# BLE Mesh Demo

## Purpose

This document will explain the Mesh Demo and the motivations behind the design decisions.

## Architecture

The mesh network realized in this demo has the following points to show:

1. Create a mesh network with two groups of Thingy52s. The two groups are on the same Mesh network (common network key) but they cannot decode the other group’s messages because they have separate application keys. The purpose of creating and bifurcating two groups such as this is to simulate two separate drawers on a station.
2. Demonstrate one Thingy52 from one group to be able to “move” to the other group and be part of the new group. This would demonstrate the cubies switching drawers by wirelessly being reconfigured to belong in the new group.

The above two points should demonstrate isolation of traffic between two drawers while allowing for any cubie to be reconfigured for re-use with either drawer.

Below is a diagram of a node and the capabilities built into it via the firmware:

Please see BLE Mesh profile document for definitions of Elements.

1. Configuration Server
2. Health Server
3. Generic On/Off Server
4. Generic On/Off Client
5. Generic On/Off Server
6. Generic On/Off Client

Element 1

Element 2

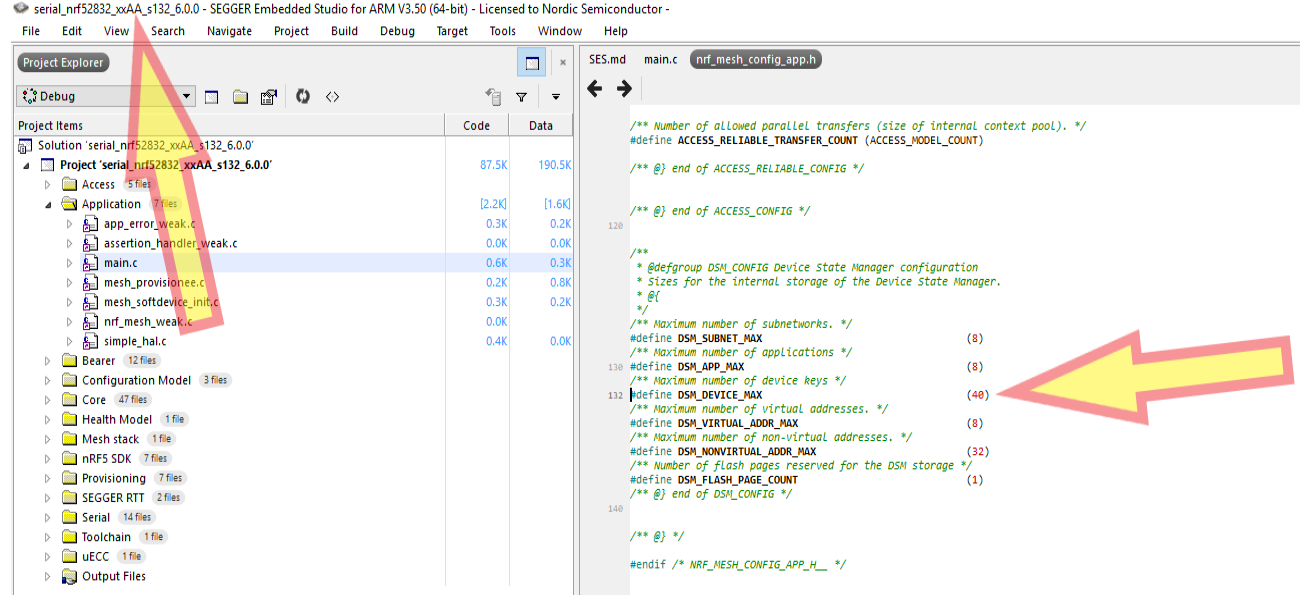
In this demo, we only use the Generic On/Off Server on Element 1. In a mesh network, a server controls access to a resource. In this case, the resource is the LED. We will command the Generic On/Off Server via a Generic On/Off Client. In this case, the Client will be the same as the Provisioner but they do not have to be. In other words, another node in the mesh network with the proper network and application key can successfully command the server to switch the LED on and off. The Provisioner is a special node that assigns each node its role. So, in this instance, a provisioner assigns which application key and which group an unprovisioned node belongs to. The demo comes as pre-provisioned nodes, ready to run. The diagram below illustrates

## Provisioning

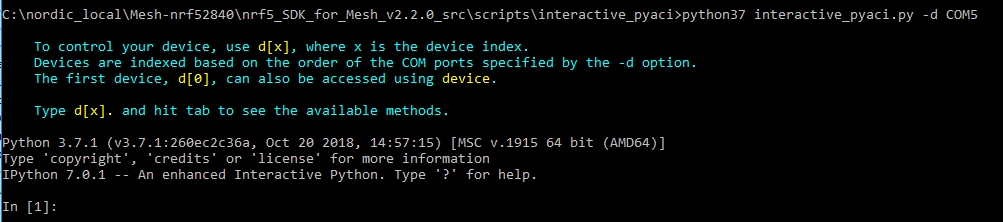
1. All nodes should be programmed with the firmware and they should come up blinking red:



1. Connect a nRF52-DK to the PC; launch SES with the project in <Mesh SDK 2.2>/examples/serial/. Open the nrf\_mesh\_config\_app.h file and modify DSM\_DEVICE\_MAX to surpass or equal the maximum number of nodes you intend to provision using this nRF 52-DK provisioner:



1. Compile, flash, and run the firmware on the nRF52-DK.
2. Follow instructions [here](http://infocenter.nordicsemi.com/topic/com.nordic.infocenter.meshsdk.v2.2.0/md_scripts_interactive_pyaci_README.html?cp=4_1_0_3_0) to install Python 3 and launch the PyACI tool under <Mesh SDK 2.2>/scripts/interactive\_pyaci:



1. Now we will provision each node one-at-a-time. The easiest way is to turn off all the nodes except one which is the one that shall be provisioned. So turn off all nodes except one which would be blinking red indicating that it is unprovisioned.
2. Now we will provision the first node. In PyACI shell, type in the highlighted input; example output shown for clarity (not highlighted):

In **[**1**]:** db **=** MeshDB**(**"database/example\_database.json"**)**

In **[**2**]:** db**.**provisioners

Out**[**2**]:** **[{**'name'**:** 'BT Mesh Provisioner'**,** 'UUID'**:** \_UUID**(**b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'**),** 'allocated\_unicast\_range'**:** **[{**'low\_address'**:** 0010**,** 'high\_address'**:** 7fff**}],** 'allocated\_group\_range'**:** **[{**'low\_address'**:** c000**,** 'high\_address'**:** feff**}]}]**

In **[**3**]:** p **=** Provisioner**(**device**,** db**)**

In **[**4**]:** 2018**-**11**-**13 02**:**31**:**25**,**175 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 02**:**31**:**25**,**181 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 02**:**31**:**25**,**186 **-** INFO **-** COM5**:** SubnetAdd**:** **{**'subnet\_handle'**:** 0**}**

2018**-**11**-**13 02**:**31**:**25**,**197 **-** INFO **-** COM5**:** AppkeyAdd**:** **{**'appkey\_handle'**:** 0**}**

2018**-**11**-**13 02**:**31**:**25**,**202 **-** INFO **-** COM5**:** AppkeyAdd**:** **{**'appkey\_handle'**:** 1**}**

In **[**4**]:**

In **[**4**]:** p**.**scan\_start**()**

In **[**5**]:** 2018**-**11**-**13 02**:**31**:**40**,**555 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 02**:**31**:**51**,**343 **-** INFO **-** COM5**:** Received UUID 0059ffff00000000a5cb9cc547cdcea9 **with** RSSI**:** **-**48 dB

In **[**5**]:**

In **[**5**]:** p**.**scan\_stop**()**

In **[**6**]:** 2018**-**11**-**13 02**:**32**:**03**,**714 **-** INFO **-** COM5**:** Success

In **[**6**]:**

In **[**6**]:** p**.**provision**(**name**=**"Light bulb #1"**)**

In **[**7**]:** 2018**-**11**-**13 02**:**32**:**15**,**090 **-** INFO **-** COM5**:** Provision**:** **{**'context'**:** 0**}**

2018**-**11**-**13 02**:**32**:**15**,**102 **-** INFO **-** COM5**:** Link established

2018**-**11**-**13 02**:**32**:**15**,**144 **-** INFO **-** COM5**:** Received capabilities

2018**-**11**-**13 02**:**32**:**15**,**146 **-** INFO **-** COM5**:** Number of elements**:** 3

2018**-**11**-**13 02**:**32**:**15**,**149 **-** INFO **-** COM5**:** OobUse**:** **{**'context'**:** 0**}**

2018**-**11**-**13 02**:**32**:**15**,**368 **-** INFO **-** COM5**:** ECDH request received

2018**-**11**-**13 02**:**32**:**15**,**384 **-** INFO **-** COM5**:** EcdhSecret**:** **{**'context'**:** 0**}**

2018**-**11**-**13 02**:**32**:**15**,**749 **-** INFO **-** COM5**:** Provisioning complete

2018**-**11**-**13 02**:**32**:**15**,**753 **-** INFO **-** COM5**:** Address**(**es**):** 0x10**-**0x12

2018**-**11**-**13 02**:**32**:**15**,**754 **-** INFO **-** COM5**:** Device key**:** ef48f9898373a1cef29db185ee4058b4

2018**-**11**-**13 02**:**32**:**15**,**756 **-** INFO **-** COM5**:** Network key**:** 18eed9c2a56add85049ffc3c59ad0e12

2018**-**11**-**13 02**:**32**:**15**,**759 **-** INFO **-** COM5**:** Adding device key to subnet 0

2018**-**11**-**13 02**:**32**:**15**,**766 **-** INFO **-** COM5**:** Adding publication address of root element

2018**-**11**-**13 02**:**32**:**15**,**783 **-** INFO **-** COM5**:** DevkeyAdd**:** **{**'devkey\_handle'**:** 8**}**

2018**-**11**-**13 02**:**32**:**15**,**786 **-** INFO **-** COM5**:** AddrPublicationAdd**:** **{**'address\_handle'**:** 0**}**

2018**-**11**-**13 02**:**32**:**15**,**853 **-** INFO **-** COM5**:** Provisioning link closed

In **[**7**]:**

In **[**7**]:** cc **=** ConfigurationClient**(**db**)**

In **[**8**]:** cc**.**force\_segmented **=** **True**

In **[**9**]:** device**.**model\_add**(**cc**)**

In **[**10**]:** cc**.**publish\_set**(**8**,** 0**)**

In **[**11**]:** cc**.**composition\_data\_get**()**

In **[**12**]:** 2018**-**11**-**13 02**:**33**:**01**,**420 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 02**:**33**:**01**,**521 **-** INFO **-** COM5**.**ConfigurationClient**:** Received composition data **(**page 0x00**):** **{**

"cid"**:** "0059"**,**

"pid"**:** "0000"**,**

"vid"**:** "0000"**,**

"crpl"**:** 40**,**

"features"**:** **{**

"relay"**:** 0**,**

"proxy"**:** 2**,**

"friend"**:** 2**,**

"low\_power"**:** 2

**},**

"elements"**:** **[**

**{**

"index"**:** 0**,**

"location"**:** "0000"**,**

"models"**:** **[**

**{**

"modelId"**:** "0000"

**},**

**{**

"modelId"**:** "0002"

**},**

**{**

"modelId"**:** "1000"

**},**

**{**

"modelId"**:** "1001"

**}**

**]**

**},**

**{**

"index"**:** 1**,**

"location"**:** "0000"**,**

"models"**:** **[**

**{**

"modelId"**:** "1000"

**},**

**{**

"modelId"**:** "1001"

**}**

**]**

**}**

**]**

**}**

In **[**12**]:**

In **[**12**]:** cc**.**appkey\_add**(**0**)**

In **[**13**]:** 2018**-**11**-**13 02**:**33**:**09**,**562 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 02**:**33**:**09**,**642 **-** INFO **-** COM5**.**ConfigurationClient**:** Appkey status**:** AccessStatus**.**SUCCESS

2018**-**11**-**13 02**:**33**:**09**,**645 **-** INFO **-** COM5**.**ConfigurationClient**:** Appkey add 0 succeded **for** subnet 0 at node 0010

In **[**13**]:**

In **[**13**]:** cc**.**appkey\_add**(**1**)**

In **[**14**]:** 2018**-**11**-**13 02**:**33**:**13**,**850 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 02**:**33**:**13**,**936 **-** INFO **-** COM5**.**ConfigurationClient**:** Appkey status**:** AccessStatus**.**SUCCESS

2018**-**11**-**13 02**:**33**:**13**,**939 **-** INFO **-** COM5**.**ConfigurationClient**:** Appkey add 1 succeded **for** subnet 0 at node 0010

In **[**14**]:**

In **[**14**]:** cc**.**model\_app\_bind**(**db**.**nodes**[**0**].**unicast\_address**,** 0**,** mt**.**ModelId**(**0x1000**))** #bind to appkey0

In **[**15**]:** 2018**-**11**-**13 02**:**33**:**24**,**401 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 02**:**33**:**25**,**361 **-** INFO **-** COM5**.**ConfigurationClient**:** Model app bind status**:** AccessStatus**.**SUCCESS

2018**-**11**-**13 02**:**33**:**25**,**367 **-** INFO **-** COM5**.**ConfigurationClient**:** Appkey bind 0 to model 1000 at 0010

In **[**15**]:**

In **[**15**]:** cc**.**model\_subscription\_add**(**db**.**nodes**[**0**].**unicast\_address**,** 0xc001**,** mt**.**ModelId**(**0x1000**))** #Add to group 0xC001

2018**-**11**-**13 02**:**33**:**35**,**254 **-** INFO **-** COM5**:** Success

In **[**16**]:** 2018**-**11**-**13 02**:**33**:**36**,**299 **-** INFO **-** COM5**.**ConfigurationClient**:** Model subscription status**:** AccessStatus**.**SUCCESS

2018**-**11**-**13 02**:**33**:**36**,**306 **-** INFO **-** COM5**.**ConfigurationClient**:** Added subscription 'c001' to model 1000 at element 0010

1. We will now provision the second node. We will leave the first node switched on since it has already been provisioned and its LED should be off. Switch on the second node and observe its red LED breathing to signify that it is unprovisioned.
2. In the same PyACI shell from step 6 above, we type in the highlighted Python commands:

In **[**16**]:** p**.**scan\_start**()**

In **[**17**]:** 2018**-**11**-**13 03**:**01**:**21**,**404 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 03**:**01**:**22**,**906 **-** INFO **-** COM5**:** Received UUID 0059ffff00000000d4164b7a7253618c **with** RSSI**:** **-**46 dB

In **[**17**]:**

In **[**17**]:** p**.**scan\_stop**()**

2018**-**11**-**13 03**:**01**:**32**,**809 **-** INFO **-** COM5**:** Success

In **[**18**]:**

In **[**18**]:** p**.**provision**(**name**=**"Light bulb #2"**)**

In **[**19**]:** 2018**-**11**-**13 03**:**01**:**55**,**598 **-** INFO **-** COM5**:** Provision**:** **{**'context'**:** 0**}**

2018**-**11**-**13 03**:**01**:**55**,**609 **-** INFO **-** COM5**:** Link established

2018**-**11**-**13 03**:**01**:**55**,**665 **-** INFO **-** COM5**:** Received capabilities

2018**-**11**-**13 03**:**01**:**55**,**666 **-** INFO **-** COM5**:** Number of elements**:** 3

2018**-**11**-**13 03**:**01**:**55**,**669 **-** INFO **-** COM5**:** OobUse**:** **{**'context'**:** 0**}**

2018**-**11**-**13 03**:**01**:**55**,**879 **-** INFO **-** COM5**:** ECDH request received

2018**-**11**-**13 03**:**01**:**55**,**890 **-** INFO **-** COM5**:** EcdhSecret**:** **{**'context'**:** 0**}**

2018**-**11**-**13 03**:**02**:**00**,**318 **-** INFO **-** COM5**:** Provisioning complete

2018**-**11**-**13 03**:**02**:**00**,**321 **-** INFO **-** COM5**:** Address**(**es**):** 0x13**-**0x15

2018**-**11**-**13 03**:**02**:**00**,**325 **-** INFO **-** COM5**:** Device key**:** f2efd4734a036568fda5d6f499292071

2018**-**11**-**13 03**:**02**:**00**,**331 **-** INFO **-** COM5**:** Network key**:** 18eed9c2a56add85049ffc3c59ad0e12

2018**-**11**-**13 03**:**02**:**00**,**337 **-** INFO **-** COM5**:** Adding device key to subnet 0

2018**-**11**-**13 03**:**02**:**00**,**340 **-** INFO **-** COM5**:** Adding publication address of root element

2018**-**11**-**13 03**:**02**:**00**,**361 **-** INFO **-** COM5**:** DevkeyAdd**:** **{**'devkey\_handle'**:** 9**}**

2018**-**11**-**13 03**:**02**:**00**,**362 **-** INFO **-** COM5**:** AddrPublicationAdd**:** **{**'address\_handle'**:** 1**}**

2018**-**11**-**13 03**:**02**:**00**,**422 **-** INFO **-** COM5**:** Provisioning link closed

In **[**19**]:**

In **[**19**]:** cc**.**publish\_set**(**9**,** 1**)**

In **[**20**]:** cc**.**composition\_data\_get**()**

In **[**21**]:** 2018**-**11**-**13 03**:**02**:**23**,**056 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 03**:**02**:**23**,**143 **-** INFO **-** COM5**.**ConfigurationClient**:** Received composition data **(**page 0x00**):** **{**

"cid"**:** "0059"**,**

"pid"**:** "0000"**,**

"vid"**:** "0000"**,**

"crpl"**:** 40**,**

"features"**:** **{**

"relay"**:** 0**,**

"proxy"**:** 2**,**

"friend"**:** 2**,**

"low\_power"**:** 2

**},**

"elements"**:** **[**

**{**

"index"**:** 0**,**

"location"**:** "0000"**,**

"models"**:** **[**

**{**

"modelId"**:** "0000"

**},**

**{**

"modelId"**:** "0002"

**},**

**{**

"modelId"**:** "1000"

**},**

**{**

"modelId"**:** "1001"

**}**

**]**

**},**

**{**

"index"**:** 1**,**

"location"**:** "0000"**,**

"models"**:** **[**

**{**

"modelId"**:** "1000"

**},**

**{**

"modelId"**:** "1001"

**}**

**]**

**}**

**]**

**}**

In **[**21**]:**

In **[**21**]:** cc**.**appkey\_add**(**0**)**

In **[**22**]:** 2018**-**11**-**13 03**:**02**:**44**,**436 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 03**:**02**:**44**,**515 **-** INFO **-** COM5**.**ConfigurationClient**:** Appkey status**:** AccessStatus**.**SUCCESS

2018**-**11**-**13 03**:**02**:**44**,**519 **-** INFO **-** COM5**.**ConfigurationClient**:** Appkey add 0 succeded **for** subnet 0 at node 0013

In **[**22**]:**

In **[**22**]:** cc**.**appkey\_add**(**1**)**

In **[**23**]:** 2018**-**11**-**13 03**:**02**:**48**,**471 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 03**:**02**:**48**,**562 **-** INFO **-** COM5**.**ConfigurationClient**:** Appkey status**:** AccessStatus**.**SUCCESS

2018**-**11**-**13 03**:**02**:**48**,**568 **-** INFO **-** COM5**.**ConfigurationClient**:** Appkey add 1 succeded **for** subnet 0 at node 0013

In **[**23**]:**

In **[**23**]:** cc**.**model\_app\_bind**(**db**.**nodes**[**1**].**unicast\_address**,** 0**,** mt**.**ModelId**(**0x1000**))** #bind to appkey0

In **[**24**]:** 2018**-**11**-**13 03**:**03**:**02**,**567 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 03**:**03**:**02**,**632 **-** INFO **-** COM5**.**ConfigurationClient**:** Model app bind status**:** AccessStatus**.**SUCCESS

2018**-**11**-**13 03**:**03**:**02**,**636 **-** INFO **-** COM5**.**ConfigurationClient**:** Appkey bind 0 to model 1000 at 0013

In **[**24**]:**

In **[**24**]:** cc**.**model\_subscription\_add**(**db**.**nodes**[**1**].**unicast\_address**,** 0xc001**,** mt**.**ModelId**(**0x1000**))** #Add to group 0xC001

In **[**25**]:** 2018**-**11**-**13 03**:**03**:**11**,**058 **-** INFO **-** COM5**:** Success

2018**-**11**-**13 03**:**03**:**11**,**189 **-** INFO **-** COM5**.**ConfigurationClient**:** Model subscription status**:** AccessStatus**.**SUCCESS

2018**-**11**-**13 03**:**03**:**11**,**192 **-** INFO **-** COM5**.**ConfigurationClient**:** Added subscription 'c001' to model 1000 at element 0013

1. Switch on the third node. Now we will provision it, same as above, now shown for brevity without the output:

In **[**27**]:** p**.**scan\_start**()**

In **[**28**]:** p**.**scan\_stop**()**

In **[**29**]:** p**.**provision**(**name**=**"Light bulb #3"**)**

In **[**30**]:** cc**.**publish\_set**(**10**,** 2**)**

In **[**31**]:** cc**.**composition\_data\_get**()**

In **[**32**]:** cc**.**appkey\_add**(**0**)**

In **[**33**]:** cc**.**appkey\_add**(**1**)**

In **[**34**]:** cc**.**model\_app\_bind**(**db**.**nodes**[**2**].**unicast\_address**,** 0**,** mt**.**ModelId**(**0x1000**))** #bind to appkey0

In **[**35**]:** cc**.**model\_subscription\_add**(**db**.**nodes**[**2**].**unicast\_address**,** 0xc001**,** mt**.**ModelId**(**0x1000**))** #Add to group 0xC001

1. Now we will provision the fourth node so switch that node on and it should signal unprovisioned state. However, we will bind it to Application Key 1 instead of Application Key 0. We will note a pattern here that we can use to provision any number of devices:

