



APPLICATION NOTE

AN60301

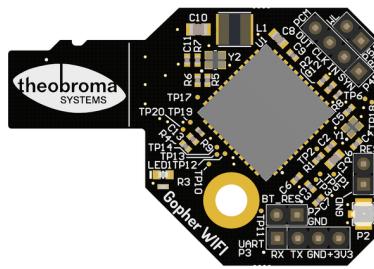
Using the Gopher Wifi module on
the RK3399-Q7 Devkit

Document revision: Release v1.1
Issue date: Jan 31, 2018

Contents

1	Introduction	1
1.1	Prerequisites	1
1.2	Firmware Blobs	1
1.3	Kernel Configuration	2
1.4	Kernel Compile	2
1.5	NetworkManager Configuration	2
1.6	Reboot & wlan0 Check	3
1.7	Testing Wifi Connectivity	3
2	Troubleshooting	5
2.1	Antenna & Signal Strength	5
2.2	MicroSD Slot Electrical Connection	5
2.3	Expected dmesg Output	5

1 Introduction



The Gopher WiFi module is a high-performance WiFi adapter based on the Broadcom BCM43455. Featuring an SDIO interface and a MicroSD connector, it can be used in the MicroSD slot on a RK3399-Q7 development kit.

1.1 Prerequisites

It is assumed that compiling the standard kernel, as described in the RK3399-Q7 User Manual, is already working. The relevant section of the User Manual is available on the web at:

<https://www.theobroma-systems.com/rk3399-q7-user-manual/04-software.html>

The most recent PDF version is always listed at:

<https://www.theobroma-systems.com/rk3399-q7/resources>

1.2 Firmware Blobs

The WiFi module needs two firmware files to work:

```
brcmfmac43455-sdio.bin  
brcmfmac43455-sdio.txt
```

The firmware will be sent to you on request via email. Both files must be placed in the kernel source tree into the folder:

```
firmware/brcmn
```

Kernel compilation fails if these files are not placed in the right folder.

1.3 Kernel Configuration

Compared the the standard RK3399-Q7 kernel, the BRCMFMAC driver has to be enabled and the firmware blobs for the Wifi module should be built into the kernel. Including the firmware in the kernel image avoids the problem that the Wifi module may be detected before the root filesystem is mounted.

The kernel source tree at <https://git.theobroma-systems.com/puma-linux.git> contains a branch that has the neccessary changes:

```
git checkout puma-release-4.4-2018-01-11-gopher
make puma-rk3399_defconfig
```

1.4 Kernel Compile

Compile the kernel as usual:

```
make -j4
```

and copy the Image file to your RK3399-Q7 module.

1.5 NetworkManager Configuration

Due to a pathological interaction between MAC address randomization and the BRCMFMAC driver (https://bugzilla.redhat.com/show_bug.cgi?id=1382741), it is neccessary to disable this function on your RK3399-Q7 module. Add this snippet to /etc/NetworkManager/NetworkManager.conf:

```
[device]
wifi.scan-rand-mac-address=no
```

1.6 Reboot & wlan0 Check

After a reboot, the wlan0 device should show up the ip a output:

```
# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 1000
    link/ether 76:b9:4c:99:9e:59 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::b550:31c4:48aa:3da6/64 scope link
        valid_lft forever preferred_lft forever
3: sit0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen 1
    link/sit 0.0.0.0 brd 0.0.0.0
4: wlan0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000
    link/ether cc:b8:a8:c3:54:30 brd ff:ff:ff:ff:ff:ff
5: can0: <NOARP> mtu 16 qdisc noop state DOWN group default qlen 10
    link/can
```

If it does not, check the Troubleshooting chapter for advice. Otherwise, continue to the next section.

1.7 Testing Wifi Connectivity

Use nmcli device wifi list to list available Wifi networks. Below you see the networks available at the Theobroma Systems office:

```
# nmcli device wifi list
*  SSID           MODE   CHAN  RATE      SIGNAL  BARS  SECURITY
  Theobroma Systems 2.4GHz  Infra   6   54 Mbit/s  44      **  WPA1 WPA2
  Theobroma Guest 2.4GHz  Infra   2   54 Mbit/s  42      **  WPA2
  ITandTEL HOTSPOT    Infra   1   54 Mbit/s  39      ** 
  GETec 2_4           Infra   1   54 Mbit/s  35      **  WPA2
  GETecGuest          Infra   1   54 Mbit/s  35      **  WPA2
  Theobroma Systems 5GHz  Infra  44   54 Mbit/s  29      *  WPA1 WPA2
  Theobroma Guest 5GHz  Infra  44   54 Mbit/s  29      *  WPA2
```

Connect to a network using `nmcli device wifi connect`. Connecting to the Theobroma Guest 2.4GHz network works like this:

```
# nmcli device wifi connect "Theobroma Guest 2.4GHz" password "xxxxxx"
```

And results in this output:

```
[ 136.957676] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
Device 'wlan0' successfully activated with 'd5f50081-d17c-4205-8e49-ee10c730e975'.
```

You should now be able to ping your Wifi router:

```
# ping 10.5.0.1
PING 10.5.0.1 (10.5.0.1) 56(84) bytes of data.
64 bytes from 10.5.0.1: icmp_seq=1 ttl=64 time=6.91 ms
64 bytes from 10.5.0.1: icmp_seq=2 ttl=64 time=10.0 ms
64 bytes from 10.5.0.1: icmp_seq=3 ttl=64 time=14.9 ms
64 bytes from 10.5.0.1: icmp_seq=4 ttl=64 time=7.33 ms
```

Congratulations, you now have a fully functional Wifi connection.

2 Troubleshooting

2.1 Antenna & Signal Strength

The Gopher Wifi module has excellent receive sensitivity and will in many cases be able to connect to Wifi networks even without an external antenna.

To provide better signal strength, an UFL connector is available to connect a standard antenna. Good results have been achieved with the Pulse W1039B030 dual-band antenna:

- Manufacturer: Pulse
- Manufacturer part number: W1039B030
- Farnell order number: 1900063

Direct link: <http://uk.farnell.com/pulse-engineering/w1039b030/antenna-ext-2-4-5-5ghz-ipex-artic/dp/1900063>

2.2 MicroSD Slot Electrical Connection

Despite the “click” sound on insertion, MicroSD slots do not have a mechanical retention. When inserting the Gopher module, make sure it is inserted completely, you hear the click, and the module springs back only about 1mm. If it springs back about 5mm, the slot is in the “ejected” state and there will be no electrical connection.

2.3 Expected dmesg Output

Should the Wifi module not be detected on your setup, comparing with the expected kernel output can give clues as to what went wrong. Below you see the output from a successful detection:

```
[ 20.342271] rockchip-iodomain ff770000.syscon:io-domains: Setting to 3300000 done
[ 20.349867] rockchip-iodomain ff770000.syscon:io-domains: Setting to 3300000 done
[ 20.401722] mmc0: queuing unknown CIS tuple 0x80 (2 bytes)
[ 20.411068] mmc0: queuing unknown CIS tuple 0x80 (3 bytes)
[ 20.420028] mmc0: queuing unknown CIS tuple 0x80 (3 bytes)
[ 20.430434] mmc0: queuing unknown CIS tuple 0x80 (7 bytes)
```

```
[ 20.441377] mmc0: queuing unknown CIS tuple 0x81 (9 bytes)
[ 20.507376] mmc_host mmc0: Bus speed (slot 0) = 10000000Hz (slot req 10000000Hz, actual 10000000Hz
→div = 0)
[ 20.531544] mmc0: queuing unknown CIS tuple 0x91 (3 bytes)
[ 20.537163] mmc0: new high speed SDIO card at address 0001
[ 20.547565] brcmfmac: F1 signature read @0x18000000=0x15264345
[ 20.560238] brcmf_sdio_drivestrengthinit: No SDIO Drive strength init done for chip 4345 rev 6
→pmurev 27
[ 21.135778] brcmf_c_preinit_dcmds: Firmware version = wl0: Oct 23 2017 14:52:12 version 7.45.96.6
→(2af1ebc@sh-git) (r) FWID 01-bc98efaf es7.c5.n4.a3
[ 21.223821] brcmf_cfg80211_reg_notifier: not a ISO3166 code
[ 21.298954] IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
[ 21.341959] brcmf_add_if: ERROR: netdev:wlan0 already exists
[ 21.347693] brcmf_add_if: ignore IF event
[ 21.373717] IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
[ 21.504140] brcmf_add_if: ERROR: netdev:wlan0 already exists
[ 21.509902] brcmf_add_if: ignore IF event
[ 21.594529] IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
```