AI-01881 Home Assistant Zigbee2MQTT

Owner	Ⓡ Rigel Arcayan
Tags	Project
Created time	@July 26, 2025 12:12 AM

I have a Zigbee3.0 Setup at home and running with Scenes via eWeLink App with the following devices:

ZBBridge-P-1 pc

SNZB-06P-6 pcs

ZBMINIL2-2 pcs

ZBMINIR2-4 pcs

I would like to migrate to Home Assistant using Zigbee2MQTT. I already have a working HA via a Hyper-V setup in my notebook pc.

Provide a step by step guide from start to finish.

Migrating your Zigbee setup from eWeLink to Home Assistant with Zigbee2MQTT is a great move for more local control and flexibility. Here's a step-by-step guide to help you through the process:

Important Considerations Before You Start:

- **You will need a dedicated Zigbee Coordinator.** Your existing ZBBridge-P is likely flashed with custom firmware (e.g., Tasmota) to work with eWeLink locally, or it's tied to the eWeLink cloud. To use Zigbee2MQTT, you'll generally need a dedicated Zigbee USB dongle (like the SONOFF Zigbee 3.0 USB Dongle Plus-P or -E, or another compatible coordinator). While some attempts have been made to use the ZBBridge with Zigbee2MQTT remotely, a direct USB dongle is generally more reliable and recommended for a stable Zigbee2MQTT setup.

- **Re-pairing is almost always required.** Unfortunately, there's no direct migration tool to transfer your devices and their network keys from the eWeLink hub to a new Zigbee2MQTT coordinator without re-pairing each device. This means you'll need to reset and pair each of your SNZB-06P, ZBMINIL2, and ZBMINIR2 devices individually.
- **Plan your migration.** If you have critical automations, consider doing this in phases or at a time when you can tolerate some downtime for your smart home devices.

Here's the step-by-step guide:

Phase 1: Prepare Home Assistant and Install Necessary Add-ons

1. **Ensure Home Assistant is running:** You mentioned you have a working HA via Hyper-V. Make sure it's up and accessible.
2. **Install an MQTT Broker (Mosquitto):** Zigbee2MQTT communicates with Home Assistant via MQTT.
    - In Home Assistant, go to **Settings > Add-ons**.
    - Click on **Add-on Store** (bottom right corner).
    - Search for "Mosquitto broker" and click on it.
    - Click **Install** and wait for it to complete.
    - Once installed, go to the **Configuration** tab for Mosquitto. You likely won't need to change much here for a basic setup.
    - Go to the **Info** tab and click **Start**.
    - Enable "Start on boot" and "Watchdog" for reliability.
3. **Install Zigbee2MQTT:**
    - In Home Assistant, go to **Settings > Add-ons > Add-on Store**.
    - Click the three dots in the top right corner and select **Repositories**.
    - Add the following repository URL: `https://github.com/zigbee2mqtt/hassio-zigbee2mqtt` and click **Add**.
    - Close the repository window. You might need to refresh the Add-on Store page.

- Search for "Zigbee2MQTT" and click on it.
- Click **Install** and wait for it to complete.

Phase 2: Configure Zigbee2MQTT

1. **Identify your Zigbee Coordinator's Serial Path:**
    - Connect your new Zigbee USB dongle (e.g., SONOFF ZBDongle-P/E) to your host PC where Hyper-V is running.
    - You'll need to pass this USB device through to your Home Assistant Hyper-V virtual machine. The exact steps for Hyper-V USB passthrough can vary, but generally involve:
        - Disabling "Enhanced Session Mode" for the VM if it's enabled.
        - Going to the Hyper-V Manager, right-clicking your Home Assistant VM, and selecting **Settings**.
        - Under "Hardware" > "SCSI Controller" (or similar), click "Add Hardware" and select "COM Port" or "Physical Disk" depending on how Hyper-V exposes USB devices. You might need to use a tool like USB/IP for more reliable USB passthrough in Hyper-V if direct COM port mapping isn't sufficient.
        - Once passed through, you need to find the serial path *within* the Home Assistant OS.
        - Go to **Settings > System > Hardware** in Home Assistant.
        - Click on **All Hardware**. Look for entries like `/dev/ttyUSB0`, `/dev/ttyACM0`, or `/dev/serial/by-id/usb-xxxx_your_dongle_name-if00-port0`. This is the path you'll use in Zigbee2MQTT. The `by-id` path is preferred as it's more persistent.
2. **Configure Zigbee2MQTT Add-on:**
    - In Home Assistant, go to **Settings > Add-ons > Zigbee2MQTT**.
    - Go to the **Configuration** tab.
    - Under the `serial:` section, you'll need to specify your adapter type and the port.
        - **For SONOFF ZBDongle-P:**YAML

```
serial:
 adapter: zstack
 port: /dev/ttyUSB0 # Or your actual serial path, e.g., /dev/serial/by-id/usb-ITEAD_SONOFF_Zigbee_3.0_USB_Dongle_Plus_XXXXXXX-if00-port0
```

- **For SONOFF ZBDongle-E:**YAML

```
serial:
 adapter: ember
 port: /dev/ttyUSB0 # Or your actual serial path
```

- **Important:** Replace `/dev/ttyUSB0` with the actual path you found in step 1.

- Under the `mqtt:` section, ensure the `broker` is set to your Home Assistant IP (or `mqtt://core-mosquitto` if using the HA Add-on, which is usually default), and `base_topic` is `zigbee2mqtt`.YAML

```
mqtt:
 broker: mqtt://core-mosquitto
 base_topic: zigbee2mqtt
 # optional: user: your_mqtt_username
 # optional: password: your_mqtt_password
```

- Click **Save**.

3. **Start Zigbee2MQTT:**
   - Go back to the **Info** tab of the Zigbee2MQTT add-on.
   - Click **Start**.
   - Enable "Start on boot" and "Watchdog".
   - Wait a minute or two for it to fully start.
   - Click **Open Web UI**. This will open the Zigbee2MQTT frontend, which is invaluable for managing your Zigbee network.

Phase 3: Reset and Pair Your Zigbee Devices

This is the most time-consuming part. You'll need to factory reset each device and then pair it with your new Zigbee2MQTT network.

1. **Enable Permitted Joining in Zigbee2MQTT:**
   - In the Zigbee2MQTT Web UI, go to **Settings > Tools**.

- Toggle **Permit join (all)** to `On`. This will allow new devices to join your network for a few minutes. You can also permit joining for a specific amount of time.

2. **Reset and Pair Each Device:**
    - **SNZB-06P (Human Presence Sensor):**
        - Typically, to reset, press and hold the pairing button for 5 seconds until the LED flashes. Then release. It should start searching for a network.
        - Check the Zigbee2MQTT Web UI. Once discovered, it will appear.
        - You can then rename it if desired and configure its specific settings (like sensitivity, occupancy timeout) from the device page in Zigbee2MQTT.
    - **ZBMINIL2 (No Neutral Wire Smart Switch):**
        - There are two main ways to reset:
            - **Button:** Press and hold the physical button on the ZBMINIL2 for 5 seconds until the LED flashes.
            - **Switch Toggle:** If an external switch is connected, toggle it 10 times consecutively.
        - The device will remain in "OFF" state while in pairing mode. It will exit pairing mode after 180 seconds if not successfully paired.
        - Once paired, you can control its state (ON/OFF/TOGGLE) and configure power-on behavior via Zigbee2MQTT.
    - **ZBMINIR2 (Smart Switch):**
        - Similar to ZBMINIL2:
            - **Button:** Press and hold the device button for 5 seconds until the LED flashes.
            - **Power Cycle:** If not easily accessible, you can try powering the device off for 7 seconds, then on for 1 second, and repeat this 4 times. After the 4th cycle, power it off again for at least 7 seconds. When you power it back on, it should be in pairing mode.

    - Once paired, you can control its state and set power-on behavior, network indicator, and turbo mode through Zigbee2MQTT.

3. **Name Your Devices:** As each device joins Zigbee2MQTT, it will appear with a default name (e.g., `0x001234...`). It's highly recommended to give them user-friendly names in the Zigbee2MQTT Web UI (e.g., "Living Room Light," "Hallway Motion Sensor"). Ensure "Update Home Assistant entity ID" is toggled on when renaming in Zigbee2MQTT to reflect the changes in HA.

Phase 4: Integrate with Home Assistant

Zigbee2MQTT automatically creates Home Assistant entities for your paired devices via MQTT discovery.

1. **Check Home Assistant Devices & Entities:**
    - In Home Assistant, go to **Settings > Devices & Services**.
    - You should see a new integration for **MQTT**. If not, you might need to add it (it should auto-discover if Mosquitto is running).
    - Under the MQTT integration, you will find all your Zigbee2MQTT devices and their associated entities (switches, sensors, etc.).
2. **Verify Functionality:**
    - Test controlling your ZBMINIL2 and ZBMINIR2 switches from Home Assistant.
    - Check if your SNZB-06P sensors are reporting presence correctly.

Phase 5: Recreate Scenes and Automations

1. **Review your eWeLink scenes:** Make a list of all your scenes and automations from the eWeLink app.
2. **Recreate in Home Assistant:**
    - Go to **Settings > Automations & Scenes > Automations** or **Scenes** in Home Assistant.
    - Use the Home Assistant automation and scene editors to recreate the logic you had in eWeLink. You'll now use your newly integrated Zigbee devices as triggers and actions.

- For example, if you had a scene that turned on a light when the SNZB-06P detected presence, you would create an automation in Home Assistant:
    - **Trigger:** `State` of `binary_sensor.your_snzb06p_occupancy` changes from `off` to `on`.
    - **Action:** `Call service` > `Light: Turn on` (or `Switch: Turn on`) for your ZBMINIL2/ZBMINIR2.
- Home Assistant's flexibility allows for much more complex and powerful automations than most proprietary apps.

Troubleshooting Tips:

- **Device not pairing:**
    - Ensure "Permit join" is enabled in Zigbee2MQTT.
    - Double-check the device's factory reset procedure. Some devices require specific timings or repetitions.
    - Bring the device closer to your Zigbee coordinator during pairing.
    - Consider any potential 2.4GHz Wi-Fi interference if your Zigbee network is on the same channel.
- **Device showing as "offline" or not updating:**
    - Check the Zigbee2MQTT logs (in the Zigbee2MQTT add-on, Info tab).
    - Ensure the device has a strong signal (check LQI/RSSI in Zigbee2MQTT Web UI). If it's a battery-powered device, consider placing it closer to a mains-powered Zigbee router (like your ZBMINIL2/ZBMINIR2, once they are paired and acting as routers).
    - Power cycle the device.
- **Home Assistant not showing devices:**
    - Restart the Mosquitto broker and Zigbee2MQTT add-ons.
    - Restart Home Assistant.
    - Check the Home Assistant logs for any MQTT-related errors.
    - Verify your MQTT configuration in Zigbee2MQTT.

By following these steps, you should successfully migrate your Sonoff Zigbee devices to Home Assistant using Zigbee2MQTT, gaining full local control and integrating them into your broader smart home ecosystem. Good luck!

Compare side by side Zigbee 3.0 USB Dongle Plus-P or -E, which one to choose?

When choosing between the SONOFF Zigbee 3.0 USB Dongle Plus-P and -E, it comes down to a trade-off between **maturity/stability** and **newer technology/potential future compatibility**.

Here's a side-by-side comparison:

Feature	SONOFF Zigbee 3.0 USB Dongle Plus-P (ZBDongle-P)	SONOFF Zigbee 3.0 USB Dongle Plus-E (ZBDongle-E)
Chipset	Texas Instruments CC2652P	Silicon Labs EFR32MG21
Driver (Z2M)	`zstack`	`ember`
Maturity/Stability	**Highly mature and widely recommended.** Has been the "gold standard" for a long time. Very stable with Zigbee2MQTT and ZHA.	Newer chipset. Initially had "experimental" support in Zigbee2MQTT, but has become much more stable.
Performance	Excellent range and network capacity (up to 200 children total).	Excellent range and network capacity (can support more direct devices, up to 300 in some firmwares). Often cited as having slightly better raw performance.
Firmware Updates	Generally straightforward, typically via a flashing tool.	Can be updated over the web (OTA) with the correct firmware, which is convenient.
Matter/Thread Compatibility	**Not designed for native Matter/Thread.** While some chips can support one or the other, it typically can't do both simultaneously. You'd likely need a separate dongle for Matter/Thread.	**More likely to be future-proof for Matter/Thread.** The EFR32MG21 chip is more modern and capable of supporting these protocols, often simultaneously with Zigbee (though usually requires specific multi-protocol firmware).

Availability	Can be harder to find due to chip shortages affecting the CC2652P.	More readily available as it uses a more current chipset.

Which one to choose?

- **For maximum stability and proven performance with Zigbee2MQTT/ZHA *right now*:**
    - **Go with the ZBDongle-P.** It's a rock-solid choice that has been the community favorite for a reason. If you prioritize "set it and forget it" reliability for your Zigbee network, the P is still a fantastic option.
- **For a slightly newer chipset, easier firmware updates, and potential future Matter/Thread compatibility (even if you don't use it now):**
    - **Consider the ZBDongle-E.** While it had some initial growing pains with Zigbee2MQTT, most of those have been resolved, and it's now considered very stable. The EFR32MG21 chip is a powerful and capable chip that offers more future-proofing.

Recommendations for your specific situation (migrating from eWeLink):

Given you're migrating to Home Assistant with Zigbee2MQTT, both are good choices.

- If you can **easily find a ZBDongle-P**, it's a very safe and reliable bet that will likely give you the smoothest experience from the start.
- If the **ZBDongle-P is hard to find or significantly more expensive**, don't hesitate to get the **ZBDongle-E**. Its performance is excellent, and its support in Zigbee2MQTT has matured significantly. You'll gain the benefit of a newer chipset and potentially easier firmware updates down the line.

General advice for any Zigbee dongle:

- **Always use a USB extension cable:** This helps reduce interference from your computer's USB port and improves Zigbee signal quality.
- **Place it strategically:** Try to place your dongle in a central location, not crammed behind other electronics, to optimize your Zigbee mesh network.

Ultimately, both are capable Zigbee coordinators for Home Assistant and Zigbee2MQTT. The "P" is the battle-tested veteran, while the "E" is the powerful, increasingly stable newcomer with a brighter future in terms of multi-protocol support.