Obsah

Konfigurace jednotky NPA R04 **1**
Síťová nastavení 1
Klíče v configu 1
VPN (OpenVPN klient) 2
Zkoušky dle RfG 2
Kontrola konfigurace 3

Konfigurace jednotky NPA R04

Platí pro verzi NPA R04: 2.15.beta12 · Vytvořeno: 2026-06-19 · Aktualizováno: 2026-09-08

Tento dokument popisuje, jak se jednotlivá nastavení jednotky ukládají do konfiguračního souboru. Postupně sem budou přibývat další části konfigurace — **zatím je popsáno ukládání síťových nastavení.**

Hlavní konfigurace je JSON soubor `/data/NPA_R04.cnf` (resp. číslovaná verze `/data/NPA_R04.cnf.<N>`, viz Odesílání konfigurace §Bezpečné ukládání). V kódu k němu přistupuje `NConfig`; raw JSON drží člen `mC2`, zápis na disk dělá `NConfig::Save()`.

Síťová nastavení

Síťová nastavení se zadávají ve webovém rozhraní na stránce **Nastavení sítě** (`html/v2/net_settings.html`). Ukládají se na **dvě místa**:

1. do hlavního JSON configu `/data/NPA_R04.cnf` pod klíče s prefixem `ns_`,
2. do systémového síťového souboru `/data/npa_lan` (formát `interfaces`), který se z těch klíčů **vygeneruje** a teprve ten se reálně promítne do systému.

Klíče v configu

Klíč (`mC2`)	Pole ve formuláři	Význam
`ns_Dhcp`	`ns_Dhcp` (radio)	režim adresace: `ns_dhcp` (DHCP, **výchozí** – chybějící klíč se doplní při startu) / `ns_static` (statická IP)
`ns_net_ip`	`ns_net_ip`	IP adresa (statický režim); pod DHCP se při uložení stránky přepíše živou hodnotou ze systému
`ns_net_mask`	`ns_net_mask`	maska sítě – prefix (`24`) nebo tečkovaně (`255.255.255.0`); do `npa_lan` jde vždy prefix (`NConfig::MaskToPrefix`)
`ns_net_gate`	`ns_net_gate`	výchozí brána; pod DHCP živá hodnota

Klíč (mC2)	Pole ve formuláři	Význam
ns_net_dns	ns_net_dns	DNS servery, seznam oddělený čárkou; pod DHCP skutečné servery z DHCP (NState::GetDNS: systemd-resolved → resolv.conf → lease dhclient/networkd, nikdy stub 127.0.0.53); do npa_lan jdou oddělené mezerou bez loopbacku
ns_netPort	ns_netPort	port (Modbus TCP); prázdné = 5002 (doplní se při startu)
ns_netId	ns_netId	Modbus ID jednotky; prázdné = 1 (doplní se při startu, stejně ns_485IdSl)
ns_vpn	ns_vpn	povolení VPN (true / false)

> **Pozn.:** pole ns_m-s (NPA master/slave) z formuláře je **mrtvé** — v C++ se nikde nečte (vstup je ve formuláři navíc disabled).

VPN (OpenVPN klient)

ns_vpn jen přepíná symlink /data/npa_r04_cli.conf mezi /data/npa_r04_cli.conf.en (zapnuto) a .dis (vypnuto). Vlastní OpenVPN config (server v řádku remote, klíče) je v .en a config jednotky ho neobsahuje. Přepnutí na jiný VPN server s automatickým návratem při neúspěchu popisuje vpn--presun.md (stav v /data/vpn_switch.json).

Zkoušky dle RfG

Stránka **Zkoušky dle RfG** (formulář POST rfg_save) ukládá do mC2:

Klíč (mC2)	Význam
rfg_src	spouštění sekvencí: sw (tlačítka na stránce; **výchozí**, prázdné = sw) / hw (binární vstupy RfG-enable, RfG-seq1..3)
rfg_s1_pct, rfg_s2_pct, rfg_s3_pct	kroky sekvence – celá čísla 0--100 oddělená čárkou, nejvýše 50; prázdné/neplatné = výchozí sekvence z FW
rfg_s1_sec, rfg_s2_sec, rfg_s3_sec	setrvání na kroku [s], 1--3600; prázdné = výchozí (190 / 90 / 70)

Tentýž formulář ukládá i fajfky zařazení rozpadových míst do sekvence – ty ale nejdou do top-level klíčů, nýbrž **do objektu místa** (location[]):

Klíč (objekt místa)	Význam
`lo_rfgEn`	`true` = místo zařazené do zkoušky (fajfka), `false` = vynecháno; **chybí = zařazeno** (nové místo, dosud neuloženo z této stránky). Formulář posílá `rfg_lo_<uuid>=true` jen za zaškrtnutá, `Server` je ukládá přes `NRfgTest::SaveLocationEnable`

Výchozí hodnoty se do configu neseedují – FW je použije, když klíč chybí nebo je prázdný (`NRfgTest::SeqPct/SeqSec`). Stav běžící zkoušky není v configu, ale v `/data/rfg_test.json`. Podrobnosti `zkousky-rfg.md`.

Kontrola konfigurace

Jednotka umí ověřit, jestli v configu nechybí důležité hodnoty, jejichž absence by mohla narušit řízení, regulaci nebo běh FW. Kontrolu provádí `NConfig::ConfigWarnings()`, která vrací seznam problémů (český text); prázdný seznam = vše v pořádku.

Kde se výsledek zobrazí / odešle:

- **Hlavní stránka (Přehled).** Při každém načtení se případné problémy vypíšou **červeně úplně nahoře**. Když je vše v pořádku, nezobrazí se nic.
- **Odeslání na server.** Při každém uložení configu se kontrola spustí i v `NRemoteLog::SendConfig()` a výsledek se přibalí do odesílané JSON zprávy:

  ```
  { "sn": "…", "config": { … }, "config_ok": true, "warnings": [] }
  ```

  `config_ok` je `false`, pokud je seznam `warnings` neprázdný.

Co se kontroluje:

Kontrola	Hlášení (důvod)
BESS zařízení bez kapacity (`bess=true`, `dd_BESSCapacity ≤ 0`)	kapacita nenačtena z Modbusu ani zadána ručně; navíc zkresluje kapacitně vážené SOC (viz Řízení BESS §6)
Výkonové zařízení bez jmenovitého výkonu (`pwr=true`, `dd_Power ≤ 0`)	výkon nenačten z Modbusu ani zadán ručně (`dd_PowerHand`)
Chybí hlavní elektroměr TML (`dd_elmeterTml` ani `dd_meas="TML"`)	regulace nemá z čeho měřit
Regulované místo (FVE/MVE/KGJ/BPS/VTE/DA/HE) s nulovým výkonem	součet `dd_Power` zařízení v místě i `lo_insPower` = 0 → nelze počítat %
BESS místo s nulovým výkonem měničů (součet `dd_Power` = 0)	dělení nulou v regulaci BESS
Duplicitní adresa výstupu (`dd_modbusAddr`)	dvě zařízení píší do stejného bloku NPA Modbus serveru
Stejná Modbus slave adresa (`dd_idNet`) na téže sběrnici (RTU) / `ip:port` (TCP)	kolize při čtení měničů

Seznam je společný pro web i odeslání na server (jeden zdroj pravdy) a snadno se rozšiřuje o další kontroly.