

# LimeNET Micro 2.0

Software Guide: Amarisoft Configuration



---

**Model:** LimeNET Micro 2.0

**Type:** Indoor standalone

**Document date:** February 2026

**Document revision:** 1.0

## DOCUMENT REVISION HISTORY

Version	Date	Summary of main changes
1.0	19/02/2026	Initial release
	22/03/2026	Final release

## SAFETY GUIDELINES

Always read the technical guide before setting up or using the product. It is your responsibility to familiarise yourself with the product reference guide, other associated documentation, safety warnings, and use of the product properly and for the purpose for which it was designed.

### CRITICAL SAFETY WARNINGS

- ⊖ **CRITICAL:** Never operate the unit without antennas connected to all active RF ports. Operating without proper antenna loads may cause permanent damage to the RF front-end circuitry and will void the warranty.
- ⊖ **CRITICAL:** Do not exceed the maximum safe input power of +10 dBm on any RF or GNSS connector. Exceeding this limit may cause permanent damage to the LMS7002M transceiver IC.
- ⊖ **CRITICAL:** Use only the supplied 12V DC power adapter. Using third-party or incorrectly rated adapters may damage the unit, cause overheating, or create a fire hazard.

### CAUTION — POTENTIAL EQUIPMENT DAMAGE

Follow these recommendations to protect yourself and the Product from harm:

- Do not insert any sharp object into any of the Product openings.
- Do not place the Product near a heat source. Avoid placing the Product in direct sunlight.
- Do not place the Product in damp or wet environments, for example, near a bathtub, washbowl, kitchen sink or laundry tub, in a wet basement, or near a swimming pool. Do not spill any liquid on the Product.
- Do not touch the Product or its power adapter or cord with wet hands.
- Do not place the Product on an unstable surface or support.
- Do not place anything on top of the Product.
- Do not use liquid or aerosol cleaners; unplug the Product and use a soft, dry cloth for cleaning.
- When connecting a PC or other electronic devices to the Product, make sure you use the correct cable types and connect the device to the port of the Product as instructed. Incorrect links may damage the device and/or the Product.
- Do not open or attempt to open the Product. Opening or removing covers can expose you to dangerous high voltage elements or other potential risks.
- The Product contains components that emit electromagnetic waves and could interfere with pacemakers or other electronic medical devices. If you wear a pacemaker, other implantable or personal medical device, please consult your physician or medical device manufacturer for the required minimum safe distance between the Product and your medical device.
- If the Product malfunctions or fails to perform as expected, stop using it immediately.
- The Product is for indoor use only.

### WARNING — RISK OF ELECTRIC SHOCK OR FIRE

- Plug in the Product power adaptor or cord to the supply voltage with correct ratings (100–240V AC, 50/60 Hz).
- Pay attention to the power load of the electrical outlet and possible extension cord. An overburdened power outlet or damaged cords and plugs may cause electric shock or fire. Check the power cords regularly. If you find any damage, replace the cord immediately.
- Do not connect the plug into an extension cord, receptacle, or other outlet unless the plug can be fully inserted with no part of the blades exposed.
- Leave adequate space for ventilation to avoid any damage caused by overheating of the Product. Do not cover the Product or its ventilation holes. Blocking these may cause fire.
- Use the power adapter provided with your Product and do not fasten the power cable to building surfaces. Ensure the cable can move freely. Do not place heavy objects on the cable.
- Do not use the Product outside, and make sure all the connections are indoors.
- Refer to the Product Summary for the standby power consumption of the device and the maximum power consumption.

## REGULATORY NOTICE

**⚠ WARNING:** Operating RF transmitting equipment requires appropriate licensing in most jurisdictions. Users are solely responsible for ensuring compliance with all local, regional, and national radio frequency regulations. Unauthorized transmission may result in severe penalties including fines and criminal prosecution.

## ENVIRONMENTAL COMPLIANCE

RoHS (Restriction of Hazardous Substances Directive): Restriction of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment (2011/65/EU). The product complies with WEEE Directive for proper end-of-life disposal, EMC compliance for electromagnetic interference minimization, and CE marking for EU conformity.

The product should not be disposed of with household waste. It should be deposited in an appropriate facility to enable recovery and recycling. For more information regarding the requirements for recycling/disposal of the Product you may contact LimeNET support team.

## TABLE OF CONTENTS

1	System Software Overview .....	6
1.1	Operating System and Firmware .....	6
1.2	Amarisoft Software Suite Overview .....	6
	Amarisoft Stack Components .....	6
1.3	Open-Source Software Components .....	7
	Software Stack — Full Component List .....	7
1.4	Version Compatibility Requirements .....	7
2	Configuration File Locations and Structure .....	7
2.1	Master Directory Layout .....	7
	Complete Directory Structure .....	7
2.2	Key Configuration Files — Detailed Reference .....	8
	Configuration File Reference .....	8
2.3	Configuration File Syntax and Conventions .....	8
3	Editing and Customizing Configurations .....	9
3.1	Safe Modification Procedure .....	9
3.2	Configuration Backup and Restore .....	9
3.3	Frequency and Bandwidth Configuration .....	9
	5G NR Frequency Parameters .....	9
	4G LTE Frequency Parameters .....	10
3.4	RF Power and Attenuation Configuration .....	10
3.5	Subcarrier Spacing Configuration (5G NR) .....	10
4	Provisioning User Equipment (UE) .....	10
4.1	UE Database Overview .....	10
4.2	Adding a New UE — File Method .....	10
4.3	Adding/Removing UE — Remote API Method .....	11
4.4	UE Database Field Reference .....	11
	UE Database Fields .....	11
4.5	UE Provisioning Workflow .....	11
5	Network Operation and Testing .....	12
5.1	Service Lifecycle Management .....	12
	Service Control Commands .....	12
5.2	WebGUI Monitoring Interface .....	12
5.3	Remote API — Programmatic Control .....	12
	API Instance Ports .....	12
5.4	Validating RF Operation .....	13
5.5	Call Flow Testing Procedures .....	13
5.6	Performance Monitoring .....	14
5.7	Troubleshooting Common Software Issues .....	14
	Software Issue Resolution Guide .....	14
6	Integration Notes .....	15
6.1	LimeSDR Hardware Layer Interoperability .....	15
6.2	Timing and Synchronization Settings .....	15
7	Security Considerations .....	15
7.1	Credential and Access Management .....	15
7.2	Network Security Best Practices .....	15
7.3	Remote Support Configuration .....	15
8	Headless Operation Guide .....	16
8.1	Setup for Headless Operation .....	16
8.2	Ongoing Headless Management .....	16
9	Appendices .....	16
9.1	Complete Command Reference .....	16
	All Commands .....	16
9.2	Tested and Verified UE Devices .....	17
	Verified 5G SA Devices .....	17
9.3	UE APN Configuration Reference .....	17
	APN Settings for UE Devices .....	17
9.4	Software Update Procedures .....	18

# 1 System Software Overview

## 1.1 Operating System and Firmware

The LimeNET Micro runs Raspberry Pi OS (Linux-based, ARM64 architecture). The system boots to a command-line interface (CLI) by default for optimal real-time performance. The graphical desktop environment is available on demand but is not recommended for production deployments as it consumes CPU and memory resources needed by the Amarisoft stack.

Key system characteristics:

- Kernel: Linux ARM64 (verify version with: `uname -a`)
- Default boot: CLI terminal interface (no graphical desktop)
- Default user: credentials provided separately with unit documentation
- Sudo access: passwordless for default user
- Graphical desktop: `sudo systemctl start lightdm.service` (on-demand)
- Package manager: apt (Debian-based, ARM64 repository)

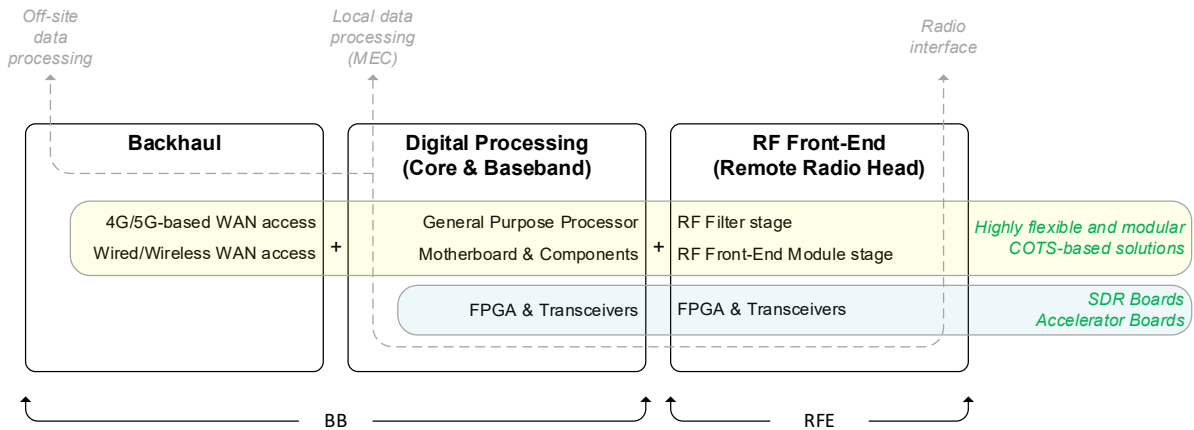


Figure 1-1: System architecture showing software layers — Backhaul connectivity, Digital Processing (Core and Baseband software), and RF Front-End interface.

## 1.2 Amarisoft Software Suite Overview

The Amarisoft software suite provides the complete end-to-end 4G/5G cellular network stack, including radio access network (RAN) and core network (CN) components (see Figure 1-1):

### Amarisoft Stack Components

Parameter	Specification
eNodeB (eNB)	4G LTE base station software — implements E-UTRA air interface, MAC, RLC, PDCP, RRC layers
gNodeB (gNB)	5G NR base station software — implements NR air interface, MAC, RLC, PDCP, SDAP, RRC layers
EPC	4G Evolved Packet Core — includes MME (Mobility Management Entity), SGW (Serving Gateway), PGW (PDN Gateway)
5GC	5G Core Network — includes AMF (Access and Mobility Management), SMF (Session Management), UPF (User Plane Function)
IMS	IP Multimedia Subsystem — enables VoLTE and VoNR voice services
WebGUI	Browser-based real-time monitoring, log analysis, and diagnostic interface
Remote API	WebSocket-based programmatic control interface for runtime configuration

The Amarisoft stack is commercial (proprietary) software licensed per Raspberry Pi unit. The license is cryptographically bound to the Pi's serial number, CPU ID, and MAC address. All other components in the software stack are open-source.

**⊖ CRITICAL:** The Amarisoft license is permanently bound to the specific Raspberry Pi hardware. Transferring to a different Pi INVALIDATES the license. If the Pi fails, contact Lime Microsystems support immediately with the original serial number before attempting replacement. Record your serial:  
`cat /proc/cpuinfo | grep Serial.`

## 1.3 Open-Source Software Components

### Software Stack — Full Component List

Program	Description	Source
TRX Driver	Radio driver bridging Amarisoft stack with SDR hardware (IQ sample transfer)	Open-source (GitHub)
LimeSuiteNG	Comprehensive library for LimeSDR hardware control and configuration	Open-source (Apache 2.0)
PCIe Kernel Module	Kernel driver for PCIe communication between RPi5 and LimeSDR XTRX	Open-source (GitHub)
FPGA Gateway	Artix-7 FPGA bitstream implementing Digital Front-End (DFE) processing	Open-source (GitHub)
Amarisoft eNB/gNB	LTE/NR base station software stack	Commercial (licensed)
Amarisoft EPC/5GC/IMS	Core network and IMS software stack	Commercial (licensed)
WebGUI	Web-based monitoring and log analysis GUI	Commercial (bundled)
FEM Control Script	Front-End Module control software (in ~/LimeNET/fem/)	Lime Microsystems

## 1.4 Version Compatibility Requirements

All software components must be version-compatible. The TRX driver, LimeSuiteNG, PCIe kernel module, and FPGA gateway form a tightly coupled set. Updating one without the others can cause sample corruption, PCIe errors, or system instability.

**i Note:** Version information is available in ~/LimeNET/ directory. The 'Version control' file shows the Amarisoft stack version and remaining upgrade support timeline. Always check compatibility before updating any component.

# 2 Configuration File Locations and Structure

## 2.1 Master Directory Layout

All LimeNET Micro network services and configurations reside in the home directory under ~/LimeNET/. The following table provides the complete directory tree with descriptions:

### Complete Directory Structure

Path	Type	Description
~/LimeNET/	Directory	Root folder for all 4G/5G stack programs and configuration
~/LimeNET/autostart/	Directory	Auto-start scripts for limenet_rpi.service
~/LimeNET/core/	Directory	MME/5GC and IMS stack program binaries
~/LimeNET/core/config/	Directory	Configuration files for core network and IMS
~/LimeNET/core/config/mme-ims.cfg	Config	Main MME/5GC configuration file
~/LimeNET/core/config/ims.cfg	Config	IMS service configuration
~/LimeNET/core/config/ue_db-ims2.cfg	Config	UE/SIM database (authorized subscribers)

Path	Type	Description
~/LimeNET/doc/	Directory	Documentation for MME/5GC, IMS, eNB, gNB stacks
~/LimeNET/doc/webGUI_access	File	Link/instructions for web GUI access
~/LimeNET/egnb/	Directory	eNodeB and gNodeB stack program binaries
~/LimeNET/egnb/config/	Directory	Configuration files for eNB/gNB
~/LimeNET/egnb/config/rf_driver/	Directory	RF driver configuration files
~/LimeNET/webGUI/	Directory	Web-based GUI files and documentation
~/LimeNET/fem/	Directory	FEM control program scripts
~/LimeNET/trx/	Directory	Amarisoft API program, IQ sample analysis tools
~/LimeNET/libs_api/	Directory	Remote API libraries (ws.js WebSocket client)
~/4G_mobile_service.sh	Script	Start/stop 4G LTE mobile service
~/5G_mobile_service.sh	Script	Start/stop 5G NR mobile service

## 2.2 Key Configuration Files — Detailed Reference

### Configuration File Reference

File	Location	Purpose and Contents
xtrx_LimeNET_5G_TDD.cfg	egnb/config/	Main gNodeB config: 5G TDD (Band 38). Defines NR_arfcn, NR_BAND, NR_BANDWIDTH, NR_DUPLEX, subcarrier_spacing, cell identity, scheduling
xtrx_LimeNET_5G_FDD.cfg	egnb/config/	Main gNodeB config: 5G FDD (Band 7). Same parameters, FDD duplex mode
xtrx_LimeNET_4G_TDD.cfg	egnb/config/	Main eNodeB config: 4G TDD. Defines LTE_arfcn, LTE_DUPLEX, LTE_RB
xtrx_LimeNET_4G_FDD.cfg	egnb/config/	Main eNodeB config: 4G FDD
drb_nr.cfg	egnb/config/	Data Radio Bearer config for 5G NR (QoS, PDCP, RLC settings)
drb_lte.cfg	egnb/config/	Data Radio Bearer config for LTE (EPS bearer mapping)
LimeNET_sib*.asn	egnb/config/	System Information Block definitions for 4G (ASN.1 format)
xtrx_5G_TDD.cfg	egnb/config/rf_driver/	RF driver config: sample rate, tx_gain, rx_gain, calibration
mme-ims.cfg	core/config/	MME/5GC config: PLMN, TAC, APN, PDN, QoS, S1/NG-AP settings
ims.cfg	core/config/	IMS config: SIP, RTP, codec settings for VoLTE/VoNR
ue_db-ims2.cfg	core/config/	Subscriber database: IMSI, K, OPc, IMPU, IMPI per UE

## 2.3 Configuration File Syntax and Conventions

Configuration files use a C-preprocessor-style syntax with the following conventions:

- `#define PARAM_NAME value` — macro-style parameter definitions (uppercase names)
- `include "path/to/file.cfg"` — include other configuration files (relative paths)
- `//` or `/* */` — single-line and multi-line comments
- JSON-like nested structures — for complex configuration objects (cells, bearers, etc.)

- Semicolons — required at end of JSON-like object definitions
- .asn files — use ASN.1 notation for System Information Block (SIB) structures

Full parameter documentation with valid ranges, defaults, and descriptions is available in: `~/LimeNET/doc/egnb/` (for eNB/gNB parameters) and `~/LimeNET/doc/core/` (for core network parameters). The Amarisoft online documentation provides the most comprehensive parameter reference.

## 3 Editing and Customizing Configurations

### 3.1 Safe Modification Procedure

Always follow this procedure when modifying any configuration file to minimize the risk of service disruption:

1. Navigate to the configuration directory: `cd ~/LimeNET/egnb/config/`
2. Create a timestamped backup: `cp filename.cfg filename.cfg.$(date +%Y%m%d_%H%M%S)`
3. Open with a text editor: `sudo nano filename.cfg`
4. Make ONLY ONE parameter change at a time
5. Save the file: `Ctrl+X, Y, Enter`
6. Restart the mobile service: `cd ~ && sudo ./5G_mobile_service.sh (select 's')`
7. Verify operation: check RF LEDs, connect test UE, check terminal for errors
8. If errors occur, restore from backup: `cp filename.cfg.backup filename.cfg`

**⚠ WARNING:** Common syntax errors that prevent service startup: missing semicolons at end of JSON objects, mismatched brackets or braces, unclosed quotation marks, undefined macro references. Always check terminal error output — it usually shows the line number of the error.

### 3.2 Configuration Backup and Restore

Comprehensive backup procedure (recommended before any changes):

```
# Backup all eNB/gNB configs
cp -r ~/LimeNET/egnb/config ~/backups/egnb_config_$(date +%Y%m%d)

# Backup all core network configs
cp -r ~/LimeNET/core/config ~/backups/core_config_$(date +%Y%m%d)

# Verify backup exists
ls -la ~/backups/
```

To restore from backup:

```
cp -r ~/backups/egnb_config_YYYYMMDD/* ~/LimeNET/egnb/config/
```

For factory reset: contact Lime Microsystems support for original configuration files, or re-image the system with the original OS image.

### 3.3 Frequency and Bandwidth Configuration

To change the operating frequency and bandwidth, edit the main configuration file for the desired technology:

```
sudo nano ~/LimeNET/egnb/config/xtrx_LimeNET_5G_TDD.cfg
```

5G NR Key Parameters:

#### 5G NR Frequency Parameters

Parameter	Specification
<code>#define NR_arfcn &lt;value&gt;</code>	NR Absolute Radio Frequency Channel Number (center frequency)
<code>#define NR_DUPLEX &lt;mode&gt;</code>	Duplex mode: "TDD" or "FDD"
<code>#define NR_BAND &lt;number&gt;</code>	Band number (e.g., 38 for TDD, 7 for FDD)
<code>#define NR_BANDWIDTH &lt;MHz&gt;</code>	Channel bandwidth in MHz (max 20 TDD, 10 FDD)
<code>subcarrier_spacing</code>	Data channel SCS: 15, 30, 60, or 120 kHz
<code>ssb_subcarrier_spacing</code>	SSB subcarrier spacing (typically 15 or 30 kHz)

Parameter	Specification
ss_pbch_block_power	SSB reference signal power in dBm

### 4G LTE Frequency Parameters

Parameter	Specification
#define LTE_earfcn <value>	E-UTRA ARFCN (center frequency identifier)
#define LTE_DUPLEX <mode>	Duplex mode: "TDD" or "FDD"
#define LTE_RB <value>	Number of Resource Blocks (determines bandwidth: 25=5MHz, 50=10MHz, 100=20MHz)

**[i] Note:** Maximum bandwidth: 20 MHz for TDD bands, 10 MHz for FDD bands. Ensure the selected frequency is authorized for your use or in an ISM band. Bandwidth must be compatible with the selected band. Consult the Amarisoft documentation in ~/LimeNET/doc/ for complete valid parameter ranges.

### 3.4 RF Power and Attenuation Configuration

RF output power is controlled through two mechanisms that can be used independently or combined:

**tx\_gain [0–50] — Analog RF IC Gain:** Controls the LMS7002M transmitter analog chain. 50 = max power (min attenuation). Step size: 1 dB from 50→40, 2 dB from 40→0. Note: every second step below 40 has no effect (odd steps are no-ops).

**cell\_gain [-200–0] — Digital Signal Gain:** Controls the digital attenuation applied to the transmitted signal. 0 = no attenuation, -200 = max attenuation (effectively silence). This is the PREFERRED method for most use cases. Important: does NOT update SS PBCH block power in SIB1. Applies per cell.

For permanent changes, edit the RF driver config file:

```
sudo nano ~/LimeNET/egnb/config/rf_driver/xtrx_5G_TDD.cfg
```

Locate the tx\_gain parameter and set the desired value. Save and restart the service.

**⚠ WARNING:** Ensure output power complies with regulatory limits for your region and license conditions.

### 3.5 Subcarrier Spacing Configuration (5G NR)

5G NR supports multiple subcarrier spacing options:

```
sudo nano ~/LimeNET/egnb/config/xtrx_LimeNET_5G_***.cfg
```

- subcarrier\_spacing: Defines SCS for data channels (15, 30, 60, or 120 kHz)
- ssb\_subcarrier\_spacing: Defines SCS for SSB/PBCH (typically 15 or 30 kHz)

**[i] Note:** SCS must be supported by the connected UE devices. Restart service after changes.

## 4 Provisioning User Equipment (UE)

### 4.1 UE Database Overview

The UE database stores authentication information for all authorized subscriber devices (SIM cards). Only devices with matching entries in this database can register on the network. The database is a text file using JSON-like syntax.

**File location:** ~/LimeNET/core/config/ue\_db-ims2.cfg

### 4.2 Adding a New UE — File Method

1. Open the UE database: `sudo nano ~/LimeNET/core/config/ue_db-ims2.cfg`
2. Add a new entry block at the end of the ue\_db array (before the closing bracket):

```
{
  sim_algo: "milenage",
  imsi: "001010000138XXX",
  opc: "DC8A4C00423DB8DD5EDD6C9520707YYY",
  amf: 0x9001,
  sqn: "000000000000",
}
```

```
K: "5E5B31B71A3EEFF070A7160D451C5ZZZ",
  impu: ["001010000138XXX", "tel:138XXX"],
  impi: "001010000138XXX@ims.mnc001.mcc001.3gppnetwork.org",
},
```

3. Replace all XXX/YYYY/ZZZ values with actual SIM card parameters
4. Save the file (Ctrl+X, Y, Enter)
5. Restart the core network service for changes to take effect

**⚠ WARNING:** IMSI, K, and OPc values MUST exactly match the physical SIM card. Even a single character mismatch will prevent authentication.

### 4.3 Adding/Removing UE — Remote API Method

UEs can be managed at runtime without editing files:

```
# Add a UE via Remote API
./ws.js mme '{"message":"ue_add","ue_db": [{
  "sim_algo": "milnagen",
  "imsi": "001010000138XXX",
  "K": "5E5B31B71A3EEFF070A7160D451C5ZZZ",
  "opc": "DC8A4C00423DB8DD5EDD6C9520707YYY"
}]}'
```

**[i] Note:** API-added UEs are volatile — they are lost on service restart unless also added to the config file.

### 4.4 UE Database Field Reference

#### UE Database Fields

Field	Type	Description
sim_algo	String	Authentication algorithm: "milnagen" (standard), "xor" (testing), "tuak"
imsi	String	International Mobile Subscriber Identity (15 digits)
K	String	Subscriber secret key (128-bit hex, 32 characters)
opc	String	Operator Variant Algorithm Configuration (128-bit hex, 32 chars)
amf	Hex	Authentication Management Field (typically 0x9001 for 5G)
sqn	String	Sequence Number (48-bit hex, 12 characters, "000000000000" initial)
impu	Array	IMS Public User Identity: ["IMSI", "tel:number"]
impi	String	IMS Private User Identity: IMSI@ims.mnc.mcc.3gppnetwork.org

To remove a UE: open the database file, locate the entry by IMSI, delete the entire block (including braces and trailing comma), save, restart core network.

### 4.5 UE Provisioning Workflow

1. Obtain SIM card parameters from SIM manufacturer (IMSI, K, OPc, AMF)
2. Record all parameters securely (K and OPc are secret keys)
3. Add entry to ue\_db-ims2.cfg (file method) or via Remote API
4. Configure IMS parameters (impu, impi) if VoLTE/VoNR is required
5. Restart core network service
6. Insert SIM into UE, power on, verify registration
7. Test data connectivity and voice (if IMS configured)

## 5 Network Operation and Testing

### 5.1 Service Lifecycle Management

Complete service control reference:

#### Service Control Commands

Action	Command	Notes
Start 5G service	<code>cd ~ &amp;&amp; sudo ./5G_mobile_service.sh</code>	Select 's', then band
Start 4G service	<code>cd ~ &amp;&amp; sudo ./4G_mobile_service.sh</code>	Select 's', then band
Stop service	Same script, select 'k'	Graceful stop, UEs disconnect
Stop auto-started	<code>sudo systemctl stop limenet_rpi.service</code>	Stops the systemd service
Disable auto-start	<code>sudo systemctl disable limenet_rpi.service</code>	Persists across reboots
Enable auto-start	<code>sudo systemctl enable limenet_rpi.service</code>	Default: 5G Band 38
Check auto-start	<code>sudo systemctl status limenet_rpi.service</code>	Shows active/inactive

**Note:** Only one technology mode (4G or 5G) can be active per unit. The service scripts handle starting both the eNB/gNB and the core network (MME/5GC/IMS) processes.

### 5.2 WebGUI Monitoring Interface

The web-based GUI provides real-time monitoring and diagnostics. Access from any browser on the same network:

```
http://<limenet-ip-address>/lte
```

WebGUI capabilities:

- Real-time cell status: active cells, operating band, bandwidth, power
- Connected UE list: IMSI, IP address, signal quality, throughput per UE
- RF metrics: DL/UL throughput, BLER, CQI, MCS, RSRP, RSRQ
- Event log: registration, handover, bearer establishment, errors
- Historical log analysis: load and analyze previously stored log files
- Cell configuration view: current parameter settings

WebGUI documentation is located in `~/LimeNET/doc/webGUI/` and `~/LimeNET/webGUI/doc/`.

### 5.3 Remote API — Programmatic Control

The Remote API enables programmatic, real-time control and monitoring via WebSocket using JSON messages. The `ws.js` client script is located at `~/LimeNET/libs_api/ws.js`.

Prerequisites (pre-installed, re-install if missing):

```
sudo apt install nodejs npm
cd ~/LimeNET/libs_api/ && npm install nodejs-websocket
```

Instance ports and aliases:

#### API Instance Ports

Instance	Port / Alias
eNB / gNB	Port 9001 (alias: enb)
MME / 5GC	Port 9000 (alias: mme)
IMS	Port 9003 (alias: ims)

Common API commands:

```
# Read current eNB configuration
./ws.js enb '{"message": "config_get"}'

# Read current MME configuration
```

```
./ws.js mme '{"message": "config_get"}'

# Change SS PBCH block power at runtime
./ws.js enb '{"message": "sib_set", "cells": {"1": {"sib1": {"ss_pbch_block_power": -30}}}'

# Change analog TX gain
./ws.js enb '{"message": "rf", "tx_gain": 50, "tx_channel_index": 0}'

# Change digital cell gain
./ws.js enb '{"message": "cell_gain", "cell_id": 1, "gain": -20}'

# Add UE to database at runtime
./ws.js mme '{"message": "ue_add", "ue_db": [{"...}]}'
```

API latency: typical config\_get back-to-back execution on a single cell takes approximately 0.06–0.08 seconds. Latency increases with connected UE count and number of active cells.

**Note:** Remote API calls are given priority by the Amarisoft stack. The ws.js script can be copied to any folder.

## 5.4 Validating RF Operation

After starting the service, verify RF operation through multiple methods:

1. Visual check: LED 3 (Ch B) and LED 4 (Ch A) should be solid green
2. UE registration: power on test UE — 5G/4G indicator within 15 seconds (see Figure 5-1)
3. Data test: open web browser on UE, load webpage
4. Remote API check: `./ws.js enb '{"message": "config_get"}'` — verify cell parameters
5. WebGUI check: `http://<ip>/lte` — verify cell status and connected UEs
6. Signal strength: verify coverage at various distances (typical: up to 50m indoor)

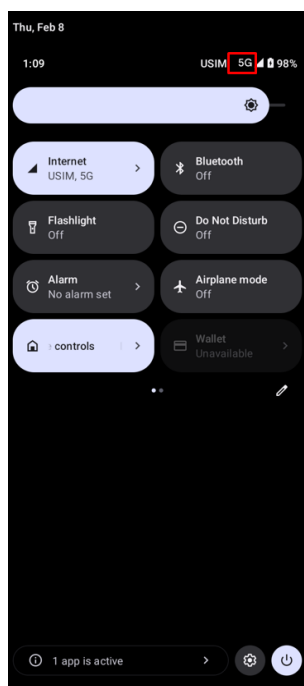


Figure 5-1: UE successfully registered on the 5G SA network — the 5G indicator appears in the status bar (highlighted).

## 5.5 Call Flow Testing Procedures

**Registration Test:** Power on UE. Verify 5G/4G indicator appears in status bar within 15 seconds. If no indicator, check SIM, APN, band support, and UE database.

**Data Test:** Open browser on UE, navigate to google.com. Verify page loads. Run speed test for throughput measurement. Check WebGUI for per-UE throughput statistics.

**VoNR/VoLTE Test (if IMS configured):** Register two UEs. Place voice call between them. Verify call setup, audio quality, and call release. Check IMS logs for SIP signaling.

**Handover Test (multi-unit):** Move UE between coverage areas of two cells. Verify seamless handover without call drop. Check eNB/gNB logs for handover events.

**Stress Test:** Register maximum number of UEs simultaneously. Monitor CPU, memory, and RF metrics. Verify service stability over extended periods.

## 5.6 Performance Monitoring

- CPU utilization: top or htop (Amarisoft processes should use <80% total CPU)
- Memory usage: free -h (watch for swap usage — indicates memory pressure)
- Disk space: df -h (keep eMMC utilization below 80%)
- Temperature: vcgencmd measure\_temp (keep below 80°C)
- Network: iperf3 between UE and backhaul server for throughput measurement

## 5.7 Troubleshooting Common Software Issues

### Software Issue Resolution Guide

Issue	Likely Cause	Resolution
Service won't start	Config syntax error	Check terminal for line numbers. Common: missing ;, ), brackets. Restore backup
Service crashes	Memory exhaustion	Check: free -h. Reduce bandwidth/UEs. Disable desktop GUI
UE not connecting	SIM not in database	Verify IMSI/K/OPc in ue_db-ims2.cfg match SIM exactly. Restart core
UE authenticates but no IP	Core network PDN issue	Check mme-ims.cfg: APN settings, IP pool, default bearer config
No internet on UE	NAT/routing issue	Check: ping 8.8.8.8 on LimeNET. Verify: iptables -L. Check APN on UE: 'internet'
Poor throughput	CPU overloaded	Run: top. Disable desktop. Reduce bandwidth. Fewer concurrent UEs
License invalid	Hardware changed	Contact support with Pi serial numbers (original + new)
WebGUI won't load	Service not running	Start mobile service. Check IP. Try: http://<ip>/lte. Clear browser cache
Config file corrupted	Edit error	Restore from backup. Factory reset: contact support for original files
PCIe errors in log	Driver/gateway mismatch	Verify versions match. Re-flash gateway. Reboot. Check: lspci

## 6 Integration Notes

### 6.1 LimeSDR Hardware Layer Interoperability

The TRX driver serves as the abstraction layer between the Amarisoft software stack and the LimeSDR XTRX hardware. It handles: IQ sample format conversion (Amarisoft internal format ↔ LimeSDR 12-bit I/Q), sample rate configuration and resampling, timestamp management for TDD switching, gain control translation (tx\_gain/cell\_gain → LMS7002M registers), and calibration trigger at startup.

The PCIe kernel driver manages low-level bus communication: DMA buffer allocation, interrupt handling, and bus error recovery. All driver components must be version-compatible.

### 6.2 Timing and Synchronization Settings

The RF driver configuration file (in `~/LimeNET/egnb/config/rf_driver/`) contains timing-critical parameters that control the sample clock relationship between the FPGA and the baseband processor. These include TX/RX timing advance offsets, TDD switching guard periods, and timestamp alignment parameters.

**⚠ WARNING:** RF driver timing parameters are pre-configured for optimal performance on the LimeNET Micro hardware. Do not modify these parameters without guidance from Lime Microsystems engineering.

#### Calibration and Tuning

The LMS7002M performs automatic calibration routines at service startup:

- TX DC offset calibration: minimizes LO leakage
- TX IQ imbalance calibration: corrects phase/amplitude mismatch between I and Q
- RX DC offset calibration: removes DC bias from received signal
- RX IQ imbalance calibration: corrects receive path I/Q mismatch
- LO frequency calibration: fine-tunes PLL synthesizer for target frequency

Calibration runs automatically and takes approximately 1–2 seconds per channel. Manual tuning can be performed via LimeSuiteNG GUI tools if needed for non-standard frequency bands.

## 7 Security Considerations

### 7.1 Credential and Access Management

- Change default system passwords immediately after first setup
- Use SSH key-based authentication; disable password authentication for production
- Implement unique passwords for each unit in multi-unit deployments
- Record and securely store the Raspberry Pi serial number: `cat /proc/cpuinfo`
- Protect the UE database file — it contains subscriber authentication keys (K, OPc)
- Set restrictive file permissions on sensitive files: `chmod 600 ue_db-ims2.cfg`

### 7.2 Network Security Best Practices

- Configure iptables firewall to restrict management interface access
- Use VPN for remote management over untrusted networks
- Regularly review system access logs: `journalctl --since today`
- Monitor for unauthorized UE connections via WebGUI
- Keep all software components updated with security patches
- Disable unused services and ports
- Implement network segmentation between management and user plane traffic

### 7.3 Remote Support Configuration

For remote support from Lime Microsystems, Raspberry Pi Connect is available:

```
sudo apt install rpi-connect
rpi-connect on signin
```

Note the verification code (XXXX-YYYY) and provide it to support. After acknowledgment:

```
rpi-connect shell on      # Enable remote shell access
rpi-connect vnc on       # Enable remote desktop access
```

**Note:** Disable rpi-connect after support session is complete for security.

## 8 Headless Operation Guide

Headless operation (no monitor, keyboard, or mouse) is fully supported and recommended for production deployments.

### 8.1 Setup for Headless Operation

1. Complete initial setup with display and keyboard (Sections 5–6 of Quick Start Guide)
2. Configure Ethernet with static IP (recommended) or verify DHCP lease
3. Verify SSH access works from remote computer
4. Configure auto-start for mobile services (enabled by default)
5. Test all functionality via SSH and WebGUI
6. Disconnect monitor and keyboard

### 8.2 Ongoing Headless Management

- SSH for all command-line operations: `ssh username@<ip-address>`
- WebGUI for monitoring: `http://<ip-address>/lte`
- Remote API for programmatic control: `./ws.js enb '{...}'`
- Configure syslog forwarding for centralized log collection
- Set up email or webhook alerts for critical events

Benefits: reduced power consumption, better system performance (no GPU overhead), suitable for rack-mount or remote installations.

## 9 Appendices

### 9.1 Complete Command Reference

#### All Commands

Category	Command	Description
System	<code>sudo poweroff</code>	Graceful system shutdown
System	<code>sudo reboot</code>	System restart
System	<code>uname -a</code>	OS version and kernel info
System	<code>vcsencmd measure_temp</code>	CPU temperature
System	<code>df -h</code>	Disk usage
System	<code>free -h</code>	Memory usage
System	<code>top / htop</code>	Process monitor
System	<code>cat /proc/cpuinfo   grep Serial</code>	Pi serial number
Service	<code>cd ~ &amp;&amp; sudo ./5G_mobile_service.sh</code>	5G service start/stop script
Service	<code>cd ~ &amp;&amp; sudo ./4G_mobile_service.sh</code>	4G service start/stop script
Service	<code>sudo systemctl stop limenet_rpi.service</code>	Stop auto-started service
Service	<code>sudo systemctl disable limenet_rpi.service</code>	Disable auto-start
Service	<code>sudo systemctl enable limenet_rpi.service</code>	Enable auto-start
Service	<code>sudo systemctl status limenet_rpi.service</code>	Service status check
Network	<code>ip addr show</code>	IP addresses
Network	<code>ping -c 4 8.8.8.8</code>	Internet connectivity test

Category	Command	Description
Network	<code>ping -c 4 www.google.com</code>	DNS resolution test
Network	<code>ip route show</code>	Routing table
Network	<code>sudo iptables -L -n -v</code>	Firewall rules
Network	<code>ethtool eth0</code>	Ethernet link status
GUI	<code>sudo systemctl start lightdm.service</code>	Start desktop
GUI	<code>sudo systemctl stop lightdm.service</code>	Stop desktop
GUI	<code>http://&lt;ip&gt;/lte</code>	Access WebGUI (browser)
API	<code>./ws.js enb '{"message":"config_get"}'</code>	Read eNB/gNB config
API	<code>./ws.js mme '{"message":"config_get"}'</code>	Read core network config
Config	<code>sudo nano ~/LimeNET/egnb/config/file.cfg</code>	Edit eNB/gNB config
Config	<code>sudo nano ~/LimeNET/core/config/ue_db-ims2.cfg</code>	Edit UE database
Debug	<code>lspci</code>	Verify PCIe device (LimeSDR XTRX)
Debug	<code>journalctl -p warning -n 100</code>	Last 100 warnings/errors
Debug	<code>dmesg   tail -50</code>	Last 50 kernel messages

## 9.2 Tested and Verified UE Devices

The following UE devices have been tested for 5G SA mode compatibility:

### Verified 5G SA Devices

Device	Status
Huawei P40 Pro	Band 38, 5G SA verified
OnePlus Nord	Band 38, 5G SA verified
OnePlus Nord 2 5G	Band 38, 5G SA verified
Oppo Find X3 Lite	Band 38, 5G SA verified
Umidigi G9 5G	Band 38, 5G SA verified

**Note:** Other UEs may work but require separate testing. Verify the UE supports your operating band. Keep UE firmware up to date. Contact Lime Microsystems for enterprise UE recommendations.

## 9.3 UE APN Configuration Reference

### APN Settings for UE Devices

Setting	Value
APN Name	internet (or apn)
APN	internet (or apn)
Bearer	4G and 5G (or unspecified)
Authentication	None
Protocol	IPv4
Roaming Protocol	IPv4
MVNO Type	None

## 9.4 Software Update Procedures

LimeSuiteNG and Open-Source Drivers:

```
sudo apt update
cd ~/develop/LimeSuiteNG
git pull
mkdir -p build && cd build
cmake ..
make -j4
sudo make install
```

FPGA Gateway: download latest release from GitHub, flash using limeUtil or limeGUI, verify with lspci after reboot.

Amarisoft Stack: contact Lime Microsystems for licensed updates. Updates are provided as complete packages with installation instructions. The 'Version control' file in ~/LimeNET/ shows remaining upgrade support timeline.

**⚠ WARNING:** Always backup ALL configurations before any update. Updates may require reboot. Test thoroughly after every update. Ensure all component versions remain compatible.

## End of Document

This concludes the LimeNET Micro 2.0 series Software Guide: Amarisoft Configuration. For additional support, please contact Lime Microsystems.

**Technical Support:** [support@limemicro.com](mailto:support@limemicro.com)

**Sales Inquiries:** [sales@limemicro.com](mailto:sales@limemicro.com)

**Website:** <https://limemicro.com/>

*Document Revision 1.0 | February 2026 |*

*© 2026 Lime Microsystems Ltd. All rights reserved.*