

Step by step guide mesctl with Zephyr

Introduction

This step-by-step guide is based on the "Tutorial-How-to-set-up-BlueZ_Part2-3" guide [1], but uses the Raspberry Pi 4 and an nRF52-DK specifically. In the following chapters, all steps are given to switch on and off an LED on the nRF52-DK, using bluez meshctl on the Raspberry Pi 4. In addition, pictures are added to visualise the steps. Also, PuTTY will be running on a Windows 10 laptop.

Within all chapters, the left column shows the (terminal/meshctl) commands that should typed in the Raspberry Pi terminal. The right column shows a short description about the left command. Pictures about the steps are given below the specific command(s) for this step.

Note 1: During long in-active behaviour, the meshctl will disconnect the provisioned device. Step 6b (type: connect) should be re-done before continuing.

Note 2: After stopping/exiting the meshctl, it may be needed to turn the nRF52-DKs off and on before reconnecting.

Starting meshctl

Step	Raspberry Pi terminal - command	Description
1.	cd	Go back to main directory
2.	cd bluez-5.50/mesh	Go to directory with meshctl
3.	meshctl	Launch meshctl

Provisioning

First turn on the nRF52-DK and connect its micro-USB input to the USB input of the Windows 10 laptop (can be any other device with a serial monitor, like PuTTY, installed).

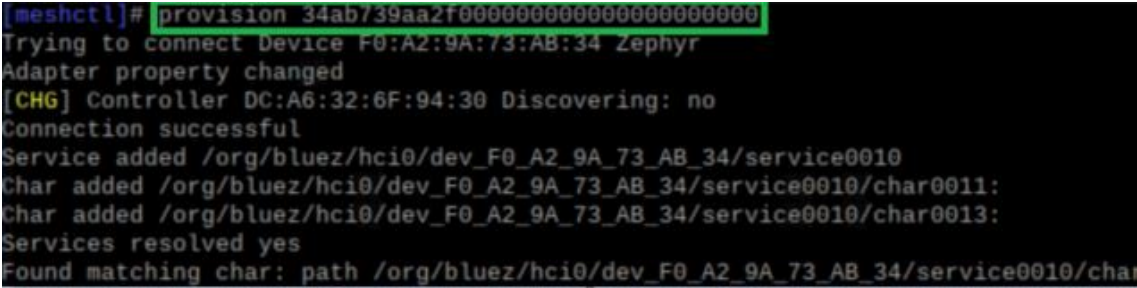
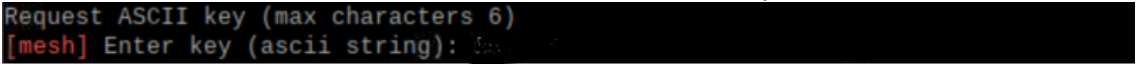
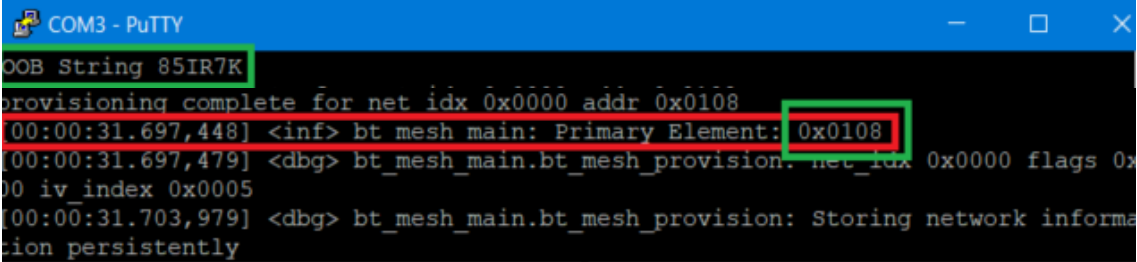
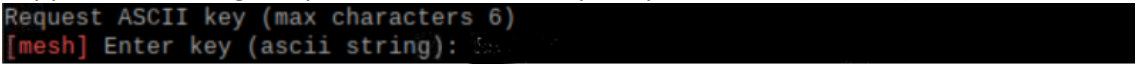
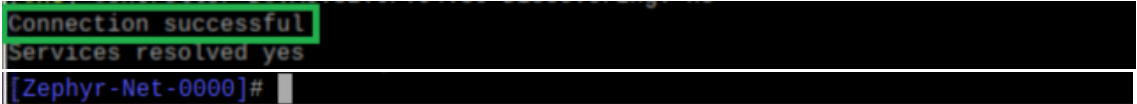
Step	meshctl - command(s)	Description
4.	Press Enter	Reveal meshctl prompt
5.	discover-unprovisioned on	Discover unprovisioned BLE mesh devices

When a new device appears, make sure it states the correct name after [NEW] (Here: Zephyr) and copy the Device UUID: → continue with step 6a-1.

```
[meshctl]# discover-unprovisioned on
SetDiscoveryFilter success
Discovery started
Adapter property changed
[CHG] Controller DC:A6:32:6F:94:30 Discovering: yes
Services resolved no
    Mesh Provisioning Service (00001827-0000-1000-8000-00805f9b34fb)
        Device UUID: 34ab739aa2f000000000000000000000
        00B: 0000
[NEW] Device F0:A2:9A:73:AB:34 Zephyr
[meshctl]#
```

When no new device appears, it may already be provisioned: → continue with step 6b.

```
[meshctl]# discover-unprovisioned on
SetDiscoveryFilter success
Discovery started
Adapter property changed
[CHG] Controller DC:A6:32:6F:94:30 Discovering: yes
[meshctl]#
```

Step	meshctl - command(s)	Description
6a-1.	provision 34ab739aa2f000000000000000000000	Provision device with its UUID (see step 5). (Note: this is 1 command, do not press enter in between).
<p>When done correctly, this can be seen on the terminal:</p>  <p>Somewhere in between that, it should now ask for an (ASCII) key:</p>  <p>The serial terminal should look like this:</p>  <p>Copy the OOB String (here: 851R7K – for step 6a-2) and the Primary Element without 0x (0108 – for step 8). Lastly, save the net_idx (above the Primary Element – for reconnecting = step 6b) without 0x, here 0x0000. The next connected device's, net_idx should be 0x0001 (1 higher). It may be that the net_idx will be given as 0x0000, but you must use 1 higher for each added device. Continue with step 6a-2.</p>		
6a-2.	851R7K	<p>This is the OOB string from the serial terminal (see step 6a-1.) Note that this will be different every time, for security purpose.</p> <p>Provisioning is only accepted when the correct key is provided.</p>
<p>Copy the OOB String and paste it here in the Raspberry Pi terminal (meshctl):</p>  <p>Something similar to this should appear:</p>  <p>If it fails, try again from step 5 (most likely you must go to step 6b now). If it continuous to fails, restart at step 1. Continue with step 7 if it worked.</p>		
6b.	connect OR: connect <net_idx>	<p>Connect to the nRF52-DK, which was previously provisioned and not changed.</p> <p><net_idx> can be found in the serial monitor above the line with the Primary</p>

Step	meshctl - command(s)	Description
		Element (given in step 6a-1 during provisioning of a new device).

Something similar should appear. **Note:** This can take some time. If nothing happens or it didn't connect properly (i.e. showing the new command request from meshctl), try again several times.

```
[meshctl]# connect
Looking for mesh network with net index 0000
SetDiscoveryFilter success
Discovery started
Adapter property changed
[CHG] Controller DC:A6:32:6F:94:30 Discovering: yes

      Mesh Proxy Service (00001828-0000-1000-8000-00805f9b34fb)
      Network Beacon for net index 0000
Trying to connect to mesh
Adapter property changed
[CHG] Controller DC:A6:32:6F:94:30 Discovering: no
Connection successful
Services resolved yes
Found matching char: path /org/bluez/hci0/dev_F0_A2_9A_73_AB_34/service0010/char
```

Note the line with "Connection successful". The new command request look like this:

```
[Zephyr-Net-0000]#
```

Note that it contains the device name (Zephyr) and the net idx (Net-0000) between []. The next will be 1 higher than this, 0001 (although it will probably state 0000).

Something similar to this should appear on the serial terminal (PuTTY):

```
COM3 - PuTTY
provisioning complete for net idx 0x0000 addr 0x0108
[00:00:31.697,448] <inf> bt mesh main: Primary Element: 0x0108
[00:00:31.697,479] <dbg> bt_mesh_main.bt_mesh_provision: net_idx 0x0000 flags 0x
00 iv_index 0x0005
[00:00:31.703,979] <dbg> bt_mesh_main.bt_mesh_provision: Storing network informa
tion persistently
```

Copy the Primary Element without 0x (0108 - step 8).

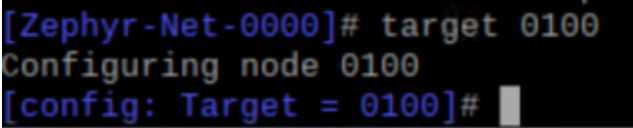
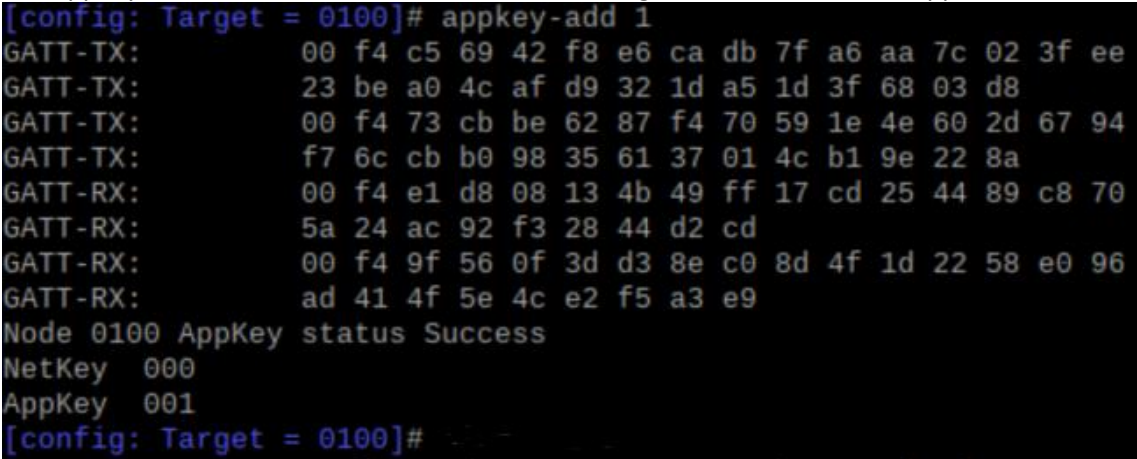
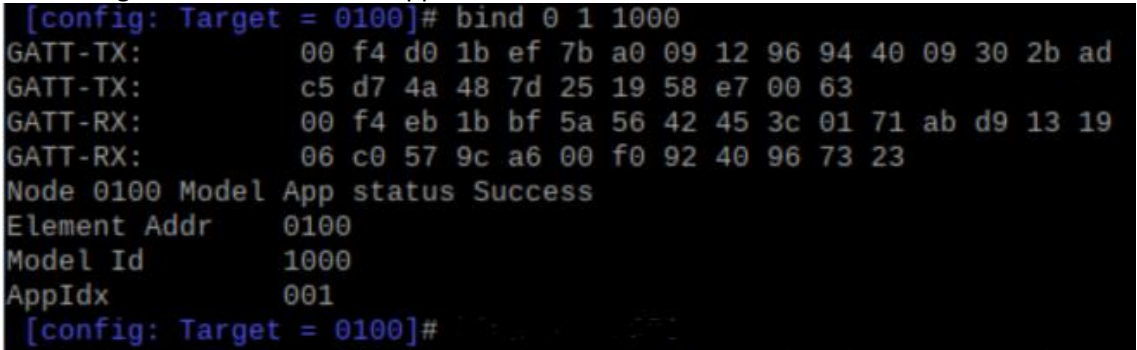
If it fails, try again (it can take some time to re-establish the connection). If it continues to fail, restart at step 1. Continue with step 7 if it worked.

Configuration

This should only be done once per LED and like-wise elements. When you already did this for the specific LED/like-wise element, continue with step 15 in the next chapter.

Step	meshctl - command(s)	Description
7.	menu config	Entering Configuration Menu of meshctl
8.	target 0108	This Unicast Address (Primary Element from serial terminal) will from now on be used for any target command. (Always refer to the serial monitor, PuTTY, for the correct target. Errors, like "Destination not set" or "Node xxxx not found" – Change this target device when another must be selected).

The name should change to [config: Target = 01008]:

Step	meshctl - command(s)	Description
	 <pre>[Zephyr-Net-0000]# target 0100 Configuring node 0100 [config: Target = 0100]#</pre> <p>Note line 2 indicating configuration on target node 0100: <i>"Configuring node 0100"</i>. Also, note that the Unicast Address in the screenshot is 0100, because another Unicast Address was given via the serial terminal (step 6a-2 or step 6b). All following screenshots will contain Unicast Address 0100, but keep in mind that you should use your own Unicast Address.</p>	
9.	appkey-add 1	Load AppKey #1 from prov_db.json
	<p>The appkey should not be used before, and something similar to this should appear:</p>  <pre>[config: Target = 0100]# appkey-add 1 GATT-TX: 00 f4 c5 69 42 f8 e6 ca db 7f a6 aa 7c 02 3f ee GATT-TX: 23 be a0 4c af d9 32 1d a5 1d 3f 68 03 d8 GATT-TX: 00 f4 73 cb be 62 87 f4 70 59 1e 4e 60 2d 67 94 GATT-TX: f7 6c cb b0 98 35 61 37 01 4c b1 9e 22 8a GATT-RX: 00 f4 e1 d8 08 13 4b 49 ff 17 cd 25 44 89 c8 70 GATT-RX: 5a 24 ac 92 f3 28 44 d2 cd GATT-RX: 00 f4 9f 56 0f 3d d3 8e c0 8d 4f 1d 22 58 e0 96 GATT-RX: ad 41 4f 5e 4c e2 f5 a3 e9 Node 0100 AppKey status Success NetKey 000 AppKey 001 [config: Target = 0100]#</pre> <p>Note the line <i>"Node 0100 AppKey status Success"</i>.</p>	
10.	bind 0 1 1000	Expose Generic On/Off server, locked by AppKey #1. Syntax: bind <ele_idx> <app_idx> <mod_id>. Where: <ele_index> = 0, <app_idx> = appkey nr from step 9, and <mod id> = 1000
	<p>Something similar to this should appear on the terminal:</p>  <pre>[config: Target = 0100]# bind 0 1 1000 GATT-TX: 00 f4 d0 1b ef 7b a0 09 12 96 94 40 09 30 2b ad GATT-TX: c5 d7 4a 48 7d 25 19 58 e7 00 63 GATT-RX: 00 f4 eb 1b bf 5a 56 42 45 3c 01 71 ab d9 13 19 GATT-RX: 06 c0 57 9c a6 00 f0 92 40 96 73 23 Node 0100 Model App status Success Element Addr 0100 Model Id 1000 AppIdx 001 [config: Target = 0100]#</pre> <p>Note Success in the line: <i>"Node 0100 Model App status Success"</i>. This is enough for turning on/off the LED, so continue with step 14. For other functions, it may be necessary to do more (steps 11. to 13.). So if your project isn't the LED, continue with step 11.</p>	
11.	bind 0 1 1001	Repeating previous step, but with <mod_id> = 1001
	Something similar to step 10, but other in GATT-xx, should appear on the terminal:	



Step	meshctl - command(s)	Description
	<pre>[config: Target = 0100]# bind 0 1 1001 GATT-TX: 00 f4 64 8a 77 96 a6 f2 fd 7b cf a5 c2 d0 ab e9 GATT-TX: 7c c6 14 c2 60 31 ad 29 ed 00 63 GATT-RX: 00 f4 2f 90 2c f0 8c 1b 03 f4 0c 23 74 11 c5 8f GATT-RX: 7e 87 a0 43 d6 98 c2 a8 66 b2 68 a5 Node 0100 Model App status Success Element Addr 0100 Model Id 1001 AppIdx 001 [config: Target = 0100]#</pre>	
Note Success in the line: "Node 0100 Model App status Success".		
12.	sub-add 0100 c000 1000	<p>Add subscribe to On/Off Updates at Group Address c000. Where:</p> <p><ele_addr> = target's Unicast Address, <sub_addr> = Group Address = c000, and <model_id> = <mod_id> from step 10</p>
Something similar to step 10 and 11, but other in GATT-xx, should appear on the terminal:		
	<pre>[config: Target = 0100]# sub-add 0100 c000 1000 GATT-TX: 00 f4 05 b5 f6 25 1c e5 07 88 ee 76 40 5b f8 52 GATT-TX: ea 68 33 a3 ea 5e 73 5b f3 f8 dc GATT-RX: 00 f4 01 24 d0 76 11 cc dd fa df 57 6e 0d 9c 14 GATT-RX: 00 56 ee ad a4 65 fc 5c ea e9 c1 3a Node 0100 Subscription status Success Element Addr 0100 Model Id 1000 Subscr Addr c000 [config: Target = 0100]#</pre>	
Note Success in the line: "Node 0100 Subscription status Success".		
13.	pub-set 0100 c000 1 0 5 1001	<p>Publish On/Off Updates at Group Address c000. Where:</p> <p><ele-addr> = target's Unicast Address, <pub_addr> = Group Address = c000, <app_id> = appkey nr from step 9, <per (setp res)> = period in ms, <re-xmt (cnt per)> = rexmit count, and <mod_id> = <mod_id> from step 11</p>
Something similar to step 10 to 12, but other in GATT-xx, should appear on the terminal:		


Step	meshctl - command(s)	Description
	<pre> [config: Target = 0100]# pub-set 0100 c000 1 0 5 1001 GATT-TX: 00 f4 77 f6 79 35 fb 83 f2 43 53 3b a1 94 f9 76 GATT-TX: 0e 67 bd 58 6d 4f 23 e4 87 e7 4f b1 6d c7 GATT-TX: 00 f4 df 2a 34 d1 2c 59 28 70 26 fb bd 23 42 7d GATT-TX: 5d f0 83 8e 9b 16 GATT-RX: 00 f4 f8 b0 e5 92 27 3a bd 90 49 6a 4d 0e 83 78 GATT-RX: 0c 36 d8 3e e0 04 5c 58 6f GATT-RX: 00 f4 2b f6 44 23 d8 fe b8 e2 d9 27 17 0d 2a a5 GATT-RX: 0d 41 fa 2d 69 60 97 86 d7 ed 0d 59 8f 37 GATT-RX: 00 f4 0c e0 ff 6c 14 87 d4 a2 f0 80 5b e6 8d c3 GATT-RX: b9 65 a9 34 32 fb e9 97 Node 0100 Publication status Success Element Addr 0100 Model Id 1001 Pub Addr c000 Period 0 ms Rexmit count 0 Rexmit steps 5 GATT-TX: 00 f4 f0 ef 94 27 9d ac 24 c5 c5 8c 4e a5 66 88 GATT-TX: 6a 0f b7 15 94 48 83 ea 27 [config: Target = 0100]# </pre> <p>Note Success in the line: "Node 0100 Publication status Success".</p>	
14.	back	Go back to the main menu of meshctl

Turning on/off the LED

When all configurations and connections are (re-)established, steps 17 and 18 can be repeated to turn on/off the LED several times. Changing between devices (nRF52-DKs) can be done by repeating step 6b and continuing with step 17 and 18 (**note**: the configurations from step 7 onwards should be already done. If not, do this first). Changing between LEDs and like-wise elements, can be done by targeting a different Unicast Address in step 16 (**note**: this is on the same device). Step 7 and onwards should be repeated when not yet done for this target.

Step	meshctl - command(s)	Description
15.	menu onoff	Entering onoff Menu of meshctl
16.	target 0108	Target the new node (use specific Unicast Address).
<p>This should appear on the terminal: (Unicast Address be different, but consistent).</p> <pre> [config: Target = 0108]# target 0108 Controlling ON/OFF for node 0108 [on/off: Target = 0108]# █ </pre> <p>Notify the change from [config: Target = 0108] to [on/off: Target = 0108]. Now the node 0108 can be controlled on/off instead of changing configurations for this node.</p>		
17.	onoff 0	Set on/off client to 0 (= off). LED1 on nRF52-DK should turn off.
The Raspberry Pi terminal should show this:		

Step	meshctl - command(s)	Description
	<pre>[on/off: Target = 0108]# onoff 0 GATT-TX: 00 f4 06 2f 74 ab f6 dc 1b d0 9f 80 ff 4c 24 2c GATT-TX: 38 e2 db 5f 3e 19 db GATT-RX: 00 f4 c4 45 9a a5 3d cc e6 65 47 a9 c7 48 ed 65 GATT-RX: 58 50 79 43 4e e9 On Off Model Message received (1) opcode 8204 00 [on/off: Target = 0108]#</pre> <p>Note the last line copies the <code>onoff</code> statement 0.</p> <p>The serial monitor, PuTTY, shows this:</p>  <p>Note the last line showing <code>onoff 0x00</code>, which indicates a <code>onoff</code> of 0 was commanded. Also note the Unicast Address, before <code>onoff 0x00</code>, corresponds with the targeted Unicast Address.</p>	
18.	<code>onoff 1</code>	Set on/off client 1 (= on). LED1 on nRF52-DK should turn on.
	<pre>[on/off: Target = 0108]# onoff 1 GATT-TX: 00 f4 f3 be 5b 4b 93 31 98 c3 bf dd a7 50 4c 1b GATT-TX: a3 f9 db 6a d5 2d 90 GATT-RX: 00 f4 b1 15 21 01 9d f1 84 26 d6 c9 86 6d 5c 6a GATT-RX: 5d e0 3a ba 58 77 On Off Model Message received (1) opcode 8204 01 [on/off: Target = 0108]#</pre> <p>Note the last line copies the <code>onoff</code> statement 1.</p> <p>The serial monitor, PuTTY, shows this:</p>  <p>Note the last line showing <code>onoff 0x01</code>, which indicates a <code>onoff</code> of 1 was commanded. Also note the Unicast Address, before <code>onoff 0x01</code>, corresponds with the targeted Unicast Address. Other commands, like <code>0xA0</code> or <code>0x08</code>, can also be typed, but do not effect the LED other than using <code>0x01</code>.</p>	
19.	<code>get</code>	Show on/off Client status. Can be done at any time, after configuration. Shows the LED status (0 or 1).
	<pre>[on/off: Target = 0108]# get GATT-TX: 00 f4 26 57 40 4e 30 cc 4d a7 a7 40 a5 20 a7 d2 GATT-TX: bf 4f ae c0 aa GATT-RX: 00 f4 3a c3 53 9f c4 00 91 a9 74 29 25 9f 02 a1 GATT-RX: d2 ec 02 fa 68 5e On Off Model Message received (1) opcode 8204 01 [on/off: Target = 0108]#</pre> <p>Note the last shows 01, which means that the LED is on (this was previously commanded).</p>	

Step	meshctl - command(s)	Description
The serial monitor, PuTTY, shows this:		
		
Note this line shows <i>onoff 0x01</i> , which means that the LED is on (this was previously commanded). Also note the Unicast Address, before <i>onoff 0x01</i> , corresponds with the targeted Unicast Address.		
20.	exit	Exit meshctl. Mesh details have been automatically updated into <code>prov_db.json</code>

References

- [1] K. Ren, "Step-by-Step Guide How to deplo w BlueZ v5.50 on Raspberry Pi 3 and Use it," 12 October 2018. [Online]. Available: https://3pl46c46ctx02p7rzdsvsg21-wpengine.netdna-ssl.com/wp-content/uploads/2019/03/Tutorial-How-to-set-up-BlueZ_Part2-3.pdf. [Accessed 9 July 2020].