How to Create a 6LoWPAN Network using Raspberry Pi and nRF52840 USB Dongle

**Prerequisites**

1. Set up a fresh Raspberry Pi device with the rpi-imager tool by following the instructions on [raspberrypi.org](https://www.raspberrypi.org/software/).
2. Download and install the nRF Connect Desktop app from [nordic semiconductors](https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-Desktop) , and then install the programmer utility inside the app.
3. Install a tool to transfer files over ssh . Ex - Filezilla
4. On the raspberry pi install “screen” by using the command  
   sudo apt-get install screen.
5. Install node and npm on the raspberry pi.

**Setting Up the Border Router**

1.Clone and install OpenThread. The script/bootstrap commands make sure the toolchain is installed and the environment is properly configured:

git clone --recursive https://github.com/openthread/openthread.git

cd openthread

./script/bootstrap

2.Build the OpenThread nRF52840 example:

git clone --recursive https://github.com/openthread/ot-nrf528xx.git

cd ot-nrf528xx

./script/build nrf52840 USB\_trans DOT\_BOOTLOADER=USB

3.Convert the built file into flashable firmware

cd ~/ot-nrf528xx/build/bin

arm-none-eabi-objcopy -O ihex ot-rcp ot-rcp.hex

4. Transfer the ot-rcp.hex firmware file from the raspberry pi to your desktop using a ssh transfer tool.

5. Flash the firmware onto the nrf52840 dongle using nRF Desktop connect programmer.

**Install Open Thread Border Router**

1. Log in to your Raspberry Pi and clone ot-br-posix from GitHub:

git clone https://github.com/openthread/ot-br-posix.git --depth 1

2. OTBR has two scripts that bootstrap and set up the Thread Border Router we will run them :

cd ot-br-posix

./script/bootstrap

INFRA\_IF\_NAME=wlan0 WEB\_GUI=1 ./script/setup

3. Check if OTBR is successfully installed:

sudo service otbr-agent status

4. Reboot the raspberry pi:

sudo reboot

5. Check if you have the correct connection string to the nrf52840 dongle, first we will check the address of the nrf52840 on the raspberry pi

ls /dev/ttyACM\*

This command will give you the address of the nrf dongle (/dev/ttyACM0, /dev/ttyACM1 e.t.c.)

Now we will edit the connection string on the otbr-agent service:

sudo nano /etc/default/otbr-agent

add/edit the line

OTBR\_AGENT\_OPTS="-I wpan0 -B OTBR\_INFRA\_IF\_NAME spinel+hdlc+uart:///dev/ttyACM0 trel://OTBR\_INFRA\_IF\_NAME"

**Configure Thread Network**

1. Open the cli for open thread:

sudo ot-ctl

2. Create new network data(settings) :

dataset init new

3. Choose a channel:

dataset channel 11

4. Choose a network key:

dataset networkkey 00112233445566778899aabbccddeeff

5. Start the network :

dataset commit active

ifconfig up

thread start

6. Check Router state and information:

state

netdata show

ipaddr

**Setup End Device**

1.Clone and install OpenThread. The script/bootstrap commands make sure the toolchain is installed and the environment is properly configured:

git clone --recursive https://github.com/openthread/openthread.git

cd openthread

./script/bootstrap

2. Build the ot-cli :

script/cmake-build posix -DOT\_DAEMON=ON

3. Build the OpenThread nRF52840 example:

git clone --recursive https://github.com/openthread/ot-nrf528xx.git

cd ot-nrf528xx

./script/build nrf52840 USB\_trans -DOT\_SRP\_CLIENT=ON -DOT\_ECDSA=ON -DOT\_BOOTLOADER=USB

4. Convert the compiled binary to hex

cd ~/ot-nrf528xx/build/bin

arm-none-eabi-objcopy -O ihex ot-cli-ftd ot-cli-ftd.hex

5. Flash the firmware to the end device dongle using nrfConnect Programmer

6. Reboot the raspberry pi:

sudo reboot

7. Check if you have the correct connection string to the nrf52840 dongle, first we will check the address of the nrf52840 on the raspberry pi

ls /dev/ttyACM\*

This command will give you the address of the nrf dongle (/dev/ttyACM0, /dev/ttyACM1 e.t.c.)

Now we will edit the connection string on the otbr-agent service:

sudo nano /etc/default/otbr-agent

add/edit the line

OTBR\_AGENT\_OPTS="-I wpan0 -B OTBR\_INFRA\_IF\_NAME spinel+hdlc+uart:///dev/ttyACM0 trel://OTBR\_INFRA\_IF\_NAME"

**Connect End Device to Network**

1. First we will get the network joining dataset from the router device:

sudo ot-ctl

sudo ot-ctl dataset active -x

Example output :

0e0800000000000100004a0300001635060004001fffe00208b373a8805891d4e70708fd0c27129c14cbf3030f4f70656e5468726561642d666537650102fe7e0410935e6c41976627f08e436ad71fd6745c0c0402a0f7f8000300000b051000112233445566778899aabbccddeeff

2. In the end device we will add this dataset to join the network:

sudo ot-ctl

dataset set active 0e0800000000000100004a0300001635060004001fffe00208b373a8805891d4e70708fd0c27129c14cbf3030f4f70656e5468726561642d666537650102fe7e0410935e6c41976627f08e436ad71fd6745c0c0402a0f7f8000300000b051000112233445566778899aabbccddeeff

3. We will verify that we have joined the network by checking state and netdata:

state

netdata show

State of the end device should be child.

Netdata should be the same as the netdata of the router.