

Run Thread networks

The following guide explains how to enable **Thread networking** on your **SMLIGHT SMHUB** device.

This process involves updating the operating system, flashing Thread firmware to the radio module, and installing the OpenThread Border Router (OTBR) app.

“ ⚠ **Note:** This is an *early-access feature*. Future (expected v0.3.5+ versions) of SMHUB-OS will make it fully automatic.
For now, please follow the steps carefully in the order below.

A new complete manual is created here:

<https://smlight.tech/support/manuals/books/smhubs/page/using-smhub-as-thread-border-router-for-matter-devices>

~~? A. Update SMHUB-OS to v0.3.4dev2~~

~~Before enabling Thread, update your device to the latest development version supporting Thread services.~~

~~1. Download the update file~~

- ~~• Open a terminal on your SMHUB or connect via SSH.~~
- ~~• Run:~~

```
wget --0 smhub-update-0.3.4.dev2.raucb  
https://updates.smlight.tech/firmware/smhubs/os/smhubs-update-0.3.4.dev2.raucb
```

~~2. Install the update~~

```
sudo rauc install smhub-update-0.3.4.dev2.raucb
```

~~After installation, please reboot your device.~~

? B. Flash Thread Firmware to the EFR32 Chip

The EFR32 radio inside your SMHUB needs a special firmware to operate as a **Thread radio co-processor (RCP)**.

1. Install the flashing tool:

```
pip install universal-silabs-flasher
```

2. Download the flashing script:

```
wget -O flash-efr.sh https://updates.smlight.tech/firmware/smhubs/flash-efr.sh
```

Web Terminal

Access the built-in Linux shell of your SMHUB directly from your SMHUB-OS. You can use this console to execute system commands, inspect logs, manage services, or perform debugging - all without SSH.

Run commands with caution — changes here affect the live system.

✓ Connected

```
Welcome to SMHUB-OS Web Terminal
smlight@SMHUB:~$ sudo wget -O flash-efr.sh https://updates.smlight.tech/firmware/smhubs/flash-efr.sh
Password:
--2025-11-11 14:13:40-- https://updates.smlight.tech/firmware/smhubs/flash-efr.sh
Resolving updates.smlight.tech... 193.111.63.30
Connecting to updates.smlight.tech|193.111.63.30|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3725 (3.6K) [application/x-sh]
Saving to: 'flash-efr.sh'

flash-efr.sh          100%[=====>]  3.64K  --.-KB/s  in
2025-11-11 14:13:41 (747 KB/s) - 'flash-efr.sh' saved [3725/3725]
```

3. Make the script executable:

```
sudo chmod 777 flash-efr.sh
```

4. Run the flasher script and select Thread firmware:

```
sudo ./flash-efr.sh
```

Web Terminal

Access the built-in Linux shell of your SMHUB directly from your SMHUB-OS. You can use this console to execute system commands, inspect logs, manage services, or perform debugging - all without SSH.

Run commands with caution — changes here affect the live system.

✓ Connected

```
Length: 3725 (3.6K) [application/x-sh]
Saving to: 'flash-efr.sh'

flash-efr.sh                               100%[=====>]  3.64K  --KB/s  in

2025-11-11 14:13:41 (747 KB/s) - 'flash-efr.sh' saved [3725/3725]

smlight@SMHUB:~$ sudo chmod +x flash-efr.sh
smlight@SMHUB:~$ sudo ./flash-efr.sh
[SMHUB] SMHUB EFR32 Flasher
[SMHUB] We will flash EFR32 radio on /dev/ttyS2
Select firmware to flash:
  1) Thread NCP
  2) Zigbee Coordinator
  3) Zigbee Router
Enter 1/2/3: 1
[SMHUB] Selected: Thread NCP
[SMHUB] Downloading firmware file: thread.gbl ...
[SMHUB] Downloaded thread.gbl
[SMHUB] Activating EFR32 bootloader via GPIO...
[SMHUB] EFR32 bootloader mode activated.
[SMHUB] Flashing firmware to EFR32 on /dev/ttyS2 (as user 'smlight') ...
2025-11-11 14:15:00.930 SMHUB universal_silabs_flasher_flasher INFO Extracted GBL metadata: NabuCasaMetadata(metadata_version=1, sdk_version='4.4.1', ezsp_version='7.4.1.0',
p_version=None, cpc_version=None, fw_type=<FirmwareImageType.ZIGBEE_NCP: 'zigbee_ncp'>, fw_variant=None, baudrate=115200)
2025-11-11 14:15:00.941 SMHUB universal_silabs_flasher_flasher INFO Probing ApplicationType.EZSP at 115200 baud
2025-11-11 14:15:08.955 SMHUB universal_silabs_flasher_flasher INFO Probing ApplicationType.GECKO_BOOTLOADER at 115200 baud
2025-11-11 14:15:08.611 SMHUB universal_silabs_flasher_flasher INFO Detected bootloader version '2.3.1'
2025-11-11 14:15:08.621 SMHUB universal_silabs_flasher_flasher INFO Detected ApplicationType.GECKO_BOOTLOADER, version '2.3.1' at 115200 baudrate (bootloader baudrate 1152
thread.gbl [#####-----] 16% 00:01:42
```

Once flashing completes, the EFR32 chip will act as a Thread RCP.

? C. Install the OpenThread Border Router (OTBR) App

This application enables SMHUB to act as a **Thread Border Router**, bridging your Thread network with Ethernet or Wi-Fi.

1. Download the OTBR package:

```
wget -O openthread_0.3.0-1-dev1_riscv64.ipk
https://updates.smlight.tech/firmware/smhubs/os/other/openthread_0.3.0-1-dev1_riscv64.ipk
```

2. Install it:

```
sudo opkg install openthread_0.3.0-1-dev1_riscv64.ipk
```

3. Start the OTBR service:

```
sudo /etc/init.d/S99openthread start
```

4. Check OTBR logs (optional):

```
nano /var/log/otbr-agent.log
```

If the service starts successfully, your SMHUB is now broadcasting as a Thread Border Router.

? D. Integrate with Home Assistant

To connect your Thread network to **Home Assistant**:

1. Open **Home Assistant** → **Devices & Services**.
2. Search for “**Open Thread Border Router**” and install it.
3. In the configuration, enter:

```
tcp://<SMHUB_IP>:8081
```

(replace `<SMHUB_IP>` with your actual device IP address)

4. Add the **Thread integration**.
5. Open its settings and confirm a **Preferred Thread Network** appears.

Once visible, your SMHUB is successfully integrated with Home Assistant as a Thread Border Router.

You can now pair **Thread or Matter** devices through the standard Home Assistant interface.

Revision #6

Created 8 November 2025 16:15:55 by Support3

Updated 12 December 2025 12:22:06 by Support3