ModusToolbox™

MeshClient and ClientControlMesh App User Guide

Document Number. 002-26575 Rev. *A

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com

Contents

About This Document

This document provides quick start instructions for the MeshClient and the ClientControlMesh applications, which are part of WICED® Studio and ModusToolbox™.

Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use. For a comprehensive list of acronyms and other terms used in Cypress documents, go to www.cypress.com/glossary.

IoT Resources and Technical Support

Cypress provides a wealth of data at http://www.cypress.com/internet-things-iot to help you to select the right IoT device for your design, and quickly and effectively integrate the device into your design. Cypress provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates. Customers can acquire technical documentation and software from the Cypress Support Community website (http://community.cypress.com/).

1 Overview

Bluetooth SDK in WICED® Studio and ModusToolbox offer wide variety of Bluetooth SIG Mesh 1.0-related products. One of them is a set of portable libraries that can be used on any platform to create an application to provision and control the mesh. WICED Studio supports Bluetooth Mesh on CYW20706, CYW20735, CYW20719 and Cypress modules based on these silicon devices. ModusToolbox includes support for Bluetooth Mesh on CYW20819 and Cypress modules such as CYBT-213043-02 based on this device.

The MeshClient and the ClientControlMesh applications provide a sample Windows implementation that show how to use interfaces exposed by the mesh libraries. The MeshClient works only with Windows 10. The MeshClient application uses PC's built-in Bluetooth radio, or an external Bluetooth dongle to communicate with Bluetooth mesh. The ClientControlMesh application implements all layers of the mesh stack. The ClientControlMesh application implements only the application layer. It uses the Mesh Models and Mesh Core libraries residing on the embedded device that requires a Cypress device to act as a client and hence requires an extra evaluation board to be connected to the PC for mesh operation. Any of the Cypress devices that support Bluetooth mesh can be used for this application irrespective of the device used by the mesh nodes. The ClientControlMesh can be used with any version of Windows operating system.

The MeshClient and the ClientControlMesh Windows applications are installed with WICED Studio and ModusToolbox installation as part of Bluetooth SDK.

- **App paths in WICED Studio** (Change 6.4 to appropriate version based on the WICED Studio version being used)

  If the default path for the installation is used, the **MeshClient** project is in:

  C:\Users\<user>\Documents\WICED-Studio-6.4\common\apps\snip\mesh\peerapps\Windows\MeshClient

  To open the application on Windows machine:

  ☐ Go to: *C:\Users\<user>\Documents\WICED-Studio-6.4\common\apps\snip\mesh\peerapps\Windows\MeshClient\Release\x86*

  ☐ Double-click the **MeshClient** application.

  The **ClientControlMesh** project is in:

  C:\Users\<user>\Documents\WICED-Studio-6.4\common\apps\snip\mesh\ClientControl

  To open the application on Windows machine:

  ☐ Go to: *C:\Users\<user>\Documents\WICED-Studio-6.4\common\apps\snip\mesh\ClientControl\Release*

  ☐ Double-click the **ClientControlMesh** application.

- **App paths in ModusToolbox** (Change ModusToolbox_1.1 and bt_sdk-1.1 to the appropriate versions based on the ModusToolbox version being used)

  If the default path for the installation is used, the **MeshClient** project is in:

  C:\Users\<name>\ModusToolbox_1.1\libraries\bt_sdk-1.1\components\BT-SDK\common\apps\snip\mesh\peerapps\Windows\MeshClient

  To open the application on Windows machine:

  ☐ Go to: *C:\Users\<name>\ModusToolbox_1.1\libraries\b t_sdk-1.1\components\BT-SDK\common\apps\snip\mesh\peerapps\Windows\MeshClient\Release\x86.*

  ☐ Double-click the **MeshClient** application.

  The **ClientControlMesh** project is in:

  C:\Users\<name>\ModusToolbox_1.1\libraries\bt_sdk-1.1\components\BT-SDK\common\apps\snip\mesh\ClientControl

  To open the application on Windows machine

  ☐ Go to *C:\Users\<name>\ModusToolbox_1.1\libraries\bt_sdk-1.1\components\BT-SDK\common\apps\snip\mesh\ClientControl\Release*

  ☐ Double-click the **ClientControlMesh** application.

These apps can also be downloaded online from Software tab at www.cypress.com/ble-mesh.

The ClientControlMesh and the MeshClient applications can be built using Microsoft Visual Studio 2017 or later release.

Operating System (OS) Requirements

- **MeshClient** - The MeshClient application relies upon the Windows Bluetooth stack version which is available only in Windows 10 Creators Update. While launching the application, if you see the error "This application requires Windows 10 Creator Updates", install these updates to run this application. Go to the following link click **Update now** to download and install these updates:
  https://www.microsoft.com/en-us/software-download/windows10
- **ClientControlMesh** - The ClientControlMesh does not use Windows stack and can be executed on any version of Windows OS.

Figure 1 shows the software block diagram of the MeshClient (left) and ClientControlMesh (right) applications.

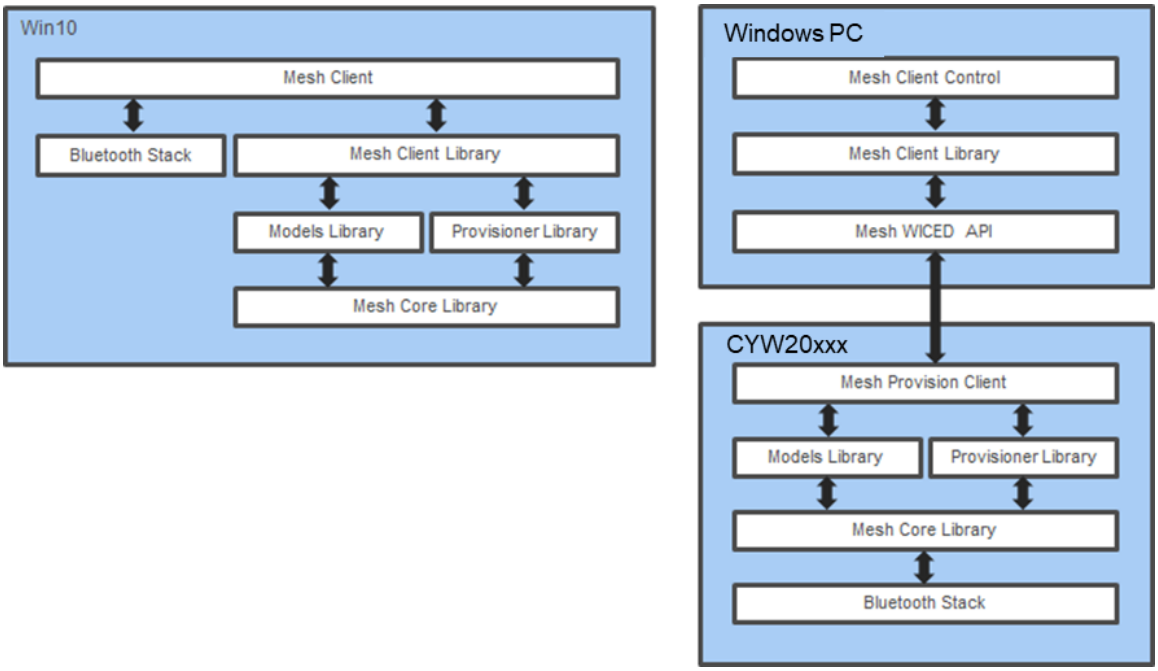


Figure 1. MeshClient (Left) and ClientControlMesh (Right) Software Block Diagram

The MeshClient application uses Bluetooth stack as it exists on Windows 10 OS. It uses GATT Proxy connection [1] to control the mesh. The ClientControlMesh application uses Bluetooth stack of the Cypress silicon. It can supports both GATT Proxy and advertising channel to provision and control mesh devices.

The MeshClient and ClientControlMesh applications expose the functionality of various client models defined in the Mesh specifications including Configuration, Health, Default Transition Time, OnOff, Level, Power OnOff, Light Lightness, Light HSL, Light CTL, Sensor, and a sample Vendor-Specific client models. Other client and server models can be added in future releases.

1.1 Mesh Libraries

The MeshClient library executes the state machines required for provisioning and configuration. It provides an interface to the application to test the mesh functionality. The library maintains the database for the mesh network.

In the MeshClient application, to execute Bluetooth functionality such as starting BLE scan, establishing a connection to a specific device, or sending a data packet, the MeshClient library executes methods that are provided by the MeshClient application, which in turn uses Bluetooth Stack of the OS. On the other hand, in the ClientControlMesh application, the MeshClient library uses Mesh WICED API to control embedded application to perform all the mesh related work.

The Mesh Core, Models, Provisioner and Core libraries implement all the functionality as defined in the Bluetooth SIG Mesh Profile [1] and Mesh Models [2] specifications.

2 MeshClient Applications Overview

2.1 Provisioning

Provisioning is a process of adding new nodes into a Mesh network. Provisioning is performed by a special node called a “Provisioner”. The MeshClient/ClientControlMesh applications perform as a Provisioner in the mesh network. MeshClient and ClientControlMesh applications maintain the database for the network, initiate the scan for unprovisioned devices, and perform the provisioning procedure as defined in Mesh Profile specification [1]. As a result of execution of the provisioning procedure, the Provisioner provides to the new node a bare minimum of the information to be a part of the mesh network (network key, IV index) and establishes the Device Key for the new node that is used between the Provisioner and the node during the configuration stage.

While the Mesh specification allows provisioning over the Advertising (PB-ADV) and GATT (PB-GATT) bearers, the MeshClient uses the GATT bearer only because it relies on the Microsoft Bluetooth stack as transport. The MeshClient Control can be configured to use any bearer.

2.2 Configuration

It is not enough just to provision a device to make it a fully functional mesh node of the mesh network. The following is a partial list of things that the Provisioner needs to perform during the configuration:

- Read the new node’s composition data to find out the device capabilities. For example, based on the information in the composition data, the Provisioner can figure out if it is a switch, a light bulb, or some other device.
- Set up the features that the new node should support. For example, if the node supports GATT Proxy or Friend role, the Provisioner needs to specify if the node should use the feature.
- Add network keys if the node should also be a part of other subnets, and add application keys for use with various models.
- Bind appropriate application keys to appropriate models of the new node. For example, the Provisioner can specify one application key to be used to configure the bulb and different application key to control the bulb.
- Configure various network parameters. For example, the Provisioner can specify the number of times the node should retransmit the message if it performs as a relay, and the number of times and frequency at which the node should publish the status messages.
- Configure the new device to be a part of a group.
- Configure clients, for example on/off switch, to control a specific server such as a light bulb, or a group of servers such as all light bulbs in a room.

2.3 Control

After the new node has been provisioned and configured, it can send and receive massages to and from devices in the same mesh network. For example, when you provision and configure a switch, the switch can send ON/OFF commands to a bulb or to all bulbs in the room.

The MeshClient and ClientControlMesh can act as various actuators including an on/off switch, a dimmer and a color control. For that purpose, they support corresponding client models and can send various Get/Set commands to control mesh devices. For example, the application can send a command to dim the light bulb to a certain level, or to adjust the color temperature of the light bulb.

Similar to any other client, the application can send messages to a single device or to a group of the devices. The replies are typically received from each device. When the application addresses the group with acknowledged message, each device in the group would send a reply. The mesh stack monitors how many replies have been received, and if reply from some specific node is not received, the Device Unreachable message is sent to the application.

Depending on the type of the device, some devices may act purely as clients, others like servers, and some can act simultaneously as client and servers. A simple generic on/off switch is an example of a clean client. An HSL light bulb is an example of a pure server. There can be a node which is wired to two bulbs. There can be a power strip with one switch and several outlets, and the switch can be configured to control one of the outlets, or all outlets or the strip, or several strips.

3 Using the MeshClient Application

- If WICED Studio is being used for application development and if the default path for WICED Studio installation is used, open the following application:
  C:\Users\<user>\Documents\WICED-Studio-6.4\common\apps\snip\mesh\peerapps\Windows\MeshClient\Release\x86\MeshClient.exe.
- To run the ClientControlMesh application, open the following application:
  C:\Users\<user>\Documents\WICED-Studio-6.4\common\apps\snip\mesh\ClientControl\Release\x86\ ClientControlMesh.exe.
- If ModusToolbox is being used for the application development and if default path for ModusToolbox installation is used, open the following application:
  C:\Users\<user>\ModusToolbox_1.1\libraries\bt_sdk-1.1\components\BT-SDK\common\apps\snip\mesh\peerapps\Windows\MeshClient\Release\x86\ MeshClient.exe
- To run the ClientControlMesh application, open the following application:
  C:\Users\<user>\ModusToolbox_1.1\libraries\bt_sdk-1.1\components\BT-SDK\common\apps\snip\mesh\ClientControl\Release\ClientControlMesh.exe

If a Windows 10 PC is used, it is recommended to use MeshClient application as it does not require an external device to run Bluetooth stack.

The user interface of the MeshClient and the ClientControlMesh applications are very similar. The only key difference between the two applications is COM port selection and Baud rate setting. These fields are not available in the MeshClient app as it uses PC's in-built Bluetooth. The ClientControlMesh application requires to talk to an external Cypress Evaluation board/device over the HCI UART. Hence these fields are provided in the ClientControlMesh application. Refer section 3.1 to learn how to select COM port and baud rate. Rest of this document uses screenshots from the MeshClient app as most of other fields and buttons are similar in both apps.

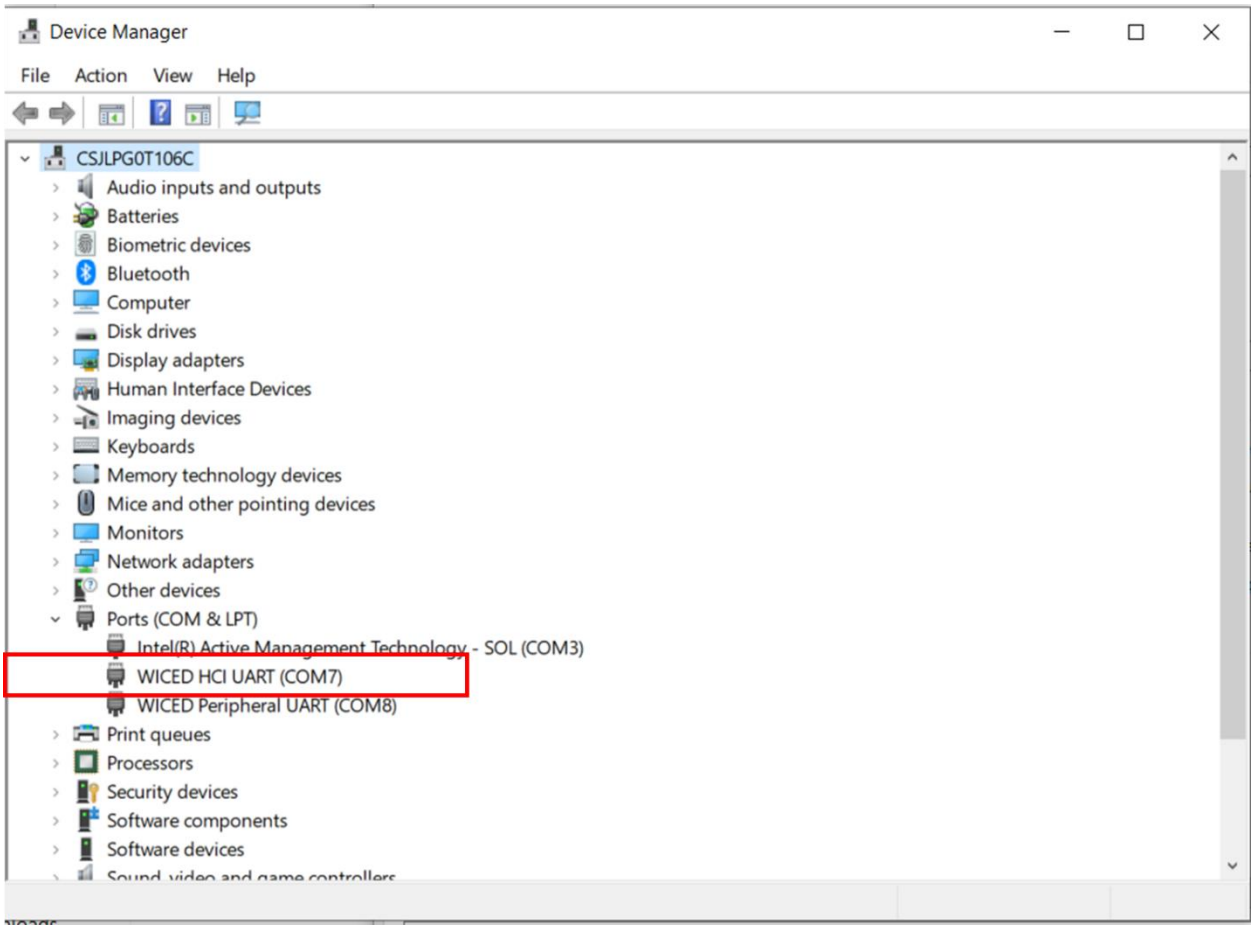
3.1 Creating and Opening a Mesh Network

Step 1: Jump to step 2 if you are using the MeshClient application. Continue here if using the ClientControlMesh application.

Program one evaluation board with mesh_provision_client snip application. This snip is available as part of WICED Studio installation under *apps\snip\mesh\mesh_provison_client* and on GitHub for CYW20819 device. If using WICED Studio, refer to the respective kit's user guide to learn how to build and download an application on to the board. If using ModusToolbox while designing your application with CYW20819 or CYW20819-based EZ-BT module such as CYBT-213043-02, refer to the Getting Started with Bluetooth Mesh Application Note to learn how to download code examples from GitHub and program the board.

Once the board is programmed and connected to the PC, check the COM port number for the HCI UART. To check the COM port number, go to **Device Manager** on your PC and expand **Ports (COM & LPT)**. Here, look for **WICED HCI UART**. This is the COM port number to be used in the ClientControlMesh application. See the following screenshot.

Note: If the PC is detecting HCI and PUART ports as **USB Serial Port** without any distinction, then the lower COM port number is likely to be HCI UART's COM port number.



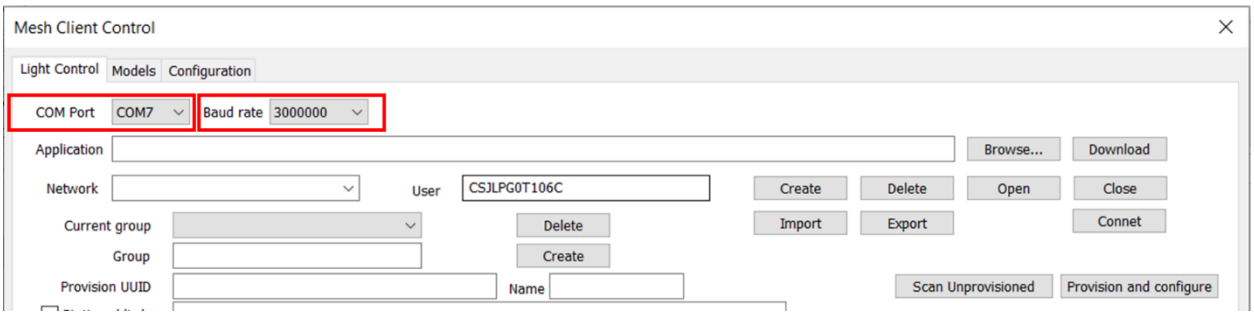
Once the COM port is identified, open the ClientControlMesh application.

(**Note**: If the board is connected to PC after opening the ClientControlMesh application, the ClientControlMesh application will not detect the COM port. So, make sure you open application after the board is connected and enumerated.)

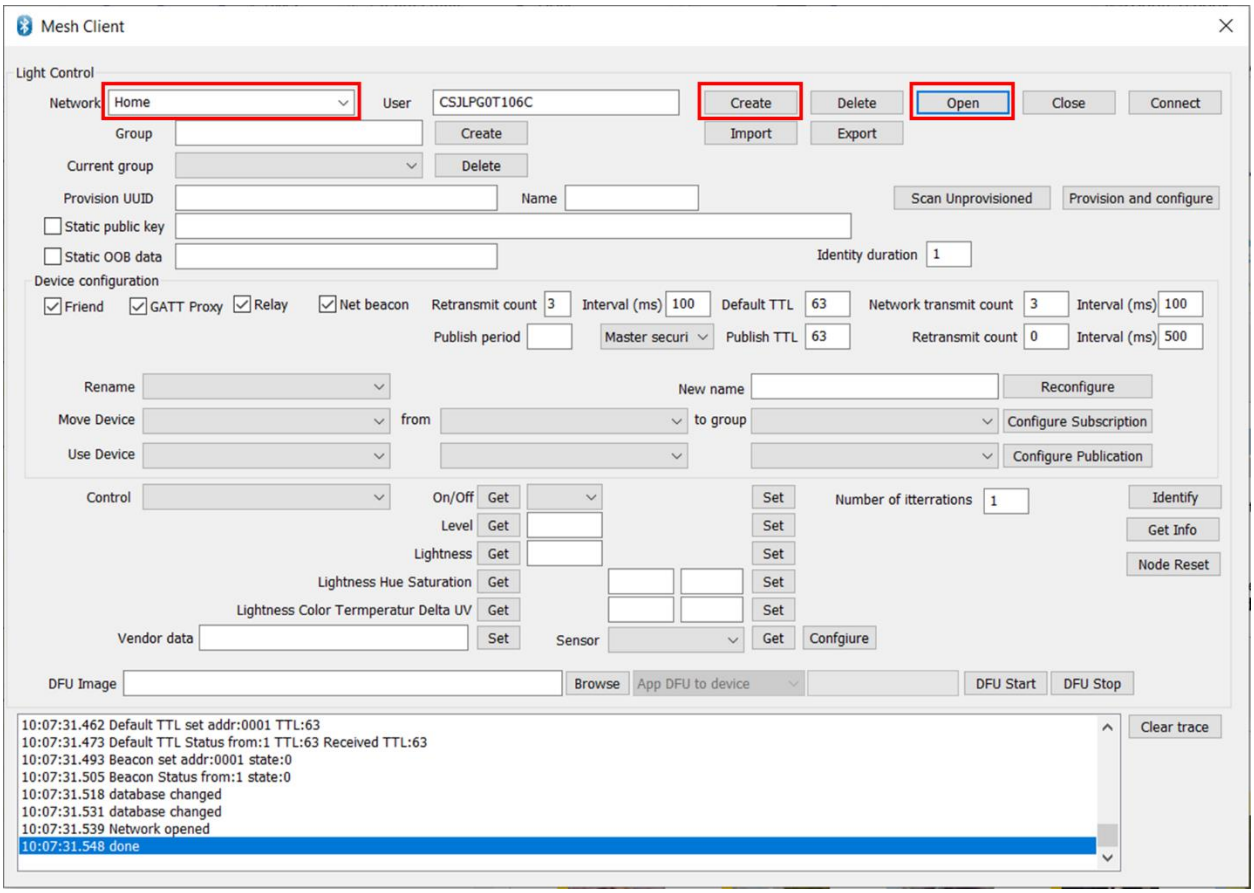
Select this HCI-UART COM port form the **COM port** dropdown menu.

Then, select **Baud rate** 300000 if using CYW20819EVB-02 or any other chip-on-board evaluation boards.

Select **Baud rate** 921600 if the CYBT-213043-EVAL or CYBT-213043-MESH EZ-BT Mesh evaluation kits are being used to run provision control client.



Step 2: In the **Network** field, type the string that you want to use as the network name. Click **Create** and then, click **Open**. See the following screenshot.



When a network is created, the MeshClient creates the required network attributes such as the mesh UUID, network and application keys, and saves the information in the mesh database which is stored in a JSON file in the directory where the application is started from. The schema of the Mesh Provisioner database is described in corresponding document from Bluetooth SIG [3].

There can be multiple networks controlled by the same PC, for example “Home”, “Office”, “Parent’s house”.

When you click **Open**, the MeshClient configures the stack with the parameters of the selected network. Similarly, the ClientControlMesh talks over the selected COM port to configure the stack running on the embedded platform. The “done” trace at the end of the configuration process indicates that the stack has been configured successfully.

3.2 Adding a Node

Use WICED Studio or ModusToolbox to build and download one of the mesh samples to a WICED evaluation board. In the following description, the BLE_Mesh_LightDimmable code example is used. See the respective kits’ user guide/getting started guide to learn how to program the board.

Do the following to provision a new node and also see the following screenshot:

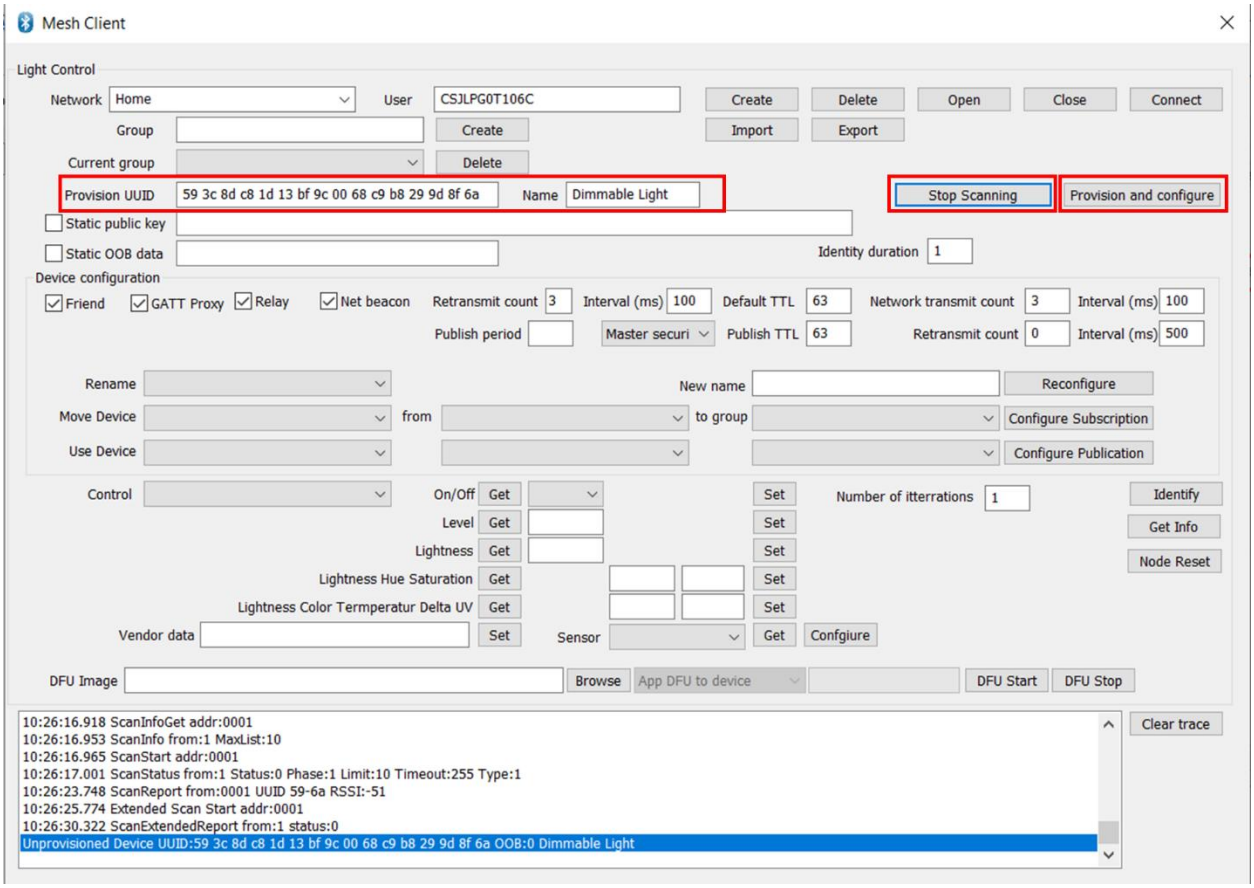
1. In the MeshClient window, click **Scan Unprovisioned**.

   The title of the button changes to **Stop Scanning** to indicate that the scan is active.

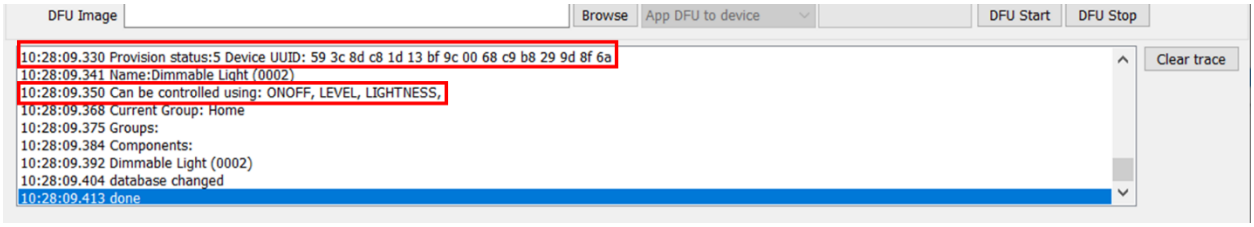
2. The trace window displays the UUIDs of the devices that are in the radio range. The **Provisioned UUID** field is automatically filled with the UUID of the last discovered device.

3. When you see the device that you want to work with, click **Stop Scanning**.

4. Click **Provision and configure**.



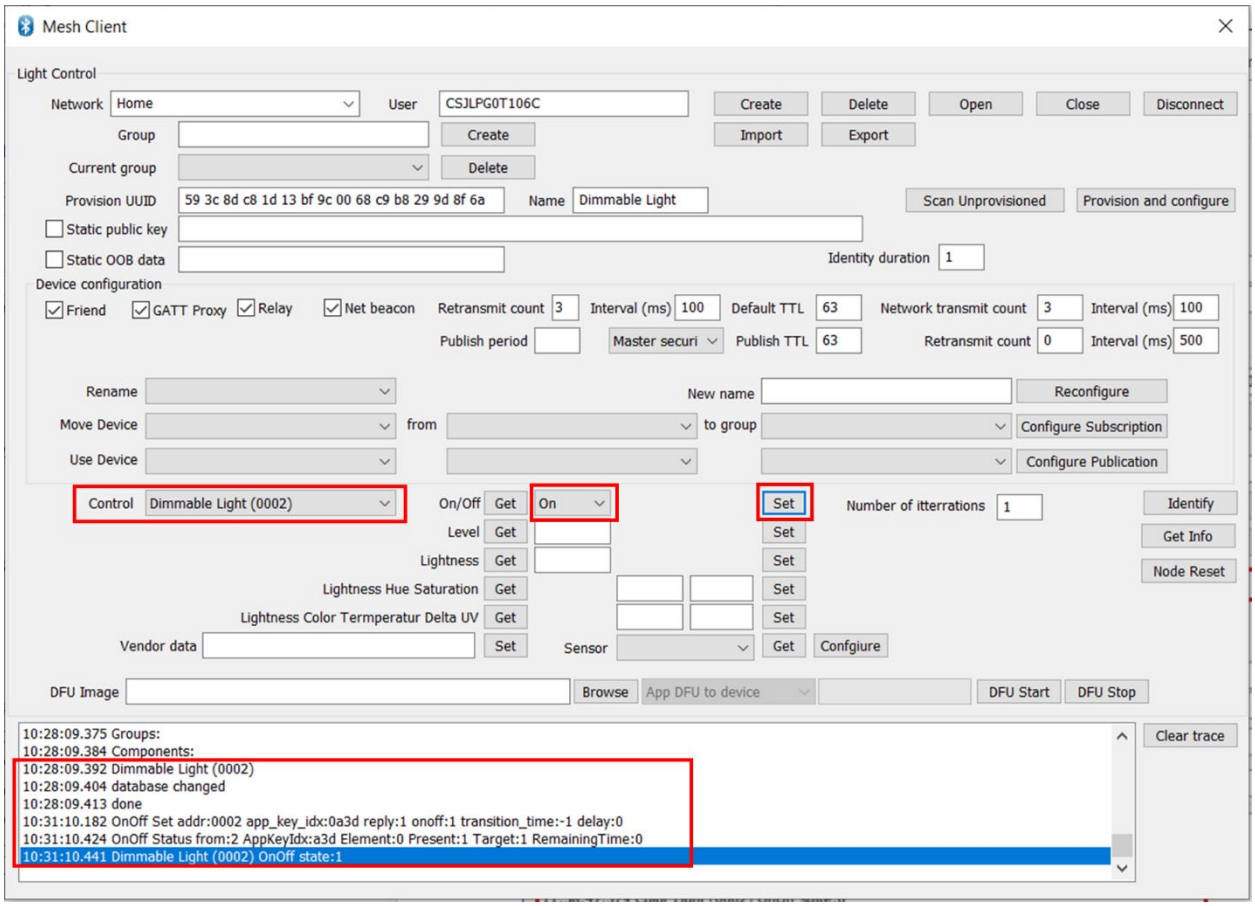
The provisioning and configuration process consists of several of steps. While the process is being executed, the status is displayed in the trace window. At the end of the process, **Provision status:5** appears in the trace window, indicating that the process has been completed successfully. The MeshClient also queries the library for the methods available for the application to control the device. For example, the provision device in the trace below can be controlled using On Off, Level as well as Lightness.



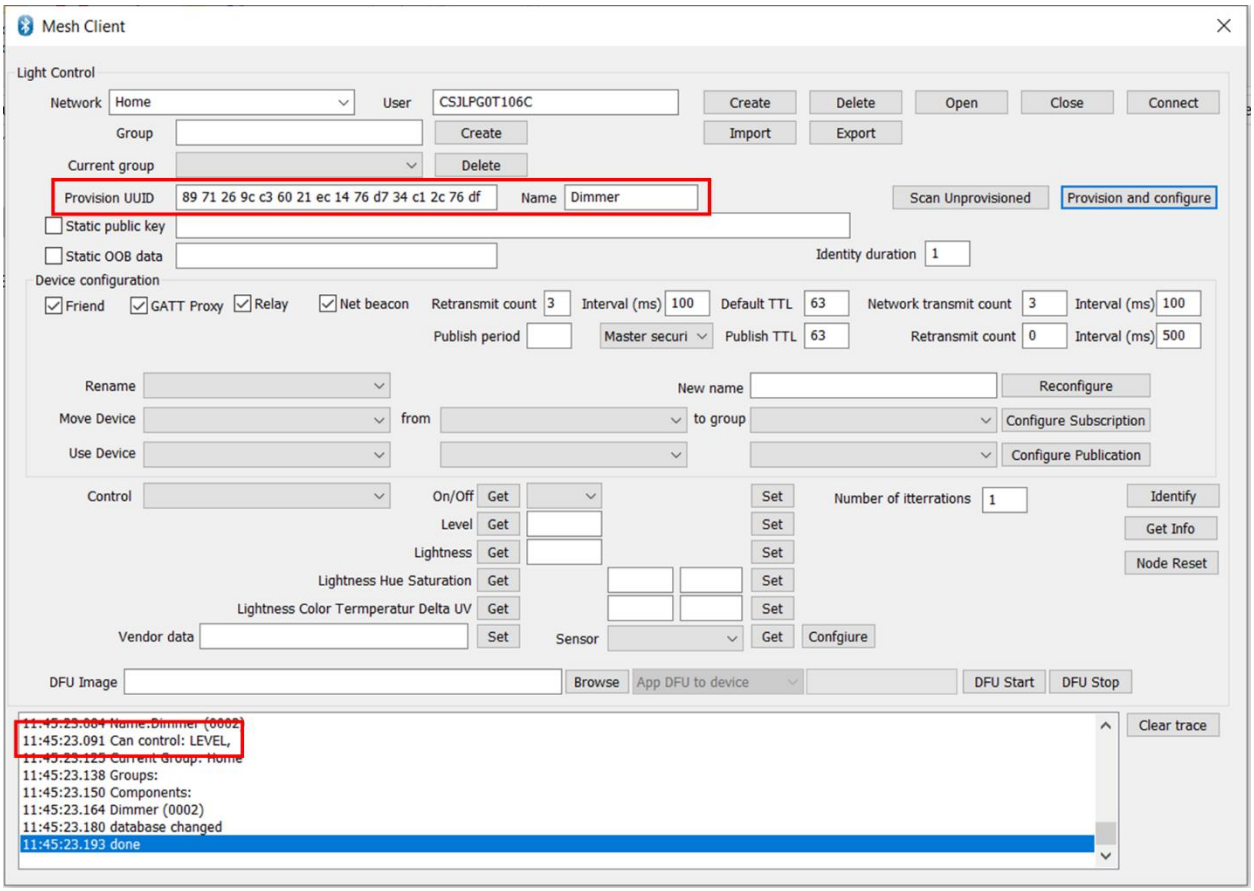
The trace window will print out the results of the operation.

If the device has been configured to be a GATT Proxy, the MeshClient will keep the connection to the new device open. If needed, click **Disconnect** to drop the GATT connection. When MeshClient is not connected to the mesh network, click **Connect** to establish the GATT connection.

The device is ready to use. For example, you can select the device and issue a **Get** the **On/Off status** command, or set a desired state to **On** and issue a **Set** command. Before sending any command to the node, ensure that app is connected to the proxy node i.e. the **Connect/Disconnect** button must shows **Disconnect**.

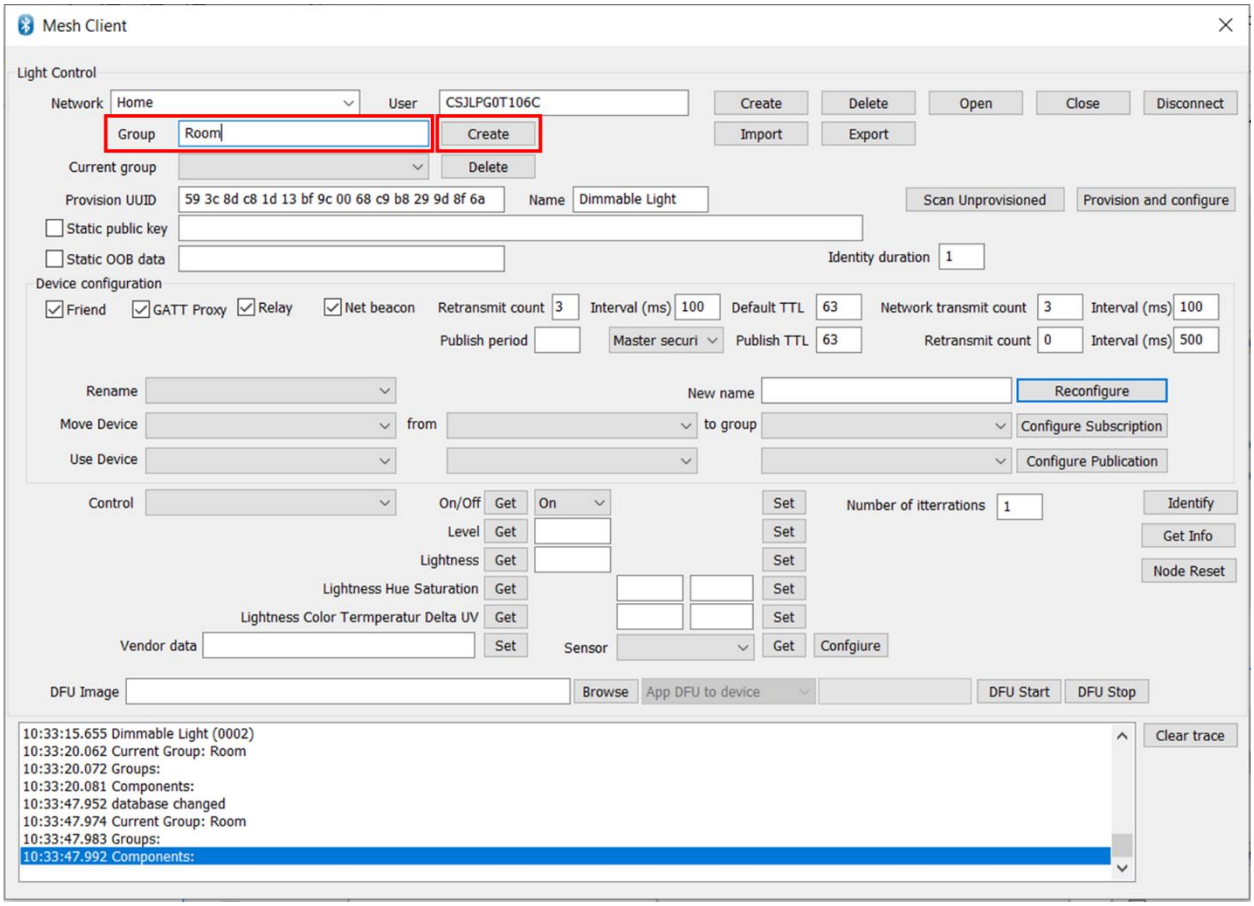


When a switch/dimmer or any other client is provisioned, instead of the “Can be controlled using” tag, the trace window will show “Can control”. For example, following screenshot shows the “Can Control” trace when a dimmer (level client model) is provisioned. As it is a level client, it can control level as shown in the traces.



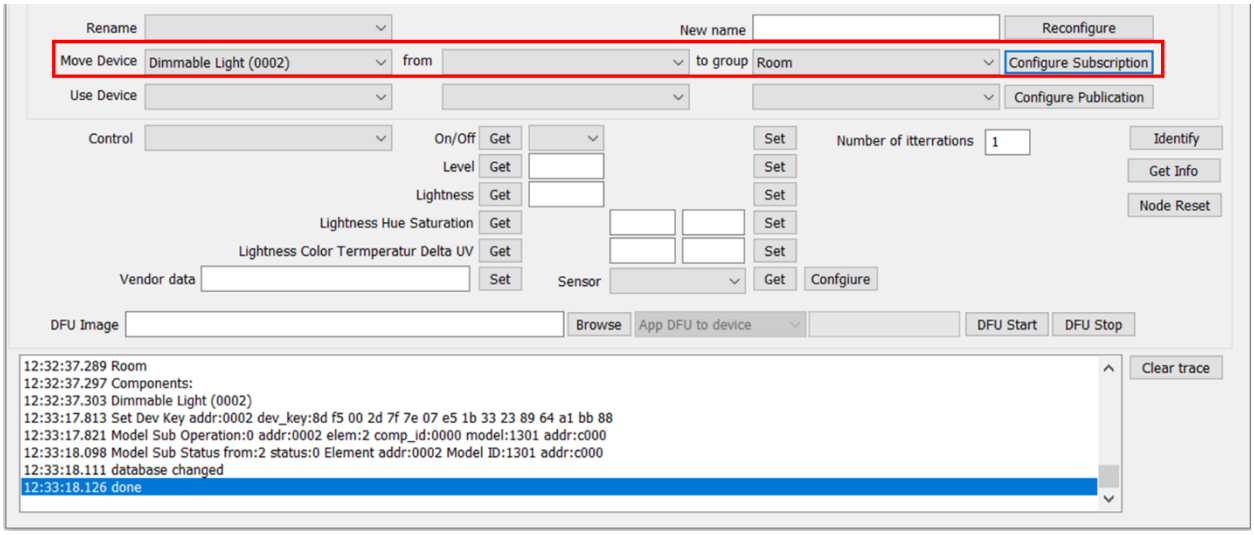
3.3 Creating and Managing Groups

A group can be used to issue commands to several devices at the same time. To create a group in the current network, type in a string in the **Group** field, and click **Create** next to it.



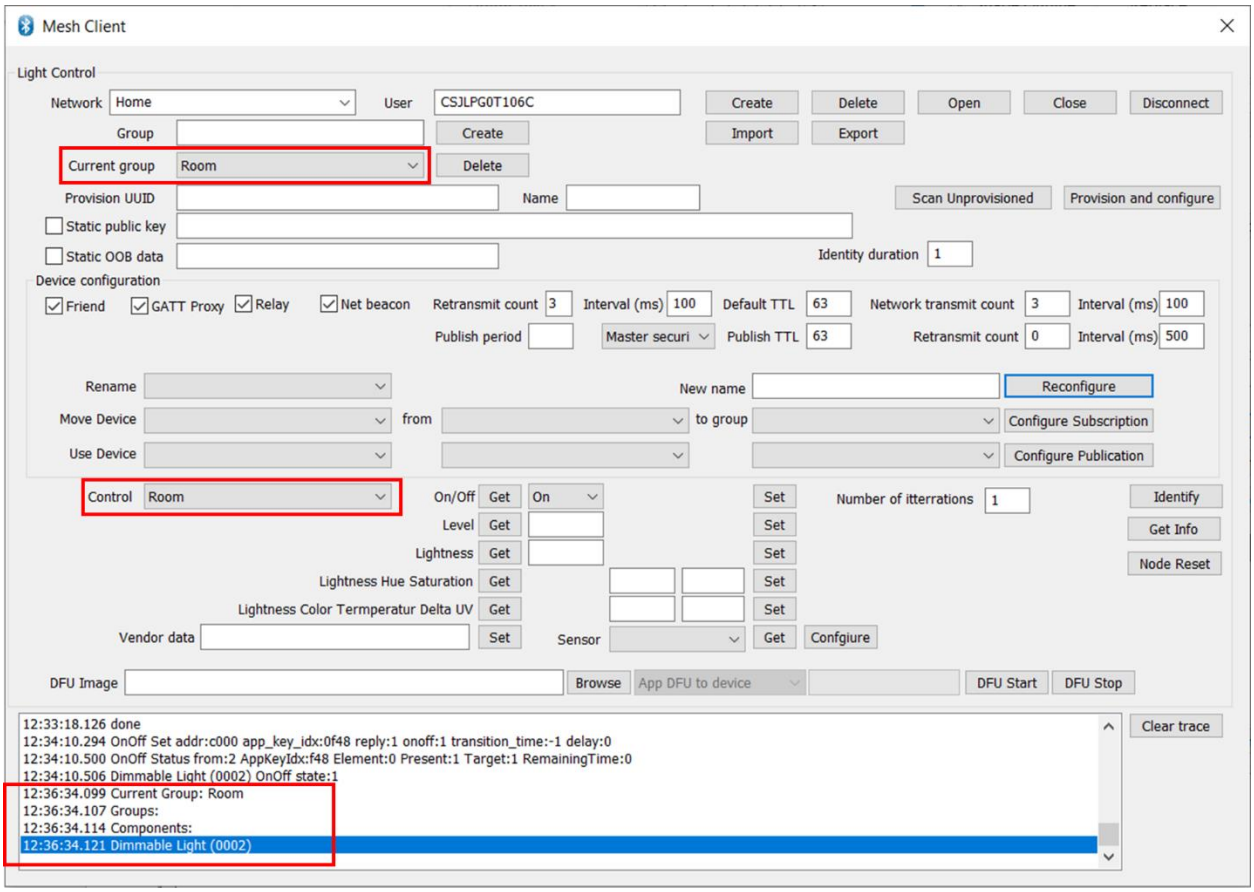
You can then select the group in the Current group field. The nodes now will be provisioned in to the created group.

You can also move devices between previously created groups. **Configure Subscriptions** puts the device in to a requested group, the device is now subscribed to process unicast messages destined to that device as well as messages addressed to the group.

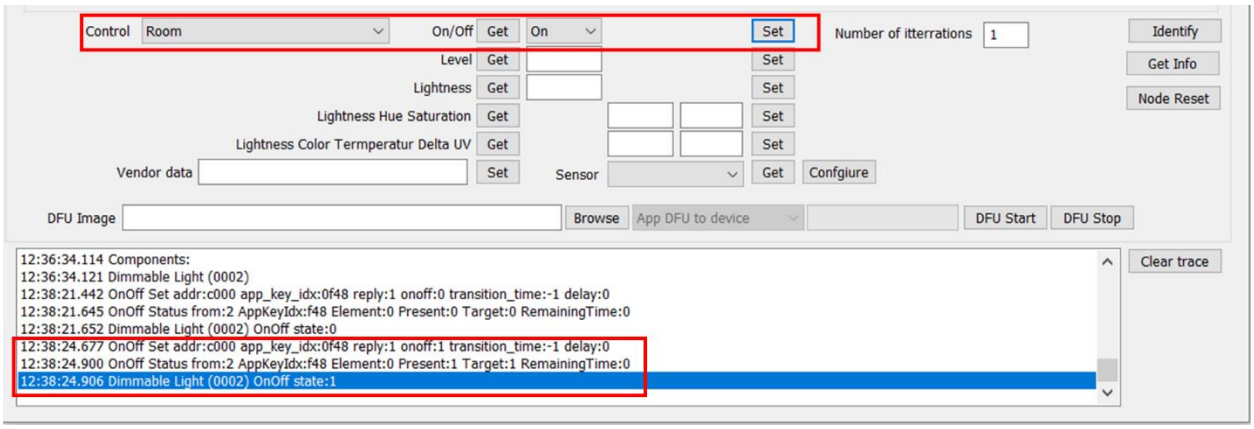


If a new device is provisioned while the current selected group is 'Room', the device will be assigned to this group. You do not need to perform the Move operation. In the Control field, you can select an individual device, a group address, or the name of the network to unicast, multicast, or broadcast mesh control messages respectively.

When the current group is changed, the trace window will display the content of the new current group.

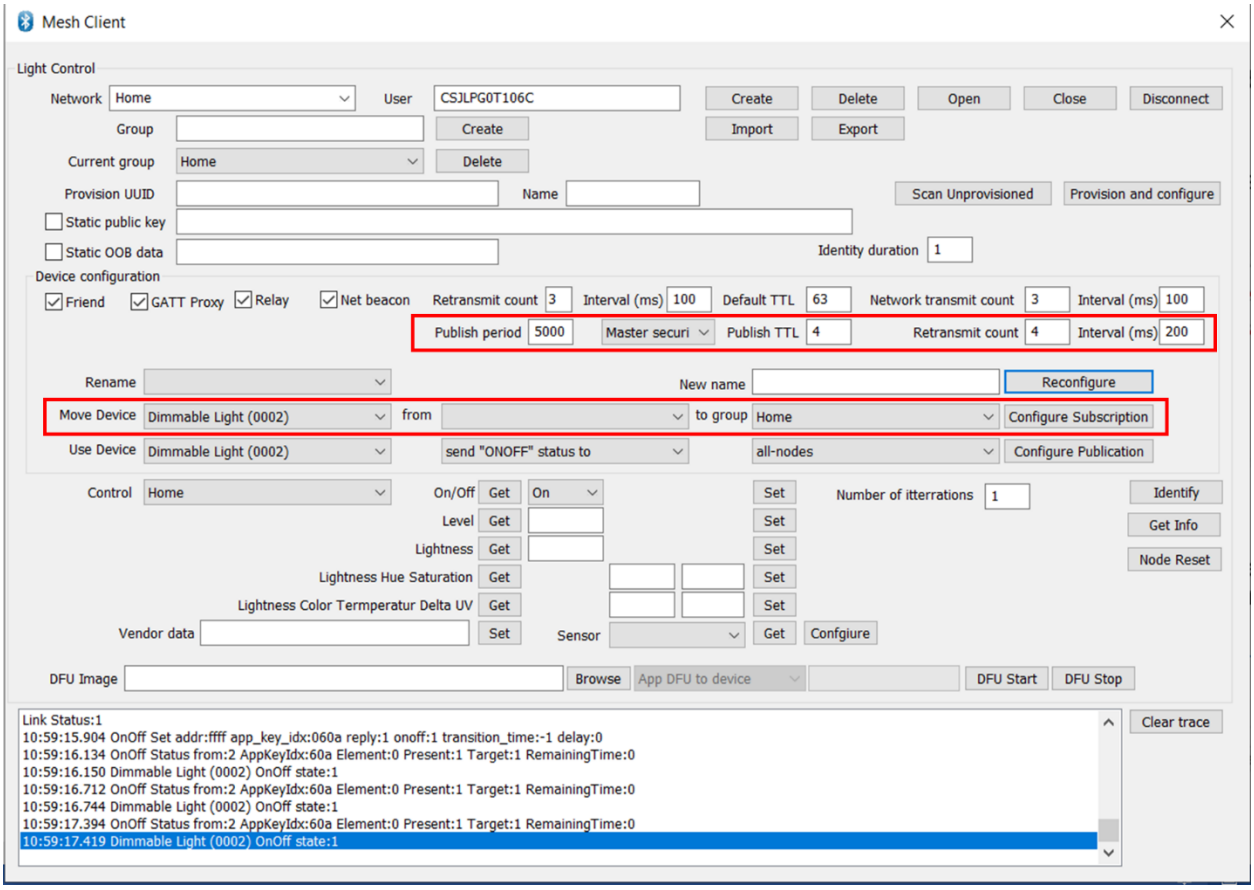


Entire group can be controlled using a single command. Select the group name to be controlled in **Control** field, select the action, and click **Set**.



3.4 Configuring Devices

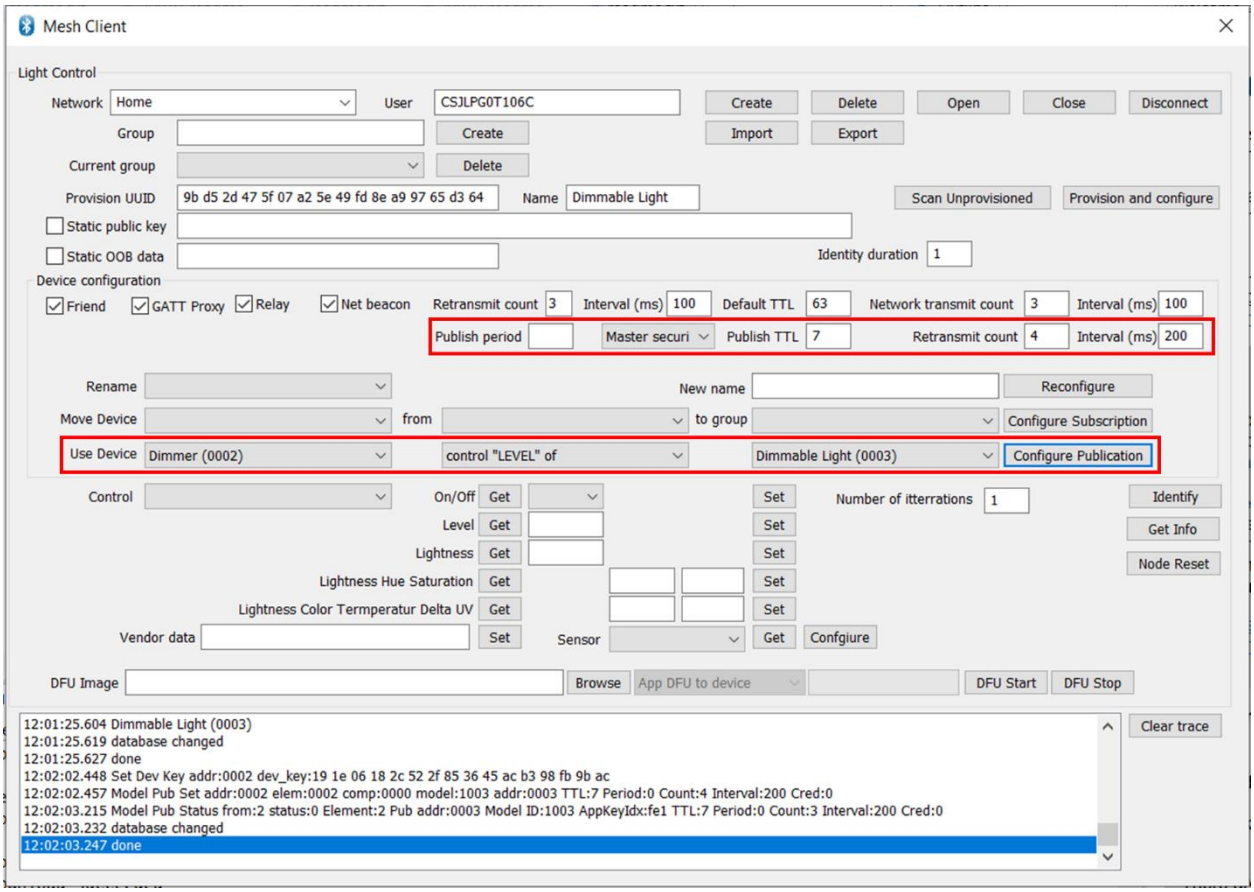
The device configuration section allows you to configure multiple parameters for sending or relaying messages. By default, parameters from the initial provisioning are used; you can specify a new name for an already configured device, select the device in the **Rename** field, and click **Reconfigure**.



A device can also be configured to publish status change to a specific node, and specific group, or to all devices in the network. For example, a light bulb can publish hue/saturation/lightness state periodically.

Similarly, a controlling device can be configured to send messages to individual devices or to a group. This allows you to configure a switch to control one or more lights.

To configure a destination for the messages originated by the device, select the device in the **Use Device** field, select the method (for example, if this is a Dimmer, a LEVEL methods will be available), desired destination, and click **Configure Publication**.



If the Dimmer was provisioned while the group room was selected as the current group, the Dimmer will already be configured to send level to all devices in the group room. However, it can be reconfigured to send messages to any single device, or to any other group in the network.

The publication parameters are used from the Publish period, Publish TTL, Retransmit count, and Interval fields.

3.5 Over-The-Air Device Firmware Upgrade

The device firmware is updated from the build PC to the development kit using the MeshClient DFU interface. Open the mesh **Network**, named **home** in Figure 2. Scan unprovisioned devices, then Provision and Configure. Then, click **Disconnect** (Connect toggles as shown in Figure 2). Next, select the device from the **Control** dropdown list and click **Connect**. Use **Browse** to select the update image and the image information file. Select **App DFU to device** from the transfer type dropdown list. Click **DFU Start** to begin the transfer.

The image information file is a plain text file formatted as follows:

```
# Company ID (2 bytes)
CID=0x0131
# Firmware ID (2 bytes Product ID + 2 bytes HW Version ID + 4 bytes FW Version)
FWID=0x3026000101010002
```

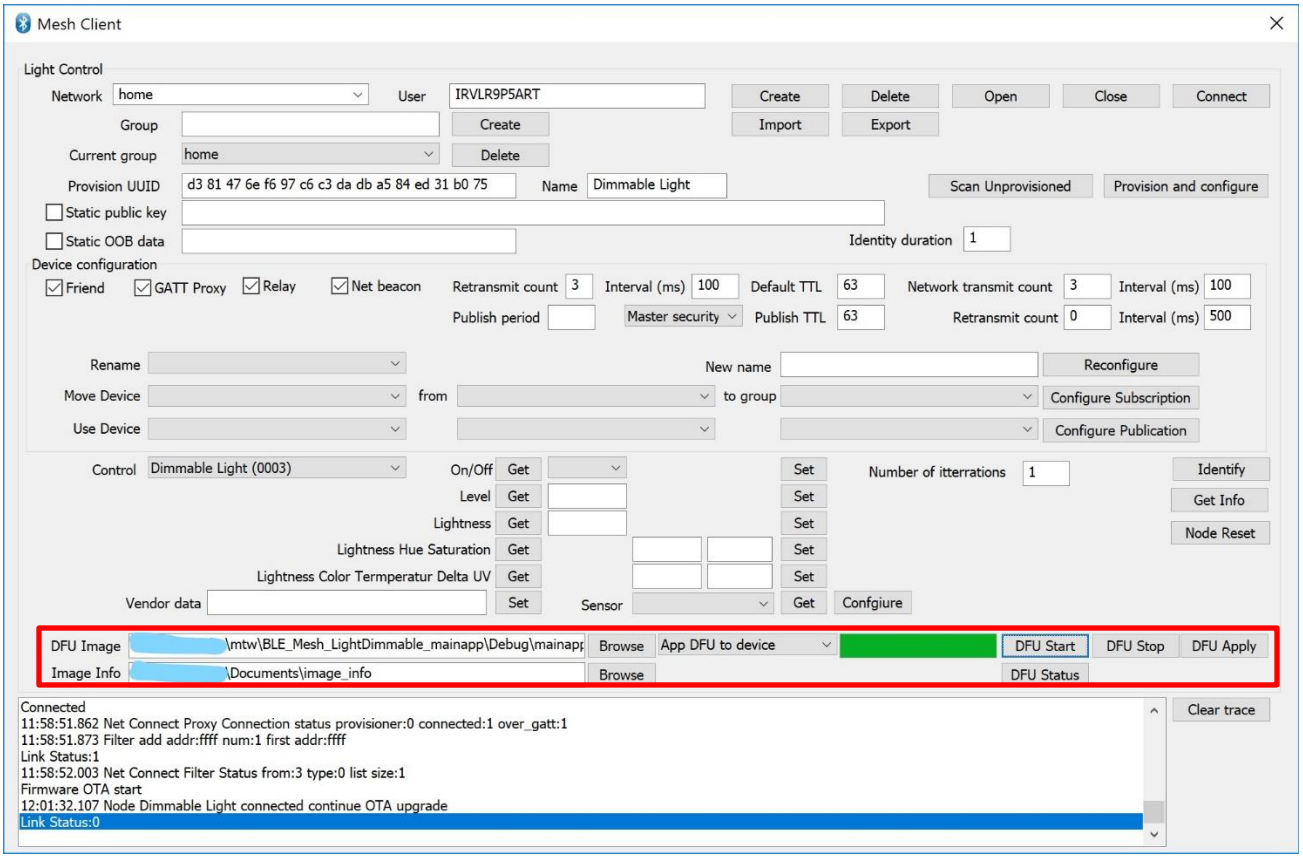


Figure 2. Mesh Client DFU

4 References

[1] Mesh Profile Specification v1.0

[2] Mesh Models Specification v1.0

[3] Mesh Provisioner Database Specification v1.0

Document Revision History

Document Title: MeshClient and ClientControlMesh App User Guide

Document Number: 002-26575

Revision	ECN	Issue Date	Description of Change
**	6489357	05/01/2018	Initial release
*A	6556129	04/24/2019	Removed Associated Part Family Updated for BT SDK release

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709