

Mesh Provisioning

Hardware academy mesh workshop

1 Scope

The scope of this hands-on is to get experience with the mesh provisioning app, available for iOS and Android.

The session is based on a mesh demo developed for the nRF52 development kits, that allows the button on the devkit to act as a generic on off client, and the LED on the devkit to act as a generic on off server.

Using the provisioning app on the phone, the user can provision the devkits into a mesh, and configure the generic on off client and server.

The client can be configured to publish messages to an address every time the button is pushed, and the server can be configured to subscribe to addresses so that the LED is updated whenever someone publishes to the address.

By using group addresses multiple clients can publish and multiple servers can subscribe to the same address.

Since each participant only have a single devkit, everyone should form up in groups of 2-4 for this exercise.

The first task is to set up a simple mesh with two devkits and configure the client in one to publish to the same group address as the server in the other. This will have the LED on one devkit be controlled by the button on the other:



The second task requires all the devkits to be provisioned into the mesh, and two separate group addresses to be used. The devkits will be split into two groups that can be individually controlled, and two of the devkits should be set to publish into these group addresses. This will allow two buttons to turn on and off either half of the LED's.



1.1.1 PLEASE NOTE:

• There is a troubleshooting section at the end of this document if you experience issues during the hands-on.

1.2 Prerequisites

- 1) Each participant will receive one nRF52840 DK.
- 2) Each group needs a single smart phone with the provisioning app installed, available for <u>Android</u> and <u>iOS</u>

1.3 Setting up the Mesh SDK

1) Download the modified mesh SDK from here: https://github.com/NordicPlayground/nrf52-mesh-light-switch-client-server-proxy

Download the nRF5 SDK v15.0.0 from here: https://developer.nordicsemi.com/nRF5_SDK/nRF5_SDK_v15.x.x/

It is recommended to put both the mesh SDK and the nRF5 SDK in the following folder: c:\NordicSemi $\$

Make sure to use Segger Embedded Studio v3.40 or newer.

- Navigate to: C:\NordicSemi\nrf52-mesh-light-switch-client-serverproxy\examples\light_switch_london\proxy_server\ & open the light_switch_proxy_server_nrf52840_xxAA_s140_6_0_0.emproject file
- 3) Click on *Tools -> Options -> Building tab ->* double click on Global Macros

Make sure that the SDK_ROOT & MESH_ROOT point to the correct locations:



<u>E</u> dit <u>V</u> iew <u>S</u> earch	n <u>N</u> avigate	Project	Build	Debug	Target	Tools	Window	Help				(
ject Explorer			main.c	simple_p	wm.c	simple_pw	vm.h			🔁 🧧	¥ ×	Contents			é	>
Debug ·	- 🗆 🗀	S O	← →								-	< →		Searc	ch Help	
			· · ·		nput ena	ble(app)	rtt input har	dler, RTT_INPUT_POLL_PER	IOD MS):					Prove		-
ect Items	Code	Data ^		ERROR	CHECK(esh_stack	k_start());						Embedded S	tudio for ARN	4	
olution 'light_switch_pro:	103.4K	188.7K		1£ (1	m device	provisio	(hence					Intro	duction			
Project 'light_switch_p A Ccess 5 files	103.4K	188.7K [2.0K]		< `	-	-	,					👂 🍝 SEGO	ER Embedde	d Studio User	r Guide	
Access offes	(20.4K) 6.8K	0.5K						th_data[NRF_MESH_KEY_SI] t prov start params =	<pre>E] = STATIC_AUTH_DATA;</pre>			🕨 🥏 Com				
Access_loopbi access_loopbi	0.0K	0.5K		(_	_									d Studio for ARI	М
access_toopb access_publis	1.2K	0.0K	410		.p_st	atic_data	a = statio	_auth_data,				ARM				
	1.6K	0.0K			.prov	vice uni	e_cb = provis	ioning_complete_cb,				🛛 🖉 Usin				
access_reliabl	1.5K	0.1K		}	;	****C_0**	- 1000					D C Lib				
SEGGER Embedded Studi	in for ARM V	3 50 - Propert	v Editor					<pre>start(&prov_start_pa</pre>	rams));			Mult				
· Secontempedata staa		siste i repen	y canton									🖒 🍏 Inpu	t and output			
Set Global Macros								igure_device_uuid_get	0;			👂 🧼 Local				
Set Global Macros								"Device UUID ", p_uu	id, NRF_MESH_UUID_SIZE);			👂 🧼 Com				
lobal Macros:												🔷 C++				
													ies Reference	2		
SDK_ROOT=C:/NordicSemi/n MESH_ROOT=C:/NordicSemi	RF5_SDK_15	0.0_a53641a						C			_	🕞 🧼 Appe	ndices			_
mesh_noon-ci/noraicseiii				, bioy				SEGGER Embedd	ed Studio for ARM V3.50 - Options							
		-		, prost				SEGGER Embedd	ed Studio for ARM V3.50 - Options							
				proxy				 SEGGER Embedd Environment Option 						Search		
		-		proxy										Search		
				proxy						Value				Search		
				proxy				e Environment Option t D Building	,	Value				Search		
				, prost				e Environment Option	Option	Value				Search		
		-		, prosy				e Environment Option t Building Debug	Option					Search		
				, prosy				e Environment Option t D Building	Option Build Automatically Build Before Debug Confirm Debugger Stop Display EfA	Yes No Yes				Search		
				, prosy				p Building Debug Environment	Option Build Automatically Build Before Debug Confirm Debugger Stop Display FTA Display Progress Bar	Yes No Yes Yes				Search		
n				, prosy				e Environment Option t Building Debug	Option Build Automatically Build Before Debug Contim Debugger Stop Display TrGA Display Progress Bar Echo Build Command Lines	Yes No Yes Yes No				Search		
				, prosy				e Environment Option 9 Building Debug Environment Languages	Option	Yes No Yes No Yes				Search		
				, , , , , , , , , , , , , , , , , , , ,				p Building Debug Environment	Option Build Automatically Build Before Debug Contim Debugger Stop Display Progress Bar Echo Build Command Lines Echo Raw Error/Warning Output Find Error Arte Building	Yes No Yes No Yes Yes						
				, , , , , , , , , , , , , , , , , , , ,				P Building Debug Environment Languages Source Control	Option Automatically Build Before Debug Automatically Build Before Debug Conting UTA Debugger Stop Debugger Approx San Echo Build Command Lines Echo Raw Ecro/Warning Output Find Error Arter Building Global Marcos	Yes No Yes No Yes Yes SDK_R(DOT=C:/I	iordicSemi/nR	F5_SDK_15.0		ESH_ROOT=C:**	
				, , , , , , , , , , , , , , , , , , , ,				e Environment Option 9 Building Debug Environment Languages	Option Build Automatically Build Before Debug Contim Debugger Stop Display PTA Display Progress Bar Echo Build Command Lines Echo Raw Error/Warning Output Find Error Arter Building Global Macros Keep Going On Error	Yes No Yes Yes Yes SDK_R(No	DOT=C:/I	lordicSemi/nR	F5_SDK_15.0		ESH_ROOT=C:**	
		_						Environment Option Building Debug Environment Languages Source Control Text Editor	Option Automatically Build Before Debug Automatically Build Before Debug Conting UTA Debugger Stop Debugger Approx San Echo Build Command Lines Echo Raw Ecro/Warning Output Find Error Arter Building Global Marcos	Yes No Yes No Yes Yes SDK_R(DOT=C:/I	lordicSemi/nR	F5_SDK_15.0		ESH_ROOT=C(
		-						P Building Debug Environment Languages Source Control	Option Build Automatically Build Before Debug Automatically Build Before Debug Continm Debugger Stop Desploy ITA Ench Build Command Lines Since Policy Crit Before Building Size Policy Crit Before Building Size Policy Crit Before Building	Yes No Yes Yes Yes SDK_R(No Yes Yes					ESH_ROOT= C	
		-				ОК	Cancel	Environment Option Building Debug Environment Languages Source Control Text Editor	Option Build Automatically Build Before Debug Contrim Debugger Stop Display FIA Display Progress Bar Erch Build Command Lines Erch Baru Error/Warning Output Find Error Arter Building Global Macros Keep Going On Error Save Project File Before Building Show Build Information Toolchain Root Directory	Yes No Yes Yes Yes SDK_R(No Yes Yes		lordicSemi/nR /S(GCCTarget)			ESH_ROOT=C	
		-				ОК	Cancel	Environment Option Building Debug Environment Languages Source Control Text Editor	Option Build Automatically Build Before Debug Contim Debugger Stop Display FIA Display Progress Bar Erch Build Command Lines Erch Raw Error/Warning Output Find Error Arter Building Global Macros Keep Going On Error Save Project File Before Building Show Build Information	Yes No Yes Yes Yes SDK_R(No Yes Yes					ESH_ROOT=C	
	across all col	utions and m			brary file		Cancel	Environment Option Building Debug Environment Languages Source Control Text Editor Windows	Option Build Automatically Build Before Debug Contrim Debugger Stop Display FIA Display Progress Bar Erch Build Command Lines Erch Baru Error/Warning Output Find Error Arter Building Global Macros Keep Going On Error Save Project File Before Building Show Build Information Toolchain Root Directory	Yes No Yes Yes Yes SDK_R(No Yes Yes					ESH_ROOT=C	
	i across all sol	utions and p	rojects e.g.	paths to li	brary file:			Environment Option P Building Debug Environment Languages Source Control Text Editor Windows Keyboard Color Schemes	Option Build Automatically Build Before Debug Confirm Debugger Stop Display TA Displa	Yes No Yes Yes Yes SDK_R(No Yes SIStudi	oDir)/gc	/S(GCCTarget).	'bin		ESH_ROOT=C	
	l across all so 0.2K	utions and pr		paths to li	brary file:			Environment Option Debug Environment Languages Source Control Text Ealtor Windows Keyboard	Option Automatically Build Before Debug Automatically Build Before Debug Contim Debugger Stop Display Trogers Bar Eno Build Command lang Output Find Error After Building Global Macrosi Keep Going On Error Save Project File Erlore Building Show Build Information Toolchan Boot Directogy Build Acrelenation	Yes No Yes Yes Yes SDK_R(No Yes SIStudi	oDir)/gc	/S(GCCTarget).	'bin		ESH_ROOT=C;**	
iuiid macros that are shared			rojects e.g.	paths to li	brary file:			Environment Option P Building Debug Environment Languages Source Control Text Editor Windows Keyboard Color Schemes	Option Build Automatically Build Before Debug Confirm Debugger Stop Display TA Displa	Yes No Yes Yes Yes SDK_R(No Yes SIStudi	oDir)/gc	/S(GCCTarget).	'bin		ESH_ROOT=C	

 Open the main.c file and change the DEVICE_NAME define on line 93 to something unique.
 This is to evold escidently provisioning company class, and area the bonds on start.

This is to avoid accidently provisioning someone else's node once the hands on starts.

- 5) Compile the example by pressing F7 or *Build -> Build light_switch_proxy_server_nrf52840_xxAA_s140_6.0.0*
- 6) Make sure your nRF52840 DK is connected to the PC over USB and turned on. Flash the example by selecting *Target -> Download light_switch_proxy_server_nrf52840_xxAA_s140_6.0.0*, or you can start a new debug session by pressing F5 (in this case the board will be flashed automatically).
 If asked to update the board firmware click yes, and wait for the update procedure to complete.
- 7) Go through the rest of the guide. The proxy server example contains both the generic on/off client & server models. They are slightly simplified compared to the generic models, but still contain most of the same functionality. This enables flexibility to use a device as either a client (e.g. switch) and/or server (e.g. light).



2 Hands-on

Unless otherwise noted, all the following steps should be performed by one member of the group. Try to make sure that everyone gets to see what is happening, and that everyone in the group gets some experience using the app.

2.1 Mesh provisioning basic example

 Download and run the nRF Mesh app for iOS or Android on a single phone in the group. Navigate to the scanner section, and verify that all the devkits with your group ID are present in the list:





 Select one of the devkits in your group. Click "IDENTIFY", and next click "PROVISION". Wait for the provisioning process to complete and click "OK" when the Configuration Complete dialog appears.



3) Once a devkit is provisioned and configured we can finalize the configuration by binding app keys and assigning addresses to the models.

From the network overview click "Configure" on the devkit you just provisioned to open the Node Configuration screen.

You should see two elements that can be expanded by clicking the arrow symbol, revealing three models in the first element and one element in the second as shown in the picture below.



Click on the item called "Generic On Off Server"

Telenor 🖾 🖏 🖬 🕸	ঙ	≵ I□I 44% 💷 17:33	Telenor	co tal 🕫 🗹	@\$1014	8 🕏 17:33		Telenor 🚥 🖏 🖬 🕸 🕅		@\$\$10144	% 💷 17:34
nRF Mesh		DISCONNECT	÷	Node Conf MeshGroup02			÷	Node Con MeshGroup(on	
	MeshGroup Address: 0001 Elements: 2 Models: 4 ConFiGURE	c Semicondu	:: • •	Elements Element : 0001 3 Models Element : 0002 1 Models Added App H App key 1 63964771734FBD760		✓ ✓	:: ~ (Elements Element : 0001 3 Models SIG Model ID: 0x00 Health Server SIG Model ID: 0x10 Generic On Off S SIG Model ID: 0x10 Element : 0002 1 Models Generic On Off C SIG Model ID: 0x10	erver 00 02 eerver 00 2 2		^
<mark>ਨੋਰ</mark> Network	≱) Scanner	Č Settings			s by Nordic		67 07	Added App App key 1 63964771734FBD7	6E3B40519D		ADD
\bigtriangledown	0			\bigtriangledown	0 🛛			\triangleleft	0		

4) Before the model can publish or subscribe to an address we need to bind it to an app key. Click "BIND KEY" under "App Key", select "App key 1", and confirm that the key was bound successfully to the model as shown below:

Telenor 🚾 ^않	Telenor 🛅 🐑 🕼 🍁 🚺 🥸 🕉 🕼 (44% 💷) 17:34	Telenor 🔤 🛗 非 🖬 🛛 🕹 🕏 10:44% 💷 17:34
← Generic On Off Server	← Select App Key	← Generic On Off Server
от Арр Кеу		•• Арр Кеу
Bound App Keys	App key 1 63964771734FBD76E3B40519D1D94A48	Bound App Keys
BIND KEY		BIND KEY
T Publish		Publish
Publish Address		Publish Address
SET PUBLISH ADDRESS		SET PUBLISH ADDRESS
Subscribe		
Not subscribed to any group addresses.		Not subscribed to any group addresses.
SUBSCRIBE		SUBSCRIBE
Wireless by Nordic		Wireless by Nordic
\triangleleft O \square	\triangleleft O \square	



5) Now we can subscribe to a group address. To do this click "SUBSCRIBE" and enter the address 0xC001 (or any other address starting with 0xC).

Telenor 🖫 🛔 🕸 🕹 🕸 🕹 🕄 10144% 💷 17:34	Telenor 🚾 🖏 🖬 🗱	\$\$\$\$1⊒144% ■⊃17:34	Telenor 🕮 🖏 🖬 😸 😸 🖇 🔲 (44% 💷) 17:34
← Generic On Off Server	← Generic On	Off Server	← Generic On Off Server
от Арр Кеу	от Арр Кеу		от Арр Кеу
Bound App Keys	물 Group Add	dress	Bound App Keys
BIND KEY	Group address thi subscribe.	s model should	BIND KEY
Publish	0×C001		🚡 Publish
Publish Address	None	CANCEL OK	Publish Address
SET PUBLISH ADDRESS		SET PUBLISH ADDRESS	SET PUBLISH ADDRESS
SET PUBLISH ADDRESS	- Outrouthe		SET PUBLISH ADDRESS
▲ Subscribe	1 2 3 4 5	SET PUBLISH ADDRESS	
Subscribe Not subscribed to any group addresses.	% ^ ~ I [
▲ Subscribe	% ^ ~ I [Q W E R T @ # & * -	6 7 8 9 0	Subscribe Group Address
Subscribe Not subscribed to any group addresses.	% ^ ~ I I Q W E R T @ # & * - A S D F G	6 7 8 9 0 1 > { > Y U I O P Å + = () O Å	Subscribe Group Address OxC001
Subscribe Not subscribed to any group addresses.	$ \begin{array}{c} 3\\ Q \\ W \\ e \\ A \\ S \\ D \\ e \\ \hline z \\ c \\$	6 7 8 9 0 1 × × > > 1 × × > > 1 × × > > 1 × × > > + × > > + + × + + × + + × + + × * : : /	Subscribe Group Address OxC001

- 6) Now we should provision a second devkit into the mesh. To do this simply repeat the steps described in 1) and 2).
- 7) Go back to the network overview and click "Configure" on the second devkit in the list. Expand the second element and click on the "Generic On Off Client" item to open the



Generic On Off Client configuration screen.

	త ≵≀⊡190% 💷) 14:34	🚥 "iil 🗟 🌣 🔀 🖸	& ≵⊫0190% ■014:34	© () () () () () () () () () () () () ()	ˈ& ≵ Ⅰ□[190% 💷) 14:34
nRF Mesh	DISCONNECT	← Node Confi nRF5x Mesh L		← Generi Model II	i <mark>c On Off Client</mark> D: 1001
	RF5x Mesh Light ddress: 0001 company: Nordic Semicondu lements: 2 Addels: 4	Elements Element : 0003 Models	~		ound to this model.
	CONFIGURE DETAILS	Element : 0004	<u>^</u>		
	RF5x Mesh Light ddress: 0003 company: Nordic Semicondu	Generic On Off Clie SIG Model ID: 0x1001		Publish Publish Ac None None	Idress
	lements: 2 Iodels: 4	Added App K	leys		SET PUBLICATION
	CONFIGURE DETAILS	App key 1 06F0E1D883950BF41	5D0CDBC70C2AC79	生 Subscrit	be
			ADD	Not subscribed to	any group addresses.
					SUBSCRIBE
		 Reset Node 			
Retwork	x) x Scanner Settings	Resetting this node w provisioned state bac state.		v	Vireless by Nordic
\triangleleft	0	\triangleleft	0	\triangleleft	0

- 8) Once again, we need to bind the model to an app key, similar to step 4).
- Click "SET PUBLICATION" and select "Publish Address". Enter the same address as you 9) did in step 5, and click OK. V

Verify that the	publish	address	is updated	and	click	"APPLY".	
-----------------	---------	---------	------------	-----	-------	----------	--





10) Check that the two devkits are configured correctly by clicking the button on the second devkit and verify that the LED toggles on the first devkit.

Congratulations! You have just provisioned and configured your first mesh network using the provisioning app 🐵



2.2 Extend the mesh

Note: For the hardware academy workshop you should skip this section, as time is limited, and move to hands on exercise 3.

A mesh consisting of two nodes is not very impressive, and the next task is to add any remaining devkits to the mesh network and assign them to different groups.

Unless the group has more than 4 devkits available it might be necessary for some of the devkits to act both as buttons and LED's, by configuring both the client and server. An example of how this can be done is shown in the picture below:



1) Start by provisioning any remaining devkits, as described in the previous section.



2) Partition all the available devkits into two groups and assign each group to its own group address (0xC001 and 0xC002 for example).

This includes both the original 2 devkits, as well as the new devkits provisioned in step 1)

Make sure that the Generic On Off Server subscribes to the group address that the devkit belongs to. For a reminder on how to do this, refer to steps 3-5 in section 2.2.

3) Let two devkits publish to either of the group addresses, allowing you to control the two light groups individually.

For a refresh on setting up publication, refer to steps 7-9 in section 2.2.

Hint: A single devkit can be both a subscriber and publisher at the same time. Just remember to subscribe on the simple on off server (LED), and publish on the simple on off client (button).

4) Bonus task: Try to add all devkits to a third group (0xC003 for example) and have one of the devkits publish to this address. This will allow all the LED's to be controlled at the same time, while still allowing the two previously defined groups to be used.

Hint: A server can subscribe to several group addresses.



2.3 **PWM** integration

The purpose of this hands on is to modify parts of the code in the mesh example to change the behaviour of the application.

The task involves controlling the LED's through PWM (pulse width modulation) to allow fading or pulsing of LED's instead of just turning them on or off.

To simplify the task a small PWM driver has been provided, and the PWM interface is described through the simple_pwm.h header file.

1) Initialize the PWM driver in main(), as indicated by the TODO comment After initialization, set one of the LED's to pulse or blink repeatedly.



 Modify the on_off_server_set_cb() callback in main.c to fade LED0 in and out when the LED status is changed, rather than have it set or cleared immediately

static bool on_off_server_set_cb(const generic_on_off_server_t * p_server, bool value)



- 3) Look for any remaining references to hal_led_ functions in main.c, and replace them with calls to the PWM library.
- 4) **(optional)** Currently nothing happens in the code when the "Identify" button is pressed in the nRF Mesh app during provisioning.

In order to change this it is necessary to make changes to some of the files in the mesh SDK:

- In mesh_provisionee.c, the switch case in the prov_evt_handler() function needs to be extended to handle the NRF_MESH_PROV_EVT_INVITE_RECEIVED case, and forward an event to the application informing it of the event.

- The mesh_provisionee_start_params_t struct in mesh_provisionee.h needs to be



extended with an additional callback, which should occur when the INVITE_RECEIVED event occurs.

- Add code in prov_evt_handler() to call this callback when the NRF_MESH_PROV_EVT_INVITE_RECEIVED event occurs.

- In main.c, add a function to handle the invite received callback, and make sure to include it in the provisionee configuration in the start() function

- In the callback function you just created, add code to blink or pulse a LED when the invite callback occurs



3 Troubleshooting

3.1 Lost connection to the mesh

Sometimes the link between the phone and one of the mesh nodes could be broken. In this case it is necessary to reconnect manually from the app on the phone to get access to the mesh:

- 1) Click the "Connect" button in the app.
- 2) Click on any of the devices showing up in the list (each node in the mesh can act as a gateway to the phone)

Now the phone should be connected again, and the nodes can be configured.

3.2 Mesh stability issues

If any problems occur during the provisioning or configuration it might be necessary to reset the devkits and the app and start from the beginning.

To reset the devkit, click Button 3 while the kit is running. Alternatively, do a full erase and reflash the kit with the mesh firmware.

To reset the app, navigate to the "settings" section of the app and press the "Reset Network" button.