

AIW FAQ

How to establish a data connection using EWM-341 on RK3588 (Debian Linux)			
Category	EWM	Date	2025/12/30
Keyword	EWM-341CQGI1, Cellular, Debian, Rockchip RK3588, ASR-A501		
HW model	ASR-A501 (Rockchip RK3588)		
OS version	Debian 12 (Linux kernel version: 6.1.75-rt16) (Image version: DIV110057) BSP version: RK3588_RISC_DIV110057		

Category

1.	Build Environment on Host.....	2
2.	Enable the MBIM driver in the kernel	2
3.	Build Kernel Image.....	2
4.	Flash Image	2
5.	Data connection setup [On target device]	3

This document is for establishing a cellular data connection using the EWM-341 module on a Rockchip RK3588 system running Debian 12. It details the necessary steps for enabling MBIM driver in the kernel, rebuilding the kernel image, and verifying hardware detection on the target device. Finally, the instructions show you how to configure the network profile and executing “mbim-network” and “dhclient” commands to successfully connect to the internet.

1. Build Environment on Host

For detailed instructions on setting up the compilation environment, please refer to the [MEMO RK3588 Debian Build Process manual](#). Before proceeding with the following steps, please ensure you start and enter the docker environment

2. Enable the MBIM driver in the kernel

The RK3588 kernel configuration file needs to be modified to enable the USB_NET_CDC_MBIM driver for MBIM mode support.

Command and Response	Description
<pre>\$ cd ~/BSP/ASRA501A2EG90AIM38DBV110057_2025-04-08/kernel/arch/arm64/configs</pre>	Go to the path of RK3588 kernel config
<pre>\$ vi rk3588_adv_defconfig</pre> <pre>CONFIG_USB_NET_CDC_NCM=y # CONFIG_USB_NET_HUAWEI_CDC_NCM is not set CONFIG_USB_NET_CDC_MBIM=y # CONFIG_USB_NET_DM9601 is not set # CONFIG_USB_NET_SR9700 is not set</pre>	Open the “rk3588_adv_defconfig” file to edit. Enable the “CONFIG_USB_NET_CDC_MBIM” config and save the file.

3. Build Kernel Image

After enabling the CDC-MBIM driver, the kernel must be rebuilt to take effect.

Command and Response	Description
<pre>\$ cd ~/BSP/ASRA501A2EG90AIM38DBV110057_2025-04-08 \$./build.sh clean:kernel \$./build.sh kernel</pre>	Clean and rebuild the kernel.

4. Flash Image

For image building and flashing , you can refer to “4.2 Image building” and “5.3 Flash image” in the [MEMO RK3588 Debian Build Process manual](#).

5. Data connection setup [On target device]

Command and Response	Description
<pre>\$ sudo su # apt update # apt install libmbim-utils</pre>	Install the MBIM package.

The following steps show how to make sure the cellular module is detected before establishing a data connection

Command and Response	Description
<pre># lsusb Bus 002 Device 002: ID 1bc7:1061 Telit Wireless Solutions LN920 Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub Bus 003 Device 002: ID 046d:c548 Logitech, Inc. USB Receiver</pre>	List all USB devices to verify the module's USB PID and VID.
<pre># ls /dev/ttyUSB* /dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2 /dev/ttyUSB3 /dev/ttyUSB4</pre>	There should be 5 COM ports visible.
<pre># ls /dev/cdc-wdm* /dev/cdc-wdm0</pre>	List the CDC-WDM device node name.
<pre># ifconfig wwan0 wwan0 flags=4226<BROADCAST,NOARP,MULTICAST> mtu 1500 ether 56:34:A4:C7:FC:0E txqueuelen 1000 (Ethernet) RX bytes:0 (0.0 B) RX packets:0 errors:0 dropped:0 overruns:0 frame:0 TX bytes:0 (0.0 B) TX packets:0 errors:0 dropped:0 overruns:0 carrier:0</pre>	"wwan0" is the module's network interface.

Before establishing a data connection, ensure the module is in the MBIM mode and the MBIM configuration file "mbim-network.conf" includes APN, username and password information. If the module is already in MBIM mode, you can skip these two optional steps.

Command and Response	Description
<pre># echo -e "AT#USBCFG=2" > /dev/ttyUSB2</pre>	Configure the module to MBIM mode (optional)
<pre># reboot</pre>	A system reboot is required for the changes to take effect. (optional)
<pre># vi /etc/mbim-network.conf APN=internet APN_USER= APN_PASS= APN_AUTH= PROXY=yes</pre>	Ensure "mbim-network.conf" includes a correct APN for data connection setup.
<pre># mbim-network /dev/cdc-wdm0 start Loading profile at /etc/mbim-network.conf... APN: internet APN auth protocol: unset APN user: unset APN password: unset mbim-proxy: yes Querying subscriber ready status 'mbimcli -d /dev/cdc-wdm0 --query-subscriber-ready-status --device-open-proxy'... [/dev/cdc-wdm0] Subscriber ready status retrieved: Ready state: 'initialized' Subscriber ID: '466924133395866' SIM ICCID: '89886920041333958665' Ready info:</pre>	Establish the data connection with "mbim-network" command and the CDC-WDM device node.

```
'none' Telephone numbers: (1) '+886975569283'
Querying registration state 'mbimcli -d /dev/cdc-wdm0
--query-registration-state --device-open-proxy'...
[/dev/cdc-wdm0] Registration status: Network error: 'unknown' Register state: 'home'
Register mode: 'automatic' Available data classes: 'lte' Current
cellular class: 'gsm' Provider ID: '46692' Provider name: 'Chunghwa Telecom' Roaming
text: 'unknown' Registration flags: 'packet-service-automatic-attach'
Attaching to packet service with 'mbimcli -d /dev/cdc-wdm0
--attach-packet-service --device-open-proxy'...
[/dev/cdc-wdm0] Successfully attached to packet service [/dev/cdc-wdm0] Packet
service status: Network error: 'unknown' Packet service state: 'attached'
Available data classes: 'lte' Uplink speed: '50000000 bps' Downlink speed: '100000000
bps'
Starting network with 'mbimcli -d /dev/cdc-wdm0 --connect=apn='internet'
--device-open-proxy'...
Network started successfully
```

```
# dhclient wwan0
```

Request an IP address for the wwan0 interface using dhclient.

```
# ifconfig wwan0
```

Check whether the interface “wwan0” has obtained an IP.

```
wwan0      Link encap:Ethernet  HWaddr 56:34:A4:C7:FC:0E
            inet addr:10.93.20.64  Bcast:169.254.255.255  Mask:255.255.0.0
            inet6 addr: fe80::5434:a4ff:fc0e/64 Scope:Link
            inet6 addr: fe80::fc50:d5e3:ff77:1649/64 Scope:Link
            UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:14 errors:15 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:2186 (2.1 KiB)
```

Verify the network connection using the “ping” command.

Command and Response

Description

```
# ping 8.8.8.8 -c 4
```

A successful ping to an IP address indicates the network connection is working.

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=39.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=46.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=55.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=116 time=61.8 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 39.392/50.851/61.844/8.479 ms
```

```
# ping www.google.com -c 4
```

You can also ping a domain name successfully.

```
PING www.google.com(nctsaa-ac-in-x04.1e100.net (2404:6800:4012:6::2004))
56
data bytes
64 bytes from nctsaa-ac-in-x04.1e100.net (2404:6800:4012:6::2004):
icmp_seq=1
ttl=116 time=15.4 ms
64 bytes from nctsaa-ac-in-x04.1e100.net (2404:6800:4012:6::2004):
icmp_seq=2
ttl=116 time=57.8 ms
64 bytes from nctsaa-ac-in-x04.1e100.net (2404:6800:4012:6::2004):
icmp_seq=3
ttl=116 time=55.9 ms
64 bytes from nctsaa-ac-in-x04.1e100.net (2404:6800:4012:6::2004):
icmp_seq=4
ttl=116 time=80.2 ms
```

```
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 15.368/52.315/80.212/23.378 ms
```