# How to setup wifi connection

# How to setup wifi connection

*Stable: 09.03.2020 - 08:21 / Revision: 06.03.2020 - 13:51*

## Contents

1 ifconfig ..... 2
1.1 Check WLAN interface ..... 2
1.2 Initialize WLAN interface ..... 3
2 iw ..... 3
2.1 Scan available SSID (AP) ..... 3
3 Wlan Network attachment (without system which manage systemd-networkd configuration) ..... 4
3.1 Configure your WiFi connection ..... 4
3.1.1 Connect to SSID ..... 5
3.1.2 link to SSID ..... 5
3.1.3 Assign IP address to WLAN interface ..... 5
3.1.4 Check connectivity ..... 6
4 Automatic WiFi configuration at start up ..... 6
4.1 networkctl ..... 6
4.2 How to set a wireless configuration with networkd ..... 7

Even if the example is related to the RTL8723BU chip (this chip calls for a WiFi (minimally 802.11g/n) and Bluetooth 4.0 LE), most of the commands are generic ones and can be applied to any platform.

# 1 ifconfig

Verify the wlan0 interface is present (that means drivers have been started and the WiFi firmware loaded):

## 1.1 Check WLAN interface

```
Board $> ifconfig -a
lo        Link encap:Local Loopback
          LOOPBACK  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

```
wlan0     Link encap:Ethernet  HWaddr 60:F1:89:3F:F6:0E
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:10 (10.0 B)
```

## 1.2 Initialize WLAN interface

```
Board $> ifconfig wlan0  192.168.43.135  broadcast 192.168.43.255  netmask
255.255.255.0
```

```
Board $> ifconfig wlan0 up
```

```
Board $> ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 60:F1:89:3F:F6:0E
          inet addr:192.168.43.135  Bcast:192.168.43.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1                    /* UP
indicate that your interface is UP */
          RX packets:19 errors:0 dropped:0 overruns:0 frame:0
          TX packets:19 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1774 (1.7 KiB)  TX bytes:2326 (2.2 KiB)
```

# 2 iw

The next phase is to scan for any wireless access points with the command:

## 2.1 Scan available SSID (AP)

- Get only SSID name

```
Board $> iw dev wlan0 scan |grep SSID
        SSID: NETWORK1
        SSID: NETWORK2
```

- List full SSID information

```
Board $> iw dev wlan0 scan
BSS 00:23:5e:4a:28:f9(on wlan0)
        TSF: 0 usec (0d, 00:00:00)
        freq: 2412
        beacon interval: 100 TUs
        capability: ESS ShortPreamble ShortSlotTime (0x0421)
        signal: -72.00 dBm
        last seen: 0 ms ago
        SSID: NETWORK1
        Supported rates: 1.0* 2.0* 5.5* 6.0 9.0 11.0* 12.0 18.0
        DS Parameter set: channel 1
        TIM: DTIM Count 0 DTIM Period 1 Bitmap Control 0x0 Bitmap[0] 0x2
        Country: FR     Environment: Indoor/Outdoor
                Channels [1 - 13] @ 20 dBm
        BSS Load:
                 * station count: 1
                 * channel utilisation: 30/255
```

```
                 * available admission capacity: 23437 [*32us]
        ERP: <no flags>
        Extended supported rates: 24.0 36.0 48.0 54.0
        WMM:     * Parameter version 1
                 * u-APSD
                 * BE: CW 15-1023, AIFSN 3
                 * BK: CW 15-1023, AIFSN 7
                 * VI: CW 7-15, AIFSN 2, TXOP 3008 usec
                 * VO: CW 3-7, AIFSN 2, TXOP 1504 usec
BSS 00:23:5e:96:57:20(on wlan0)
        TSF: 0 usec (0d, 00:00:00)
        freq: 2412
        beacon interval: 100 TUs
        capability: ESS Privacy ShortPreamble ShortSlotTime (0x0431)
        signal: -66.00 dBm
        last seen: 0 ms ago
        SSID: NETWORK2
        Supported rates: 1.0* 2.0* 5.5* 6.0 9.0 11.0* 12.0 18.0
        DS Parameter set: channel 1
        Country: FR     Environment: Indoor/Outdoor
                Channels [1 - 13] @ 20 dBm
        BSS Load:
                 * station count: 8
                 * channel utilisation: 48/255
                 * available admission capacity: 23437 [*32us]
        ERP: <no flags>
        RSN:     * Version: 1
                 * Group cipher: CCMP
                 * Pairwise ciphers: CCMP
                 * Authentication suites: IEEE 802.1X 00-40-96:0
                 * Capabilities: 4-PTKSA-RC 4-GTKSA-RC (0x0028)
        Extended supported rates: 24.0 36.0 48.0 54.0
        WMM:     * Parameter version 1
                 * u-APSD
                 * BE: CW 15-1023, AIFSN 3
                 * BK: CW 15-1023, AIFSN 7
                 * VI: CW 7-15, AIFSN 2, TXOP 3008 usec
                 * VO: CW 3-7, AIFSN 2, TXOP 1504 usec
```

# 3 Wlan Network attachment (without system which manage systemd-networkd configuration)

## 3.1 Configure your WiFi connection

Configure WiFi connection by using **wpa_supplicant** tool

- Check current configuration

```
Board $> cat /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1
network={
   key_mgmt=NONE
}
```

- Set the WiFi network name and password

```
Board $> wpa_passphrase <your_ssid_name> <your_ssid_key> >> /etc/wpa_supplicant.conf
```

- Check new configuration

```
Board $> cat /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1
network={
       ssid="your_ssid_name"
       psk="your_ssid_key"
}
```

If no WPA key replace **psk=...** with **key_mgmt=NONE**

#### 3.1.1 Connect to SSID

```
Board $> wpa_supplicant -B -iwlan0 -c /etc/wpa_supplicant.conf
Successfully initialized wpa_supplicant
```

#### 3.1.2 link to SSID

```
Board $> iw wlan0 link
       SSID: NETWORK1
       freq: 2462
       RX: 501 bytes (3 packets)
       TX: 4056 bytes (22 packets)
       signal: -75 dBm
       tx bitrate: 12.0 MBit/s
       bss flags:      short-preamble short-slot-time
       dtim period:    1
       beacon int:     100
```

#### 3.1.3 Assign IP address to WLAN interface

Use the DHCP client to obtain an address (assuming wireless network (associated to) has a DHCP server):

```
Board $> dhclient wlan0
```

Use the ip command to verify the IP address assigned by the DHCP. The IP address is 192.168.43.135 from below.

```
Board $> ip addr show wlan0
3: wlan0:  mtu 1500 qdisc mq state UP qlen 1000
  link/ether 74:e5:43:a1:ce:65 brd ff:ff:ff:ff:ff:ff
  inet 192.168.43.135/24 brd 192.168.1.255 scope global wlan0
  inet6 fe80::76e5:43ff:fea1:ce65/64 scope link
     valid_lft forever preferred_lft forever
```

### 3.1.4 Check connectivity

The most basic connectivity test is to use the “ping” command. In this example, the wireless router (associated to) has an IP address of 192.168.43.1:

```
Board $> ping 192.168.43.1
PING 192.168.43.1 (192.168.43.1): 56 data bytes
64 bytes from 192.168.43.1: seq=0 ttl=64 time=14.905 ms
64 bytes from 192.168.43.1: seq=1 ttl=64 time=30.387 ms
64 bytes from 192.168.43.1: seq=2 ttl=64 time=20.462 ms
```

- Note : Enter <CTRL+C> to terminate the ping session.

# 4 Automatic WiFi configuration at start up

## 4.1 networkctl

Systemd has a specific service for the network named **systemd-networkd**, this service comes with the tool **networkctl** which allows to show the status of each network interface.

```
Board $> networkctl --no-pager
 IDX LINK             TYPE               OPERATIONAL SETUP
 1 lo               loopback           carrier     unmanaged
 2 eth0             ether              routable    configured
 3 sit0             sit                no-carrier         unmanaged
 4 ip6tnl0          tunnel6            no-carrier         unmanaged
 5 wlan0            wlan               no-carrier         unmanaged
 6 wlan1            wlan               no-carrier         unmanaged
```

6 links listed.

We can see that the **eth0** interface is managed by networked via the information **configured** and the interface is used ( **routable**).

## 4.2 How to set a wireless configuration with networkd

The goal is to configure an wlan network interface via systemd-networkd configuration.
All the network configurations are stored on **/lib/systemd/network** or **/etc/systemd/network**
Create the file dedicated to wireless interface "/lib/systemd/network/51-wireless.network" :

```
Board $> echo "[Match]" > /lib/systemd/network/51-wireless.network
Board $> echo "Name=wlan0" >> /lib/systemd/network/51-wireless.network
Board $> echo "[Network]" >> /lib/systemd/network/51-wireless.network
Board $> echo "DHCP=ipv4" >> /lib/systemd/network/51-wireless.network
```

Check content is as follow

```
Board $> cat   /lib/systemd/network/51-wireless.network
 [Match]
 Name=wlan0
 [Network]
 DHCP=ipv4
```

For attaching this wireless interface to a specific network, we need to have some information like SSID of network and password.

To see the list of wireless network available:

```
Board $> ifconfig wlan0 up
Board $> iw dev wlan0 scan |grep SSID
       SSID: NETWORK1
       SSID: NETWORK2
```

Associate the wireless network to wireless interface, here wlan0:

```
Board $> mkdir -p /etc/wpa_supplicant/
         echo "ctrl_interface=/var/run/wpa_supplicant" > /etc/wpa_supplicant
/wpa_supplicant-wlan0.conf
         echo "eapol_version=1" >> /etc/wpa_supplicant/wpa_supplicant-wlan0.conf
         echo "ap_scan=1" >> /etc/wpa_supplicant/wpa_supplicant-wlan0.conf
         echo "fast_reauth=1" >> /etc/wpa_supplicant/wpa_supplicant-wlan0.conf
         echo "" >> /etc/wpa_supplicant/wpa_supplicant-wlan0.conf
         wpa_passphrase SSID_OF_NETWORK PASSWORD_OF_NETWORK >> /etc/wpa_supplicant
/wpa_supplicant-wlan0.conf
```

Where **SSID_OF_NETWORK PASSWORD_OF_NETWORK** correspond to the SSID and password of wireless network.

the name of file on /etc/wpa_supplicant must contains the name of wireless interface:
wpa_supplicant-**wlan0**.conf

To enable and start the wireless configuration:

```
Board $> systemctl enable wpa_supplicant@wlan0.service
          systemctl restart systemd-networkd.service
          systemctl restart wpa_supplicant@wlan0.service
```

Receive

Transmit

uniprocessor