Mesh Network

The following directions are for a windows instillation of the Nordic Semiconductor Mesh SDK.

Download nRF5 SDK and nRF5 SDK for Mesh

1- Go to Nordic Semiconductor Website. Click here

2- Click on **Software and tools**.

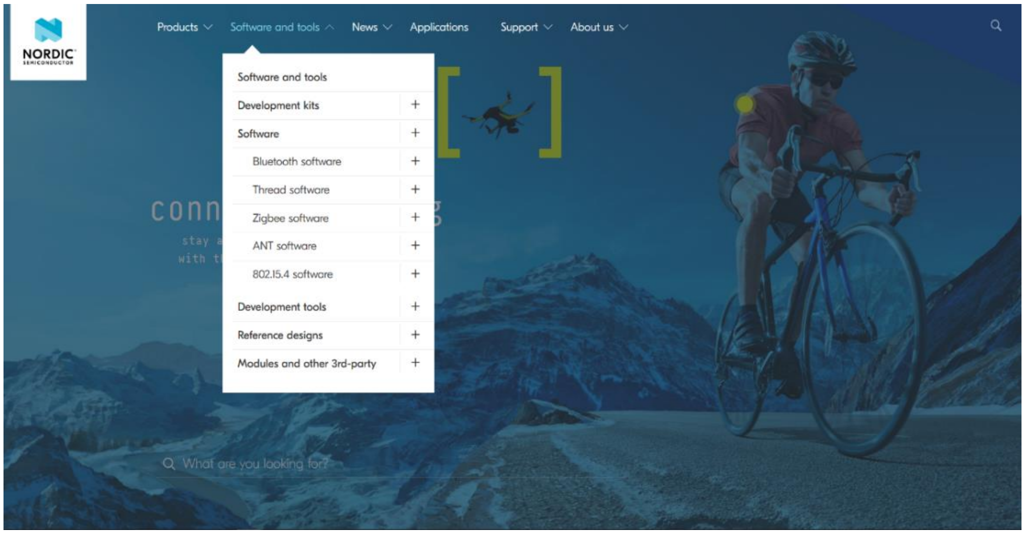


Fig 1.1: Software and Tools

3- Click on **Bluetooth software** and choose **nRF SDK**.

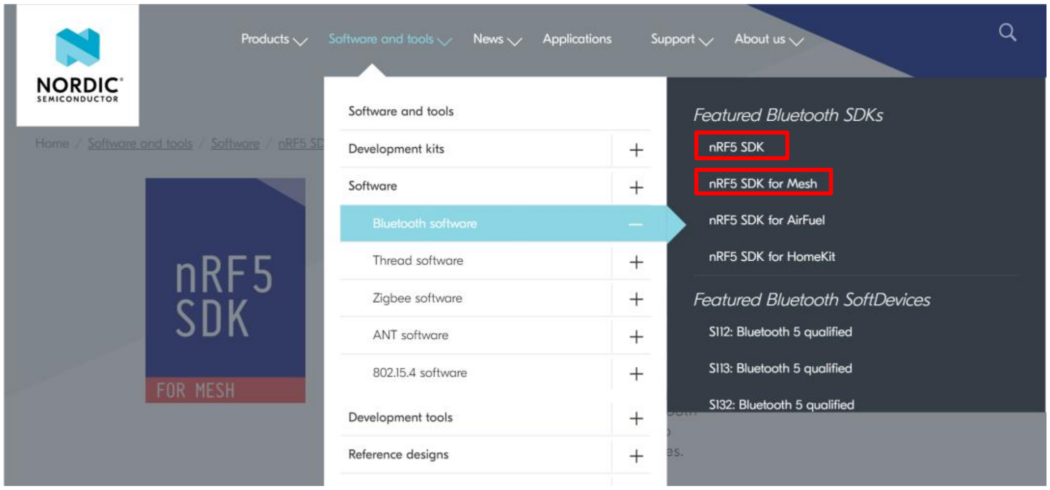
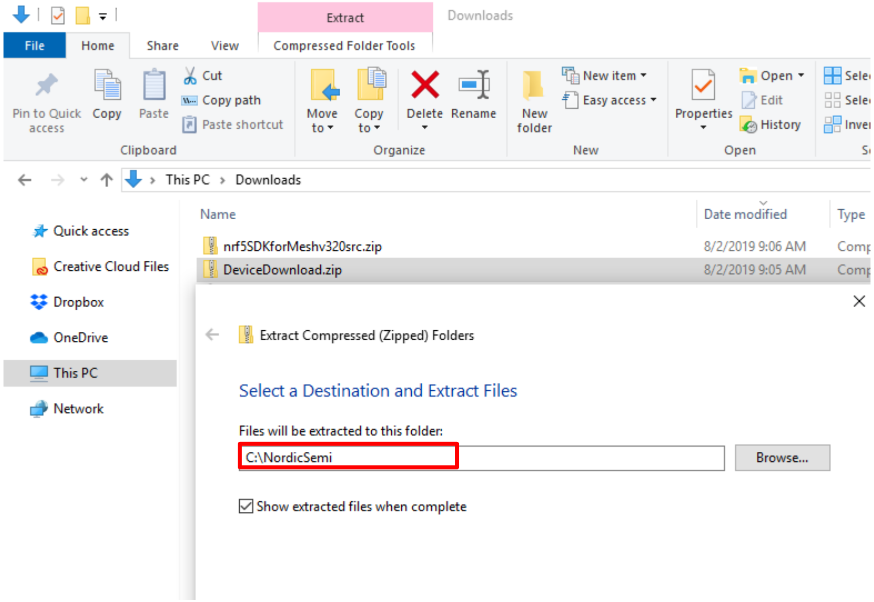


Fig 1.2: Bluetooth software

4- Once the two files have been downloaded create a directory for instillation of the SDKs such as **c:\Nordic Semi**.

5- Extract the files of **DeviceDownload.zip** into the **c:\NordicSemi** directory.



nrf5SDKforMeshv320src.zip 8/2/2019 9:06 AM Comp
DeviceDownload.zip 8/2/2019 9:05 AM Comp

Extract Compressed (Zipped) Folders

Select a Destination and Extract Files

Files will be extracted to this folder:

C:\NordicSemi

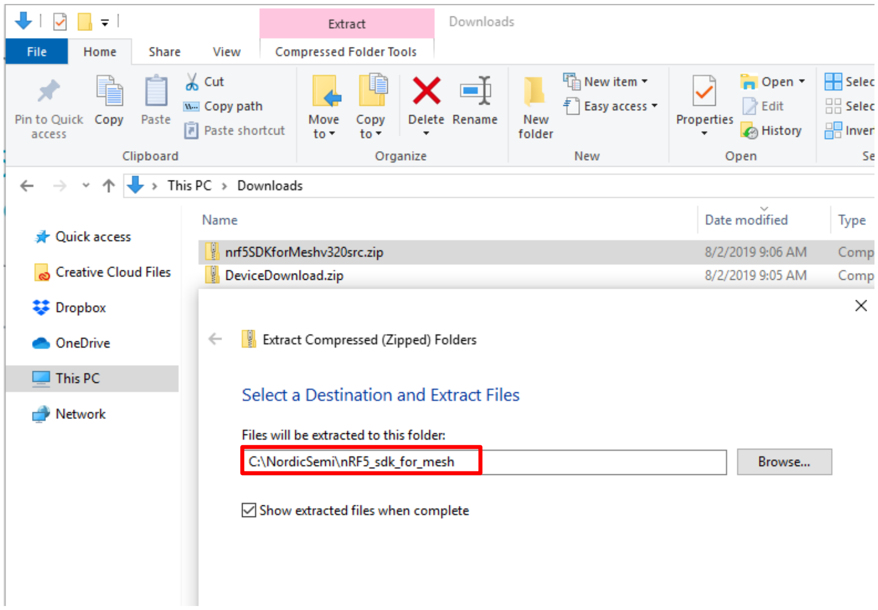
Browse...

☑ Show extracted files when complete

Fig 1.3: DeviceDownload.zip Extracting

6- In the **c:\NordicSemi** directory create a directory called **nRF5_sdk_for_mesh**.

7- Extract the contents of **nRF5SDKforMeshv310src.zip** into the **nRF5_sdk_for_mesh** directory.



nrf5SDKforMeshv320src.zip 8/2/2019 9:06 AM Comp
DeviceDownload.zip 8/2/2019 9:05 AM Comp

Extract Compressed (Zipped) Folders

Select a Destination and Extract Files

Files will be extracted to this folder:

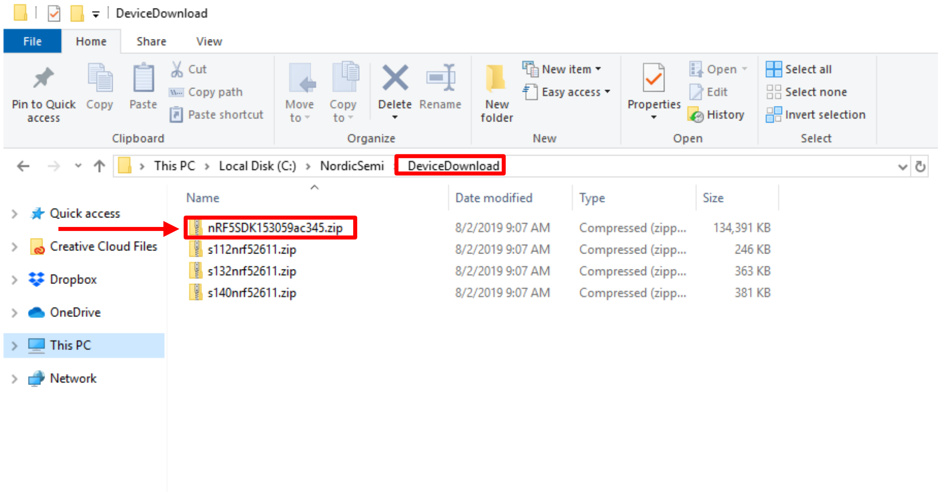
C:\NordicSemi\nRF5_sdk_for_mesh

Browse...

☑ Show extracted files when complete

Fig 1.4: nRF5SDKforMeshv310src.zip Extracting

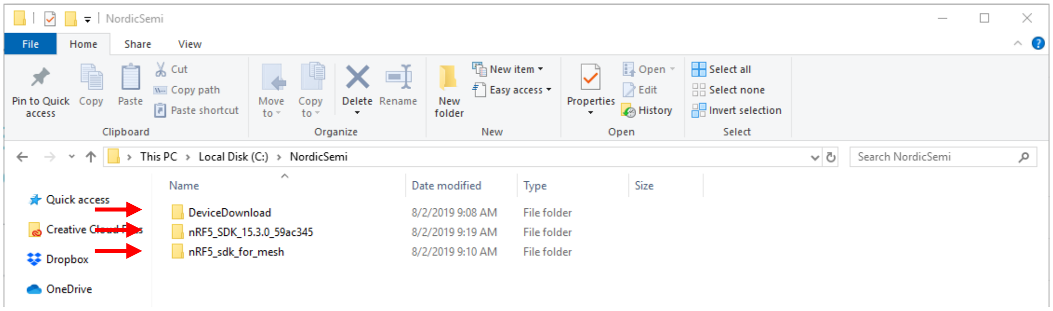
8- Go to **c:\NordicSemi** >> **DeviceDownload,** extract the contents of **nrf5SDK153059ac345.zip** into the **c:\NordicSemi** directory.



Name	Date modified	Type	Size
nRF5SDK153059ac345.zip	8/2/2019 9:07 AM	Compressed (zipp...	134,391 KB
s112nrf52611.zip	8/2/2019 9:07 AM	Compressed (zipp...	246 KB
s132nrf52611.zip	8/2/2019 9:07 AM	Compressed (zipp...	363 KB
s140nrf52611.zip	8/2/2019 9:07 AM	Compressed (zipp...	381 KB

Fig 1.5: nrf5SDK153059ac345.zip Extracting

Your directory structure should now look similar to this



Name	Date modified	Type	Size
DeviceDownload	8/2/2019 9:08 AM	File folder	
nRF5_SDK_15.3.0_59ac345	8/2/2019 9:19 AM	File folder	
nRF5_sdk_for_mesh	8/2/2019 9:10 AM	File folder	

Fig 1.6: directory structure

Install Segger Studio

Since we are using Nordic Semiconductor Chips, we can get a free version of Segger Embedded Studio for this project.

1- To download Segger Embedded Studio go back to **Nordic Semiconductor** website, select **Software and tools** >>**Development tools**, or click here.

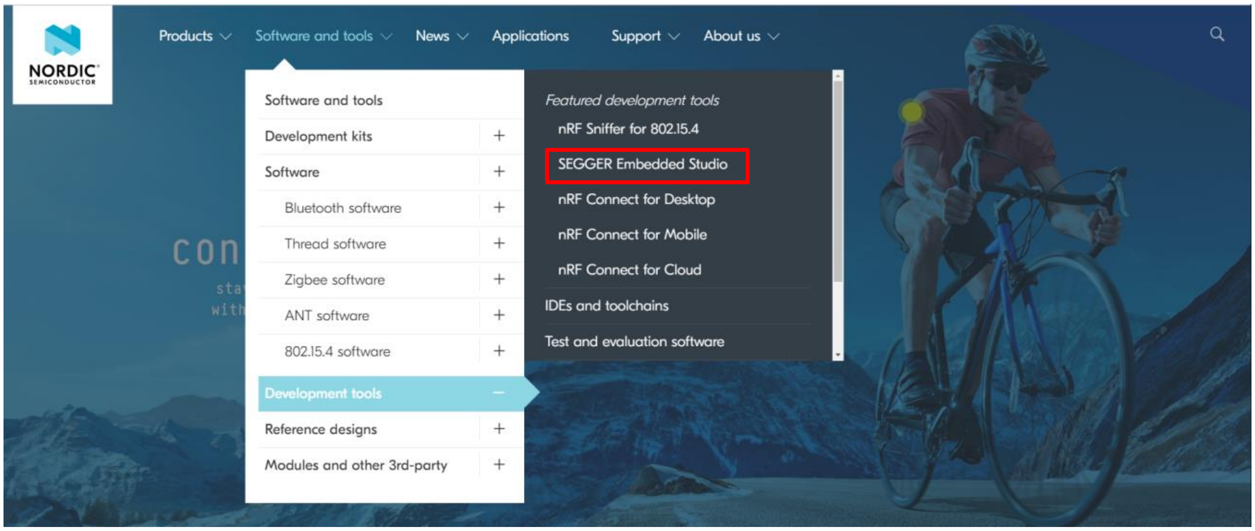
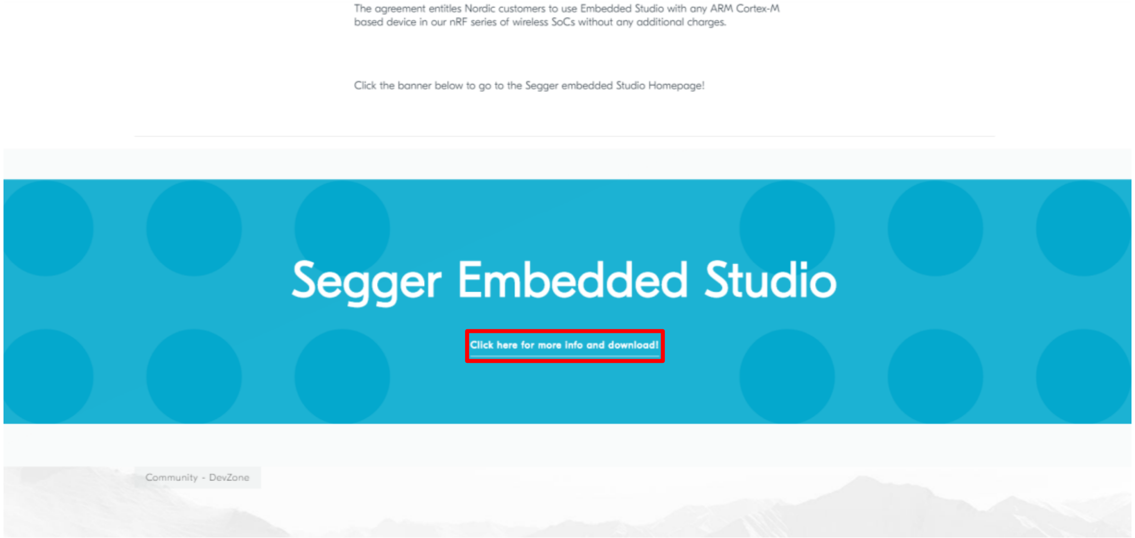


Fig 2.1: Development tools

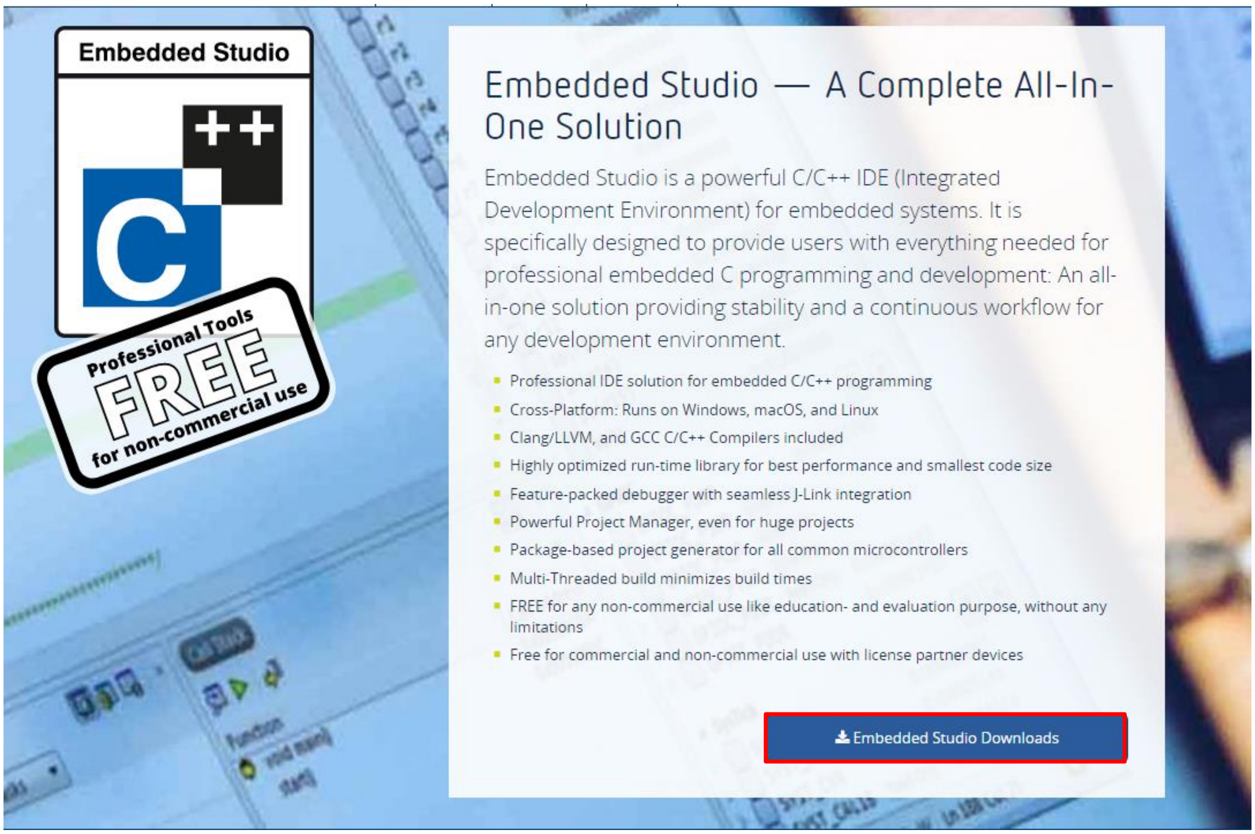


The agreement entitles Nordic customers to use Embedded Studio with any ARM Cortex-M based device in our nRF series of wireless SoCs without any additional charges.

Click the banner below to go to the Segger embedded Studio Homepage!

Fig 2.2: Segger Embedded Studio Download

2- On the Segger web site click the Segger Embedded Studio Downloads box and select the Embedded Studio for ARM version for your operating system



Embedded Studio — A Complete All-In-One Solution

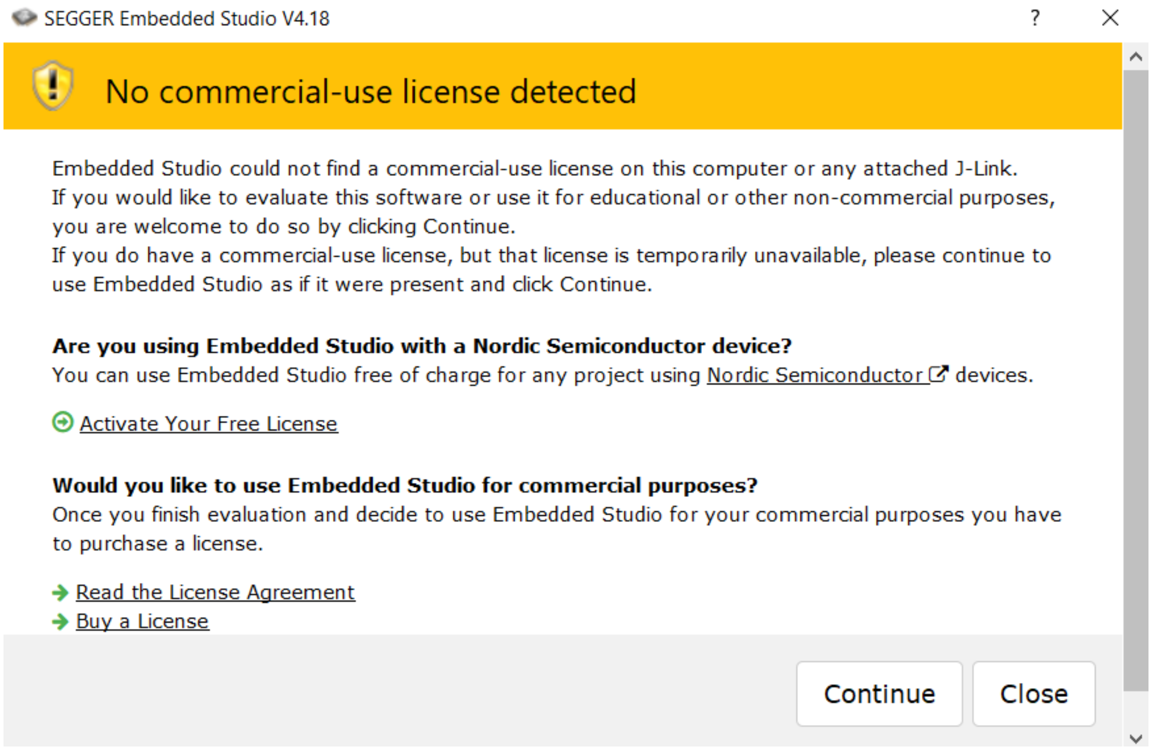
Embedded Studio is a powerful C/C++ IDE (Integrated Development Environment) for embedded systems. It is specifically designed to provide users with everything needed for professional embedded C programming and development: An all-in-one solution providing stability and a continuous workflow for any development environment.

- Professional IDE solution for embedded C/C++ programming
- Cross-Platform: Runs on Windows, macOS, and Linux
- Clang/LLVM, and GCC C/C++ Compilers included
- Highly optimized run-time library for best performance and smallest code size
- Feature-packed debugger with seamless J-Link integration
- Powerful Project Manager, even for huge projects
- Package-based project generator for all common microcontrollers
- Multi-Threaded build minimizes build times
- FREE for any non-commercial use like education- and evaluation purpose, without any limitations
- Free for commercial and non-commercial use with license partner devices

Fig 2.3: Segger Embedded Studio Downloads box

3- Run the Setup_EmbeddedStudio file Just accept all the default choices.

On the first run of Segger Studio you will see **Activate your License Dialog Box**. If you are using this for yourself, you may select the **non-commercial license** or **activate a free Nordic License**. If you are developing for a **Commercial application**, you will need to register for a free **Nordic License**



SEGGER Embedded Studio V4.18

No commercial-use license detected

Embedded Studio could not find a commercial-use license on this computer or any attached J-Link.
If you would like to evaluate this software or use it for educational or other non-commercial purposes, you are welcome to do so by clicking Continue.
If you do have a commercial-use license, but that license is temporarily unavailable, please continue to use Embedded Studio as if it were present and click Continue.

Are you using Embedded Studio with a Nordic Semiconductor device?
You can use Embedded Studio free of charge for any project using Nordic Semiconductor devices.

Activate Your Free License

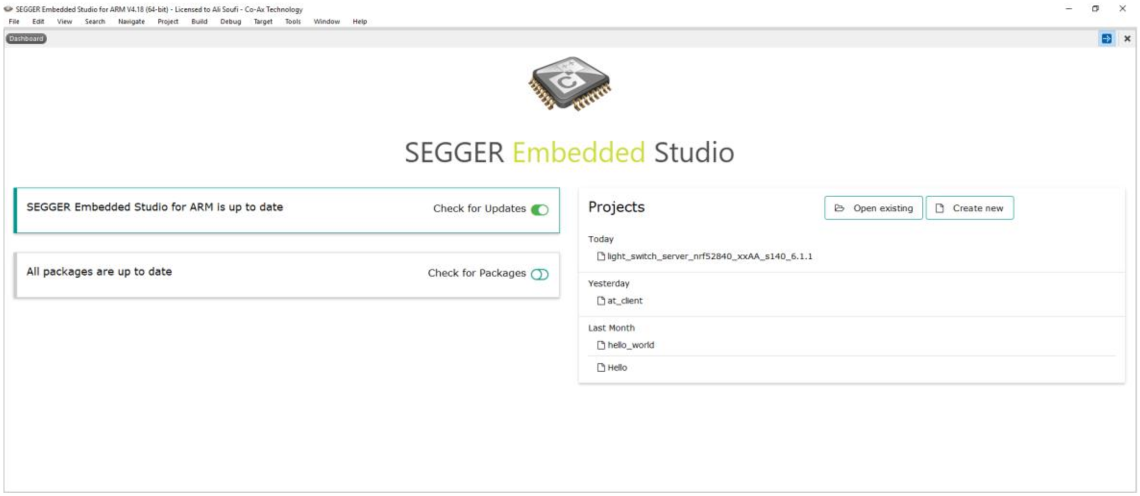
Would you like to use Embedded Studio for commercial purposes?
Once you finish evaluation and decide to use Embedded Studio for your commercial purposes you have to purchase a license.

→ Read the License Agreement
→ Buy a License

Fig 2.3: License detected box

Setting up the first Mesh Demo

1- Go to Segger Embedded Studio.



SEGGER Embedded Studio for ARM V4.18 (64-bit) - Licensed to Ali Soufi - Co-Ax Technology

SEGGER Embedded Studio for ARM is up to date

All packages are up to date

Fig 3.1: Segger Embedded Studio Dashboard

2- Select **File**>> **Open Solution,** C:\NordicSemi\nRF5_sdk_for_mesh\examples\light_switch\server\ **light_switch_server_nrf52840_xxAA_s140_6_1_1.emProject.**

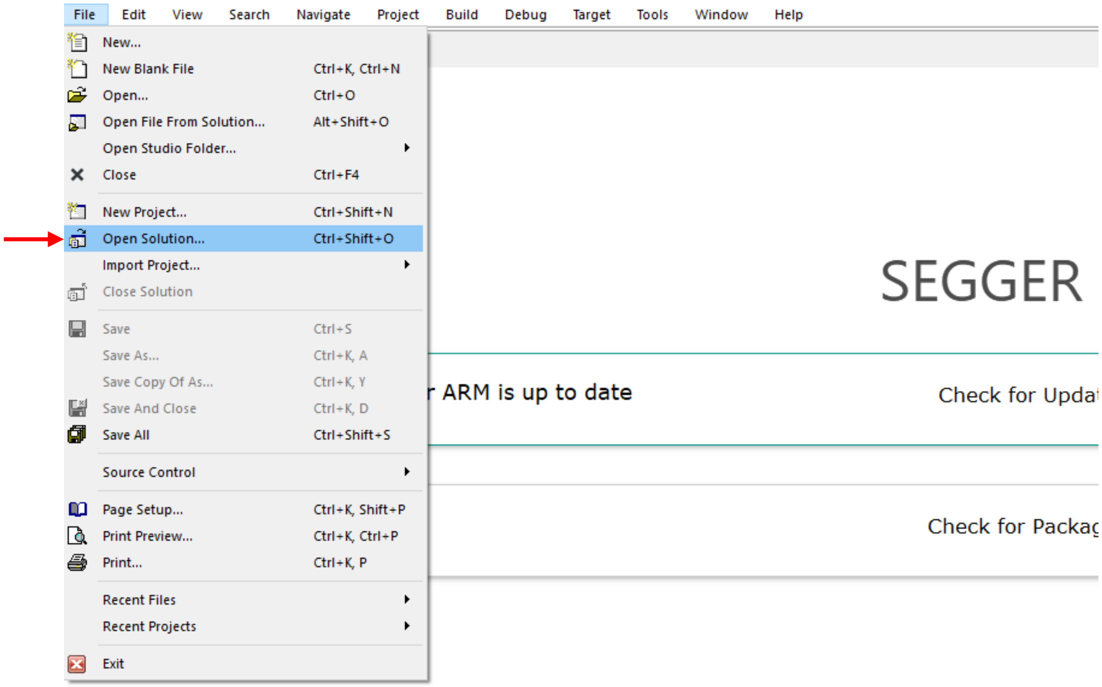
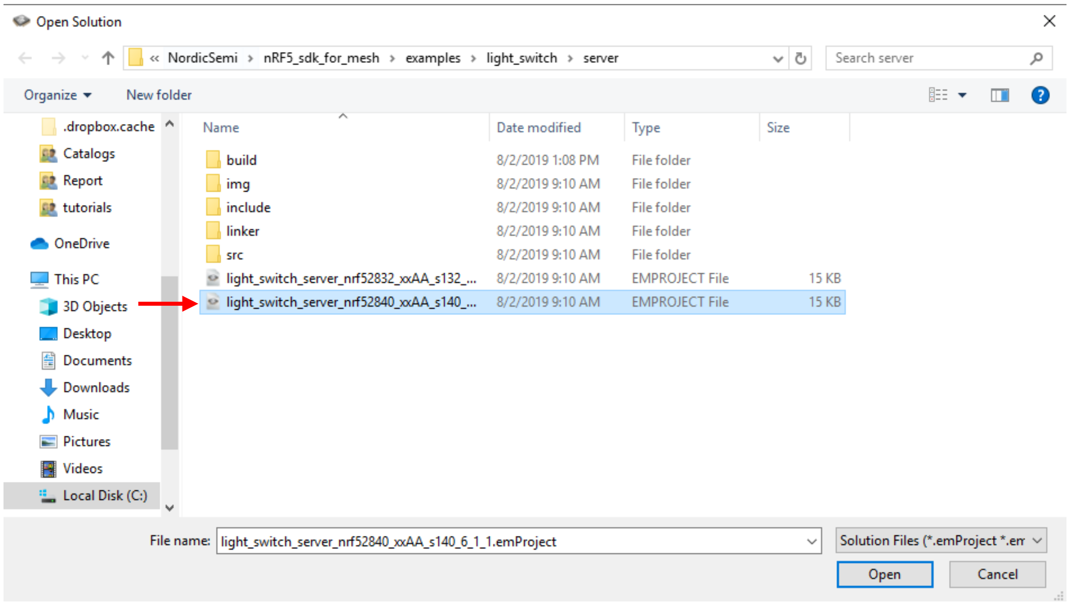


Fig 3.2: Open Solution



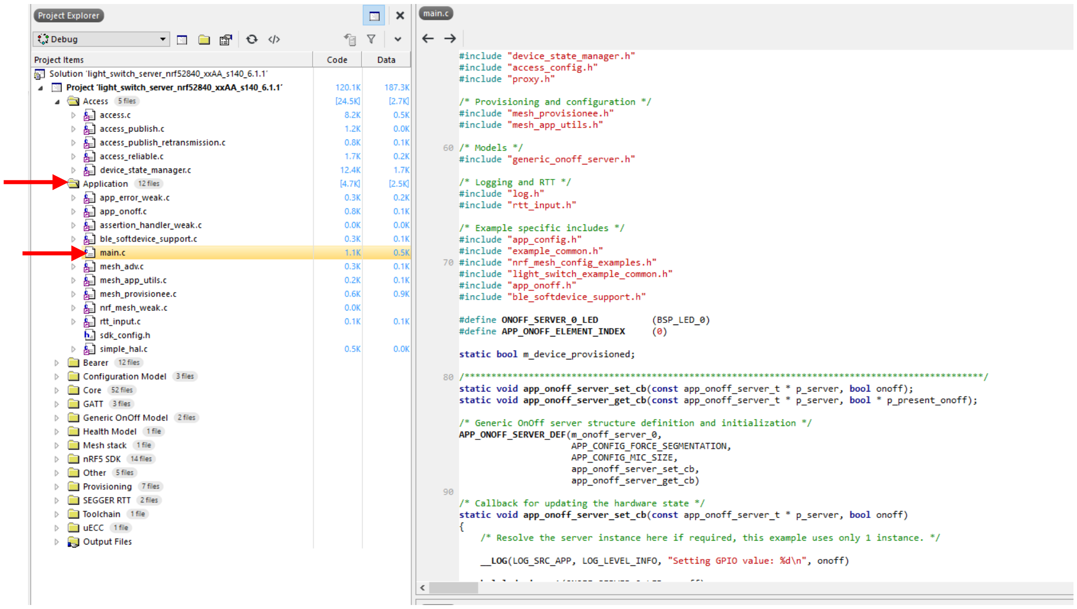
Name	Date modified	Type	Size
build	8/2/2019 1:08 PM	File folder	
img	8/2/2019 9:10 AM	File folder	
include	8/2/2019 9:10 AM	File folder	
linker	8/2/2019 9:10 AM	File folder	
src	8/2/2019 9:10 AM	File folder	
light_switch_server_nrf52832_xxAA_s132_...	8/2/2019 9:10 AM	EMPROJECT File	15 KB
light_switch_server_nrf52840_xxAA_s140_...	8/2/2019 9:10 AM	EMPROJECT File	15 KB

File name: light_switch_server_nrf52840_xxAA_s140_6_1_1.emProject

Solution Files (*.emProject *.em

Fig 3.3: Example Directory

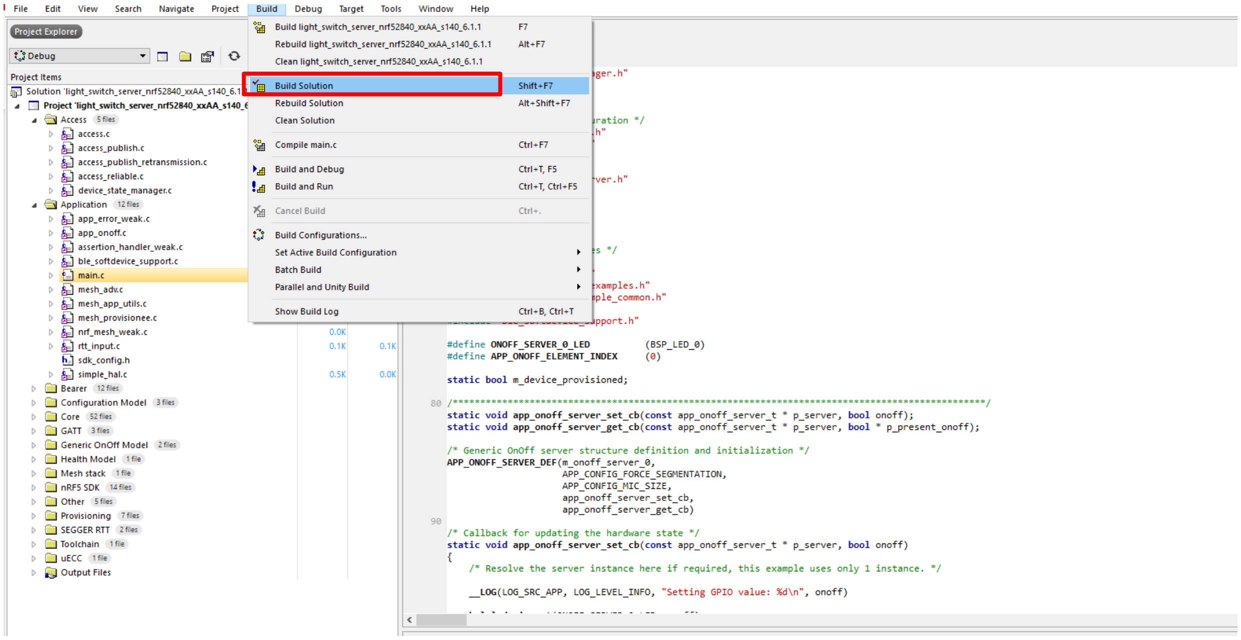
3- Go to **main.c** to open the program.



```c
#include "device_state_manager.h"
#include "access_config.h"
#include "proxy.h"

/* Provisioning and configuration */
#include "mesh_provisionee.h"
#include "mesh_app_utils.h"

/* Models */
#include "generic_onoff_server.h"

/* Logging and RTT */
#include "log.h"
#include "rtt_input.h"

/* Example specific includes */
#include "app_config.h"
#include "example_common.h"
#include "nrf_mesh_config_examples.h"
#include "light_switch_example_common.h"
#include "app_onoff.h"
#include "ble_softdevice_support.h"

#define ONOFF_SERVER_0_LED          (BSP_LED_0)
#define APP_ONOFF_ELEMENT_INDEX     (0)

static bool m_device_provisioned;

/*************************************************************************************************/
static void app_onoff_server_set_cb(const app_onoff_server_t * p_server, bool onoff);
static void app_onoff_server_get_cb(const app_onoff_server_t * p_server, bool * p_present_onoff);

/* Generic OnOff server structure definition and initialization */
APP_ONOFF_SERVER_DEF(m_onoff_server_0,
                     APP_CONFIG_FORCE_SEGMENTATION,
                     APP_CONFIG_MIC_SIZE,
                     app_onoff_server_set_cb,
                     app_onoff_server_get_cb)

/* Callback for updating the hardware state */
static void app_onoff_server_set_cb(const app_onoff_server_t * p_server, bool onoff)
{
    /* Resolve the server instance here if required, this example uses only 1 instance. */

    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "Setting GPIO value: %d\n", onoff)
```

Fig 3.4: main.c

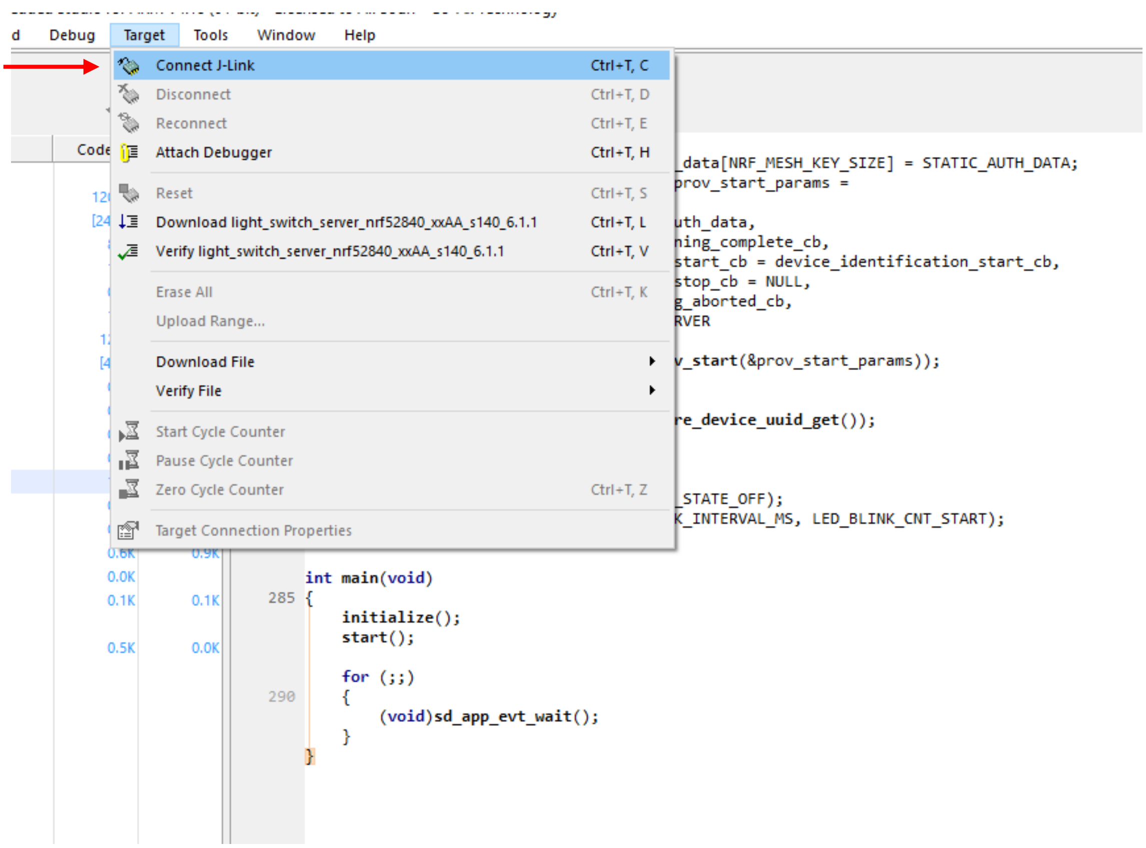
4- Select **Build**>>**Build solution**.



```c
#define ONOFF_SERVER_0_LED          (BSP_LED_0)
#define APP_ONOFF_ELEMENT_INDEX     (0)

static bool m_device_provisioned;

/*************************************************************************************************/
static void app_onoff_server_set_cb(const app_onoff_server_t * p_server, bool onoff);
static void app_onoff_server_get_cb(const app_onoff_server_t * p_server, bool * p_present_onoff);

/* Generic OnOff server structure definition and initialization */
APP_ONOFF_SERVER_DEF(m_onoff_server_0,
                     APP_CONFIG_FORCE_SEGMENTATION,
                     APP_CONFIG_MIC_SIZE,
                     app_onoff_server_set_cb,
                     app_onoff_server_get_cb)

/* Callback for updating the hardware state */
static void app_onoff_server_set_cb(const app_onoff_server_t * p_server, bool onoff)
{
    /* Resolve the server instance here if required, this example uses only 1 instance. */

    __LOG(LOG_SRC_APP, LOG_LEVEL_INFO, "Setting GPIO value: %d\n", onoff)
```

Fig 3.5: Build solution

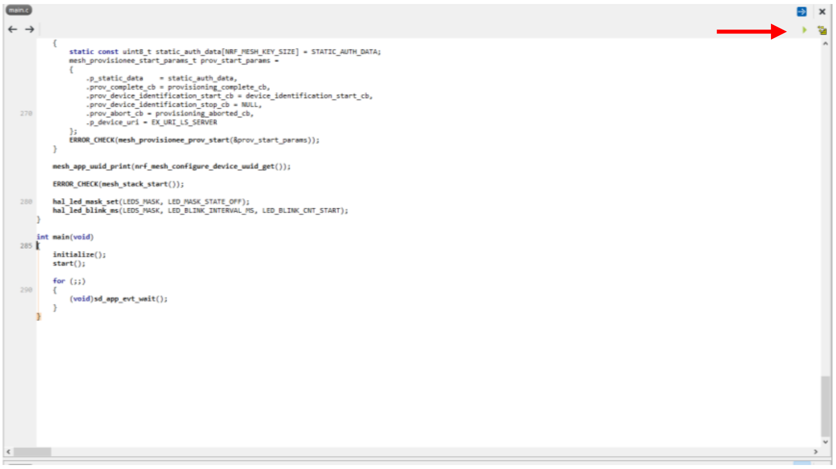
5- Connect your board to the computer you are using and select **Target**>>**Connect J-Link.**



```
_data[NRF_MESH_KEY_SIZE] = STATIC_AUTH_DATA;
prov_start_params =

uth_data,
ning_complete_cb,
start_cb = device_identification_start_cb,
stop_cb = NULL,
g_aborted_cb,
RVER

v_start(&prov_start_params));


re_device_uuid_get());



_STATE_OFF);
K_INTERVAL_MS, LED_BLINK_CNT_START);


int main(void)
{
    initialize();
    start();

    for (;;)
    {
        (void)sd_app_evt_wait();
    }
}
```

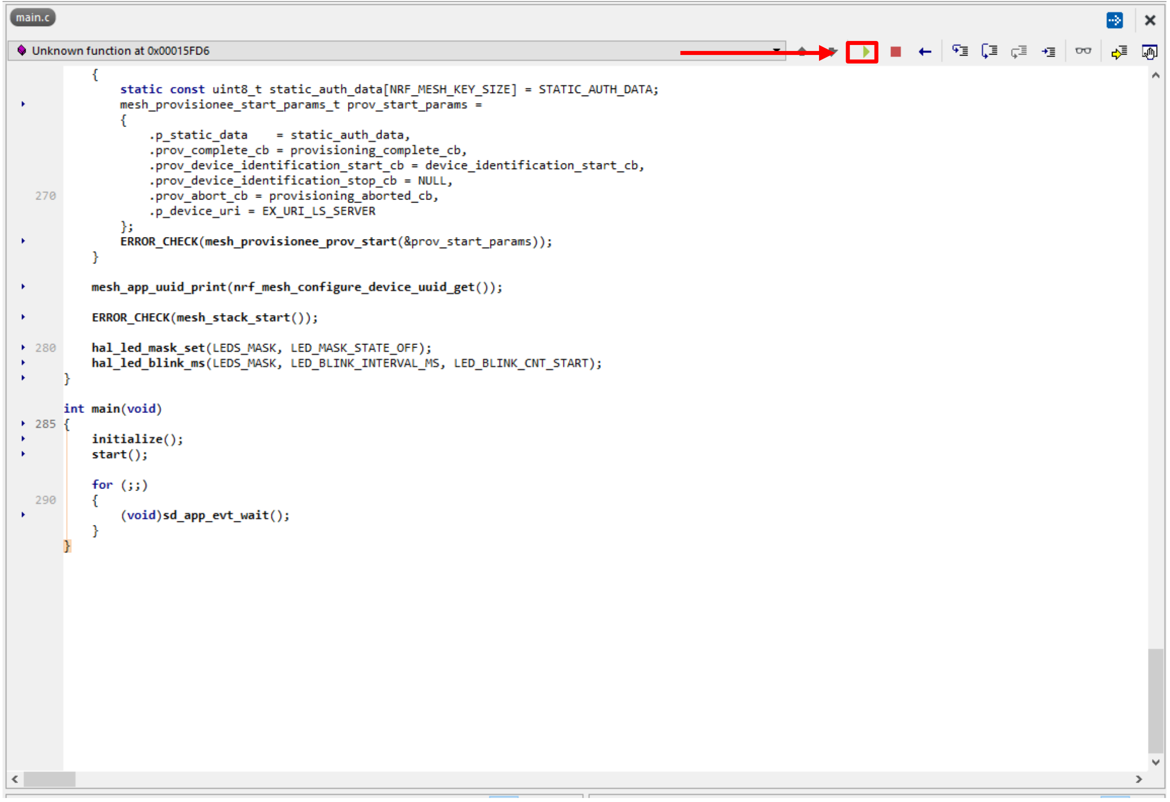
Fig 3.6: Connect J-Link

6- Click on the green arrow to program .



```
    {
        static const uint8_t static_auth_data[NRF_MESH_KEY_SIZE] = STATIC_AUTH_DATA;
        mesh_provisionee_start_params_t prov_start_params =
        {
            .p_static_data    = static_auth_data,
            .prov_complete_cb = provisioning_complete_cb,
            .prov_device_identification_start_cb = device_identification_start_cb,
            .prov_device_identification_stop_cb = NULL,
            .prov_abort_cb = provisioning_aborted_cb,
            .p_device_uri = EX_URI_LS_SERVER
        };
        ERROR_CHECK(mesh_provisionee_prov_start(&prov_start_params));
    }

    mesh_app_uuid_print(nrf_mesh_configure_device_uuid_get());

    ERROR_CHECK(mesh_stack_start());

    hal_led_mask_set(LEDS_MASK, LED_MASK_STATE_OFF);
    hal_led_blink_ms(LEDS_MASK, LED_BLINK_INTERVAL_MS, LED_BLINK_CNT_START);
}

int main(void)
{
    initialize();
    start();

    for (;;)
    {
        (void)sd_app_evt_wait();
    }
}
```

7- After the board programs press it again to run the program .

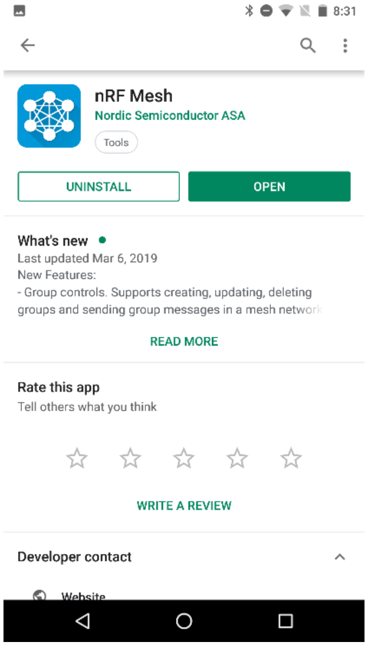


```
    {
        static const uint8_t static_auth_data[NRF_MESH_KEY_SIZE] = STATIC_AUTH_DATA;
        mesh_provisionee_start_params_t prov_start_params =
        {
            .p_static_data    = static_auth_data,
            .prov_complete_cb = provisioning_complete_cb,
            .prov_device_identification_start_cb = device_identification_start_cb,
            .prov_device_identification_stop_cb = NULL,
            .prov_abort_cb = provisioning_aborted_cb,
            .p_device_uri = EX_URI_LS_SERVER
        };
        ERROR_CHECK(mesh_provisionee_prov_start(&prov_start_params));
    }

    mesh_app_uuid_print(nrf_mesh_configure_device_uuid_get());

    ERROR_CHECK(mesh_stack_start());

    hal_led_mask_set(LEDS_MASK, LED_MASK_STATE_OFF);
    hal_led_blink_ms(LEDS_MASK, LED_BLINK_INTERVAL_MS, LED_BLINK_CNT_START);
}

int main(void)
{
    initialize();
    start();

    for (;;)
    {
        (void)sd_app_evt_wait();
    }
}
```

Fig 3.7: Program

For an Android Phone

1- Go to your Play Store, Search for Nordic Semiconductor and Install nRF Mesh.



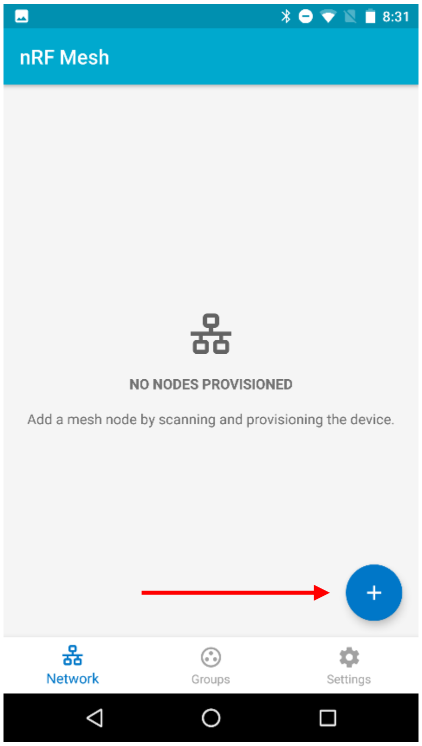
What's new
Last updated Mar 6, 2019
New Features:
- Group controls. Supports creating, updating, deleting groups and sending group messages in a mesh network

Rate this app
Tell others what you think

Developer contact

Fig 4.1: nRF Mesh Android App

2- Open nRF Mesh and press on the plus sign to scan for unprovisioned nodes.

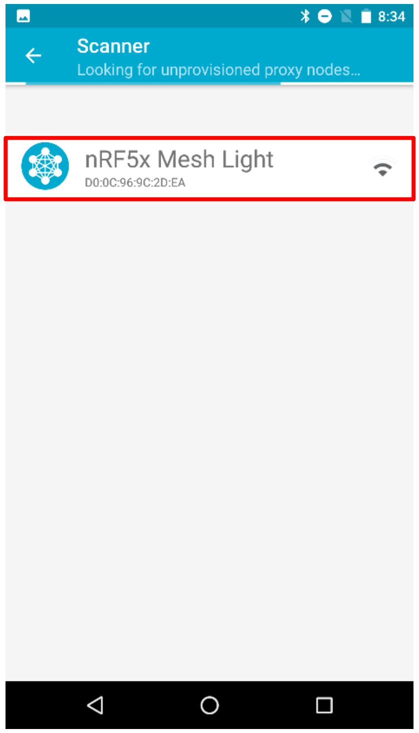


NO NODES PROVISIONED

Add a mesh node by scanning and provisioning the device.

Fig 4.2: nRF Mesh

3- You will see nRF5x Mesh Light in the list touch it to select.



Scanner

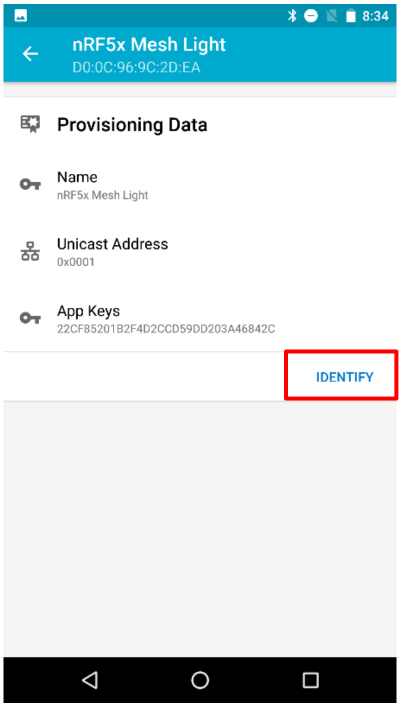
Looking for unprovisioned proxy nodes...

nRF5x Mesh Light

D0:0C:96:9C:2D:EA

Fig 4.3: nRF5x Mesh Light

4- Press **IDENTIFY** the board will flash the LEDS and change to the Provision Option.



nRF5x Mesh Light

D0:0C:96:9C:2D:EA

Provisioning Data

Name

nRF5x Mesh Light

Unicast Address

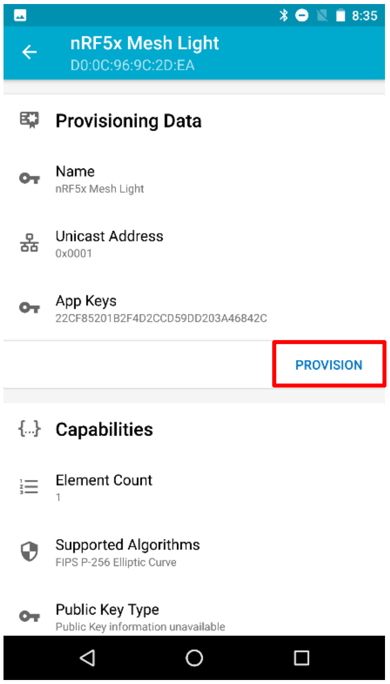
0x0001

App Keys

22CF85201B2F4D2CCD59DD203A46842C

Fig 4.4: nRF Mesh IDENTIFY

5- Press provision.



nRF5x Mesh Light
D0:0C:96:9C:2D:EA

Provisioning Data

Name
nRF5x Mesh Light

Unicast Address
0x0001

App Keys
22CF85201B2F4D2CCD59DD203A46842C

Capabilities

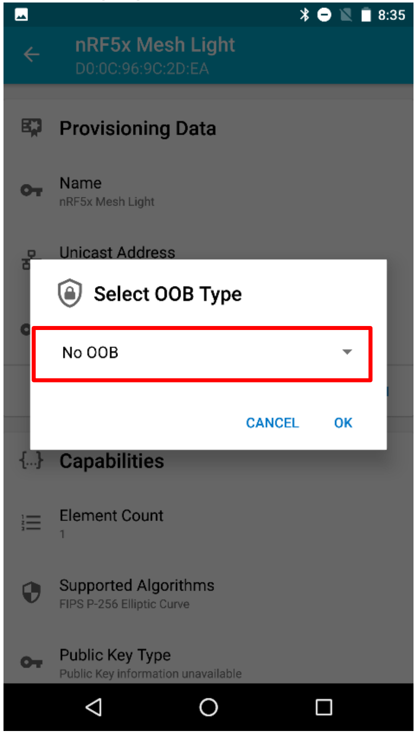
Element Count
1

Supported Algorithms
FIPS P-256 Elliptic Curve

Public Key Type
Public Key information unavailable

Fig 4.5: nRF Mesh PROVISION

6- Select no OOB for the provisioning option and observe the steps on the screen.



nRF5x Mesh Light
D0:0C:96:9C:2D:EA

Provisioning Data

Name
nRF5x Mesh Light

Unicast Address

Select OOB Type

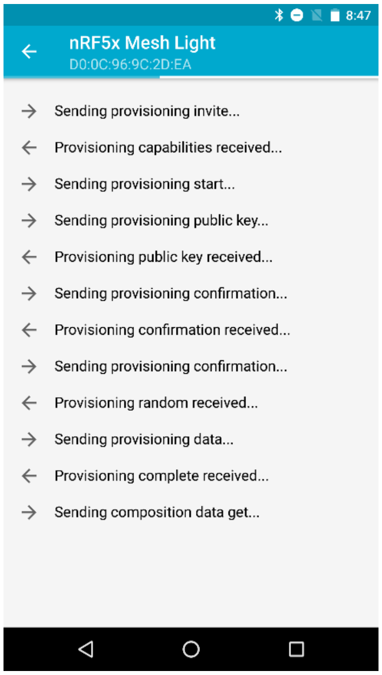
No OOB

Capabilities

Element Count
1

Supported Algorithms
FIPS P-256 Elliptic Curve

Public Key Type
Public Key information unavailable

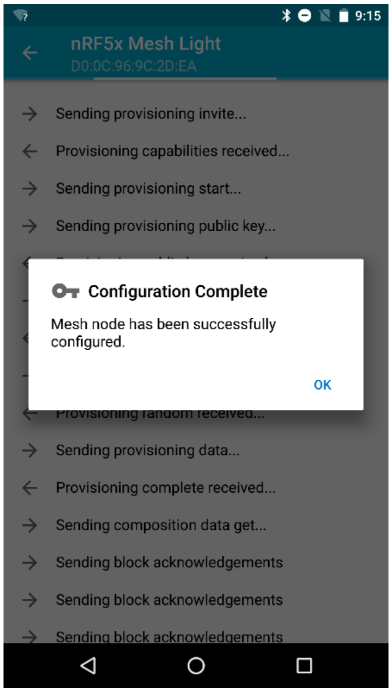


nRF5x Mesh Light
D0:0C:96:9C:2D:EA

→ Sending provisioning invite...
← Provisioning capabilities received...
→ Sending provisioning start...
→ Sending provisioning public key...
← Provisioning public key received...
→ Sending provisioning confirmation...
← Provisioning confirmation received...
→ Sending provisioning confirmation...
← Provisioning random received...
→ Sending provisioning data...
← Provisioning complete received...
→ Sending composition data get...

Fig 4.7: nRF Mesh provision options
Fig 4.6: nRF Mesh No OOB

7- Once the node has been provisioned connect to the node by touching gear in the nRf52x Mesh Light.



nRF5x Mesh Light
D0:0C:96:9C:2D:EA

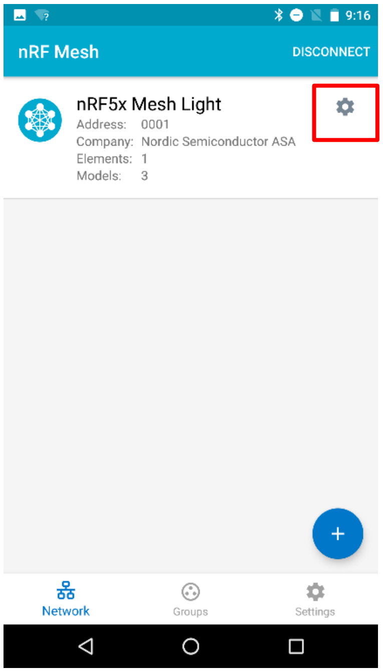
→ Sending provisioning invite...
← Provisioning capabilities received...
→ Sending provisioning start...
→ Sending provisioning public key...

Configuration Complete

Mesh node has been successfully configured.

← Provisioning random received...
→ Sending provisioning data...
← Provisioning complete received...
→ Sending composition data get...
→ Sending block acknowledgements
→ Sending block acknowledgements
→ Sending block acknowledgements

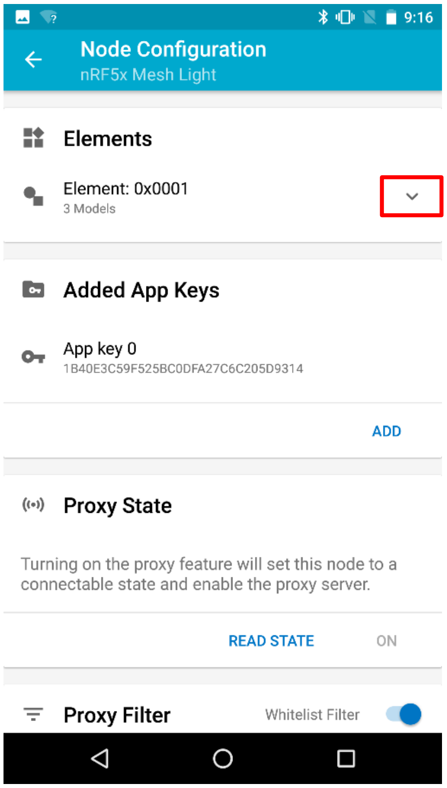
Fig 4.8: nRF Mesh Configuration



nRF5x Mesh Light
Address: 0001
Company: Nordic Semiconductor ASA
Elements: 1
Models: 3

Fig 4.9: nRF Mesh node connecting

8- Press the down arrow in the Elements.



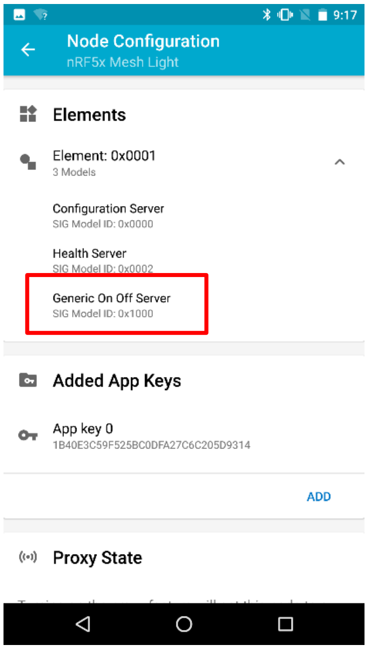
Element: 0x0001
3 Models

App key 0
1B40E3C59F525BC0DFA27C6C205D9314

Turning on the proxy feature will set this node to a connectable state and enable the proxy server.

Fig 4.10: nRF Mesh Elements

9- Three elements will be displayed the Configuration Server, the Health Server and the Generic On Off Server. Touch the Generic On Off Server.



Node Configuration
nRF5x Mesh Light

Elements

Element: 0x0001
3 Models

Configuration Server
SIG Model ID: 0x0000

Health Server
SIG Model ID: 0x0002

Generic On Off Server
SIG Model ID: 0x1000

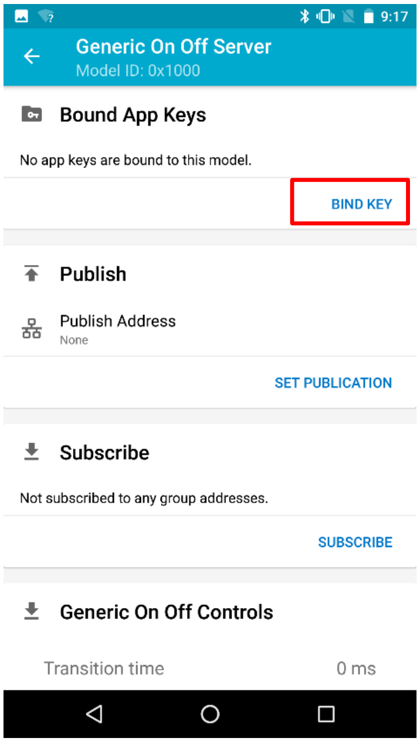
Added App Keys

App key 0
1B40E3C59F525BC0DFA27C6C205D9314

Proxy State

Fig 4.11: Generic On Off Server

10- On the top of the screen will be the Bound App Keys. Press the Bind Key and select App Key 0.



Generic On Off Server
Model ID: 0x1000

Bound App Keys

No app keys are bound to this model.

Publish

Publish Address
None

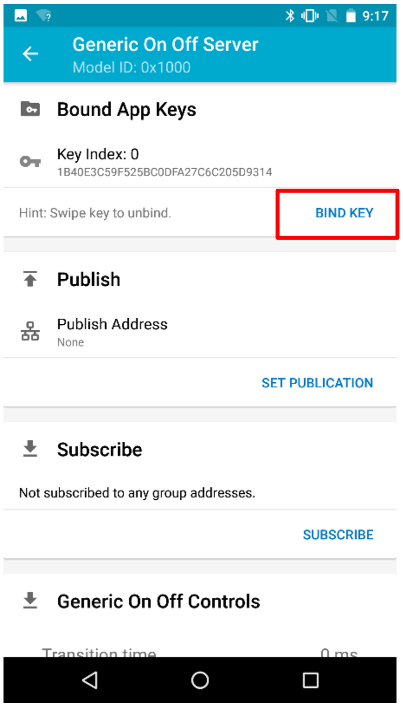
Subscribe

Not subscribed to any group addresses.

Generic On Off Controls

Transition time 0 ms

Fig 4.12: nRF Mesh Bind Key



Generic On Off Server
Model ID: 0x1000

Bound App Keys

Key Index: 0
1B40E3C59F525BC0DFA27C6C205D9314

Hint: Swipe key to unbind.

Publish

Publish Address
None

Subscribe

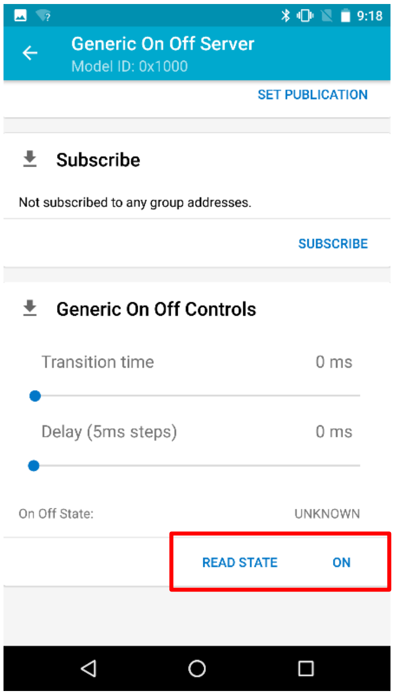
Not subscribed to any group addresses.

Generic On Off Controls

Transition time 0 ms

Fig 4.13: App Key 0

11- The Bounded keys will now display the value of Key Index 0. Scroll down to the Generic On Off Controls. You can now experiment with the controls and observe the behavior of the node.



Generic On Off Server
Model ID: 0x1000

Subscribe

Not subscribed to any group addresses.

Generic On Off Controls

Transition time 0 ms

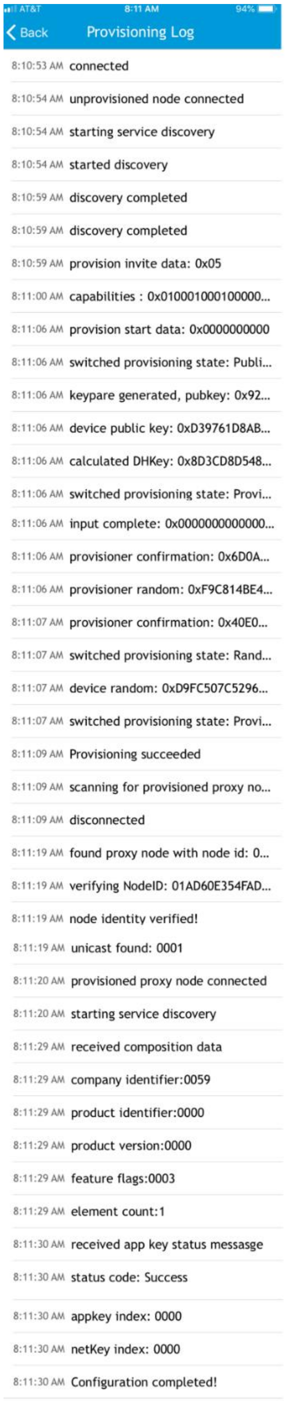
Delay (5ms steps) 0 ms

On Off State: UNKNOWN

Fig 4.14: Generic On Off Controls

The Nodes provisdioning information may be cleared by pressing button 4 on the board, the four LEDS will blink several times and the provisioning information will be cleared. You will then need to go in to the settings on the phone and reset the mesh network to reprovision the device.

If you are using an iPhone, you will need to view the provisioning log



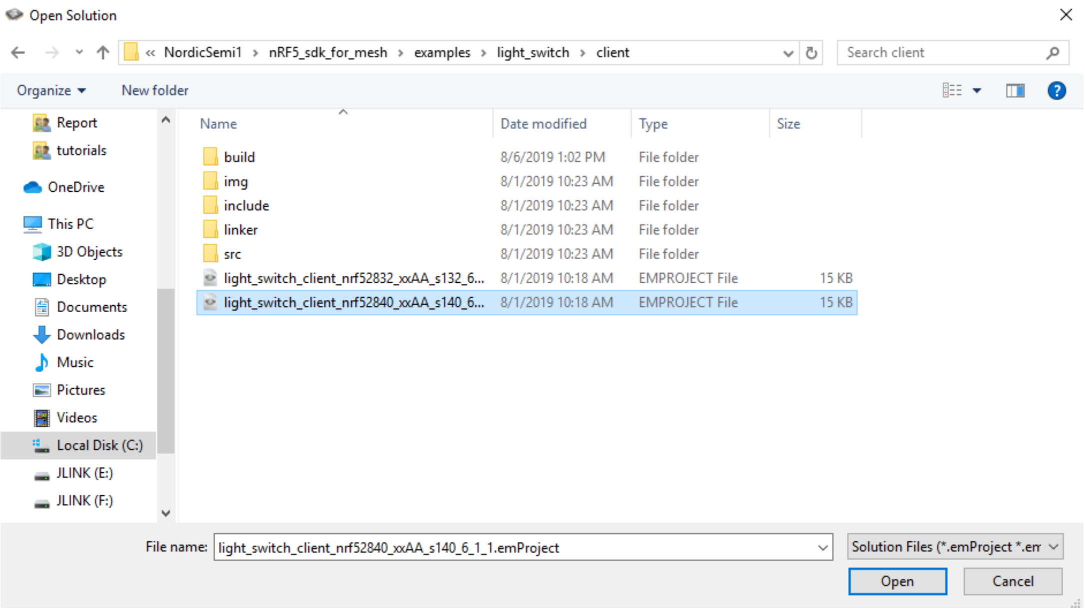
```
8:10:53 AM connected
8:10:54 AM unprovisioned node connected
8:10:54 AM starting service discovery
8:10:54 AM started discovery
8:10:59 AM discovery completed
8:10:59 AM discovery completed
8:10:59 AM provision invite data: 0x05
8:11:00 AM capabilities : 0x010001000100000...
8:11:06 AM provision start data: 0x0000000000
8:11:06 AM switched provisioning state: Publi...
8:11:06 AM keypare generated, pubkey: 0x92...
8:11:06 AM device public key: 0xD39761D8AB...
8:11:06 AM calculated DHKey: 0x8D3CD8D548...
8:11:06 AM switched provisioning state: Provi...
8:11:06 AM input complete: 0x0000000000000...
8:11:06 AM provisioner confirmation: 0x6D0A...
8:11:06 AM provisioner random: 0xF9C814BE4...
8:11:07 AM provisioner confirmation: 0x40E0...
8:11:07 AM switched provisioning state: Rand...
8:11:07 AM device random: 0xD9FC507C5296...
8:11:07 AM switched provisioning state: Provi...
8:11:09 AM Provisioning succeeded
8:11:09 AM scanning for provisioned proxy no...
8:11:09 AM disconnected
8:11:19 AM found proxy node with node id: 0...
8:11:19 AM verifying NodeID: 01AD60E354FAD...
8:11:19 AM node identity verified!
8:11:19 AM unicast found: 0001
8:11:20 AM provisioned proxy node connected
8:11:20 AM starting service discovery
8:11:29 AM received composition data
8:11:29 AM company identifier:0059
8:11:29 AM product identifier:0000
8:11:29 AM product version:0000
8:11:29 AM feature flags:0003
8:11:29 AM element count:1
8:11:30 AM received app key status messasge
8:11:30 AM status code: Success
8:11:30 AM appkey index: 0000
8:11:30 AM netKey index: 0000
8:11:30 AM Configuration completed!
```

Fig 4.15: provisioning log

Programming a Client Board

To program another board as a client, follow the directions.

1- Open Segger Embedded studio again.
2- Close the first solution by selecting **File**>>**Close solution**.
3- After closing the first solution Select **File**>>**Open solution**, C:\NordicSemi1\nRF5_sdk_for_mesh\examples\light_switch\client\light_switch_client_nrf52840_xxAA_s140_6_1_1.emProject



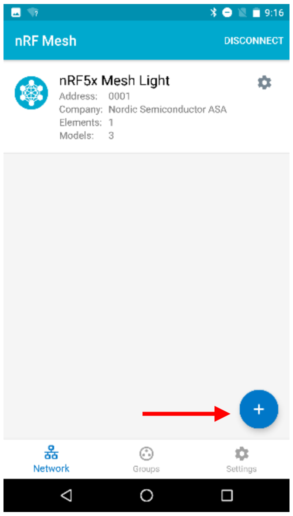
Name	Date modified	Type	Size
build	8/6/2019 1:02 PM	File folder	
img	8/1/2019 10:23 AM	File folder	
include	8/1/2019 10:23 AM	File folder	
linker	8/1/2019 10:23 AM	File folder	
src	8/1/2019 10:23 AM	File folder	
light_switch_client_nrf52832_xxAA_s132_6...	8/1/2019 10:18 AM	EMPROJECT File	15 KB
light_switch_client_nrf52840_xxAA_s140_6...	8/1/2019 10:18 AM	EMPROJECT File	15 KB

File name: light_switch_client_nrf52840_xxAA_s140_6_1_1.emProject

Solution Files (*.emProject *.em

Fig 5.1: Client Example Directory

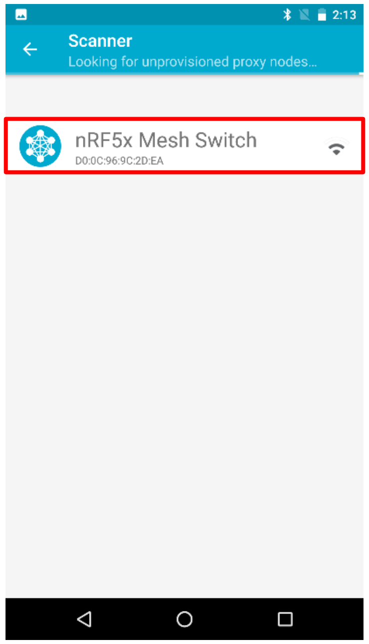
4- Connect the second board (Client), then build and program the example just like the server example.
5- To provision the client board, go to the nRF mesh app on your phone and click on the plus sign to scan for unprovisioned node.



nRF5x Mesh Light
Address: 0001
Company: Nordic Semiconductor ASA
Elements: 1
Models: 3

Fig 5.2: Screening for unprovisioned node

6- You will see nRF5x Mesh switch in the list touch it to select.



Looking for unprovisioned proxy nodes...

nRF5x Mesh Switch
D0:0C:96:9C:2D:EA

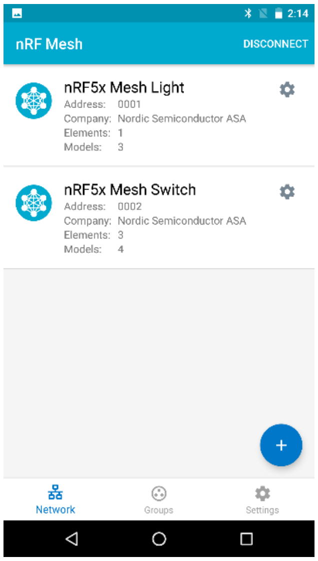
Fig 5.3: Selecting nRF5 mesh switch

7- Press **IDENTIFY** the board will flash the LEDS and change to the Provision Option.

8- Press provision.

9- Select no OOB for the provisioning option and observe the steps on the screen.

10- Once the node has been provisioned connect to the node by touching gear in the nRf52x Mesh Switch.

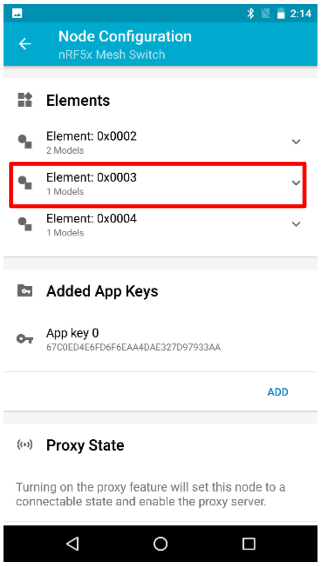


nRF5x Mesh Light
Address: 0001
Company: Nordic Semiconductor ASA
Elements: 1
Models: 3

nRF5x Mesh Switch
Address: 0002
Company: Nordic Semiconductor ASA
Elements: 3
Models: 4

Fig 5.4: nRF5 mesh

11- Press the down arrow in the Elements.



Element: 0x0002
2 Models

Element: 0x0003
1 Models

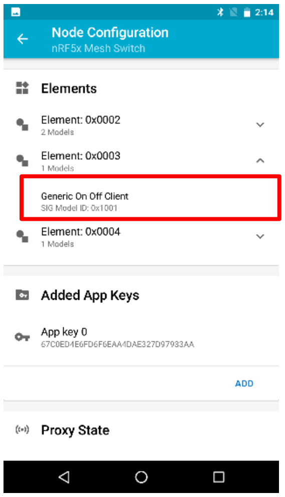
Element: 0x0004
1 Models

App key 0
67C0ED4E6FD6F6EAA4DAE327D97933AA

Turning on the proxy feature will set this node to a connectable state and enable the proxy server.

Fig 5.5: nRF5 mesh Elements

12- Three elements will be displayed press on element 3 and touch the Generic on Off Client.



2:14

Node Configuration

nRF5x Mesh Switch

Elements

Element: 0x0002
2 Models

Element: 0x0003
1 Models

Generic On Off Client
SIG Model ID: 0x1001

Element: 0x0004
1 Models

Added App Keys

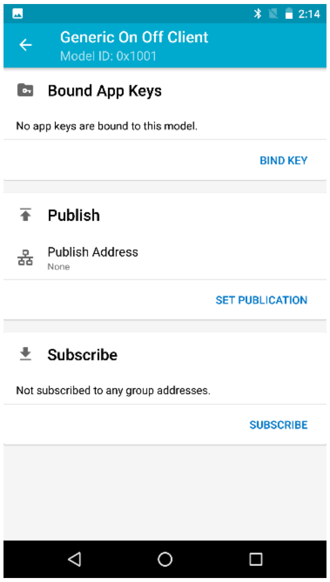
App key 0
67C0ED4E6FD6F6EAA4DAE327D97933AA

Proxy State

Fig 5.6: Generic on Off Client

13- On the top of the screen will be the Bound App Keys. Press the Bind Key and select App Key 0.

14- Select Publish to add a Publish Address, which is the nRF5 Mesh Light Address (0001) in our case and **Apply**.



2:14

Generic On Off Client

Model ID: 0x1001

Bound App Keys

No app keys are bound to this model.

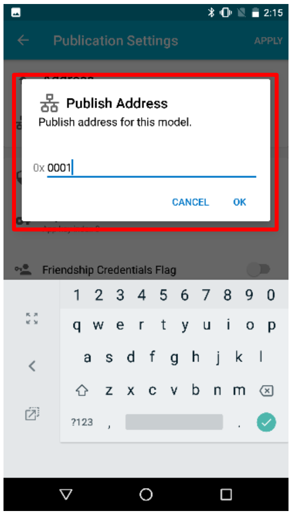
Publish

Publish Address
None

Subscribe

Not subscribed to any group addresses.

Fig 5.7: Publish, Subscribe settings



Publication Settings

Publish Address

Publish address for this model.

0x 0001

Friendship Credentials Flag

Fig 5.6: Publish Address

Now you should be able to control your Client board through your Server board.

Long Range mesh network

A long-range Bluetooth mesh networks is the ability of transferring more data with a longer distance between the transmitter and the receiver.

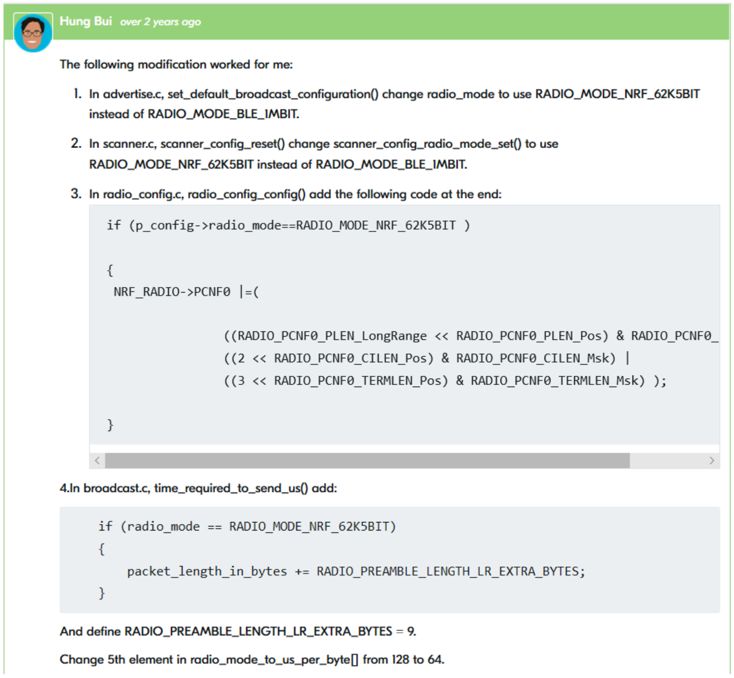
How?

As radio waves propagates over a distance the power density (Signal strength) gets reduced. Theoretically, when the antenna dimensions are very small compared to distance, doubling the distance results in ¼ th the energy density at the receiving point. To maintain the minimum energy density for a successful reception over a long distance, range can be increased by adding output power.

Bluetooth 5 long range feature provides a way to increase range without affecting output power, by reducing the baud rate. Which makes your device compatible with normal mesh devices and operating outside of the Bluetooth mesh specifications.

The following directions explain how to apply the previous example with a long range.

1- The modifications need to be made to the Nordic Mesh SDK are curtesy of Hung Bui of Nordic Semiconductor posted to the Nordic DevZone. https://devzone.nordicsemi.com/f/nordic-q-a/29813/change-phy-in-mesh.



Hung Bui over 2 years ago

The following modification worked for me:

1. In advertise.c, set_default_broadcast_configuration() change radio_mode to use RADIO_MODE_NRF_62K5BIT instead of RADIO_MODE_BLE_1MBIT.
2. In scanner.c, scanner_config_reset() change scanner_config_radio_mode_set() to use RADIO_MODE_NRF_62K5BIT instead of RADIO_MODE_BLE_1MBIT.
3. In radio_config.c, radio_config_config() add the following code at the end:

```
if (p_config->radio_mode==RADIO_MODE_NRF_62K5BIT )

{
 NRF_RADIO->PCNF0 |=(

                ((RADIO_PCNF0_PLEN_LongRange << RADIO_PCNF0_PLEN_Pos) & RADIO_PCNF0_
                ((2 << RADIO_PCNF0_CILEN_Pos) & RADIO_PCNF0_CILEN_Msk) |
                ((3 << RADIO_PCNF0_TERMLEN_Pos) & RADIO_PCNF0_TERMLEN_Msk) );

}
```

4.In broadcast.c, time_required_to_send_us() add:

```
if (radio_mode == RADIO_MODE_NRF_62K5BIT)
{
    packet_length_in_bytes += RADIO_PREAMBLE_LENGTH_LR_EXTRA_BYTES;
}
```

And define RADIO_PREAMBLE_LENGTH_LR_EXTRA_BYTES = 9.

Change 5th element in radio_mode_to_us_per_byte[] from 128 to 64.

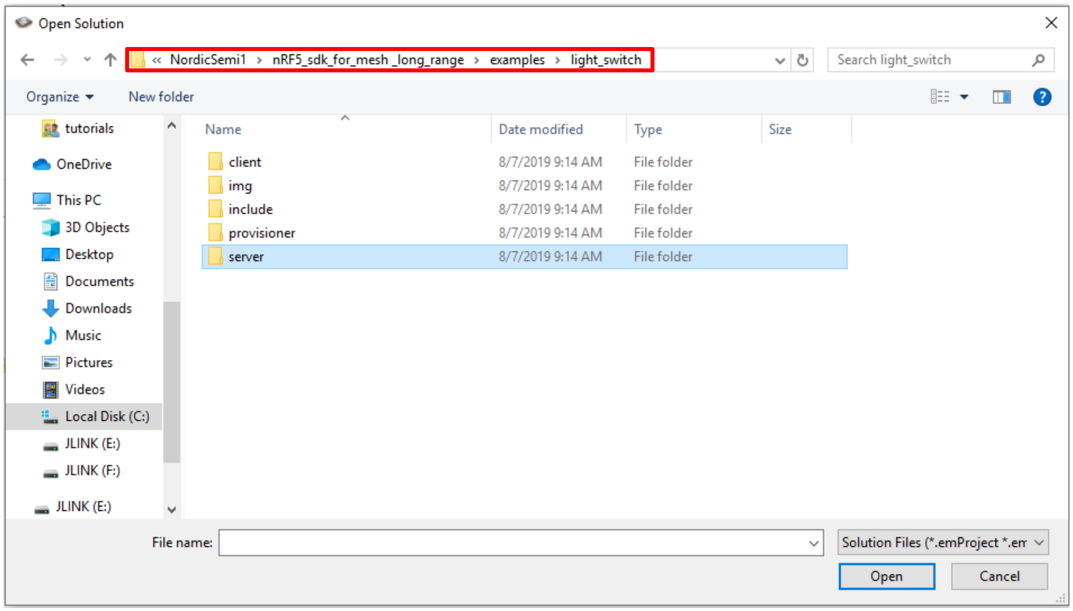
Fig 6.1: Modifications

Notice that in the first step the file to be modified is **advertiser.c**

2- Make a copy of nRF5_sdk_for_mesh and rename it to be nRF5_sdk_for_mesh_long_range

3- Open Segger Embedded Studio.

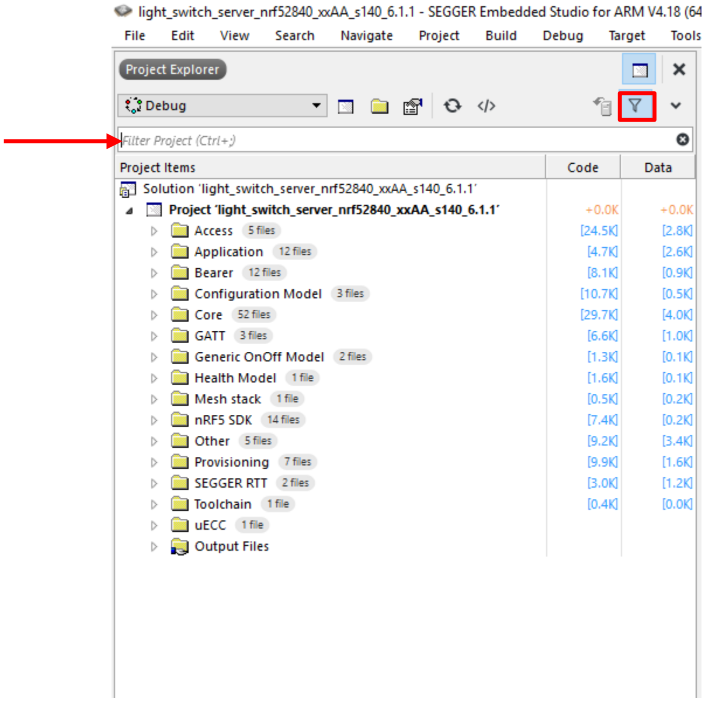
4- Select File>>Open solution>> **nRF5_sdk_for_mesh_long_range**\examples\light_switch\server\ **light_switch_server_nrf52840_xxAA_s140_6_1_1.emProject**



Name	Date modified	Type	Size
client	8/7/2019 9:14 AM	File folder	
img	8/7/2019 9:14 AM	File folder	
include	8/7/2019 9:14 AM	File folder	
provisioner	8/7/2019 9:14 AM	File folder	
server	8/7/2019 9:14 AM	File folder	

Fig 6.2: Long range example directory

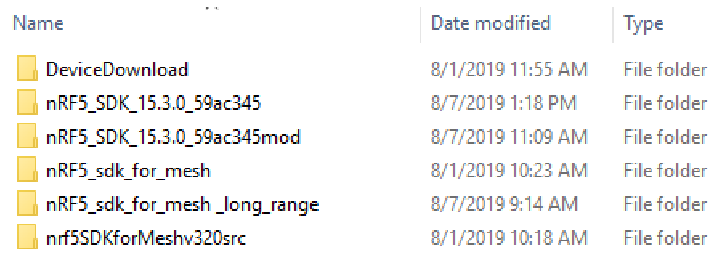
5- Find the mentioned files and modify them based on the directions.



Project Items	Code	Data
Solution 'light_switch_server_nrf52840_xxAA_s140_6.1.1'		
Project 'light_switch_server_nrf52840_xxAA_s140_6.1.1'	+0.0K	+0.0K
Access 5 files	[24.5K]	[2.8K]
Application 12 files	[4.7K]	[2.6K]
Bearer 12 files	[8.1K]	[0.9K]
Configuration Model 3 files	[10.7K]	[0.5K]
Core 52 files	[29.7K]	[4.0K]
GATT 3 files	[6.6K]	[1.0K]
Generic OnOff Model 2 files	[1.3K]	[0.1K]
Health Model 1 file	[1.6K]	[0.1K]
Mesh stack 1 file	[0.5K]	[0.2K]
nRF5 SDK 14 files	[7.4K]	[0.2K]
Other 5 files	[9.2K]	[3.4K]
Provisioning 7 files	[9.9K]	[1.6K]
SEGGER RTT 2 files	[3.0K]	[1.2K]
Toolchain 1 file	[0.4K]	[0.0K]
uECC 1 file		
Output Files		

Fig 6.3: Find files

6- Go back to the **Nordic** Website and download a new copy of the **nRF5 SDK** and extract **nRF5_SDK_15.3.0_59ac345** in your **NordicSemi** file.



Name	Date modified	Type
DeviceDownload	8/1/2019 11:55 AM	File folder
nRF5_SDK_15.3.0_59ac345	8/7/2019 1:18 PM	File folder
nRF5_SDK_15.3.0_59ac345mod	8/7/2019 11:09 AM	File folder
nRF5_sdk_for_mesh	8/1/2019 10:23 AM	File folder
nRF5_sdk_for_mesh _long_range	8/7/2019 9:14 AM	File folder
nrf5SDKforMeshv320src	8/1/2019 10:18 AM	File folder

Fig 6.4: The new directory

7- Connect the board you will use as a server to the computer and program it using the server example in the long range file.

8- Connect the board you will use as a client and program it using the client example in the long rage file.

9- Use the nRF mesh app on your phone to provision the two nodes, jus like the example in page 20.

After provisioning nodes, you will be able to control the LED in the client board using the switch in the server board.

Long Range Test with Different Antenna Patterns:

To see the difference the antenna patterns make on the overall performance and the difference between long range and normal mesh behavior we in CTi smart systems have done the following test, using 2 Rigado BMD-340 Evaluation Kits , referred to as Rigado units and 2 Nordic nRF52840-Preview-DK, referred to as Nordic units.

Many people are aware that the Long Range PHY is not a part of the BLE Mesh standard but as we only need to communicate with our devices range and penetration are the main concerns.

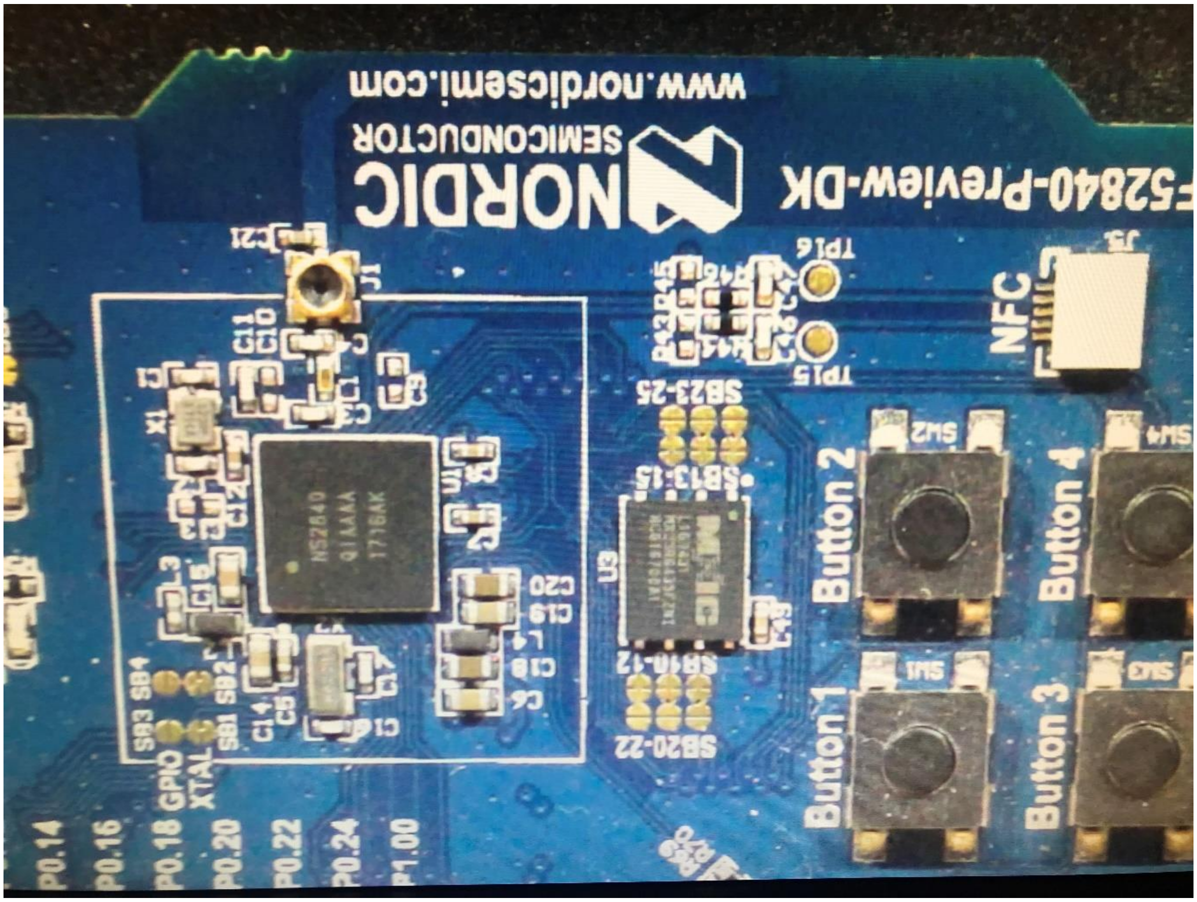


Fig 7.1: Nordic unit antenna

Looking closely at the Phone it can be seen that the Nordic nRF52840-Preview-DK uses a quarter wave monopole Antenna running under the www.nordicsemi.com

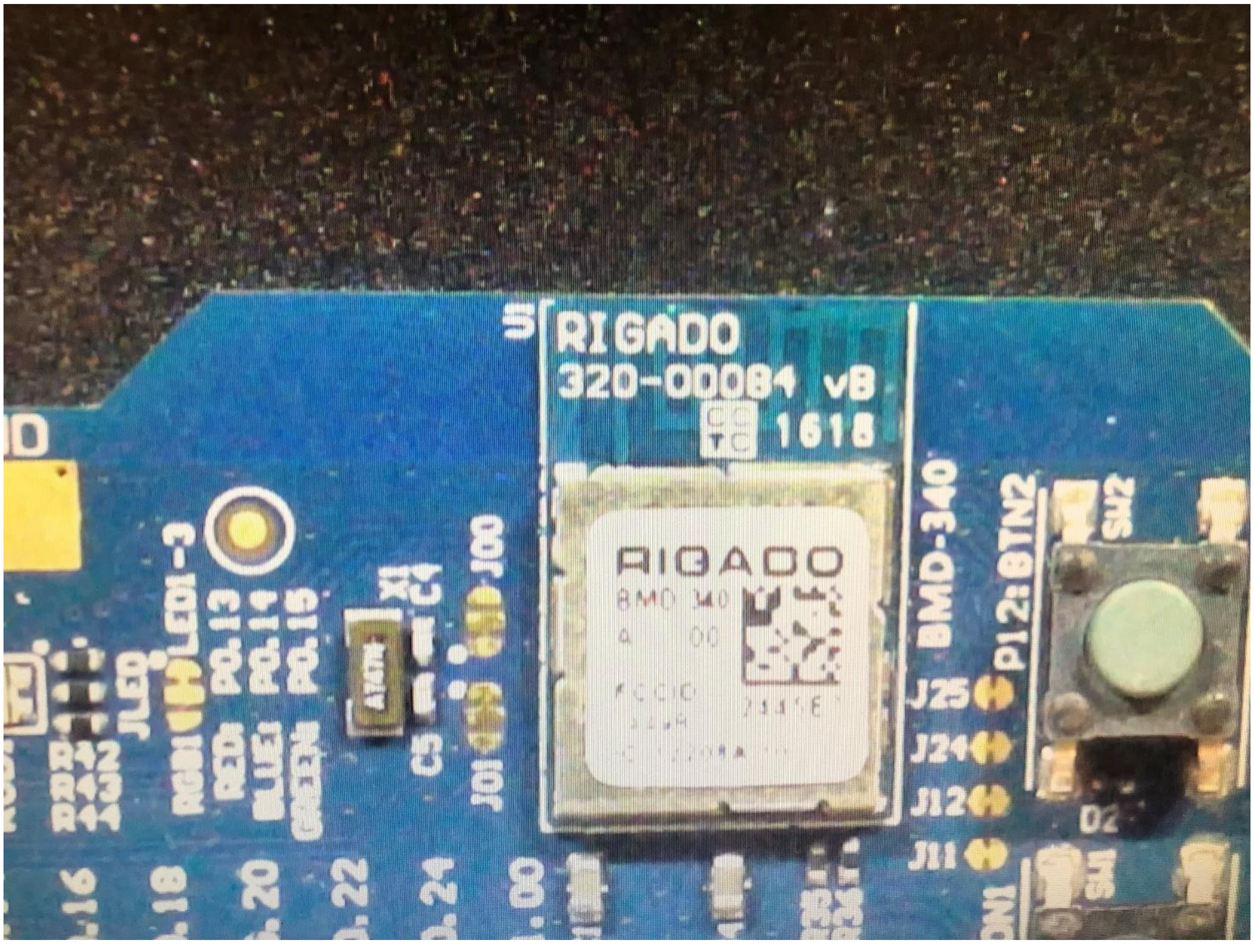


Fig 7.2: Rigado unit antenna

While the Antenna on the Rigado is bent into a Meandering design as seen under the RIGADO id.

Rigado Units Range Test:

The two **Rigado** units were programed for Long Range with the Mesh set to 62K5BIT for the physical layer and transmitting at 0dBm we configured them using nRF Mesh as the provisioner. First, we ran the test achieving a distance of **560 feet**.

The **Rigado** units were then reprogramed with the with the Mesh set to 1MBIT for the physical layer and transmitting at 0dBm for Normal Mesh Operation. Repeating the range test, we achieved **260 feet**.

Nordic Units Range Test:

Next the **Nordic** units were programed for Long Range with the Mesh set to 62K5BIT for the physical layer and transmitting at 0dBm we configured them using nRF Mesh as the provisioner. The range test provided **700 feet**.

The **Nordic** units were then reprogramed with the Mesh set to 1MBIT for the physical layer and transmitting at 0dBm for Normal Mesh Operation. The range test we achieved **560 feet**, however there is a rise where we were conducting the test, and this may have attributed to the range being less than the expected doubling.