

AN14808

Connect i.MX RTxxxx to USB 4G Module (ECM Mode)

Rev. 1.0 — 24 October 2025

Application note

Document information

Information	Content
Keywords	AN14808, USB, 4G module
Abstract	This application note discusses how to connect the i.MX RTxxxx to USB 4G module in the ECM mode.



1 Introduction

This application note discusses how to connect the i.MX RTxxxx to the USB 4G module in the **ECM** mode. For how to connect the i.MX RTxxxx to the USB 4G module in the **RNDIS** mode, see the Connect i.MX RT1060 to USB 4G Module (RNDIS Mode) (document [AN13940](#)).

Until SDK 25.06, the USB stack supports host CDC ECM only on MCUX-N and RW612. In this document, it ports the code to RT1170 for example. The USB 4G module is also EC200A-CN from Quectel. To port it into other RTxxxx MCU, the steps are slimier and the steps mentioned in this document can also be referred.

For some knowledge like **basics for USB host application by NXP SDK**, see the Connect i.MX RT1060 to USB 4G Module (RNDIS Mode) (document [AN13940](#)).

This document uses the RT1170-EVKB as an example for the board setup, as shown in [Figure 1](#). The board must be powered from an external power supply adaptor. The IDE is MCUXpresso v25.6.



Figure 1. Board setup

2 About EC200A-CN

EC200A-CN is a USB 4G modem, which supports LTE-FDD/TDD, WCDMA, and GSM/EDGE. It is provided by Quectel.

EC200A-CN supports RNDIS and ECM mode. It is configurable to run in the RNDIS/ECM mode by the AT command below:

- AT+qcfg="usbnet", 3 // RNDIS mode
- AT+qcfg="usbnet", 1 // ECM mode

To switch the mode, run the AT command only once. The configuration is recorded, and when powered on next time, it keeps the settings.

AT command for the dialing:

- AT+qnetdevctl=1,1,1

After powered on, the log below shows in AT port:

```
RDY
+CPIN: READY
+QUSIM: 1
+CFUN: 1
+QIND: SMS DONE
+QIND: PB DONE
```

When +QIND: PB DONE shows, it means that the initialization is done, and we can dial:

```
AT+qnetdevctl=1,1,1 // dial
OK
+QNETDEVSTATUS: 1
```

When +QNETDEVSTATUS: 1 shows, it means that the dial is successful, and we can send/receive MAC package to/from the EC200A-CN.

[Figure 2](#) shows the EC200A-CN interface and endpoint structure.

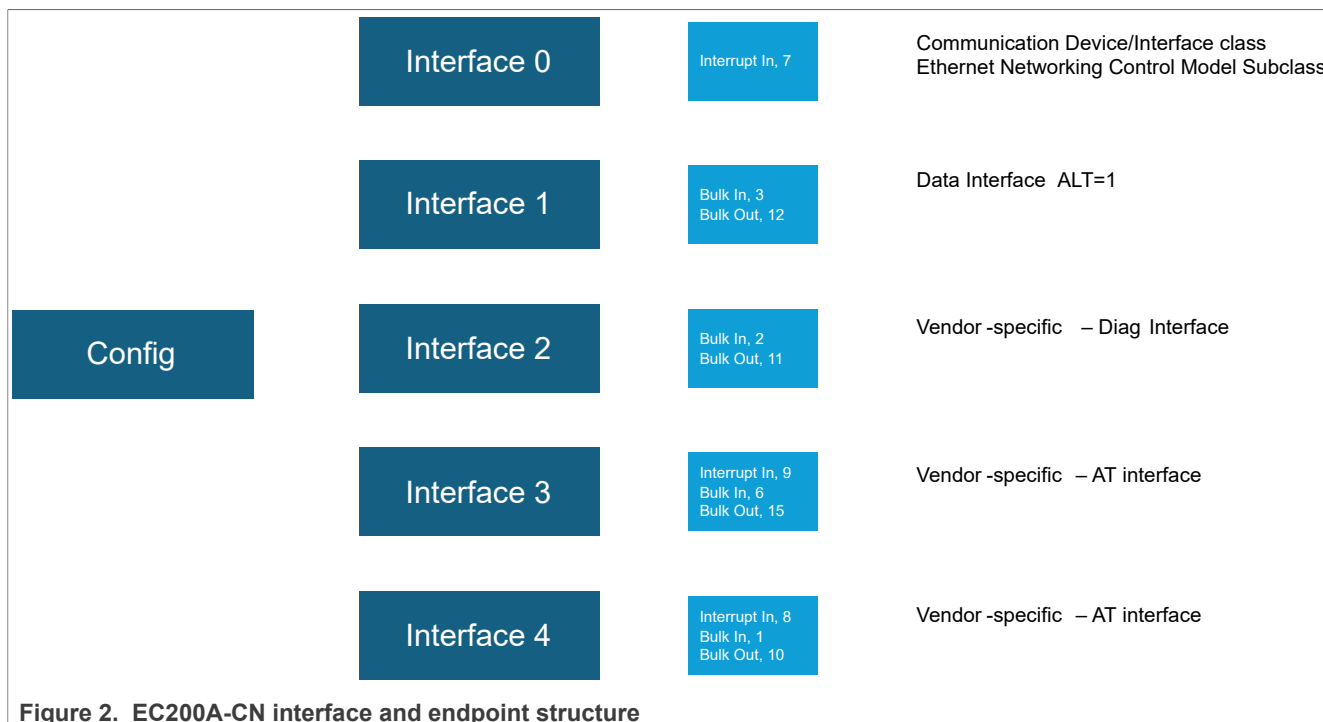


Figure 2. EC200A-CN interface and endpoint structure

The main interfaces that we use are Interface 1 (Data interface) and Interface 3 (AT interface).

3 Details for implementation

In this document, we use the `evkbmimxrt1170_lwip_dhcp_usb_freertos_cm7` example as a start point to support EC200A-CN and you must implement the changes below. Here only list the changes. For functions, parameter is not listed to make things clear and simple. For more details, see the [AN14808SW](#).

1. Remove `lwip_dhcp_usb_freertos.c`.
2. Add below files from `mcxn9xxevk_host_cdc_ecm_freertos` example.
 - `app.c/app.h`
 - `usb_host_cdc_ecm.c/usb_host_cdc_ecm.h`
 - `udpecho_raw.c/udpecho_raw.h`
3. In `usb_host_cdc.c/h`, add API `USB_HostCdcControl_ex()`.
4. In `usb_host_config.h`:

```
#define USB_HOST_CONFIG_CDC_ECM 1
#define USB_HOST_CONFIG_CDC_RNDIS 0
```

5. In `lwipopts.h`:

```
#define LWIP_MULTICAST_TX_OPTIONS 1
#define LWIP_IGMP 1
```

6. In `usb_ethernetif.h`:

- a. In `USB_HostCdcEcmRunState_t`, add:

```
USB_HostCdcEcmRunSetDataInterface_at,
USB_HostCdcEcmRunInit4GModem,
USB_HostCdcEcmRunDial,
```

- b. In `USB_HostCdcEcmInstance_t`, add:

```
usb_host_class_handle classHandle_at;
usb_host_interface_handle dataInterfaceHandle_at;
```

7. In `usb_ethernetif_freertos.c`:

- a. Add a macro definition.

```
#define EC200A_AT_INTERFACE_NUM 3
```

- b. In `USB_HostCdcEcmControlCallback()`, adjust the state machine.

```
case USB_HostCdcEcmRunSetDataInterface:
    ecmInstance->runCurState = USB_HostCdcEcmRunSetDataInterface_at;
    ecmInstance->runWaitState = USB_HostCdcEcmRunIdle;
    break;

// add for at
case USB_HostCdcEcmRunSetDataInterface_at:
    ecmInstance->runCurState = USB_HostCdcEcmRunInit4GModem;
    ecmInstance->runWaitState = (USB_HostCdcEcmRunState_t)(0);
    break;

case USB_HostCdcEcmRunInit4GModem:
    ecmInstance->runCurState = USB_HostCdcEcmRunDial;
    ecmInstance->runWaitState = (USB_HostCdcEcmRunState_t)(0);
    break;

case USB_HostCdcEcmRunDial:
```

- c. In `USB_HostCdcEcmTask()`, initialize the AT interface.

```
// init at interface
if (USB_HostCdcInit(ecmInstance->deviceHandle, &ecmInstance->classHandle_at) != kStatus_USB_Success)
{
    ecmInstance->runCurState = USB_HostCdcEcmRunIdle;
    usb_echo("USB host init failed - at.\r\n");
}
else
{
    usb_echo("USB CDC-ECM device is attached. - at\r\n");
}
```

- d. In `USB_HostCdcEcmTask()`, add code for opening AT interface, initializing 4G modem, and dialing.

```
case USB_HostCdcEcmRunSetDataInterface_at:
    //usb_echo("USB_HostCdcEcmRunSetDataInterface_at \r\n");
    ecmInstance->runPrevState = ecmInstance->runCurState;
    ecmInstance->runCurState = USB_HostCdcEcmRunIdle;
    if (USB_HostCdcSetDataInterface(ecmInstance->classHandle_at, ecmInstance->dataInterfaceHandle_at, 0,
        USB_HostCdcEcmControlCallback, ecmInstance) != kStatus_USB_Success)
    {
        usb_echo("Set data interface error. - at\r\n");
    }
    break;

case USB_HostCdcEcmRunInit4GModem:
    //usb_echo("USB_HostCdcEcmRunInit4GModem .\r\n");
    ecmInstance->runPrevState = ecmInstance->runCurState;
    // ecmInstance->runCurState = USB_HostCdcEcmRunIdle;
    // no special init step for EC200, so just skip.
    // USB_HostCdcEcmControlCallback((void *)ecmInstance, NULL, 0, kStatus_USB_Success);
    extern int init_ec200a_done;
    void init_ec200a(USB_HostCdcEcmInstance_t *ecmInstance);
    if(init_ec200a_done)
    {
```

```

        // switch to next state
        USB_HostCdcEcmControlCallback((void *)ecmInstance, NULL, 0, kStatus_USB_Success);
    }
    else
    {
        init_ec200a(ecmInstance);
    }
    break;

case USB_HostCdcEcmRunDial:
    //usb_echo("USB_HostCdcEcmRunDial .\r\n");
    ecmInstance->runPrevState = ecmInstance->runCurState;
    //ecmInstance->runCurState = USB_HostCdcEcmRunIdle;
    extern int dial_done;
    extern void lte_dial(USB_HostCdcEcmInstance_t *ecmInstance);
    if(dial_done)
    {
        // switch to next state
        USB_HostCdcEcmControlCallback((void *)ecmInstance, NULL, 0, kStatus_USB_Success);
    }
    else
    {
        lte_dial(ecmInstance);
    }
    break;

```

- e. In USB_HostCdcEcmTask(), remove netif_is_link_up().

```
if (*task_event & CDC_ECM_STATE_XFER_DATA_IN)
```

- f. In USB_HostCdcEcmEvent(), detect AT interface.

```

else if (hostInterface->interfaceDesc->bInterfaceNumber ==
        EC200A_AT_INTERFACE_NUM)
{
    g_HostCdcEcmInstance.dataInterfaceHandle_at = hostInterface;
    usb_echo("at port detected.\r\n");
}

```

- g. Add the functions below:

```

USB_HostCdcECM_ATInCallback();
USB_HostCdcECM_ATOutCallback();
dial_update_status();
dial_tx();
dial_rx();
lte_dial();
USB_HostCdcECM_EC200A_EP0_Callback();
update_ec200a_status();
ep0_communicate();
init_ec200a();

```

After all code changes listed here, build and run the code. We can see that the i.MX RT1170 EVKB is connected to the Internet successfully by USB 4G module EC200A-CN, as shown in the log below:

```

Time: 16:54:10
USB host init success.
Cannot support the USB device, please check if the device support USB CDC-ECM
class.
USB device is not supported.
AT port detected.
Device CDC-ECM attached: PID=0x6005, VID=0x2C7C, Address=1.
USB CDC-ECM device is attached.
AT Port init done.
ec200a init.
ec200a_state = 0

```

```
ok.
ec200a_state = 1
ok.
ec200a_state = 2
ok.
ec200a_state = 3
ok.
-4.
ec200a_state = 4
ok.
-5.
ec200a_state = 5
ok.
ec200a_state = 6
ok.
ec200a_state = 7
ok.
ec200a_state = 8
ok.
ec200a_state = 9
ok.
ec200a_state = 10
ok.
ec200a_state = 11
ok.
init_ec200a_done
dial.....

dial_state == 1.
>>>RX:
RDY

ec200a_rx_index = 0
dial_state == 2.
>>>RX:
+CPIN: READY

ec200a_rx_index = 1
dial_state == 3.
>>>RX:
+QUSIM: 1
Y

ec200a_rx_index = 2
dial_state == 4.
>>>RX:
+CFUN: 1

Y

ec200a_rx_index = 3
dial_state == 5.
>>>RX:
+QIND: SMS DONE

ec200a_rx_index = 4
dial_state == 6.
```

```
>>>RX:
+QIND: PB DONE

ec200a_rx_index = 5
PB DONE detected.
dial_state == 100.
AT+qnetdevctl=1,1,1
>>>TX: AT+qnetdevctl=1,1,1

dial_state == 101.
>>>RX: AT+qnetdevctl=1,1,1
ec200a_rx_index = 6
dial_state == 102.
>>>RX:
OK
tdevctl=1,1,1
ec200a_rx_index = 7
dial_state == 103.
>>>RX:
+QNETDEVSTATUS: 1

ec200a_rx_index = 8
Dial done.
Device network connection is connected.
Network speed have been changed to DL 425984000 bps / UL 425984000 bps.
Get IPv4 information from DHCP
Waiting DHCP server process... 6 10 OK!

*****
Network Interface Information
*****
IPv4 Address       : 192.168.43.100
IPv4 Subnet mask   : 255.255.255.0
IPv4 Gateway       : 192.168.43.1
IPv4 DNS 0         : 192.168.43.1
IPv4 DNS 1         : 0.0.0.0
IPv4 Group         : 239.0.0.1
MAC Address        : 2:C:29:A3:9B:6D

Server listening on UDP port 7

Start resolving domain name (quectel.com)...
Domain name resolution success

Try to ping quectel.com
Start pinging [18.220.212.242]:
ping: send 18.220.212.242
ping: recv 18.220.212.242 286 ms
ping: send 18.220.212.242
ping: recv 18.220.212.242 284 ms
ping: send 18.220.212.242
ping: recv 18.220.212.242 282 ms
ping: send 18.220.212.242
ping: recv 18.220.212.242 281 ms
ping: send 18.220.212.242
ping: recv 18.220.212.242 288 ms
ping: send 18.220.212.242
ping: recv 18.220.212.242 287 ms
... ..
```

4 Conclusion

This application note discusses how to connect the i.MX RT1170 EVKB to the Internet by USB 4G module EC200A-CN. It is helpful for user reference when they want to connect to the Internet by the same or other USB 4G modules.

5 Note about the source code in the document

The example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2025 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

6 Revision history

[Table 1](#) summarizes the revisions to this document.

Table 1. Revision history

Document ID	Release date	Description
AN14808 v1.0	24 October 2025	Initial public release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at <https://www.nxp.com/profile/terms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Microsoft, Azure, and ThreadX — are trademarks of the Microsoft group of companies.

Contents

1 Introduction2

2 About EC200A-CN3

3 Details for implementation4

4 Conclusion9

5 Note about the source code in the document9

6 Revision history9

Legal information10

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.