



# Using the UIC demo UI

---

This lab procedure walks through the basic steps to use the UIC on the Raspberry Pi platform. The first part of the lab is using the demo UI and adding a node. The second part is controlling the node. The final section is viewing Z-Wave traffic in the Zniffer.

## KEY POINTS

---

- Launch web UI
- Add a node
- Control a node
- Remove a node
- View Zniffer output
- View mqtt traffic

## Prerequisites

### 1 Prerequisites

For this lab you will need the following:


- Raspberry Pi 4
  - SD card for Raspberry Pi 4
- UZB 7 serial dongle for Z-Wave
- 1 WSTK Radio Development Board and ZG Module



- 1 IP router with built-in DHCP
- VNC viewer <https://www.realvnc.com/en/connect/download/viewer/>
- To have completed Lab 1 (Basic set up)
- Simplicity Studio
- Z-Wave Zniffer
- Z-Wave 7 SDK

#### 1.1 Open the Web UI

1. Make sure the UZB7 is connected to the Raspberry Pi USB port
2. Launch a web browser on your PC and connect to <http://raspberrypi.local:3000>
3. In the top right click the connect button
4. Make sure the node list shows the controller node



**SILICON LABS**  
Unified IoT Controller

**UIC MQTT broker connection**

Host address  Port number

Status **Connected** Disconnect

- Nodes
- Smart Start
- Door Locks
- On/Offs
- Binary Sensors
- Thermostats

### Node List

Find DSK key:

Unid	Type	Status	Security	Max Delay	State	
zw-C32C4FBC-0001	🔌	INCLUDED	Z-Wave S2 Access	0	idle	<span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 2px;">Add</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 2px;">Remove</span> <span style="border: 1px solid #ccc; padding: 2px 5px; margin-right: 2px;">Reset</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">Idle</span>

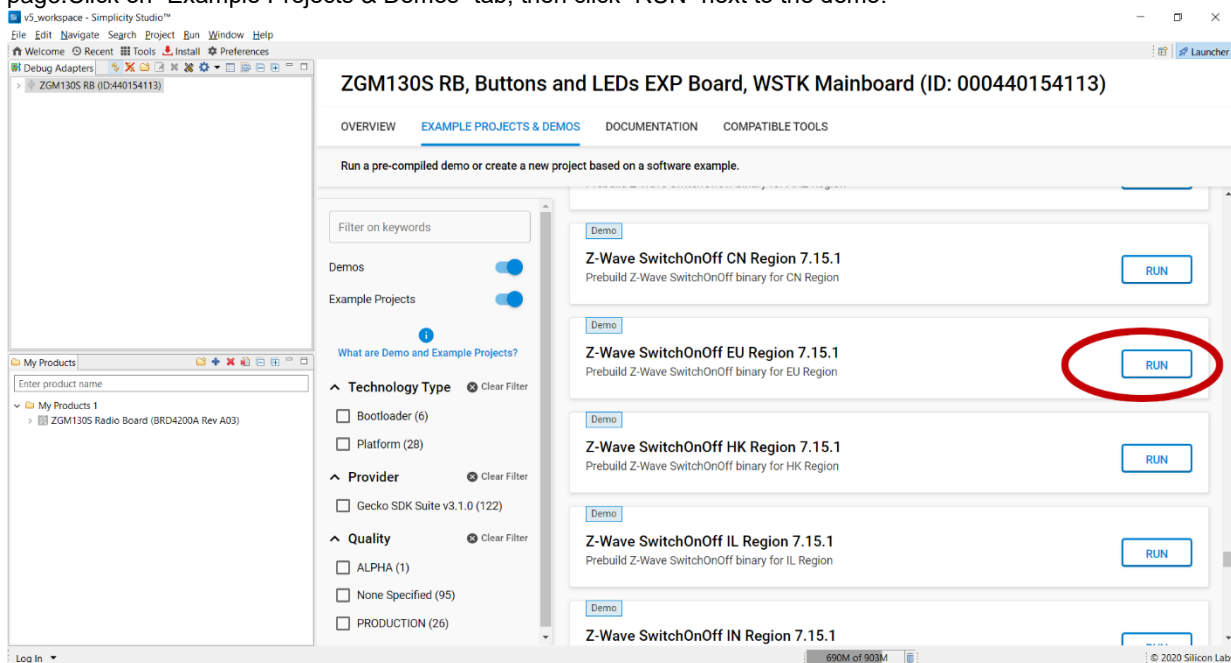
#### 1.2 Program WSTK board with Switch on/off sample application

The following section details how to program the WSTK

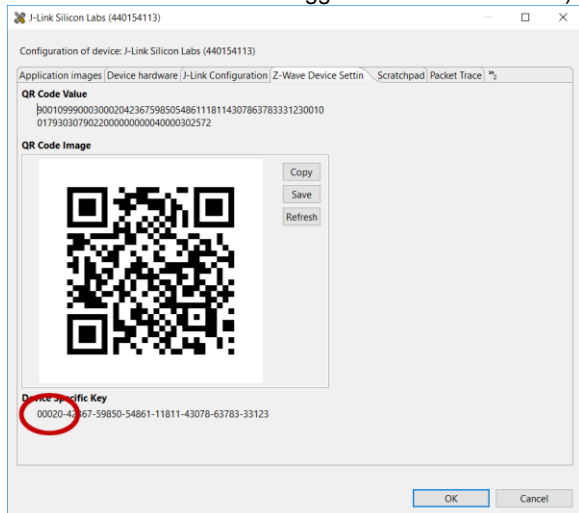
1. Connect the WSTK USB port to your PC

## Prerequisites

- Open Simplicity Studio and click on the ZGM130S in the debug adapters tab. If you don't have the Z-Wave SDK installed in Simplicity Studio, do so now.
- Then load the Z-Wave SwitchOnOff (in this case, EU Region) demo. Make sure to select the same region here that you chose in the first lab. If simplicity studio doesn't show this home screen, click on the welcome button to get back to the main page. Click on "Example Projects & Demos" tab, then click "RUN" next to the demo.





- Then right click on the ZGM130S under the debug adapters tab and choose "Device Configuration..."
- Under the "Z-Wave Device Settings" tab there is a QR code. Underneath the QR code is the Device Specific Key (you may need to make the window bigger for this to be visible). Make a note of the first five digits of the Device Specific Key.



### 1.3 Launch Zniffer to view Z-Wave traffic

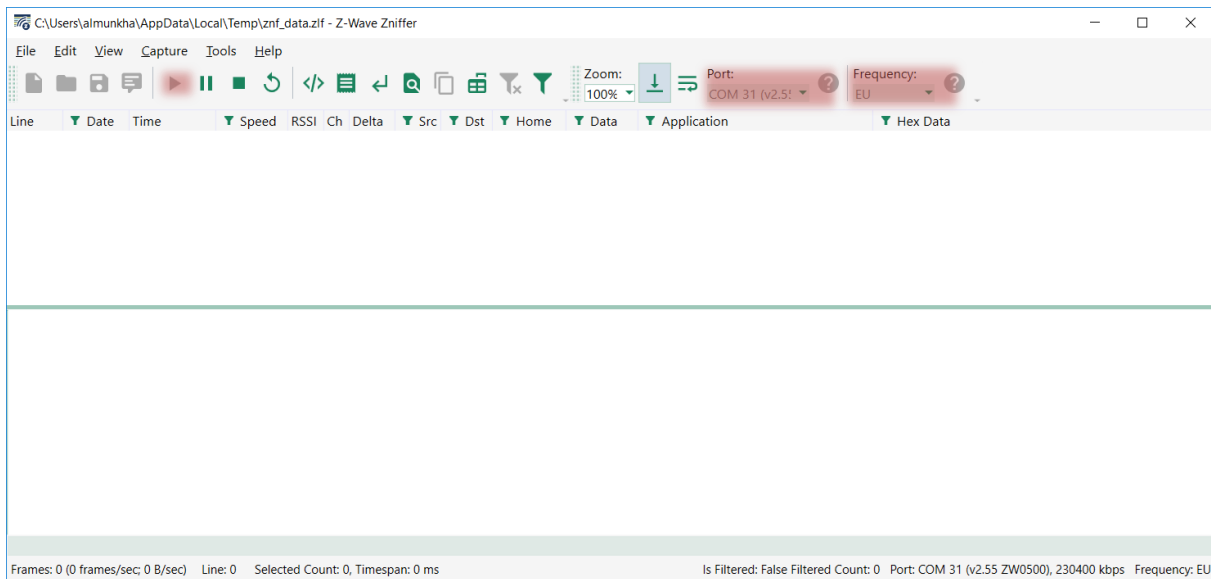
The following section details how to launch the Zniffer to view the add node traffic

- Connect the Zniffer USB device to your PC
- In Simplicity Studio, open the Z-Wave Zniffer tool from Simplicity Studio Tools menu.
- Select the COM port by clicking on the icon  next to Port. This will search for connected Zniffers.
- Select the COM port in the drop down menu.
- Select the Frequency by clicking on the icon  next to Frequency.
- Select the Frequency that matches the frequency you programmed your device with. These exercises uses "EU".

## Prerequisites



### 7. Start a new Trace .

The Ziffer tool is now ready and listening for Z-Wave RF Traffic.



**Ziffer setup and listening for Z-Wave traffic**

You might see other traffic, despite you are not sending any to your device. Currently, the Ziffer is configured for listening for any traffic in the selected region. We need to filter the trace by “Home ID” to only show traffic from your network.

1. From the UIC demo UI, find the “Home ID” in as the 8 digit hex code following the “zw-“ prefix for the controller node.
  - o In the previous figure of the demo UI, the example HomeID is C3 2C 4F BC.
2. Knowing the Home ID, click the small icon  next to “Home” in the Ziffer. Refer to **Error! Reference source not found..**
3. Write the Home ID in this field and click OK.
4. The green icon now turns red  symbolizing the Ziffer is filtering on Home Id.

#### 1.4 Subscribe to the mqtt messages

The following section details how to view the mqtt traffic

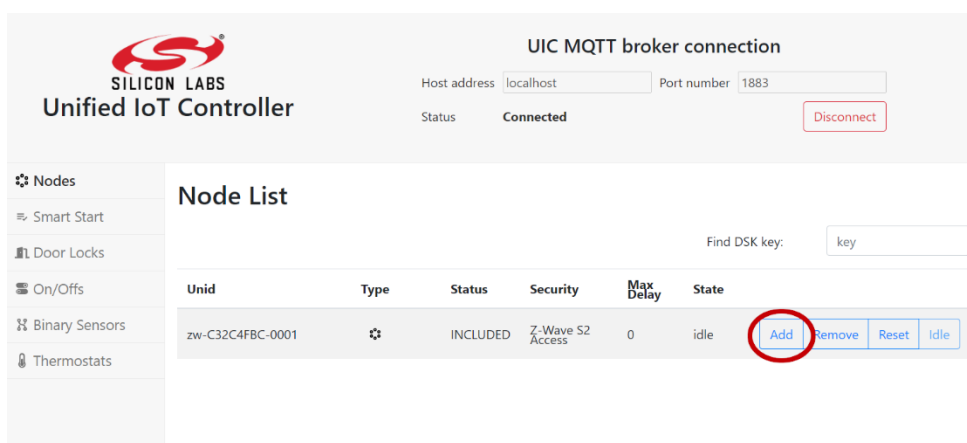
1. Open a new command line window on the Raspberry Pi
2. Type `mosquitto_sub -v -h localhost -t "ucl/#"`
3. You will now be able to see all UIC related mqtt traffic in this window

#### 1.5 Go Add the device as a node in the UIC

The following section details how to add a node using the demo UI

1. Go back to the demo UI <http://raspberrypi.local:3000>
2. Click add next to the controller node

## Prerequisites

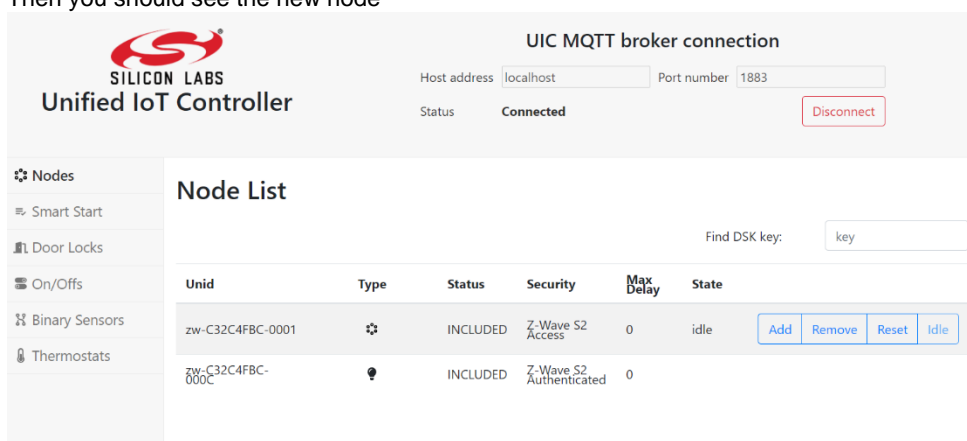


The screenshot shows the UIC MQTT broker connection interface. At the top, the status is "Connected" with a "Disconnect" button. Below, the "Node List" table contains one entry:

Unid	Type	Status	Security	Max Delay	State	
zw-C32C4FBC-0001		INCLUDED	Z-Wave S2 Access	0	idle	<a href="#">Add</a> <a href="#">Remove</a> <a href="#">Reset</a> <a href="#">Idle</a>

The "Add" button in the table is circled in red.

3. Then press BTN1 on the WSTK to start inclusion mode
4. Complete the DSK information when asked
5. Then you should see the new node



The screenshot shows the UIC MQTT broker connection interface after a second node has been added. The "Node List" table now contains two entries:

Unid	Type	Status	Security	Max Delay	State	
zw-C32C4FBC-0001		INCLUDED	Z-Wave S2 Access	0	idle	<a href="#">Add</a> <a href="#">Remove</a> <a href="#">Reset</a> <a href="#">Idle</a>
zw_C32C4FBC-000C		INCLUDED	Z-Wave S2 Authenticated	0		

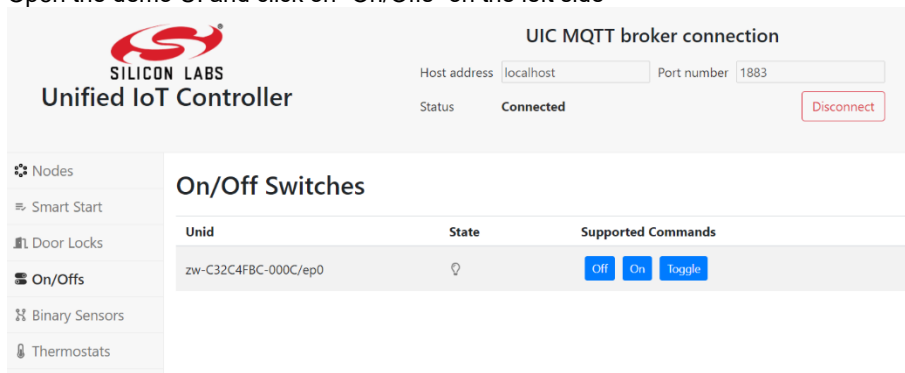
6. In the Raspberry Pi locate the security keys to be able to decrypt the zniiffer traffic
  - a. Type `journalctl | grep zpc | grep key` at the command line
  - b. When you click "Decrypt" in the Zniiffer use the keys you found in the previous step
  - c. Note. If decryption fails try power cycling the node, and toggling the LED etc, to resync the nonce
7. View the mqtt traffic in the command line window you opened earlier

## 2 Controlling a node

### 2.1 Controlling a node from the demo UI

The following section details how to turn on/off the switch example we just added to the network

1. Open the demo UI and click on “On/Offs” on the left side



2. Push the on, off, toggle buttons and you will see LED0 on the board change state
3. Go to the Zniiffer and view the Z-Wave traffic while you do this
4. Go to the command line window on the Raspberry Pi to view the mqtt traffic

### 2.2 Controlling a node from the mosquitto\_pub client

1. Subscribe to all topics by running mosquitto\_sub in the background:

```
mosquitto_sub -v -h localhost -t "ucl/#" &
```

2. Publish the toggle command seen in step 2.1

```
mosquitto_pub -h localhost -t "ucl/by-unid/zw-D581F19A-0004/ep0/OnOff/Commands/Toggle" -m "{}"
```

3. Observe the Z-Wave traffic with the Zniiffer

4. Evaluate MQTT being published via the mosquitto\_sub client:
 

```
ucl/by-unid/zw-DE2E84C0-0002/ep0/OnOff/Commands/Toggle {}
ucl/by-unid/zw-DE2E84C0-0002/ep0/OnOff/Attributes/OnOff/Desired {
  "value": false
}
```

[...]

```
ucl/by-unid/zw-DE2E84C0-0002/ep0/OnOff/Attributes/OnOff/Reported {
  "value": false
}
```

[...]

### 3 Removing a node

In this section we remove the node we just added

#### 3.1 Open web interface

Launch a web browser on your PC and connect to <http://raspberrypi.local:3000>

1. Go to the nodes area and click remove next to the controller node
2. Press the inclusion/exclusion button on the WSTK (BTN1)
3. View the traffic in the Zniffer
4. View the mqtt traffic in the command line window

## 4 Evaluating UIC logs

The UIC reference uses journalctl to handle logging, to evaluate a specific process use the following command:

```
journalctl -u <service name>
```

the following services can be evaluated:

```
uic-dev-gui-api.service      # the high-level 'middleware' api driver
uic-dev-gui-server.service  # the local gui server
uic-upvl.service            # the security provisioning list
uic-zpc.service             # the z-wave protocol controller
```