Obsah

Binární vstupy a výstupy — funkce, výpočet a odesílání na server **1**
1. Jak jsou binární I/O uložené v konfiguraci . . . 1
2. Výpočet hodnoty přiřazené funkce . . . 2
    2.1 Vstupy . . . 2
    2.2 Výstupy . . . 3
3. Globální vs. funkce přiřazená rozpadovému místu . . . 3
4. Seznam funkcí . . . 3
    4.1 Vstupy (`io_inp_func`) . . . 3
    4.2 Výstupy (`io_out_func`) . . . 4
5. Vstup jako povel (nastavení povelu) . . . 4
6. Odesílání na server . . . 5
    6.1 Mapování všech posílaných UUID a detekce nevyužitého kanálu . . . 5

Binární vstupy a výstupy — funkce, výpočet a odesílání na server

Platí pro verzi NPA R04: 2.13.beta20 · Vytvořeno: 2026-06-18

Tento dokument popisuje, jak jednotka NPA R04 pracuje s **binárními vstupy a výstupy**: jak se z **konfigurace (JSON)** počítá hodnota přiřazené funkce, jak se rozliší **globální** funkce od funkce přiřazené **rozpadovému místu**, jaký je seznam funkcí, a jak se binární I/O a **nastavený povel** odesílají na server. Popis vychází z konfigurace a chování, nezávisle na vnitřní implementaci.

---

1. Jak jsou binární I/O uložené v konfiguraci

Fyzická I/O karta je v konfiguraci uložena jako **zařízení s `dd_type = "bin_io"`**. To má dvě pole kanálů — `input[]` a `output[]` — kde každý prvek má svoje `uuid`. Vlastnosti jednotlivých kanálů (přiřazená funkce, rozpadové místo, negace) jsou v objektu zařízení uložené pod **indexovanými klíči**, kde `<i>` je pořadí kanálu:

Vlastnost	Klíč u vstupu `i`	Klíč u výstupu `i`
Přiřazená funkce	`io_inp_func<i>`	`io_out_func<i>`
Rozpadové místo (`uuid` lokace, nebo prázdné)	`lo_id_location<i>`	`lo_id_location_o<i>`
Negace (invertuje 0/1)	`io_inp_neg<i>`	`io_out_neg<i>`

Příklad (zkráceně) v JSON konfiguraci:

```
{
  "dd_type": "bin_io",
  "input":  [ { "uuid": "..." }, { "uuid": "..." } ],
  "output": [ { "uuid": "..." } ],

  "io_inp_func0": "P1 - Puls",      "lo_id_location0": "",
  "io_inp_neg0": "false",

  "io_inp_func1": "DS-FA1-ON",      "lo_id_location1": "<uuid mista>",
  "io_inp_neg1": "false",

  "io_out_func0": "Control-P1",     "lo_id_location_o0": "",
  "io_out_neg0": "false"
}
```

V příkladu má vstup 0 (`P1 - Puls`) prázdné `lo_id_location0` → je **globální**, kdežto vstup 1 (`DS--FA1-ON`) má vyplněné `lo_id_location1` → patří danému **rozpadovému místu**.

> **Pozor — nejen karta `bin_io`:** binární vstupy nemusí pocházet jen z fyzické I/O karty (`dd_type = "bin_io"`). Stejnou strukturu (`input[]` s `uuid` + indexované klíče `io_inp_func<i>`, `lo_id_location<i>`, `io_inp_neg<i>`) mají i **Modbus zařízení**, která čtou binární vstupy z registrů. Při hledání **všech** posílaných UUID (kap. 6) je proto nutné projít `input[]` / `output[]` u **všech** zařízení v poli `device`, ne jen u `bin_io`.
>
> **Menu:** Zařízení → karta I/O → *editace* → vstupy / výstupy. Funkce se vybírá ze seznamu (kap. 4), rozpadové místo z nabídky lokací; **prázdné** rozpadové místo = **globální** funkce (kap. 3).

---

2. Výpočet hodnoty přiřazené funkce

2.1 Vstupy

Hodnota funkce přiřazené vstupu vznikne ve dvou krocích:

1. **Fyzický stav** kanálu (sepnuto = 1 / rozepnuto = 0).
2. **Negace:** je-li `io_inp_neg<i> = "true"`, hodnota se invertuje.

Výsledek je „surová” hodnota jednoho kanálu. Protože **víc fyzických vstupů může mít stejnou funkci** (např. signál P1 přivedený na více míst), hodnota **funkce** se z kanálů odvodí logickou kombinací podle jejich přiřazení (funkce + rozpadové místo):

Vyhodnocení	Které kanály	Kombinace	Význam
Globálně	všechny vstupy s danou funkcí (libovolné místo)	OR	je funkce aktivní kdekoli?
Globálně (shoda)	všechny vstupy s danou funkcí	AND	je aktivní na všech?
Pro jedno místo	vstupy s danou funkcí a daným `lo_id_location`	OR (resp. AND)	stav funkce v tom místě

Není-li nakonfigurován **žádný** vstup s danou funkcí, funkce nemá platnou hodnotu.

> **Pozor — co se posílá na server:** hodnota `value` u každého kanálu je vždy hodnota **po aplikaci negace**, tj. logická (efektivní) hodnota, **ne** surový fyzický bit. U vstupu se negace `io_inp_neg<i>` aplikuje hned při čtení a teprve výsledek se uloží do stavu (a odešle). Stejně tak u výstupu (kap. 2.2) jde o hodnotu po negaci, tedy tu, která reálně jde na fyzický pin.

2.2 Výstupy

Hodnota každého výstupu se **odvozuje z jeho přiřazené funkce** `io_out_func<i>` podle stavu systému. Přehled pravidel podle skupiny funkcí:

Funkce výstupu	Hodnota (1 = sepnuto)
`Control-P0` … `Control-P4`	sepnuto, když je aktuální **povel činného výkonu** na dané úrovni P0--P4
`Control_QL…/Q0/QC…`	sepnuto, když je na dané úrovni **povel jalového výkonu**
`Control_cos-fi_…`	sepnuto, když je na dané úrovni **povel cos φ**
`Control-Ch-P1…4` / `Control-DCh1…4`	sepnuto, když je na dané úrovni povel **nabíjení / vybíjení BESS**
`GS-DS-S`, `GD_BESS` (galvanické odpojení)	nejdřív stav daného **rozpadového místa**, s **globálním fallbackem**
`Total_stop`	OR stavu **místa** a **globálního** Total_stop
`ON`	vždy 1
`PLC_Err` / `PLC_Sys`	souhrnný chybový stav
ostatní (stavové, topení, …)	odvozeno z odpovídajícího stavu systému, resp. dané funkce

Nakonec se aplikuje **negace:** je-li `io_out_neg<i> = "true"`, výsledek se invertuje. Hodnota se pak zapíše na fyzický výstup.

---

3. Globální vs. funkce přiřazená rozpadovému místu

Rozliší se vyplněním klíče **`lo_id_location<i>`** (vstup) / **`lo_id_location_o<i>`** (výstup):

- **prázdné** → funkce je **globální** (platí pro celou výrobnu),
- **`uuid` rozpadového místa** → funkce patří **tomu místu**.

Jak se to projeví:

- **Globální funkce** se vyhodnocuje přes **všechny** kanály s danou funkcí bez ohledu na místo (logický OR). Hodí se pro celovýrobní signály — povely, `Total_stop`, `SPG_HDO`.
- **Lokační funkce** se vyhodnocuje jen z kanálů, které mají vyplněné `lo_id_location` rovné danému místu. Hodí se pro stavy a ovládání konkrétního rozpadového místa (jističe, ochrany, galvanické odpojení daného místa).
- U výstupů s místně specifickými funkcemi (`GS-DS-S`, `GD_BESS`, `Total_stop`) má **přednost hodnota místa**; není-li, použije se **globální** hodnota.

---

4. Seznam funkcí

4.1 Vstupy (`io_inp_func`)

Povelové — pulsní (vyhodnocují se jako krátký puls, viz kap. 5):

Funkce	Význam
`P1 - Puls` … `P4 - Puls`	povel k regulaci činného výkonu (úroveň P1--P4)
`QL1 - Puls`, `QL2 - Puls`, `Q0 - Puls`, `QC2 - Puls`, `QC1 - Puls`	povel jalového výkonu
`cos-fi_L1`, `cos-fi_L2`, `cos-fi_1`, `cos-fi_C2`, `cos-fi_C1`	povel cos φ

Povelové — úrovňové (HDO): `SPG_HDO-P1` … `SPG_HDO-P4` — povel k regulaci činného výkonu.

Stavy / blokování (globální):

Funkce	Význam
H835L	stejnosměrné napájení ovládání (1 → ztráta napětí, 0 → OK)
DS_H298CBE_conf	potvrzení blokování zapnutí výrobny (1 → blokováno)
DS_H298CBE_BESS_conf	potvrzení blokování zapnutí BESS (1 → blokováno)
Total_stop	Total_stop výrobny (1 → aktivní)
Power_export_setting	omezení přetoků do distribuce (1 → aktivní): vstup podmiňuje ruční limit „Řídící povel obchodníka / dodavatele energie“ (fs_contractCmdPlimit, rizeni-vykonu.md §3.2); výstup = 1, když limit právě platí
RfG-enable	povolení zkoušek dle RfG – úroveň (1 → povoleno, rozepnutí = stop zkoušky)
RfG-seq1 … RfG-seq3	start sekvence 1--3 zkoušky dle RfG – náběžná hrana (viz zkousky-rfg.md)

Rozpadové místo (DS-…, vázané na lokaci): DS (dveře), DS-FA1-ON/OFF, DS-QM-ON/OFF, DS--H100NAT (suma U/f ochran), H100IF (porucha sumy ochran), DS-UPS-BattLo.

Předávací místo OM (TML-…): TML (dveře), spínací prvky TML-QM/QS/QSF/Q1/Q6/QE6/ FU1/FA1--ON|OFF, TML-H100NAT, TML-H850T, TML-UPS-BattLo.

Topení / HDO: Therm-1 … Therm-8 (termostaty), HDO-AKU, HDO-TUV (nízký tarif).

4.2 Výstupy (io_out_func)

Potvrzení aktuálního povelu (Control): Control-P0…P4, Control_QL1/QL2/Q0/QC2/QC1, Control_cos-fi_L1/L2/1/C2/C1, Control-Ch-P1…P4 (nabíjení BESS), Control-DCh1…DCh4 (vybíjení BESS).

Blokování a galvanické odpojení: H298CBE (blokování výrobny dle DŘ), H298CBE_BESS, GS-DS-S (= GD_Power_supply, odpojení zdroje), GD_BESS, GD_Wallbox.

Chyby a komunikace: PLC_Sys, PLC_Err (General_Err), PLC_Alive, DEV_Alive, Offline_Power_suma, Offline_BESS_suma, Offline_Wallbox_suma, Offline_ELM_TML_suma, Offline_ELM_DSi_suma, Offline_ELM_DS_suma, Offline_ELM_BackUp_suma.

Regulace přetoků / peak shaving: Power_export_setting, Comp-NO (vypnutí kompenzace při dodávce do DS), Peak_shaving_1…4 (vypínání zařízení po úrovních).

Topná tělesa (TUV / AKU): A1-L123 … A4-L123 (všechny fáze), A1-L1 … A4-L3 (jednotlivé fáze), Ext-Heating (vypnutí externího kotle), Heating, Cooling (vytápění/chlazení rozvaděče).

Zkoušky dle RfG: RfG-bit0 … RfG-bit3 – 4bitový kód aktuálního kroku zkoušky (zaokrouhleno(% / 10): 0000 = 0 %, 0001 = 10 % … 1010 = 100 %; mimo zkoušku 0000), viz zkousky-rfg.md.

Ostatní: ON (vždy sepnuto), Pulse (pulsní výstup [puls/kWh]), Total_stop, DS-H100NAT, DS_Door_open, DS-UPS-BattLo, H835L, TML_Door_open_suma, TML-UPS-BaterryLow.

---

5. Vstup jako povel (nastavení povelu)

Některé vstupní funkce nejsou jen stav, ale **povel**. Pulsní povelové vstupy se vyhodnotí při krátkém pulsu a nastaví **globální** povel:

Vstupní funkce	Nastaví
`P1 - Puls` … `P4 - Puls`	povel činného výkonu na úroveň P1--P4
`QL1/QL2/Q0/QC2/QC1 - Puls`	povel jalového výkonu na danou úroveň
`cos-fi_L1/L2/1/C2/C1`	povel cos φ na danou úroveň

Stejný **nastavený povel** lze zadat i z dalších zdrojů: z **webu**, z **Modbusu**, nebo **na konkrétním rozpadovém místě** (viz dokument o řízení výkonu). Globální povel přitom maže per-místní povely; povel na místě nastaví jen dané místo.

Aktuálně nastavený povel se promítá zpět do potvrzovacích výstupů `Control-…` (kap. 2.2) a posílá se na server v poli `msaved` (kap. 6).

---

6. Odesílání na server

Binární I/O i nastavený povel se posílají na server v rámci **měřicího paketu** (HTTP P0ST na `…/measuring` přes VPN; podmíněno zapnutým cloudem `ns_cloud`; viz `interfaces.md`).

Co paket (JSON) obsahuje:

Pole	Obsah
`m`	snímek živého stavu **všech binárních kanálů** — každý záznam má `value`, `function`, `location`, `dev_type` (`bin_input` / `bin_output`)
`msaved`	**nastavený povel** — úroveň činného výkonu (P), jalového výkonu, cos φ, stavy BESS …
`sn`, `time`, `plc_uuid`, `trigger`	identifikace jednotky, čas a příznak, zda jde o událostní odeslání
`events`, `events_dropped`	události zásahů od posledního úspěšného odeslání (zápisy Modbus registrů s IP klienta, ruční zásahy z webu s uživatelem) – viz `interfaces.md` §7
`old_logfile_size`, `logfile_size`	velikost `/log/npa_run.log` při prvním logu po startu a nyní

Kdy se paket odešle:

- **Okamžitě při změně binárního vstupu.** Změní-li se hodnota vstupu, jehož funkce je „platná” (viz podmínka v kap. 6.1), odešle se paket ihned s `trigger = true`. Tím server vidí příchozí povely i změny stavů prakticky okamžitě.
- **Periodicky** jinak — v nastaveném intervalu (lze měnit i ze serveru).

Tím server dostává jak každou změnu binárních vstupů (a tím i příchozí povely), tak pravidelný snímek výstupů a aktuálně nastaveného povelu.

6.1 Mapování všech posílaných UUID a detekce nevyužitého kanálu

Co se posílá: v poli `m` je **snímek celého živého stavu** — posílají se **všechny** binární kanály bez ohledu na to, zda mají přiřazenou funkci. Žádný kanál se z `m` nevyřazuje. Každý záznam přitom nese vlastní `value`, `function`, `location` a `dev_type` (`bin_input` / `bin_output`), takže **funkci i rozpadové místo lze namapovat přímo z paketu**, bez nutnosti párovat s konfigurací.

Seznam všech možných UUID z konfigurace (např. když je potřeba i kanály, které se zrovna neposlaly) se získá průchodem pole `device`: u **každého** zařízení se vezmou pole `input[]` a `output[]` a `uuid` každého prvku; funkce / místo / negace daného kanálu jsou v objektu zařízení pod indexovanými klíči podle kap. 1 (a platí to pro `bin_io` i pro Modbus zařízení — viz pozn. v kap. 1).

Detekce nevyužitého (zbytečného) kanálu. Jednotka považuje funkci za **platnou** (a jen taková spouští okamžité odeslání při změně), právě když:

`function` **není** prázdná (`""`), **není** `"---"` a **nezačíná** `"EINPUT/"`.

Kanál, jehož `function` je `""`, `"---"`, nebo začíná `"EINPUT/"`, tedy **není na nic nastavený** — touto podmínkou jej lze rozpoznat. Pozor na rozdíl: tahle podmínka řídí **jen** okamžité (event) odeslání; v periodickém snímku `m` se i takové kanály stále posílají. Pokud je má server zahazovat jako zbytečné, musí na pole `function` aplikovat **stejný test** sám.