Quick start guide: Mesh demonstration

This is a quick demonstration of some of the basic concepts of the Bluetooth Mesh network using Nordic's nRF5 SDK for Mesh.

You don't need to build any binaries for running it, as it uses pre-built binaries of the light switch example. Moreover, this example uses the the `quick_start_demo.py` file to flash the required hex files on the Development Kits.

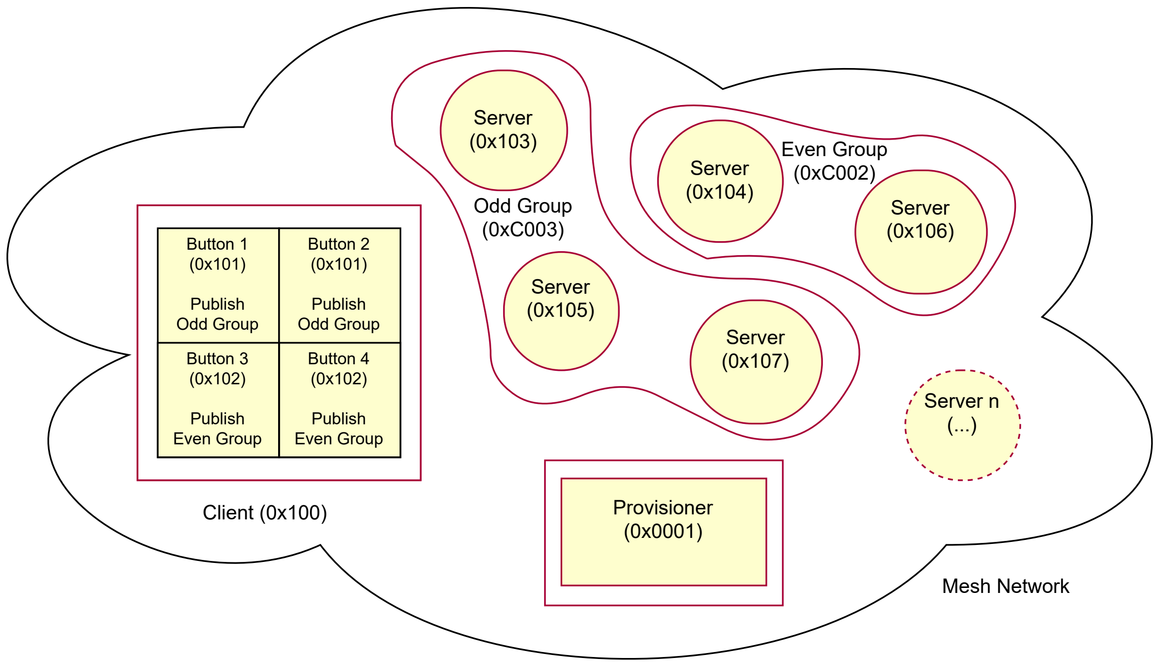
Table of contents

The light switch example, on which this example is based, demonstrates the major parts of the mesh network ecosystem. It consists of three minor examples:

- Light switch server: A minimalistic server that implements a Generic OnOff server model, which is used to receive the state data and control the state of LED 1 on the board.
- Light switch client: A minimalistic client that implements four instances of a Generic OnOff client model. When a user presses any of the buttons, an OnOff Set message is sent out to the configured destination address.
- Mesh Provisioner: A simple static provisioner implementation. This provisioner provisions all the nodes in one mesh network. Additionally, the provisioner also configures key bindings and publication and subscription settings of mesh model instances on these nodes to enable them to talk to each other.

In the following sections, these three example applications will be referred to as the server, the client, and the provisioner, respectively.

The following figure gives the overall view of the mesh network that will be set up in this example. Numbers in parentheses indicate the addresses that are assigned to these nodes by the provisioner.



Mesh network demonstration

Hardware requirements

You need at least three nRF52 Development Kit <https://www.nordicsemi.com/Software-and-Tools/Development-Kits/nRF52-DK> boards for this example:

- One nRF5 development board for the client.
- One nRF5 development board for the provisioner.
- One or more nRF5 development boards for the servers (maximum up to 30 boards).

See Compatibility for information about the supported boards.

Software requirements

Although this example does not require you to install the complete toolchain, you still need the following software:

- nRF5 SDK for Mesh. Download <https://www.nordicsemi.com/Software-and-Tools/Software/nRF5-SDK-for-Mesh> and extract the SDK archive.
- nrfjprog (recommended for flashing the devices). Download and install nRFx Command Line Tools for Windows <https://www.nordicsemi.com/Software-and-Tools/Development-Tools/nRF5-Command-Line-Tools> or nRFx Command Line Tools for Linux (64-bit) <https://www.nordicsemi.com/Software-and-Tools/Development-Tools/nRF5-Command-Line-Tools>.
- Python 3 <https://www.python.org/downloads/> or Python 2.7 <https://www.python.org/downloads/>.

Note

On Debian/Ubuntu, you must reload the udev rules after installing the nRF5x Command Line Tools:

```
$ sudo udevadm control --reload
$ sudo udevadm trigger --action=add
```

Setup

You can find the source code of the example in the following folder: `<InstallFolder>/examples/light_switch`

LED and button assignments

The buttons (1 to 4) are used to initiate certain actions, and the LEDs (1 to 4) are used to reflect the status of actions as follows:

- Server:
  - During provisioning process:
    - LED 3 and 4 blinking: Device identification active.
    - LED 1 to 4: Blink four times to indicate provisioning process is completed.
  - After provisioning and configuration is over:
    - LED 1: Reflects the value of OnOff state on the server.
      - LED ON: Value of the OnOff state is 1 (`true`).
      - LED OFF: Value of the OnOff state is 0 (`false`).
- Client:
  - During provisioning process:
    - LED 3 and 4 blinking: Device identification active.
    - LED 1 to 4: Blink four times to indicate provisioning process is completed.
  - After provisioning and configuration is over, buttons on the client are used to send OnOff Set messages to the servers:
    - Button 1: Send a message to the odd group (address: 0xC003) to turn on LED 1.
    - Button 2: Send a message to the odd group (address: 0xC003) to turn off LED 1.
    - Button 3: Send a message to the even group (address: 0xC002) to turn on LED 1.
    - Button 4: Send a message to the even group (address: 0xC002) to turn off LED 1.

    Note

    If only one server is available, pressing Button 3 turns LED 2 on the client, and Button 4 turns this LED off.
- Provisioner:
  - Button 1: Start provisioning.
  - LED 1: Reflects the state of the provisioning.
    - LED ON: Provisioning of the node is in progress.
    - LED OFF: No ongoing provisioning process.
  - LED 2: Reflects the state of the configuration.
    - LED ON: Configuration of the node is in progress.
    - LED OFF: No ongoing configuration process.

Flashing the example firmware

Before running this example, you need to flash the boards. You must specify the client and the provisioner boards. The server firmware is automatically loaded to the rest of connected boards.

To flash the example firmware on the client and the provisioner:

1. Connect the nRF5 boards to the USB ports.

   Note

   If you do not have a sufficient number of USB ports, you can program the boards one by one. In this case, switch off or disconnect the boards that you have finished programming to prevent them from being overwritten by the script.
2. Decide which board you want to use as client and which one as provisioner.

3. Execute the python script in one of the following ways:
    - let the script ask you to choose the provisioner and client boards based on their SEGGER IDs:

    ```
    nrf5_sdk_for_mesh$ python scripts/quick_start/quick_start_demo.py
    ```

    - specify the provisioner and client boards manually by providing SEGGER IDs for the provisioner (`-p`) and client (`-c`):

    ```
    nrf5_sdk_for_mesh$ python scripts/quick_start/quick_start_demo.py -p 682438729 -c 682204868
    ```

    > Note
    > You can also use the command line switch `-v` if you want to increase the verbosity of the output.

    The script flashes the required SoftDevice and example firmware on the boards.

When the flashing is complete, the script executes a reset operation to start the example applications.

Running and observing the example

After the reset, the provisioner waits for user input. Follow these steps to see the mesh network in action:

1. Press Button 1 on the provisioner board to start the provisioning process:
    - The provisioner first provisions and configures the client and assigns the address 0x100 to the client node.
    - The two instances of the OnOff client models are instantiated on separate secondary elements. For this reason, they get consecutive addresses starting with 0x101.
    - Finally, the provisioner picks up the available devices at random, assigns them consecutive addresses, and adds them to odd and even groups.

    > Note
    > You can use the J-Link RTT viewer to view the RTT output generated by the provisioner. The provisioner prints details about the provisioning and the configuration process in the RTT log. See the subsection below for details.

2. Observe the LED status on the provisioner, client, and server boards.
3. Wait until LED 1 on the provisioner board remains ON steadily for a few seconds, which indicates that all available boards have been provisioned and configured.
4. Press buttons on the client board to change the state of LED 1 on the server boards:
    a. Press Button 1 on the client board to turn ON LED 1 on all servers with ODD addresses.
    b. Press Button 2 on the client board to turn OFF LED 1 on all servers with ODD addresses.
    c. Press Button 3 on the client board to turn ON LED 1 on all servers with EVEN addresses.
    d. Press Button 4 on the client board to turn OFF LED 1 on all servers with EVEN addresses.
5. Press Button 1 on the servers to locally change the state of LED 1 and observe that the client receives the status message from the corresponding server containing the new state value.

Running the example with RTT logs

If you want to see the RTT logs printed during the provisioning and configuration process, complete the following steps:

1. Connect the nRF5 boards to the USB ports.
2. Start J-Link RTT viewer. The Configuration window appears.

> Note
> You can also press the **F2** button or select **File** > **Connect** to open the Configuration window.

3. In the Configuration window, depending on the development kit board chip number you are using, make sure that either `NRF52832_XXAA` or `NRF52840_XXAA` is selected in the Specify Target Device dropdown menu.

4. Click OK. The Emulator selection window appears.
5. Choose the desired board by selecting its USB Identification (SEGGER ID).

> Note
>
> You can open several RTT viewer sessions to observe the RTT log of all the connected boards. In this case, you have to repeat steps 2 to 5 for each board.

6. Repeat steps 2 to 4 from Flashing the example firmware.
7. Go through the steps 1 to 4 from Running and observing the example.
8. If you are monitoring the RTT log for the client, observe messages sent by the servers in response to the acknowledged Set messages.
    - The client example sends acknowledged Set messages only to odd servers, and hence only those servers respond with status messages.
    - Additionally, the provisioner enables publication for all servers, so that they can publish messages to their corresponding group client.
9. Go through the step 5 from Running and observing the example.

More information and further reading

See the Getting started section for information on environment setup, including installing the mesh toolchain, building the mesh stack and examples, running examples, and more.

Once you set up your nRF5 SDK for Mesh environment, see the example documentation for more detailed information about light switch and other examples.