egrep "tap|tun")"
while [ 1 ]; do
    if [ $(cat "$VPN_STATUS" |grep -c "") -gt 6 ]; then
        # tunnel up. White LED
        LED=$WHITE
        ACT_STRING="$(cat /proc/net/dev |egrep "tap|tun")"
        if [ "$ACT_STRING" != "$LAST_ACT" ]; then
            LED=$AMBER
            LAST_ACT=$ACT_STRING
        else
            LED=$WHITE
            LAST_ACT=$ACT_STRING
        fi
    else
        # tunnel down. All leds OFF
        LED=$BLACK
    fi
    if [ "$LED" != "$PLED" ]; then
        eval $LED
        LAST_LED=$LED
    fi
    sleep 1
done
```

Show VPN status/activity with SES/Cisco LED

```
}
vpn_leds &
```

## USB Disc Mount Status and Umount Button DIR-320

- SES red LED to indicate disc mounted
- SES button to umount disc
- SES blue LED to indicate unmounting process

Script /jffs/etc/config/mount\_status.startup (disc mount status)

```
#!/bin/sh
mp="/`nvram get usb_mntpoint`"

RED_ON='gpio disable 3'
RED_OFF='gpio enable 3'

while sleep 1; do
  if [ "`mount | grep $mp`" ]; then
    LED=$RED_OFF
  else
    LED=$RED_ON
  fi

  if [ "$LED" != "$PLED" ]; then
    eval $LED
    PLED=$LED
  fi
done
```

Script /jffs/etc/config/mount.sesbutton

```
#!/bin/sh
mp="/`nvram get usb_mntpoint`"
proftpd_enable="`nvram get proftpd_enable`"

RED_ON='gpio disable 3'
RED_OFF='gpio enable 3'
BLUE_ON='gpio disable 4'
BLUE_OFF='gpio enable 4'

if [ "`mount | grep $mp`" ]; then
  $RED_OFF
  $BLUE_ON
  if [ "$proftpd_enable" == "1" ]; then
    killall proftpd
  fi
  umount $mp
  if [ "$proftpd_enable" == "1" ]; then
    proftpd
  fi
fi

if [ "`mount | grep $mp`" ]; then
  $RED_ON
else
  $RED_OFF
```



fi

\$BLUE\_OFF

## Display signal strength over the SES light - Linksys routers

Use this script If you want to catch a better signal without computers.

Script not work in dd-wrt micro version

Linksys routers where this script will work :

- WRT54GL v1.0
- WRT54GL v1.1
- WRT54G-TM
- WRT54G v4.0
- WRT54GS V4

Linksys routers where the script does not work:

- WRT54GS v6.0
- WRT54G v5
- WRTSL54GS

has not yet been tested on other models

Discuss [here](#).

## Script Version 2

Script version two show signal more precise

### SIGNAL STRENGTH

- 0 blink = SNR < 9 No signal or very weak
- 1 blink = SNR 10-19
- 2 blinks = SNR 20-29
- 3 blinks = SNR 30-39
- 4 blinks = SNR 40-49
- 5 blinks = SNR 50-59 Excellent signal
- 6 blinks = SNR 60 > Very strong signal

Installation is pretty simple:

1. Go to "Administration -> Commands"
2. Paste script in "Command Shell"

3. Click on "Save Startup"
4. Reboot router

Works great in Client mode and Repeater mode

To work in AP mode, you must put the mac address

```
#!/bin/sh
#####
#----Created by MOJSO-----Script version 2-----#
#####
#          SIGNAL STRENGTH
# 0 blink  =  SNR < 9  No signal or very weak
# 1 blink  =  SNR 10-19
# 2 blinks =  SNR 20-29
# 3 blinks =  SNR 30-39
# 4 blinks =  SNR 40-49
# 5 blinks =  SNR 50-59  Excellent signal
# 6 blinks =  SNR 60 >  Very strong signal
#####
while sleep 3; do
BLACK=`gpio enable 3 ; gpio enable 2`
S=`wl rssi` # (put mac adress if in AP mod example: S=`wl rssi 00:11:22:33:44:55` )
N=`wl noise`

SC=`expr $S - $N`
RSC=`expr $SC / 10`

if [ $RSC -eq 0 ]; then
gpio enable 3 ; gpio enable 2 ;

elif [ $RSC -eq 1 ]; then
    gpio enable 3 ; gpio disable 2 ;
usleep 500000
    gpio enable 3 ; gpio enable 2 ;

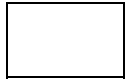
    elif [ $RSC -eq 2 ]; then
        gpio enable 3 ; gpio disable 2 ;
usleep 500000
        gpio enable 3 ; gpio enable 2 ;
        usleep 500000
        gpio enable 3 ; gpio disable 2 ;
usleep 500000
        gpio enable 3 ; gpio enable 2 ;

    elif [ $RSC -eq 3 ]; then
        gpio enable 3 ; gpio disable 2 ;
usleep 250000
        gpio enable 3 ; gpio enable 2 ;
        usleep 250000
        gpio enable 3 ; gpio disable 2 ;
usleep 250000
        gpio enable 3 ; gpio enable 2 ;
        usleep 250000
        gpio enable 3 ; gpio disable 2 ;
usleep 250000
        gpio enable 3 ; gpio enable 2 ;

    elif [ $RSC -eq 4 ]; then
        gpio disable 3 ; gpio enable 2 ;
usleep 250000
```



## Script Version 1



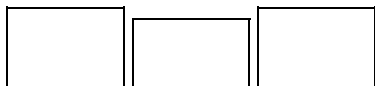
- No signal or very weak



- Low signal



- Good signal



- Excellent signal

Installation is pretty simple:

1. Go to "Administration -> Commands"
2. Paste script in "Command Shell"
3. Click on "Save Startup"
4. Reboot router

Works great in Client mode and Repeater mode

To work in AP mode, you must put the mac address

```
#!/bin/sh
#####Created by MOJSO#####
# White / Black / White - low signal      #
# Amber / Black / Amber - good signal    #
# Amber / White / Amber - excellent signal#
#####
while sleep 3; do
BLACK=`gpio enable 3 ; gpio enable 2`
S=`wl rssi` #(put mac adress if in AP mod example: S=`wl rssi 00:11:22:33:44:55` )
N=`wl noise`

SC=`expr $S - $N`
RSC=`expr $SC / 10`

if [ $RSC -eq 0 ]; then
gpio enable 3 ; gpio enable 2 ;

elif [ $RSC -eq 1 ]; then
    gpio enable 3 ; gpio disable 2 ;
sleep 1
    gpio enable 3 ; gpio enable 2 ;
sleep 1
    gpio enable 3 ; gpio disable 2 ;

elif [ $RSC -eq 2 ]; then
```

## LED\_Scripts

```
    gpio enable 3 ; gpio disable 2 ;
sleep 1
    gpio enable 3 ; gpio enable 2 ;
sleep 1
    gpio enable 3 ; gpio disable 2 ;

elif [ $RSC -eq 3 ]; then
    gpio enable 3 ; gpio disable 2 ;
sleep 1
    gpio enable 3 ; gpio enable 2 ;
    sleep 1
    gpio enable 3 ; gpio disable 2 ;

elif [ $RSC -eq 4 ]; then
    gpio disable 3 ; gpio enable 2 ;
sleep 1
    gpio enable 3 ; gpio enable 2 ;
sleep 1
    gpio disable 3 ; gpio enable 2 ;

elif [ $RSC -eq 5 ]; then
    gpio disable 3 ; gpio enable 2 ;
sleep 1
    gpio enable 3 ; gpio enable 2 ;
sleep 1
    gpio disable 3 ; gpio enable 2 ;

elif [ $RSC -ge 6 ]; then
    gpio disable 3 ; gpio enable 2 ;
sleep 1
    gpio enable 3 ; gpio disable 2 ;
sleep 1
    gpio disable 3 ; gpio enable 2 ;
else gpio enable 3 ; gpio enable 2 ;
fi
done
```

## Display signal strength over the Diagnostic light - Buffalo routers

Use this script If you want to catch a better signal without computers.

Script not work in dd-wrt micro version

Bufalo routers where this script will work :  
- WHR-HP-G54-DD

Has not yet been tested on other models

Discuss [here](#).

## Broadcom based Hardware

### SIGNAL STRENGTH

- 0 blink = SNR < 9 No signal or very weak
- 1 blink = SNR 10-19
- 2 blinks = SNR 20-29
- 3 blinks = SNR 30-39
- 4 blinks = SNR 40-49
- 5 blinks = SNR 50-59 Excellent signal
- 6 blinks = SNR 60 > Very strong signal

Installation is pretty simple:

1. Go to "Administration -> Commands"
2. Paste script in "Command Shell"
3. Click on "Save Startup"
4. Reboot router

Works great in Client mode and Repeater mode

To work in AP mode, you must put the MAC address

```
#!/bin/sh
#####
#-----Created by MOJSO-----#
#####
#          SIGNAL STRENGTH
# 0 blink  = SNR < 9 No signal or very weak
# 1 blink  = SNR 10-19
# 2 blinks = SNR 20-29
# 3 blinks = SNR 30-39
# 4 blinks = SNR 40-49
# 5 blinks = SNR 50-59 Excellent signal
# 6 blinks = SNR 60 > Very strong signal
#####
while sleep 3; do
BLACK=`gpio enable 7`
S=`wl rssi` #(put mac adress if in AP mod example: S=`wl rssi 00:11:22:33:44:55` )
N=`wl noise`

SC=`expr $S - $N`
RSC=`expr $SC / 10`

if [ $RSC -eq 0 ]; then
gpio enable 7 ;

elif [ $RSC -eq 1 ]; then
gpio disable 7 ;
usleep 500000
gpio enable 7 ;
```

## LED\_Scripts

```
    elif [ $RSC -eq 2 ]; then
        gpio disable 7 ;
    usleep 500000
        gpio enable 7 ;
        usleep 500000
        gpio disable 7 ;
    usleep 500000
        gpio enable 7 ;

    elif [ $RSC -eq 3 ]; then
        gpio disable 7 ;
    usleep 250000
        gpio enable 7 ;
        usleep 250000
        gpio disable 7 ;
    usleep 250000
        gpio enable 7 ;
        usleep 250000
        gpio disable 7 ;
    usleep 250000
        gpio enable 7 ;

    elif [ $RSC -eq 4 ]; then
        gpio disable 7 ;
    usleep 250000
        gpio enable 7 ;
        usleep 250000
        gpio disable 7 ;
    usleep 250000
        gpio enable 7 ;
        usleep 250000
        gpio disable 7 ;
    usleep 250000
        gpio enable 7 ;
        usleep 250000
        gpio disable 7 ;
    usleep 250000
        gpio enable 7 ;

    elif [ $RSC -eq 5 ]; then
        gpio disable 7 ;
    usleep 250000
        gpio enable 7 ;
        usleep 250000
        gpio disable 7 ;
    usleep 250000
        gpio enable 7 ;
        usleep 250000
        gpio disable 7 ;
    usleep 250000
        gpio enable 7 ;
        usleep 250000
        gpio disable 7 ;
    usleep 250000
        gpio enable 7 ;
        usleep 250000
        gpio disable 7 ;
    usleep 250000
        gpio enable 7 ;

    elif [ $RSC -ge 6 ]; then
        gpio disable 7 ;
```

## LED\_Scripts

```
usleep 250000
  gpio enable 7 ;
  usleep 250000
  gpio disable 7 ;
usleep 250000
  gpio enable 7 ;
  usleep 250000
  gpio disable 7 ;
usleep 250000
  gpio enable 7 ;
  usleep 250000
  gpio disable 7 ;
usleep 250000
  gpio enable 7 ;
  usleep 250000
  gpio disable 7 ;
usleep 250000
  gpio enable 7 ;
  usleep 250000
  gpio disable 7 ;
usleep 250000
  gpio enable 7 ;
  else gpio enable 7 ;
fi
done
```

## Atheros based Hardware

Here is a variant for the WHR-HP-G300N:

```
#!/bin/sh
#####
#-----Created by MOJSO-----#
#----- Altered for WHR-HP-G300N by PASOC -----#
#-----1 Adapted for Atheros compatibility --#
#-----2 New gpio mappints -----#
#-----3 Autodetect MAC -----#
#-----4 Tweaked Signal Strength Output -----#
#####
#                               SIGNAL STRENGTH
# 0 blink                       = SNR < 9      No signal or very weak
# 1 blink                       = SNR 10-19
# 2 blinks                      = SNR 20-29
# 3 blinks                      = SNR 30-39
# 4 blinks                      = SNR 40-49
# 5 blinks                      = SNR 50-59      Excellent signal
# 6 blinks                      = SNR 60-69
# 7 blinks                      = SNR 70-79
# 8 blinks                      = SNR 80 >       Sitting 3 feet away
#####
# Pin  Direction      Use
# GPIO 1      Output   DIAG LED (labeld LED8 ON PCB)
# GPIO 8      Output   Bridge LED (labeled LED10 on PCB)
#
# wl_atheros assoclist
```





## LED\_Scripts

```
usleep 150000
gpio disable 1 ;
usleep 250000
gpio enable 1 ;
usleep 150000
gpio disable 1 ;
usleep 250000
gpio enable 1 ;
usleep 150000
gpio disable 1 ;
usleep 250000
gpio enable 1 ;
elif [ $RSC -eq 6 ]; then
gpio disable 1 ;
usleep 275000
gpio enable 1 ;
usleep 125000
gpio disable 1 ;
usleep 275000
gpio enable 1 ;
usleep 125000
gpio disable 1 ;
usleep 275000
gpio enable 1 ;
usleep 125000
gpio disable 1 ;
usleep 275000
gpio enable 1 ;
usleep 125000
gpio disable 1 ;
usleep 275000
gpio enable 1 ;
usleep 125000
gpio disable 1 ;
usleep 275000
gpio enable 1 ;
elif [ $RSC -eq 7 ]; then
gpio disable 1 ;
usleep 300000
gpio enable 1 ;
usleep 125000
gpio disable 1 ;
usleep 300000
gpio enable 1 ;
usleep 125000
gpio disable 1 ;
usleep 300000
gpio enable 1 ;
usleep 125000
gpio disable 1 ;
usleep 300000
gpio enable 1 ;
usleep 125000
gpio disable 1 ;
usleep 300000
gpio enable 1 ;
usleep 125000
gpio disable 1 ;
usleep 300000
gpio enable 1 ;
usleep 125000
gpio disable 1 ;
```



## LED\_Scripts

```
mp="/`nvram get usb_mntpoint`"  
proftpd_enable="`nvram get proftpd_enable`"  
smbd_enable="`nvram get samba3_enable`"  
  
QSS_ON='gpio enable 5'  
QSS_OFF='gpio disable 5'  
USB_ON='gpio disable 1'  
USB_OFF='gpio enable 1'  
  
if [ "`mount | grep $mp`" ]; then  
    $QSS_ON  
if [ "$proftpd_enable" == "1" ]; then  
    killall proftpd  
fi  
if [ "$smbd_enable" == "1" ]; then  
    killall nmbd  
    killall smbd  
fi  
    umount $mp  
fi  
$QSS_OFF  
if [ "`mount | grep $mp`" ]; then  
    $USB_ON  
else  
    $USB_OFF  
fi
```

## See Also

[Startup Scripts](#)  
[Script Examples](#)  
[Useful Scripts](#)