RT58x Thread

Quick Start Guide

V1.0

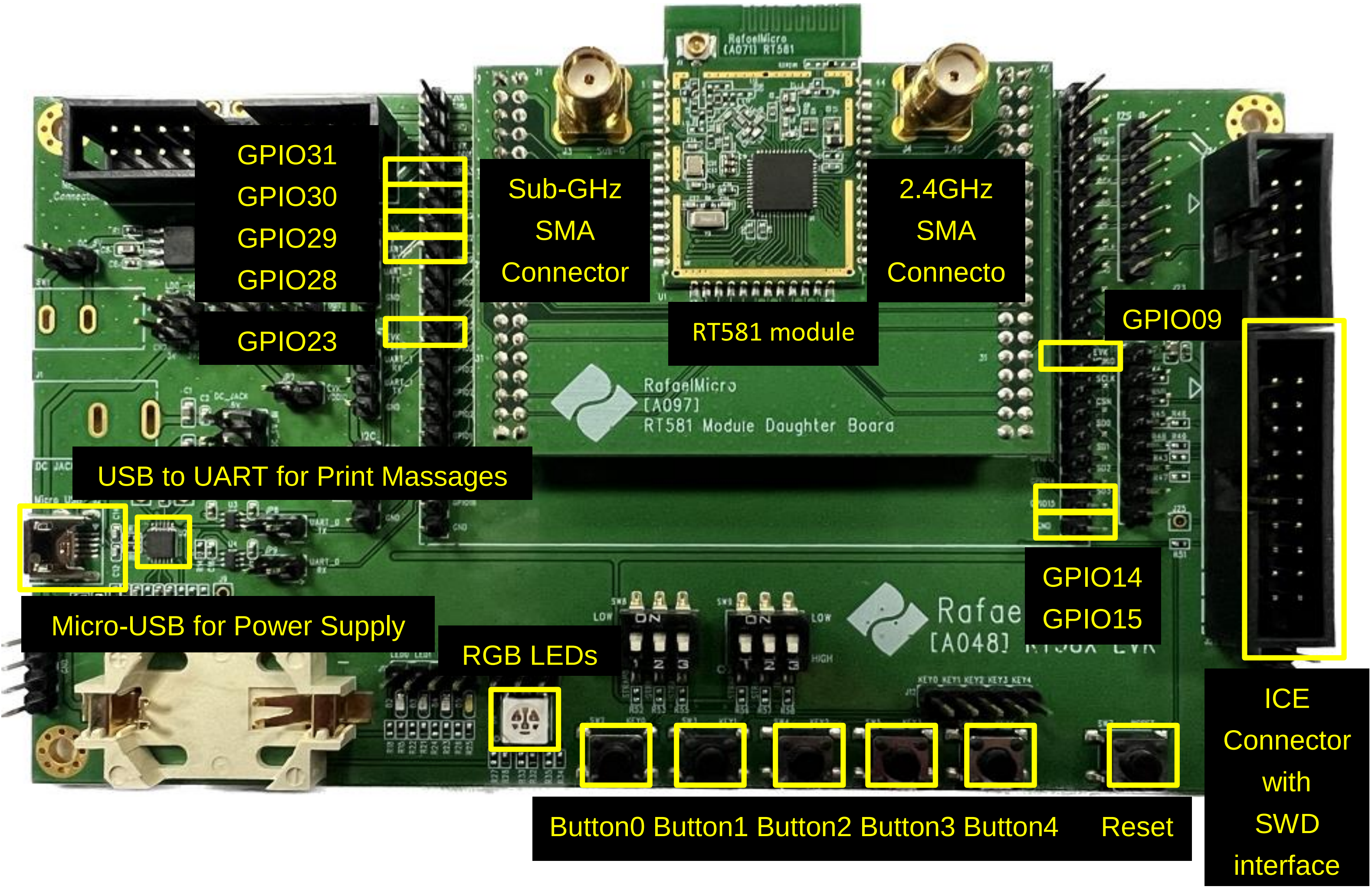
Table of Contents

1. Introduction

The content of this document aims to provide users with a clear guide to quickly mastering the use of Thread technology on RT58X devices. This document contains detailed information covering topics such as network setup and data transmission to ensure that users can fully understand and successfully apply it.

2. Board introduction

The detailed content of Rafael Micro's development board is shown in the diagram below.



3. Thread Role introduction

This section will provide a brief introduction to the role of Thread.

3.1. Device types

Full Thread Device

A Full Thread Device (FTD) always has its radio on, subscribes to the all-routers multicast address, and maintains IPv6 address mappings. There are three types of FTDs:

- Router
- Router Eligible End Device (REED) — can be promoted to a Router
- Full End Device (FED) — cannot be promoted to a Router

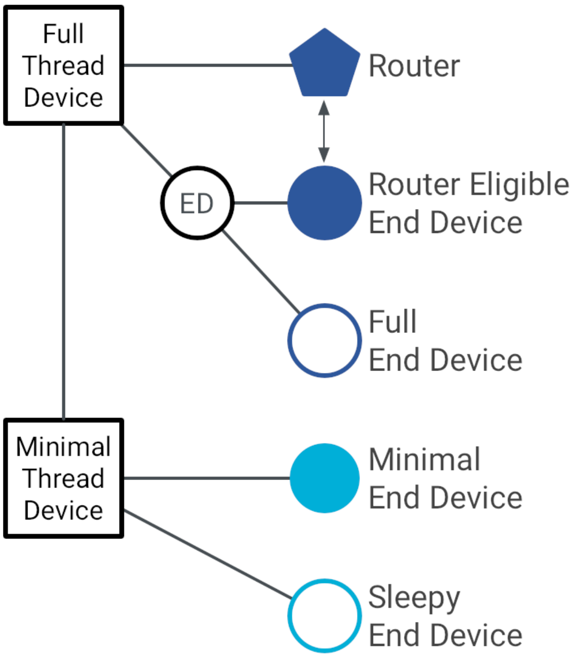
An FTD can operate as a Router (Parent) or an End Device (Child).

Minimal Thread Device

A Minimal Thread Device does not subscribe to the all-routers multicast address and forwards all messages to its Parent. There are two types of MTDs:

- Minimal End Device (MED) — transceiver always on, does not need to poll for messages from its parent
- Sleepy End Device (SED) — normally disabled, wakes on occasion to poll for messages from its parent

An MTD can only operate as an End Device (Child).



3.2. Roles

Border Router

A Border Router is a device that can forward information between a Thread network and a non-Thread network (for example, Wi-Fi). It also configures a Thread network for external connectivity.

Any device may serve as a Border Router.

Note: This SDK does not provide.

Leader

The Thread Leader is a Router that is responsible for managing the set of Routers in a Thread network. It is dynamically self-elected for fault tolerance, and aggregates and distributes network-wide configuration information.

Note: There is always a single Leader in each Thread network partition.

Router

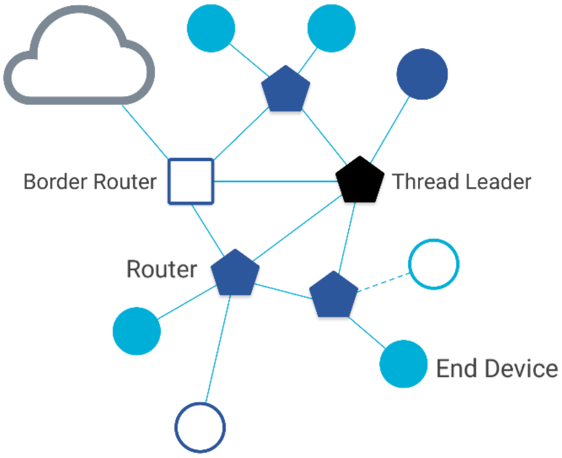
A Router is a node that:

- forwards packets for network devices
- provides secure commissioning services for devices trying to join the network
- keeps its transceiver enabled at all times

End Device

An End Device (ED) is a node that:

- communicates primarily with a single Router
- does not forward packets for other network devices
- can disable its transceiver to reduce power



4. Thread IPv6 Addressing

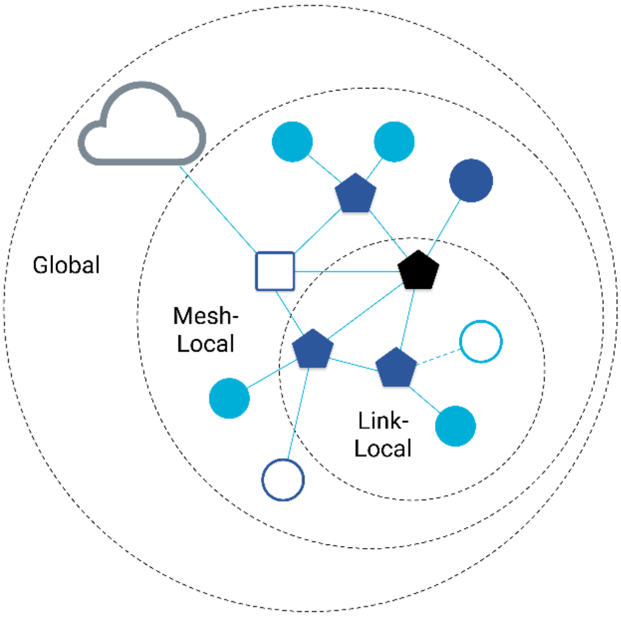
This section will provide a brief introduction to the IPv6 address of Thread.

4.1. IPv6 Scopes

There are three scopes in a Thread network for unicast addressing:

- Link-Local — all interfaces reachable by a single radio transmission
- Mesh-Local — all interfaces reachable within the same Thread network
- Global — all interfaces reachable from outside a Thread network

The first two scopes correspond to prefixes designated by a Thread network. Link-Local have prefixes of fe80::/16, while Mesh-Local have prefixes of fd00::/8.



4.2. Unicast address

Common unicast types are detailed below.

Link-Local Address (LLA)	
An EID that identifies a Thread interface reachable by a single radio transmission.	
Example	fe80::54db:881c:3845:57f4
IID	Based on 802.15.4 Extended Address
Scope	Link-Local
Details	◦ Used to discover neighbors, configure links, and exchange routing information ◦ Not a routable address ◦ Always has a prefix of fe80::/16

Mesh-Local EID (ML-EID)	
An EID that identifies a Thread interface, independent of network topology. Used to reach a Thread interface within the same Thread partition. Also called a Unique Local Address (ULA).	
Example	fde5:8dba:82e1:1:416:993c:8399:35ab
IID	Random, chosen after commissioning is complete
Scope	Mesh-Local
Details	◦ Does not change as the topology changes ◦ Should be used by applications ◦ Always has a prefix fd00::/8

Routing Locator (RLOC)	
Identifies a Thread interface, based on its location in the network topology.	
Example	fde5:8dba:82e1:1::ff:fe00:1001
IID	0000:00ff:fe00:RLOC16
Scope	Mesh-Local
Details	◦ Generated once a device attaches to a network ◦ For delivering IPv6 datagrams within a Thread network ◦ Changes as the topology changes ◦ Generally not used by applications

Anycast Locator (ALOC)	
Identifies a Thread interface via RLOC lookup, when the RLOC of a destination is not known.	
Example	fde5:8dba:82e1:1::ff:fe00:fc01
IID	0000:00ff:fe00:fcXX
Scope	Mesh-Local
Details	◦ fcXX = ALOC destination, which looks up the appropriate RLOC ◦ Generally not used by applications

Global Unicast Address (GUA)	
An EID that identifies a Thread interface on a global scope, beyond a Thread network. (This SDK does not provide)	
Example	2000::54db:881c:3845:57f4
IID	◦ SLAAC — Randomly assigned by the device itself ◦ DHCP — Assigned by a DHCPv6 server ◦ Manual — Assigned by the application layer
Scope	Global
Details	◦ SLAAC — Randomly assigned by the device itself ◦ DHCP — Assigned by a DHCPv6 server ◦ Manual — Assigned by the application layer

4.3. Multicast address

Multicast is used to communicate information to multiple devices at once. In a Thread network, specific addresses are reserved for multicast use with different groups of devices, depending on the scope.

IPv6 Address	Scope	Delivered to
ff02::1	Link-Local	All FTDs and MEDs
ff02::2	Link-Local	All FTDs
ff03::1	Mesh-Local	All FTDs and MEDs
ff03::2	Mesh-Local	All FTDs

Note: That Sleepy End Devices (SEDs) are not included as a recipient in the multicast table above.

4.4. Anycast address

Anycast is used to route traffic to a Thread interface when the RLOC of a destination is not known. An Anycast Locator (ALOC) identifies the location of multiple interfaces within a Thread partition. The last 16 bits of an ALOC, called the ALOC16, is in the format of 0xfcXX, which represents the type of ALOC.

For example, an ALOC16 between 0xfc01 and 0xfc0f is reserved for DHCPv6 Agents. If the specific DHCPv6 Agent RLOC is unknown (perhaps because the network topology has changed), a message can be sent to a DHCPv6 Agent ALOC to obtain the RLOC.

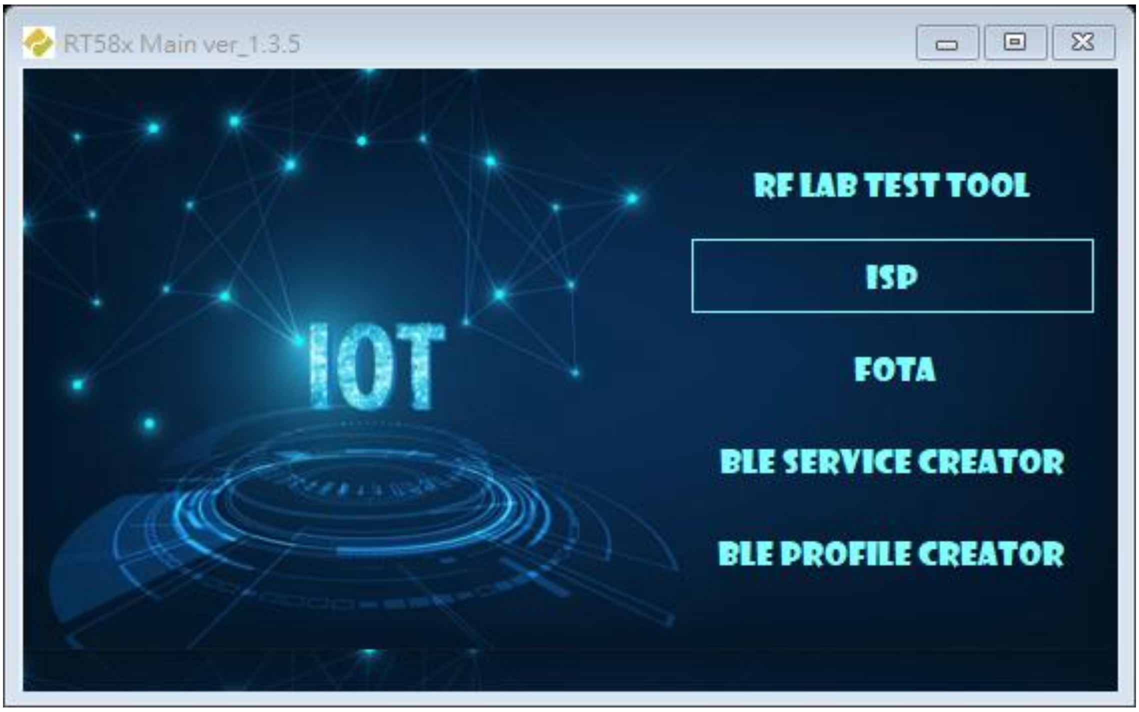
Thread defines the following ALOC16 values:

ALOC16	Type
0xfc00	Leader
0xfc01 – 0xfc0f	DHCPv6 Agent
0xfc10 – 0xfc2f	Service
0xfc30 – 0xfc37	Commissioner
0xfc40 – 0xfc4e	Neighbor Discovery Agent
0xfc38 – 0xfc3f 0xfc4f – 0xfcff	Reserved

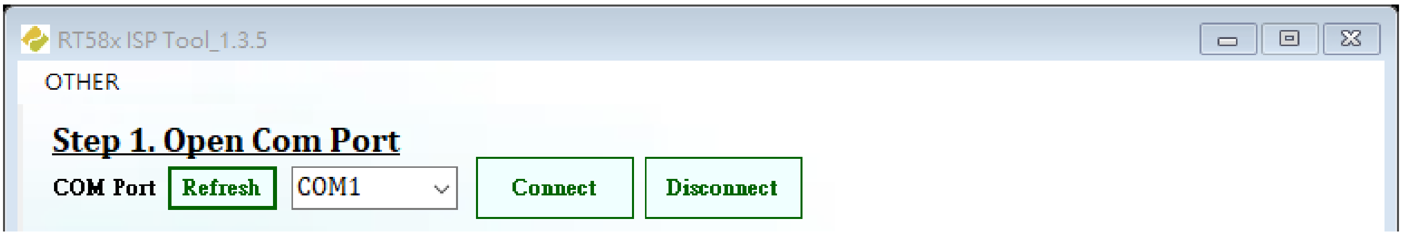
5. IoT_EVALUATION_TOOL tool

This section provides an introduction on how to use the ISP Tool to download bin files.

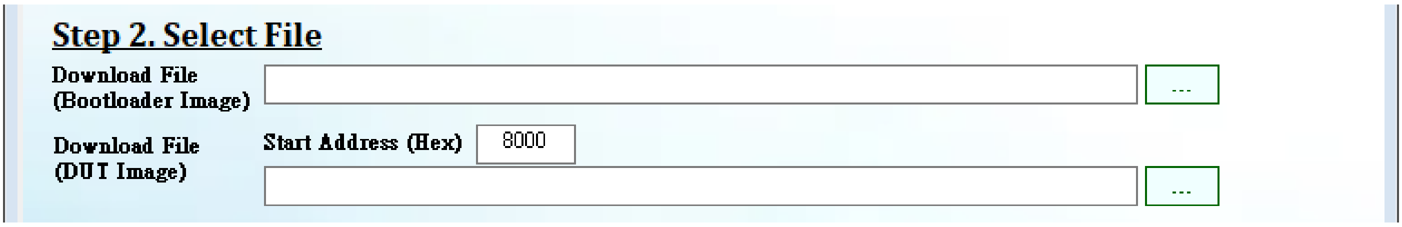
Step1. Open this tool and select “ISP” from the options.



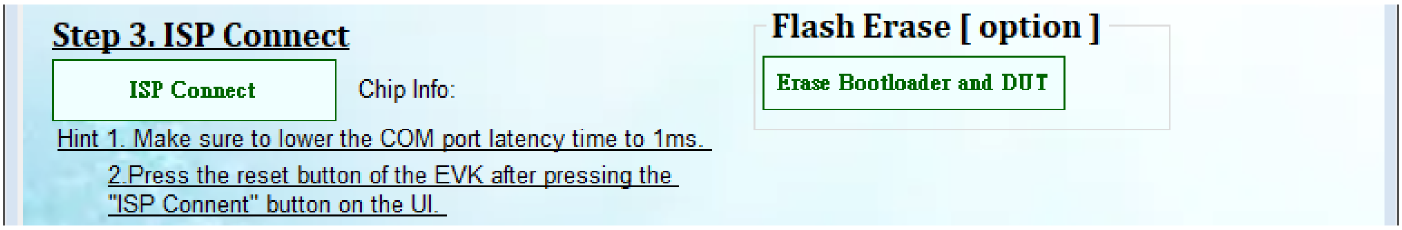
Step2. Select the USB COM port for downloading the development board.



Step3. Choose the corresponding bin file.



Step4. Execute “ISP Connect”, and when prompted, press the reset button on the board.



Step 3. ISP Connect

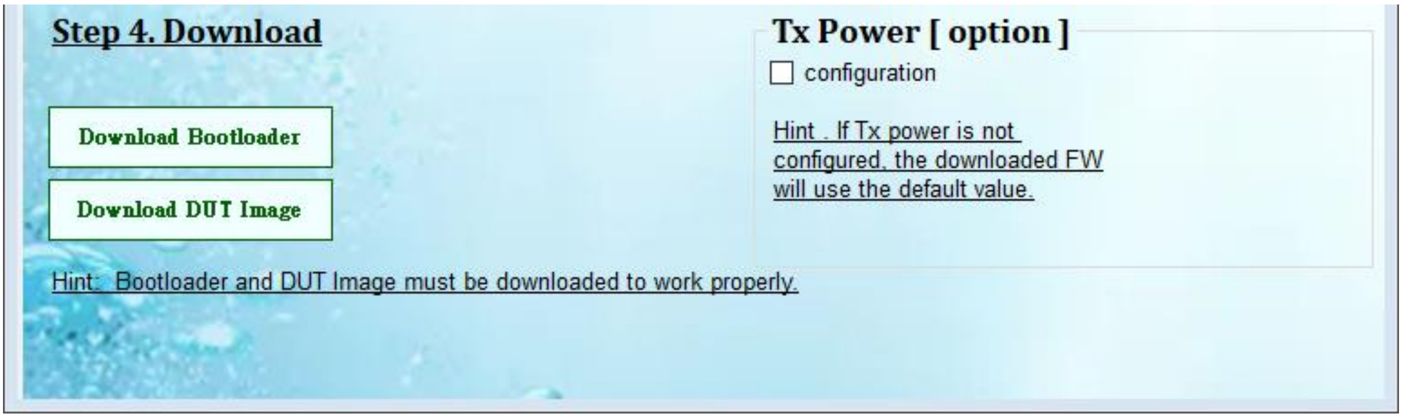
Chip Info:

Hint 1. Make sure to lower the COM port latency time to 1ms.

2.Press the reset button of the EVK after pressing the "ISP Connent" button on the UI.

Flash Erase [option]

Step5. Execute download.



Step 4. Download

Tx Power [option]

□ configuration

Hint . If Tx power is not configured, the downloaded FW will use the default value.

Hint: Bootloader and DUT Image must be downloaded to work properly.

6. Thread Networking operations

This section introduces how to create a thread network.

6.1 Automatic formation of a Thread network

Step1. Environment setup.

- 2 RT581 or RT582 ◦
  - 2 Bin file (2 Thread_2P4G.bin or 2 Thread_SubG_FTD.bin, or Thread_SubG_FTD.bin and Thread_SubG_MTD.bin)

Step2. Boot-up and waiting for an automatic formation of a Thread network.

- Device 1

Users can view on the UART log.

```
Change to detached
Change to Leader
fd00:db8:0:0:0:ff:fe00:fc00              ← Leader ALOC
fd00:db8:0:0:0:ff:fe00:1c00              ← RLOC
fd00:db8:0:0:5038:3233:3202:6cfc         ← ML-EID
fe80:0:0:0:5238:3233:3202:6cfc           ← LLA
```

Note: In the same network key, there will be only one leader.

- Device 2

Users can view on the UART log.

```
Change to detached
Change to Child
fd00:db8:0:0:0:ff:fe00:1c01              ← RLOC
fd00:db8:0:0:5038:3233:3202:99fc         ← ML-EID
fe80:0:0:0:5238:3233:3202:99fc           ← LLA
Change to router
fd00:db8:0:0:0:ff:fe00:4800              ← RLOC
fd00:db8:0:0:5038:3233:3202:99fc         ← ML-EID
fe80:0:0:0:5238:3233:3202:99fc           ← LLA
```

Note: It could be either a child or router, depending on the maximum router limit.

Step3. Special circumstances.

If you see 2 devices becoming leaders, please confirm the following:

- Ensure the network key is the same.

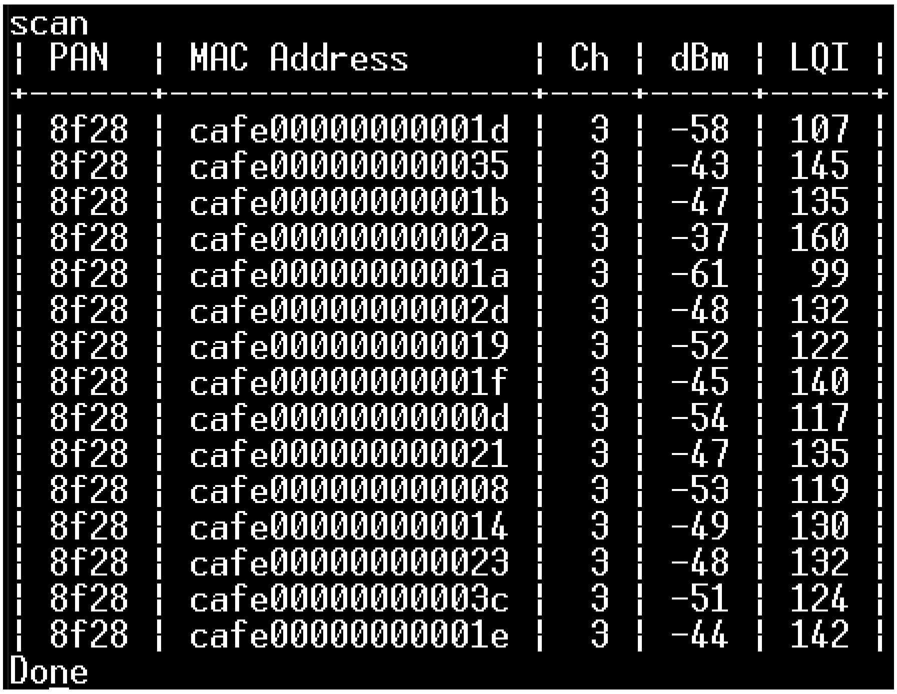
```
>networkkey
fe83448a6729feababfe29678a4483fe
Done
```

- If in the Sub-GHz band, verify if the data rate is the same.

```
=============DataRate FSK_300K==========
```

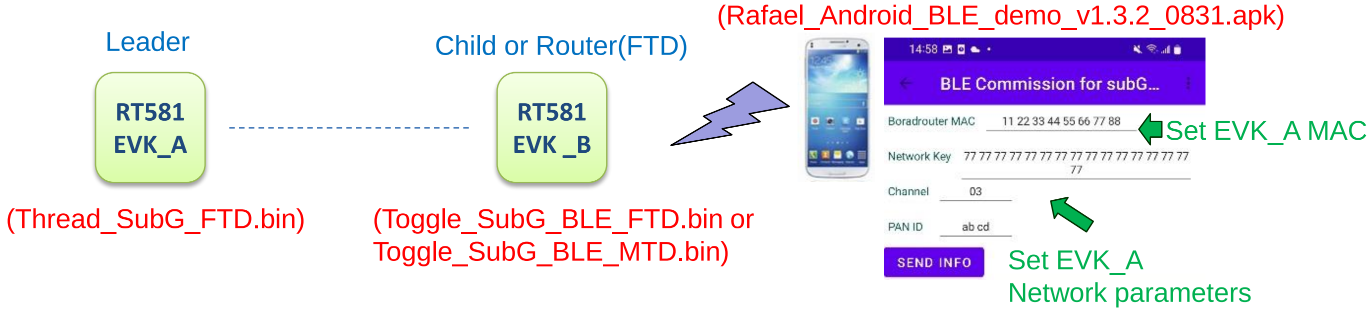
This log is displayed during boot-up.

- Use a scan to confirm if you can receive data from each other.



```
scan
| PAN  | MAC Address      | Ch | dBm | LQI |
+------+------------------+----+-----+-----+
| 8f28 | cafe00000000001d |  3 | -58 | 107 |
| 8f28 | cafe000000000035 |  3 | -43 | 145 |
| 8f28 | cafe00000000001b |  3 | -47 | 135 |
| 8f28 | cafe00000000002a |  3 | -37 | 160 |
| 8f28 | cafe00000000001a |  3 | -61 |  99 |
| 8f28 | cafe00000000002d |  3 | -48 | 132 |
| 8f28 | cafe000000000019 |  3 | -52 | 122 |
| 8f28 | cafe00000000001f |  3 | -45 | 140 |
| 8f28 | cafe00000000000d |  3 | -54 | 117 |
| 8f28 | cafe000000000021 |  3 | -47 | 135 |
| 8f28 | cafe000000000008 |  3 | -53 | 119 |
| 8f28 | cafe000000000014 |  3 | -49 | 130 |
| 8f28 | cafe000000000023 |  3 | -48 | 132 |
| 8f28 | cafe00000000003c |  3 | -51 | 124 |
| 8f28 | cafe00000000001e |  3 | -44 | 142 |
Done
```

6.2 Use BLE commissioning for network formation.



(Rafael_Android_BLE_demo_v1.3.2_0831.apk)

(Thread_SubG_FTD.bin)

(Toggle_SubG_BLE_FTD.bin or Toggle_SubG_BLE_MTD.bin)

Boradrouter MAC 11 22 33 44 55 66 77 88

Network Key 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77

Channel 03

PAN ID ab cd

Set EVK_A MAC

Set EVK_A Network parameters

Step1. Environment setup.

- 2 RT581
- 2 Bin file (Thread_SubG_FTD.bin and Toggle_SubG_BLE_FTD.bin, or Thread_SubG_FTD.bin and Toggle_SubG_BLE_MTD.bin)
- Android app (Rafael_Android_BLE_demo_v1.3.2_0831.apk)

Step2. Setting Thread_SubG_FTD device to Leader or Router.

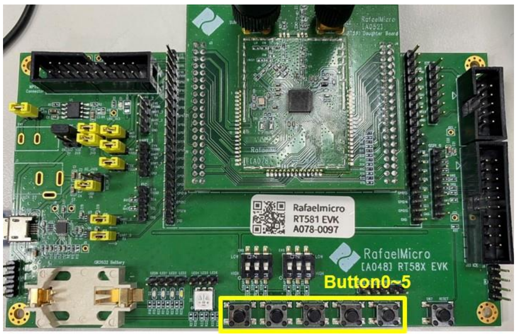
```
>thread start
Done
>state leader
Done
```

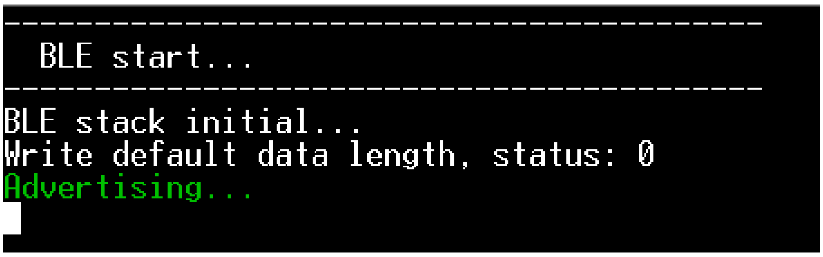
Note: If it automatically becomes a Leader or Router, skip this step.

Step3. Querying Thread_SubG_FTD device network information.

```
>extaddr
503832333202bafd
>networkkey
fe83448a6729feababfe29678a4483fe
>channel
3
Done
>panid
0xabcd
Done
```

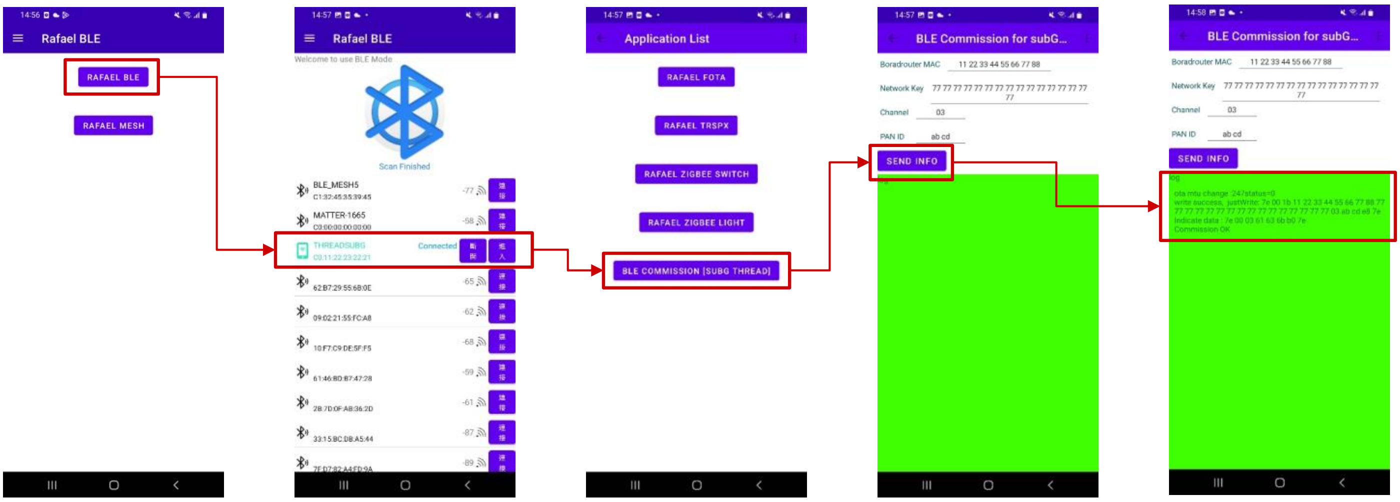
Step4. Tigger Button0 to start BLE commission.





```
-------------------------------------------
  BLE start...
-------------------------------------------
BLE stack initial...
Write default data length, status: 0
Advertising...
```

Step5. Use Rafael_Android_BLE_demo APP to commission setting.



Rafael BLE: RAFAEL BLE, RAFAEL MESH

Rafael BLE — Welcome to use BLE Mode — Scan Finished

```
BLE_MESH5        C1:32:45:35:39:45   -77
MATTER-1665      C0:00:00:00:00:00   -58
THREADSUBG       C0:11:22:23:22:21   Connected
62:B7:29:55:6B:0E                    -65
09:02:21:55:FC:A8                    -62
10:F7:C9:DE:5F:F5                    -68
61:46:8D:87:47:28                    -59
28:7D:0F:AB:36:2D                    -61
33:15:BC:DB:A5:44                    -87
7E:D7:82:A4:FD:9A                    -89
```

Application List: RAFAEL FOTA, RAFAEL TRSPX, RAFAEL ZIGBEE SWITCH, RAFAEL ZIGBEE LIGHT, BLE COMMISSION [SUBG THREAD]

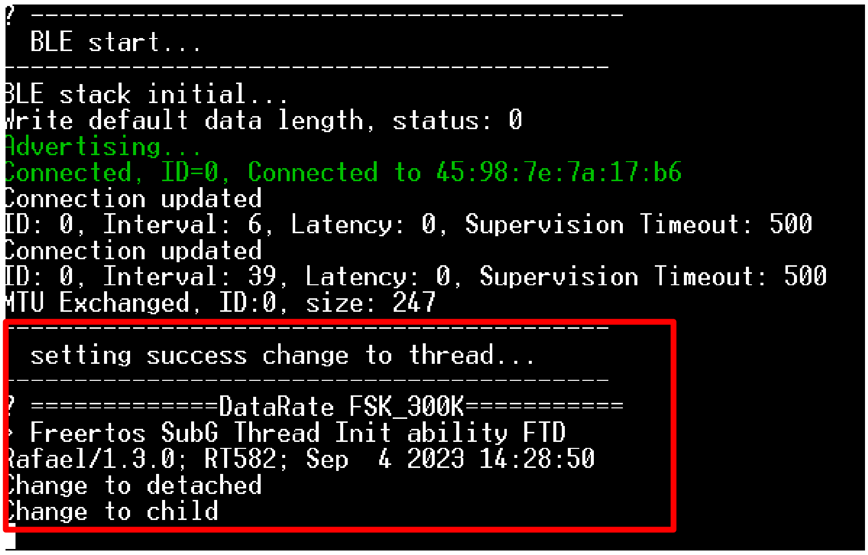
BLE Commission for subG...

Boradrouter MAC: 11 22 33 44 55 66 77 88
Network Key: 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77
Channel: 03
PAN ID: ab cd
SEND INFO

```
log
ota mtu change :247status=0
write success, justWrite: 7e 00 1b 11 22 33 44 55 66 77 88 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 77 03 ab cd e8 7e
indicate data : 7e 00 03 61 63 6b b0 7e
Commission OK
```

(MAC is different by different EVK)

Step6. Check Toggle_SubG_BLE device change to Thread child.



```
? -------------------------------------------
  BLE start...
-------------------------------------------
BLE stack initial...
Write default data length, status: 0
Advertising...
Connected, ID=0, Connected to 45:98:7e:7a:17:b6
Connection updated
ID: 0, Interval: 6, Latency: 0, Supervision Timeout: 500
Connection updated
ID: 0, Interval: 39, Latency: 0, Supervision Timeout: 500
MTU Exchanged, ID:0, size: 247
-------------------------------------------
  setting success change to thread...
-------------------------------------------
? ============DataRate FSK_300K==========
› Freertos SubG Thread Init ability FTD
Rafael/1.3.0; RT582; Sep  4 2023 14:28:50
Change to detached
Change to child
```

7. Ping operations

This section explains how to use the “ping” command.

7.1. Explanation of the ping command.

This command can send an ICMPv6 Echo Request.

ping [async] [-I source] <ipaddr> [size] [count] [interval] [hoplimit] [timeout]

- async: Use the non-blocking mode. New commands are allowed before the ping process terminates.
- source: The source IPv6 address of the echo request.
- size: The number of data bytes to be sent ; Limit size: 1280 bytes.
- count: The number of ICMPv6 Echo Requests to be sent.
- interval: The interval between two consecutive ICMPv6 Echo Requests in seconds. The value may have fractional form, for example 0.5.
- hoplimit: The hoplimit of ICMPv6 Echo Request to be sent.
- timeout: Time in seconds to wait for the final ICMPv6 Echo Reply after sending out the request. The value may have fractional form.

7.2. Ping command steps for usage.

Step1. Use the “ipaddr” command to obtain the device’s IPv6 address.

- Device 1

```
>Ipaddr
fd00:db8:0:0:0:ff:fe00:fc00          ←  Leader ALOC
fd00:db8:0:0:0:ff:fe00:1c00          ←  RLOC
fd00:db8:0:0:5038:3233:3202:6cfc     ←  ML-EID
fe80:0:0:0:5238:3233:3202:6cfc       ←  LLA
Done
```

- Device 2

```
>Ipaddr
fd00:db8:0:0:0:ff:fe00:4800          ←  RLOC
fd00:db8:0:0:5038:3233:3202:99fc     ←  ML-EID
fe80:0:0:0:5238:3233:3202:99fc       ←  LLA
Done
```

Step2. Begin using the “ping” command to transmit to the other device’s IPv6 address.

- Device 1

```
>ping fd00:db8:0:0:5038:3233:3202:99fc
> 16 bytes from fd00:db8:0:0:5038:3233:3202:99fc: icmp_seq=5 hlim=64
time=0ms
1 packets transmitted, 1 packets received. Packet loss = 0.0%. Round-trip
min/avg/max = 0/0.0/0 ms.
Done
```

- Device 2

```
>ping fd00:db8:0:0:5038:3233:3202:6cfc
> 16 bytes from fd00:db8:0:0:5038:3233:3202:6cfc: icmp_seq=5 hlim=64
time=0ms
1 packets transmitted, 1 packets received. Packet loss = 0.0%. Round-trip
min/avg/max = 0/0.0/0 ms.
Done
```

8. UDP operations

This section explains how to use the “udp” command.

8.1. Explanation of the UDP command.

udp open

Opens the example socket.

```
> udp open
Done
```

udp bind [netif] <ip> <port>

Assigns a name (i.e. IPv6 address and port) to the example socket.

- netif: the network interface to bind to.
  - not specified: Thread network interface.
  - -u: unspecified network interface.
  - -b: Backbone network interface.
- ip: the IPv6 address or the unspecified IPv6 address (::).
- port: the UDP port

```
> udp bind :: 1234
Done
> udp bind -u :: 1234
Done
> udp bind -b :: 1234
Done
```

udp send <ip> <port> <message>

Send a UDP message.

- ip: the destination address.
- port: the UDP destination port.
- message: the message to send ; Limit size: 640 characters.

```
> udp send fdde:ad00:beef:0:bb1:ebd6:ad10:f33 1234 hello
Done
```

8.2. UDP command steps for usage.

Step1. Use the “ipaddr” command to obtain the device’s IPv6 address.

- Device 1

```
>Ipaddr
fd00:db8:0:0:0:ff:fe00:fc00            ← Leader ALOC
fd00:db8:0:0:0:ff:fe00:1c00            ← RLOC
fd00:db8:0:0:5038:3233:3202:6cfc       ← ML-EID
fe80:0:0:0:5238:3233:3202:6cfc         ← LLA
Done
```

- Device 2

```
>Ipaddr
fd00:db8:0:0:0:ff:fe00:4800            ← RLOC
fd00:db8:0:0:5038:3233:3202:99fc       ← ML-EID
fe80:0:0:0:5238:3233:3202:99fc         ← LLA
Done
```

Step2. Use the “udp open” command to enable example udp socket.

- Device 1

```
>udp open
Done
```

- Device 2

```
>udp open
Done
```

Step3. Use the “udp bind” command to register udp port.

- Device 1

```
>udp bind :: 1234
Done
```

- Device 2

```
> udp bind :: 1234
Done
```

Step4. Begin using the “udp” command to transmit to the other device’s IPv6 address.

- Device 1 (Initiator)

```
>udp send fd00:db8:0:0:5038:3233:3202:99fc 1234 hello_99fc
Done
```

- Device 2

```
>10 bytes from fd00:db8:0:0:5038:3233:3202:6cfc 1234 hello_99fc
```

- Device 2 (Initiator)

```
> udp send fd00:db8:0:0:5038:3233:3202:6cfc 1234 hello_6cfc
Done
```

- Device 1

```
>10 bytes from fd00:db8:0:0:5038:3233:3202:99fc 1234 hello_6cfc
```

Revision History

Revision	Description	Owner	Date
V1.0	Initial version	Jiemin	2023/10/16

Information in this document is provided in connection with **Rafael Microelectronics, Inc.** ("**Rafael Micro**") products. These materials are provided by **Rafael Micro** as a service to its customers and may be used for informational purposes only. **Rafael Micro** assumes no responsibility for errors or omissions in these materials. **Rafael Micro** may make changes to this document at any time, without notice. **Rafael Micro** advises all customers to ensure that they have the latest version of this document and to verify, before placing orders, that information being relied on is current and complete. **Rafael Micro** makes no commitment to update the information and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to its specifications and product descriptions.

THESE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, RELATING TO SALE AND/OR USE OF **RAFAEL MICRO** PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, CONSEQUENTIAL OR INCIDENTAL DAMAGES, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. **RAFAEL MICRO** FURTHER DOES NOT WARRANT THE ACCURACY OR COMPLETENESS OF THE INFORMATION, TEXT, GRAPHICS OR OTHER ITEMS CONTAINED WITHIN THESE MATERIALS. **RAFAEL MICRO** SHALL NOT BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION, LOST REVENUES OR LOST PROFITS, WHICH MAY RESULT FROM THE USE OF THESE MATERIALS.

Rafael Micro products are not intended for use in medical, lifesaving or life sustaining applications. **Rafael Micro** customers using or selling **Rafael Micro** products for use in such applications do so at their own risk and agree to fully indemnify **Rafael Micro** for any damages resulting from such improper use or sale. **Rafael Micro, logos** and **RT568** are **Trademarks** of **Rafael Microelectronics, Inc**. Product names or services listed in this publication are for identification purposes only, and may be trademarks of third parties. Third-party brands and names are the property of their respective owners.