NAME

run - Ralink Technology USB IEEE 802.11a/g/n wireless network device

SYNOPSIS

To compile this driver into the kernel, place the following lines in your kernel configuration file:

```
device ehci
device uhci
device ohci
device usb
device run
device wlan
device wlan_amrr
```

Firmware is also needed, and provided by:

```
device runfw
```

Alternatively, to load the driver as a module at boot time, place the following lines in loader.conf(5):

```
if_run_load="YES"
runfw_load="YES"
```

DESCRIPTION

The **run** driver supports USB 2.0 wireless adapters based on the Ralink RT2700U, RT2800U, RT3000U and RT3900E chipsets.

The RT2700U chipset consists of two integrated chips, an RT2770 MAC/BBP and an RT2720 (1T2R) or RT2750 (dual-band 1T2R) radio transceiver.

The RT2800U chipset consists of two integrated chips, an RT2870 MAC/BBP and an RT2820 (2T3R) or RT2850 (dual-band 2T3R) radio transceiver.

The RT3000U is a single-chip solution based on an RT3070 MAC/BBP and an RT3020 (1T1R), RT3021 (1T2R) or RT3022 (2T2R) single-band radio transceiver.

The RT3900E is a single-chip USB 2.0 802.11n solution. The MAC/Baseband Processor can be an RT3593, RT5390, RT5392 or an RT5592. The radio can be an RT3053, RT5370, RT5372 or an RT5572. The RT3053 chip operates in the 2GHz and 5GHz spectra and supports up to 3 transmit paths and 3 receiver paths (3T3R). The RT5370 chip operates in the 2GHz spectrum and supports 1 transmit path and 1 receiver path (1T1R). The RT5372 chip operates in the 2GHz spectrum and supports up to 2 transmit paths and 2 receiver paths (2T2R). The RT5572 chip operates in the 2GHz and 5GHz spectra and supports up to 2 transmit paths and 2 receiver paths (2T2R).

These are the modes the **run** driver can operate in:

BSS mode
: Also known as *infrastructure* mode, this is used when associating with an access point, through which all traffic passes. This mode is the default.

Host AP mode
: In this mode the driver acts as an access point (base station) for other cards.

monitor mode
: In this mode the driver is able to receive packets without associating with an access point. This disables the internal receive filter and enables the card to capture packets from networks which it wouldn't normally have access to, or to scan for access points.

The **run** driver can be configured to use Wired Equivalent Privacy (WEP) or Wi-Fi Protected Access (WPA-PSK and WPA2-PSK). WPA is the de facto encryption standard for wireless networks. It is strongly recommended that WEP not be used as the sole mechanism to secure wireless communication, due to serious weaknesses in it. The **run** driver offloads both encryption and decryption of data frames to the hardware for the WEP40, WEP104, TKIP(+MIC) and CCMP ciphers.

The **run** driver can be configured at runtime with ifconfig(8).

HARDWARE

The **run** driver supports the following wireless adapters:

Airlink101 AWLL6090
ASUS USB-N11
ASUS USB-N13 ver. A1
ASUS USB-N14
ASUS USB-N66
ASUS WL-160N
Belkin F5D8051 ver 3000
Belkin F5D8053
Belkin F5D8055
Belkin F6D4050 ver 1
Belkin F9L1103
Buffalo WLI-UC-AG300N
Buffalo WLI-UC-G300HP
Buffalo WLI-UC-G300N

Buffalo WLI-UC-G301N
Buffalo WLI-UC-GN
Buffalo WLI-UC-GNM
Buffalo WLI-UC-GNM2
Corega CG-WLUSB2GNL
Corega CG-WLUSB2GNR
Corega CG-WLUSB300AGN
Corega CG-WLUSB300GNM
D-Link DWA-130 rev B1
D-Link DWA-130 rev F1
D-Link DWA-140 rev B1, B2, B3, D1
D-Link DWA-160 rev B2
D-Link DWA-162
DrayTek Vigor N61
Edimax EW-7711UAn
Edimax EW-7711UTn
Edimax EW-7717Un
Edimax EW-7718Un
Edimax EW-7733UnD
Gigabyte GN-WB30N
Gigabyte GN-WB31N
Gigabyte GN-WB32L
Hawking HWDN1
Hawking HWUN1
Hawking HWUN2
Hercules HWNU-300
Linksys WUSB54GC v3
Linksys WUSB600N
Logitec LAN-W150N/U2
Mvix Nubbin MS-811N
Panda Wireless PAU06
Planex GW-USMicroN
Planex GW-US300MiniS
Sitecom WL-182
Sitecom WL-188
Sitecom WL-301
Sitecom WL-302
Sitecom WL-315
SMC SMCWUSBS-N2
Sweex LW303

Sweex LW313
TP-LINK TL-WDN3200
TP-LINK TL-WN321G v4
TP-LINK TL-WN727N v3
Unex DNUR-81
Unex DNUR-82
ZyXEL NWD2705
ZyXEL NWD210N
ZyXEL NWD270N

EXAMPLES

Join an existing BSS network (i.e., connect to an access point):

```
ifconfig wlan create wlandev run0 inet 192.168.0.20 \
    netmask 0xffffff00
```

Join a specific BSS network with network name "my_net":

```
ifconfig wlan create wlandev run0 ssid my_net up
```

Join a specific BSS network with 64-bit WEP encryption:

```
ifconfig wlan create wlandev run0 ssid my_net \
                wepmode on wepkey 0x1234567890 weptxkey 1 up
```

Join a specific BSS network with 128-bit WEP encryption:

```
ifconfig wlan create wlandev run0 wlanmode adhoc ssid my_net \
    wepmode on wepkey 0x01020304050607080910111213 weptxkey 1
```

DIAGNOSTICS

run%d: failed load firmware of file runfw For some reason, the driver was unable to read the microcode file from the filesystem. The file might be missing or corrupted.

run%d: could not load 8051 microcode An error occurred while attempting to upload the microcode to the onboard 8051 microcontroller unit.

run%d: device timeout A frame dispatched to the hardware for transmission did not complete in time. The driver will reset the hardware. This should not happen.

SEE ALSO

intro(4), netintro(4), runfw(4), usb(4), wlan(4), wlan_amrr(4), wlan_ccmp(4), wlan_tkip(4), wlan_wep(4), wlan_xauth(4), hostapd(8), ifconfig(8), wpa_supplicant(8)

HISTORY

The **run** driver first appeared in OpenBSD 4.5.

AUTHORS

The **run** driver was written by Damien Bergamini <*damien@openbsd.org*>.

CAVEATS

The **run** driver supports some of the 11n capabilities found in the RT2800, RT3000 and RT3900 chipsets.