

# LoRa Device Configuration & AT Command Automation — Documentation

## System Overview

This setup automates the generation and management of LoRa RYLR896 module configurations by:

- Assigning unique `network_id` and `address` values
- Storing configs in a structured `JSON` file
- Printing ready-to-use `AT` command strings
- Saving script progress to resume generation later

---

## Files Structure

File Name	Description
<code>lora_nodes.json</code>	Stores all generated device entries (TX/RX)
<code>state.json</code>	Stores the index pointer of the last served device pair
<code>generate_devices.py</code>	Script to generate all TX/RX device pairs and save to JSON
<code>get_next_pair.py</code>	Script to output the next TX/RX pair with AT commands

---

## JSON Format (`lora_nodes.json`)

Each entry represents a LoRa module configuration:

```
{
  "device_id": 1,
  "role": "transmitter",
  "network_id": 1,
  "address": 1,
  "password": "LoRaSecure1234",
  "band": "868000000"
}
```

---

## Script 1: Generate All Devices (`generate_devices.py`)

**Purpose:** Create all device entries for all `network_ids` from 1 to 16, with addresses 1 to 65534. Each TX/RX pair is assigned sequential addresses within a network.

**Command to run:**

```
python3 generate_devices.py
```

---

It saves the data to `lora_nodes.json` safely without duplicating entries if they already exist.

---

## ⚙️ Script 2: Get Next Pair (`get_next_pair.py`)

**Purpose:** On each run:

- Outputs the next available TX/RX device pair
- Prints AT commands for each
- Updates `state.json` to track progress

**Command to run:**

```
./get_next_pair.py
```

**What it prints:**

- Device JSON info
- Corresponding `AT` command strings:

```
AT+BAND=868000000
AT+NETWORKID=1
AT+ADDRESS=1
AT+CPIN="LoRaSecure1234"
```

---

## 💾 State Management

PROF

File	Key	Purpose
<code>state.json</code>	<code>current_index</code>	Index of next unused TX/RX pair

If `state.json` doesn't exist, it starts from the beginning.

To **reset the state** and start from the beginning:

```
rm state.json
```

---

## 🔑 Default Configuration Values

Field	Value
Band	<code>868000000</code>

Field	Value
CPIN	"LoRaSecure1234"
NetworkID Range	1-16
Address Range	1-65534 (per NetworkID)

## Example Output



Next Device Pair:

-----

### ◆ Transmitter Info:

```
{
  "device_id": 1,
  "role": "transmitter",
  "network_id": 1,
  "address": 1,
  "password": "LoRaSecure1234",
  "band": "868000000"
}
```

### ✂ Transmitter AT Commands:

```
AT+BAND=868000000
AT+NETWORKID=1
AT+ADDRESS=1
AT+CPIN="LoRaSecure1234"
```

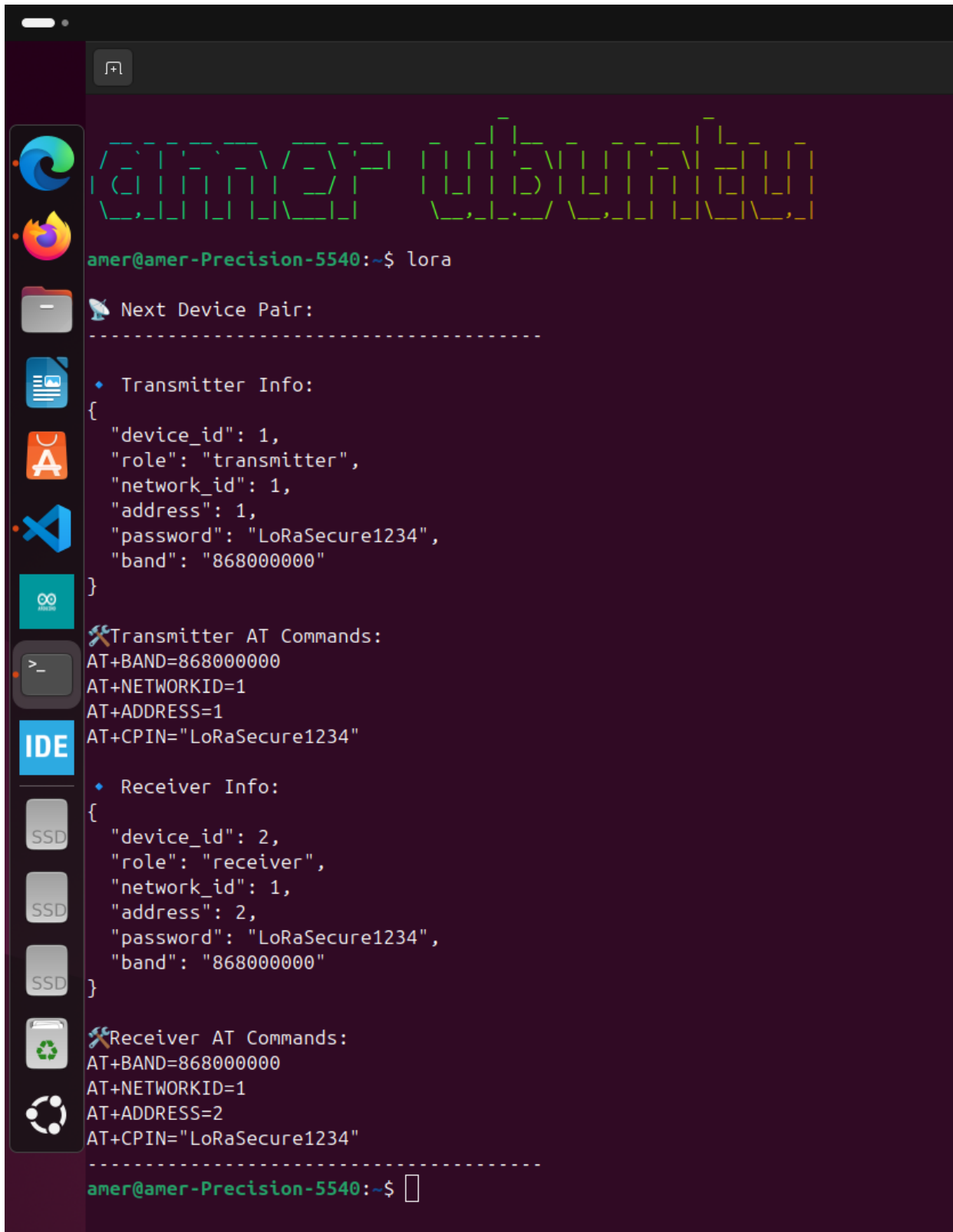
## ✓ Advantages

- No address or network reuse
- Easy recovery using state tracking
- No duplication or overwriting
- Portable and readable JSON
- Command-ready AT strings

## ⚙ linux like util **lora**

- in any place run **lora** CMD in terminal it will generate node data
- generate the TX /RX node data

- avoiding Repetition **uni node data**



A terminal window on a Linux system with a dark purple background. At the top, the text 'LoRa Developer' is displayed in a stylized, pixelated font. The user 'amer' is at the prompt 'amer@amer-Precision-5540:~\$' and has entered the command 'lora'. The output shows the configuration for the next device pair, starting with 'Next Device Pair:'. It details the 'Transmitter Info' with fields: device\_id (1), role (transmitter), network\_id (1), address (1), password (LoRaSecure1234), and band (868000000). Below this, it lists 'Transmitter AT Commands': AT+BAND=868000000, AT+NETWORKID=1, AT+ADDRESS=1, and AT+CPIN="LoRaSecure1234". Then, it shows 'Receiver Info' with fields: device\_id (2), role (receiver), network\_id (1), address (2), password (LoRaSecure1234), and band (868000000). Finally, it lists 'Receiver AT Commands': AT+BAND=868000000, AT+NETWORKID=1, AT+ADDRESS=2, and AT+CPIN="LoRaSecure1234". The terminal ends with the prompt 'amer@amer-Precision-5540:~\$' and a cursor.

```
amer@amer-Precision-5540:~$ lora

Next Device Pair:
-----

♦ Transmitter Info:
{
  "device_id": 1,
  "role": "transmitter",
  "network_id": 1,
  "address": 1,
  "password": "LoRaSecure1234",
  "band": "868000000"
}

⚡ Transmitter AT Commands:
AT+BAND=868000000
AT+NETWORKID=1
AT+ADDRESS=1
AT+CPIN="LoRaSecure1234"

♦ Receiver Info:
{
  "device_id": 2,
  "role": "receiver",
  "network_id": 1,
  "address": 2,
  "password": "LoRaSecure1234",
  "band": "868000000"
}

⚡ Receiver AT Commands:
AT+BAND=868000000
AT+NETWORKID=1
AT+ADDRESS=2
AT+CPIN="LoRaSecure1234"
-----

amer@amer-Precision-5540:~$
```