

# Komunikace v bezdrátové síti Poseidon

## Abstrakt

Popis komunikace periferií firmy ENIKA.CZ se stanicemi firmy AMiT v bezdrátové síti Poseidon.

Autor: Petr Latina, Zbyněk Říha  
Dokument: ap0051\_ap\_cz\_003.pdf

## Příloha

Obsah souboru: ap0051\_ap\_cz\_002.zip

poseidon_p1_cz_002.dsox	Příklady algoritmů obsluhy přijímačů a vysílačů.

**Obsah**


---

Obsah .....	2
Historie revizí .....	3
Související dokumentace .....	3
<b>1 Definice použitých pojmů.....</b>	<b>4</b>
<b>2 Základní vlastnosti a pravidla použití systému Poseidon.....</b>	<b>5</b>
2.1 Využití rádiového kmitočtu 868 MHz.....	5
2.2 Odezvy bezdrátového systému.....	5
2.3 Využívání ovládacích funkcí systému Poseidon .....	5
2.4 Jedinečné ID přístrojů .....	6
2.5 Zabezpečení a spolehlivost.....	6
2.6 Dosah.....	6
<b>3 Stanice AMiT v síti Poseidon .....</b>	<b>7</b>
3.1 Způsob komunikace.....	7
3.2 Časové posloupnosti komunikace, priorita.....	7
<b>4 Tvorba aplikace pro komunikaci v síti Poseidon .....</b>	<b>9</b>
4.1 Definice komunikace Poseidon v projektu .....	9
4.2 Definice periférií .....	11
4.2.1 Vlastnosti objektů na obrazovkách.....	12
Zobrazení analogových hodnot.....	12
Nastavení analogových hodnot.....	13
Zobrazení binárních hodnot .....	14
Nastavení binárních hodnot .....	15
4.2.2 Vlastnosti objektů v procesech.....	16
Zpracování analogové hodnoty od vysílače a zaslání přijímači .....	16
Zpracování stisku bezdrátového tlačítka .....	17
Ovládání žaluzií.....	17
4.2.3 Využití obrazovkových prvků v kombinaci s procesy .....	17
Stav ovládacího prvku dle skutečného stavu výstupu periferie.....	17
4.3 Vytvoření vazby mezi stanicí AMiT a perifériemi .....	20
4.3.1 SW Poseidon Asistent.....	20
4.3.2 Párovací režim .....	21
Párování periférií typu přijímač .....	21
Párování periférií typu vysílač.....	22
4.4 Přehrání aplikace .....	24
<b>5 Diagnostika komunikace .....</b>	<b>26</b>
5.1 Síla signálu .....	26
5.2 Stav komunikace s perifériemi .....	26
<b>6 Seznam podporovaných periférií .....</b>	<b>27</b>
<b>7 Univerzální aplikace.....</b>	<b>29</b>
<b>8 Technická podpora .....</b>	<b>30</b>
<b>9 Upozornění .....</b>	<b>31</b>

**Historie revizí**

Verze	Datum	Autor změny	Změny
001	29. 07. 2014	Latina Petr	Nový dokument
002	27. 01. 2021	Říha Zbyněk	Změny dle chování DetStudia verze 2.x, změna struktury dokumentu.
003	12. 02. 2021	Říha Zbyněk	Oprava odkazů v kapitolách 4.4 a 7.

**Související dokumentace**

1. Nápověda k vývojovému prostředí DetStudio – Nápověda prostředí  
soubor: Ovladani\_cs.chm
2. Nápověda k vývojovému prostředí DetStudio – Nápověda EsiDet  
soubor: Esidet\_cs.chm
3. **AMR-CP24/01** – Komunikační jednotka – Návod na obsluhu  
soubor: amr-cp2401\_g\_cz\_xxx.pdf
4. **AMR-CP44/DM RevA** – Komunikační jednotka – Návod na obsluhu  
soubor: amr-cp4xdm\_reva\_g\_cz\_xxx.pdf
5. **AMR-CP46/DM RevA** – Komunikační jednotka – Návod na obsluhu  
soubor: amr-cp4xdm\_reva\_g\_cz\_100.pdf
6. **AMR-OP70RHP/xx** – Programovatelný nástěnný ovladač – Návod na obsluhu  
soubor: amr-op70rhpxx\_g\_cz\_xxx.pdf
7. **AMR-OP71RHP/xx** – Programovatelný nástěnný ovladač – Návod na obsluhu  
soubor: amr-op71rhpxx\_g\_cz\_xxx.pdf
8. **AMR-OP87/Pxx** – Grafický operátorský panel – Návod na obsluhu  
soubor: amr-op87pxx\_g\_cz\_xxx.pdf

# 1 Definice použitých pojmů

---

**Poseidon®**

Bezdrátová síť pracující na frekvenci 868 MHz, kterou vyvinula společnost ENIKA.CZ.

**Systém Poseidon**

Souhrn periférií komunikujících v bezdrátové síti Poseidon.

**Vysílač**

Periferie systému Poseidon vysílající zprávu (např. stisk tlačítka nebo vyslaná hodnota snímače fyzikální veličiny, např. teploty).

**Přijímač**

Periferie systému Poseidon zpracovávající přijatou zprávu od vysílače.

**ID**

Jedinečné identifikační číslo každé periferie systému Poseidon.

**EsiDet**

Editor návrhového prostředí DetStudio sloužící pro tvorbu aplikací.

## 2 Základní vlastnosti a pravidla použití systému Poseidon

Bezdrátový systém Poseidon nabízí některé nové možnosti v oblasti měření a regulace, na druhou stranu je nutné zohlednit určitá omezení a vlastnosti, které se v klasické „drátové“ automatizaci nevyskytují.

### 2.1 Využití rádiového kmitočtu 868 MHz

Systém Poseidon pracuje ve **veřejném kmitočtovém pásmu 868 MHz**, pro jehož využívání jsou stanovena přesná pravidla. Pravidla jsou definována ve Všeobecném oprávnění č. VO-R/10/05.2014-3 k využívání rádiových kmitočtů a k provozování zařízení krátkého dosahu, které vydává Český telekomunikační úřad. Z technických parametrů např. vyplývá, že stanice na kmitočtu 868 MHz s maximálním vyzářeným výkonem 25 mW e.r.p. (efektivní vyzářený výkon), mohou využívat pásmo s klíčovací poměrem  $\leq 1,0 \%$ . Klíčovací poměr je podíl času, kdy zařízení vysílá, v rámci jakékoliv jedné hodiny. Laicky řečeno, nelze kmitočet obsadit vysílacím signálem neomezeně dlouho. Z tohoto důvodu všechny periferie systému Poseidon včetně stanic AMiT zajišťují splnění limitu klíčovacího poměru automaticky a nedovolí uživateli úmyslně či neúmyslně pásmo nadměru obsadit.

### 2.2 Odezvy bezdrátového systému

Jak vyplývá z předchozího odstavce, nelze na rádiové pásmo nahlížet podobně jako na „drátové“ propojení s neomezeným datovým tokem. Vysílače systému Poseidon vysílají zprávu ve velmi krátkém časovém intervalu kolem 5 ms ihned při vzniku události (stisk tlačítka, odeslání měřené hodnoty např. teploty apod.). Jinou situací je opakovaný požadavek na vyčítání stavů některých periférií (poloha žaluzií, hodnota stmívačů) nadřazeným systémem pomocí stanic AMiT. Aby se zamezilo přetížení kmitočtového pásma, je ve stanicích AMiT nastavena definovaná **prodleva opakovaného čtení** stavu. Podobná situace s časovou prodlevou je i u zápisu hodnot do bezdrátových periférií (DALI předřadníky, stmívače, ovládání žaluzií, spínací relé apod.). Aby se zabránilo opakovanému zasílání téže hodnoty, zasílá stanice AMiT hodnoty až při jejich změně (zapni/vypni nebo změna hodnoty o více než 1). Způsob komunikace a časové posloupnosti komunikace popisuje detailněji tento dokument v dalších kapitolách.

### 2.3 Využívání ovládacích funkcí systému Poseidon

Systém Poseidon je navržen tak, aby základní ovládací a regulační funkce řešil už na úrovni samotných periférií systému bez nutnosti nadřazeného systému. Například, ovládání světel může probíhat napřímo mezi vysílači (nástěnná tlačítka, klíčenky) a přijímači (spínače světel, stmívače). Reakce na stisk tlačítka je prakticky okamžitá (v řádu desítek milisekund). Nadřazený systém má přesto možnost zjistit aktuální stav osvětlení (světlo svítí / nesvítí, aktuální intenzita) a navíc může přijímače ovládat z pozice nadřazené funkce (centrální vypnutí / zapnutí). Proto doporučujeme **VŽDY** využívat všech předdefinovaných vyšších funkcí periférií systému Poseidon (které lze konfigurovat pomocí programu Poseidon Asistent) a propojení s nadřazeným systémem využívat pro globální ovládání a pro monitoring stavu a hodnot bezdrátových periférií.

Systém Poseidon na úrovni vysílačů a přijímačů řeší například tyto úlohy:

- ♦ Ovládání světel (zapnutí, vypnutí, stmívání, scény apod.).
- ♦ Časové funkce při spínání / vypínání světel a výstupních relé obecně.
- ♦ Ovládání rolet a žaluzií včetně přesného natáčení lamel.
- ♦ Regulace intenzity osvětlení na základě měření hladiny venkovního osvětlení.
- ♦ Akce na základě detekce pohybu od snímačů pohybu.
- ♦ Použití prioritních alarmových signálů pro okamžitou reakci periférií (např. ochrana žaluzií před silným větrem).

## 2.4 Jedinečné ID přístrojů

---

Každá periferie systému Poseidon má své **jedinečné identifikační číslo (ID)**, které je jí přiřazeno z výroby a je uživatelsky neměnné. Již vytvořené vazby mezi periferiemi jsou vázány na konkrétní ID, a proto je při výměně periferie nutné měnit i tyto vazby. Jedinečné ID zvyšuje bezpečnost provozu bezdrátové sítě a identifikovatelnost všech periferií.

ID každé periferie sítě Poseidon je uvedeno na samolepce dodávané s periferií. U stanic firmy AMiT je samolepka nalepena u konektoru s rozhraním Poseidon. ID lze načíst také pomocí SW Poseidon Asistent (viz kapitola 4.3.1 „SW Poseidon Asistent“).

## 2.5 Zabezpečení a spolehlivost

---

Při vysílání v daném frekvenčním pásmu jsou data volně přístupná v celém vysílacím dosahu. Proto systém Poseidon ve svém protokolu používá několik technik kódování a zabezpečení přenosu dat, aby se zvýšila spolehlivost přenosu a především, aby se zabránilo neoprávněnému zásahu do ovládání periferií. V parametrizačním programu Poseidon Asistent jsou navíc k dispozici postupy a funkce, které umožňují „zneviditelnění“ daných periferií v projektu a zaheslování projektu proti neoprávněnému použití. Doporučujeme této oblasti věnovat zvýšenou pozornost.

## 2.6 Dosah

---

Každá periferie systému Poseidon je navržena tak, aby zajistila spolehlivý přenos dat do vzdálenosti **150 m ve volném prostoru**. V případě použití systémového příslušenství k periferiím lze zaručit dosah až na 3 km. Reálný dosah je v praxi podstatně menší a záleží na počtu překážek a materiálu, přes který se signál šíří a na velikosti případného lokálního rušení od jiných zdrojů. S výhodou lze využít funkci **opakovače signálu** u každého přijímače, který je napájený ze sítě.

## 3 Stanice AMiT v síti Poseidon

Stanice AMiT se v síti Poseidon může současně chovat jako vysílač i přijímač, tzn., že může vysílat povely, dotazovat se na stav výstupů periférií nebo přijímat ze sítě informace např. v podobě analogové hodnoty (teplota, vlhkost).

Stanice AMiT může:

- ♦ zprostředkovávat výměnu informací mezi nadřazeným řídicím systémem a sítí Poseidon,
- ♦ být nadřazeným systémem pro síť Poseidon.

### 3.1 Způsob komunikace

#### Emulace stisku tlačítka

Pomocí emulace stisku tlačítka dokáže stanice AMiT vyslat povel stejně jako fyzický vysílač (tlačítko klíčenky apod.). Stanice AMiT se tedy chová přímo jako periférie z produkce firmy ENIKA. Prostřednictvím SW Poseidon Asistent (viz kapitola 4.3.1 "SW Poseidon Asistent") lze v perifériích nastavit způsob reakce vybrané periférie na emulovaný stisk tlačítka ve stanici AMiT.

#### Povelování přijímačů

Perifériím, které umožňují přijímat povely ze stanic AMiT, lze měnit stav jejich výstupů. Rozdíl tohoto povelování od emulace stisku tlačítka je takový, že lze ovládané periférie povelovat pouze v „základním režimu“. V základním režimu periférie vykonává funkci předdefinovanou výrobcem, např. periférie s reléovým výstupem by vykonávala funkci Vyp/Zap. Při povelování nelze v SW Poseidon Asistent nastavovat způsob reakce vybrané periférie na povel ze strany stanice AMiT.

#### Dotazování přijímačů

Periférie, které umožňují přijímat povely, umožňují na dotaz nebo automaticky odeslat aktuální stav (případně hodnotu) výstupů. Tato funkčnost je opět dána firmware konkrétní periférie.

#### Zpracování přijatých zpráv

V průběhu komunikace se zpracovávají přijaté zprávy od vysílačů. Přijaté zprávy mohou obsahovat následující typy hodnot:

- ♦ analogová hodnota (např. teplota, vlhkost apod.),
- ♦ procentuální hodnota v rozsahu 0 % až 100 %,
- ♦ binární hodnota (např. stav obsazeno = True u pohybového čidla nebo stav výstupu = True u relé),
- ♦ stav tlačítka na fyzickém vysílači (např. stav stisknuto = True).

### 3.2 Časové posloupnosti komunikace, priorita

V následujícím textu bude popsáno, jaké jsou dány časové posloupnosti a priority komunikace povelovacích a dotazovacích zpráv z pohledu stanice AMiT. Časové posloupnosti jsou voleny na základě doporučení tvůrce komunikace Poseidon, firmy ENIKA.CZ.

#### Povelování

Povelování má **nejvyšší** prioritu, může se jednat o emulaci stisku tlačítka nebo odeslání povelu např. pro ovládní relé Vyp/Zap. Tento typ požadavku se prakticky okamžitě vykonává. Vznikne-li více požadavků, zařadí se do fronty a postupně se vykonávají.

Mezi vykonáním dvou a více povelovacích požadavků je časová prodleva **100 ms**. Nepřijde-li po odeslání konkrétního povelu odpověď od povelované periférie, vyhodnocuje se Timeout, a **5 s** se nevysílá žádný povel (čeká se na „vyčištění“ éteru). Potom se může vysílat další povel.

**Pozor!**

*Povelovací mechanismus automaticky hlídá, zda se změnila hodnota, která se má odeslat. Jestliže nedošlo ke změně o hodnotu 1 vůči hodnotě předchozí, povel se nevyšle. Tím je zabezpečeno opakované vyslání téže hodnoty do éteru.*

**Dotazování**

Jsou-li vykonány všechny povelovací požadavky, probíhá dotazování jednotlivých periférií na stav nebo hodnotu výstupů. Dotazování má **nižší** prioritu. „**Seznam**“ dotazovaných periférií se vytváří automaticky, a to v takovém pořadí, v jakém jsou vkládány jednotlivé objekty do projektu pro stanici AMiT, viz kapitola 4.2 „Definice periférií“.

Dotazovací požadavky se vysílají postupně pro jednotlivé periferie, tak jak jsou vloženy v seznamu dotazovaných periférií. Bude-li vykonán dotaz pro poslední periférii, začíná se opět dotazovat od první periferie v seznamu.

Mezi jednotlivými dotazy je časová prodleva **1 s**. Nepřijde-li po odeslání konkrétního dotazu odpověď od periferie, vyhodnocuje se opět Timeout a **5 s** se čeká na „vyčištění“ éteru. Potom se může vysílat následující dotaz.

**Poznámka:**

*V době vydání této aplikační poznámky je protokolem Poseidon dáno, že lze uskutečnit jeden dotaz pro jeden výstup konkrétní periferie, tzn., pokud se např. dotazuje na periférii, která má 8 reléových výstupů, bude doba dotazování na tuto periférii trvat 8 s.*

**Zprávy od vysílačů**

Podobně jako vzniká seznam dotazovaných periférií, vzniká automaticky při vytváření projektu pro stanici AMiT také seznam periférií, od kterých se očekává zpráva obsahující např. analogovou hodnotu. Jsou to zprávy, které jsou generovány stiskem tlačítek nebo snímačem teploty či vlhkosti nebo vysílačem stavu kontaktů apod. Zprávy jsou po zaznamenání připraveny k okamžitému použití, např. v podobě hodnoty měřené teploty.



## 4 Tvorba aplikace pro komunikaci v síti Poseidon

Pro úspěšnou komunikaci stanice AMiT v síti Poseidon, musí být vytvořena aplikace v editoru **EsiDet** vývojového prostředí **DetStudio**. Obecně lze doporučit, aby zaslání požadavků mezi periferiemi nebylo realizováno prostřednictvím stanice AMiT (požadavek je zaslán do stanice AMiT a následně je stanicí přeposlán dále). Dochází tím ke zbytečným prodlevám při zpracování. Daleko výhodnějším řešením je realizace vazeb napřímo mezi periferiemi s tím, že stanice AMiT pouze dostává informace o nastavené a měřené hodnotě a případně ovlivňuje stav ovládaných periferií.

Veškeré ukázky, které jsou uvedeny v této kapitole, jsou také součástí přílohy ap0051\_ap\_cz\_xxx.zip.

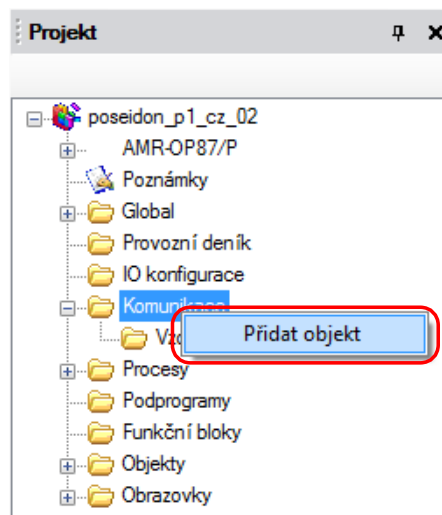
Úspěšné vytvoření komunikace v síti Poseidon lze rozdělit do několika kroků:

- ◆ definice komunikace Poseidon v projektu,
- ◆ definice (emulace) periferií, se kterými se má komunikovat,
- ◆ vytvoření vazby mezi stanicí AMiT a periferiemi.

Tvorba projektu je shodná pro celé portfolio produkce firmy AMiT, které obsahuje komunikační rozhraní Poseidon. V rámci aplikační poznámky bude komunikace naprogramovaná pro grafický operátorský panel **AMR-OP87/Pxx** s komunikačním modulem bezdrátového systému Poseidon 868 MHz **AW-P868A**.

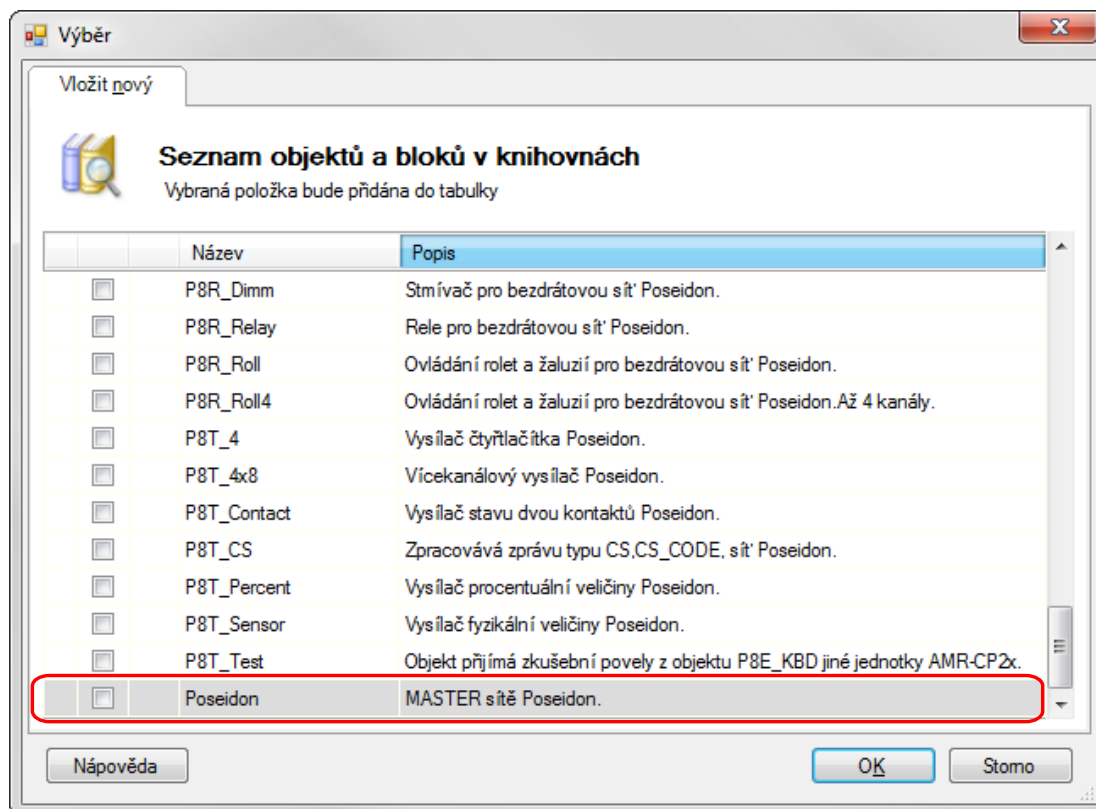
### 4.1 Definice komunikace Poseidon v projektu

Pro definici komunikace Poseidon je nutné vložit do projektu objekt **Poseidon**. Jedná se o hlavní komunikační objekt sítě Poseidon. Do projektu lze vložit z okna se seznamem komunikačních objektů. Okno se seznamem lze otevřít vyvoláním kontextového menu nad složkou „Komunikace“ a výběrem položky „Přidat objekt“.



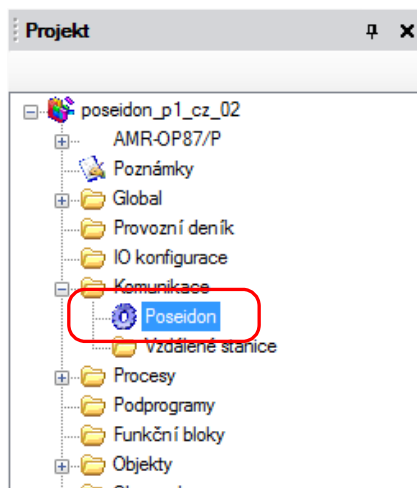
Obr. 1 – Přidání objektu do projektu

Po kliknutí na položku „Přidat objekt“ dojde k otevření okna „Seznam objektů v knihovnách“, kde lze (mimo jiné) vybrat objekt **Poseidon**.



Obr. 2 – Výběr objektu Poseidon

Jakmile bude výběr objektu potvrzen (kliknutím na tlačítko „**OK**“ nebo dvojklikem na objekt), zobrazí se v okně „Projekt“ ve složce „Komunikace“.

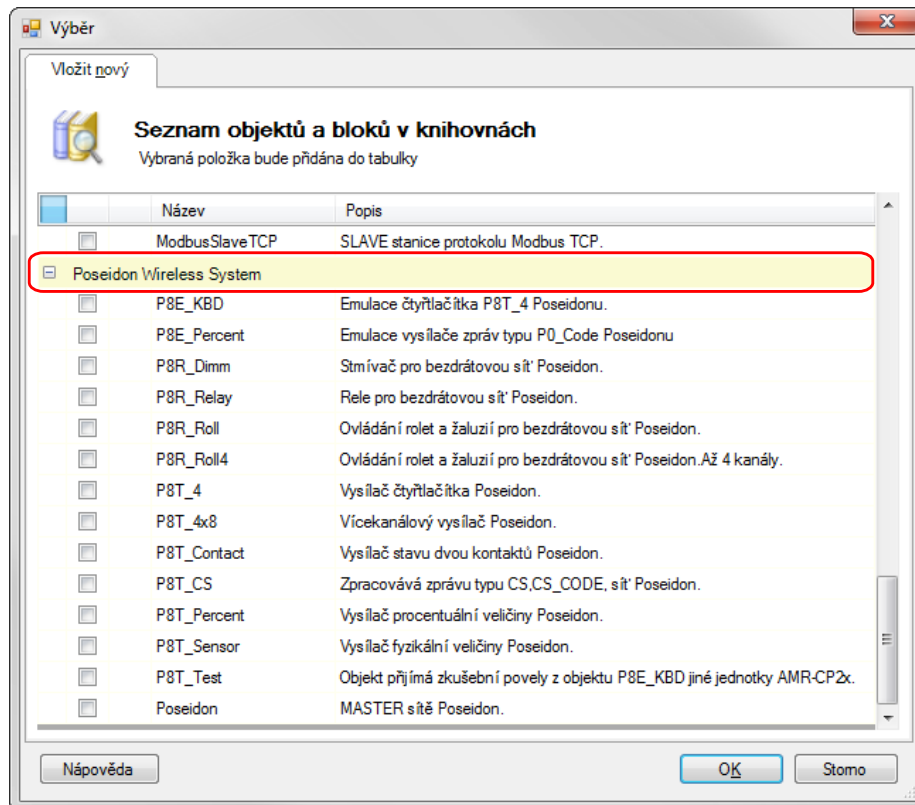


Obr. 3 – Objekt Poseidon ve složce „Komunikace“

Kliknutím levým tlačítkem myši na objekt Poseidon se v okně „Vlastnosti“ zobrazí nastavení komunikačního objektu. Nastavení lze ponechat ve výchozím stavu (vlastnost **AssistentEnable** nastavena na True), aby bylo možné používat panel pro spojení se SW „Poseidon Asistent“. SW je z produkce firmy ENIKA a slouží pro správu bezdrátové sítě Poseidon (viz kapitola 4.3.1 „SW Poseidon Asistent“).

## 4.2 Definice periferií

Periferie lze v projektu nadefinovat stejně jako vlastní komunikaci v síti Poseidon – z okna se seznamem komunikačních objektů. Pro komunikaci v síti Poseidon lze využít jakýkoliv objekt typu P8x\_xxx ze sekce „Poseidon Wireless System“.



Obr. 4 – Sekce „Poseidon Wireless System“

Seznam všech dostupných objektů, včetně periferií bezdrátové sítě Poseidon, pro které jsou objekty určeny lze nalézt v nápovědě k části **EsiDet** vývojového prostředí **DetStudio**.

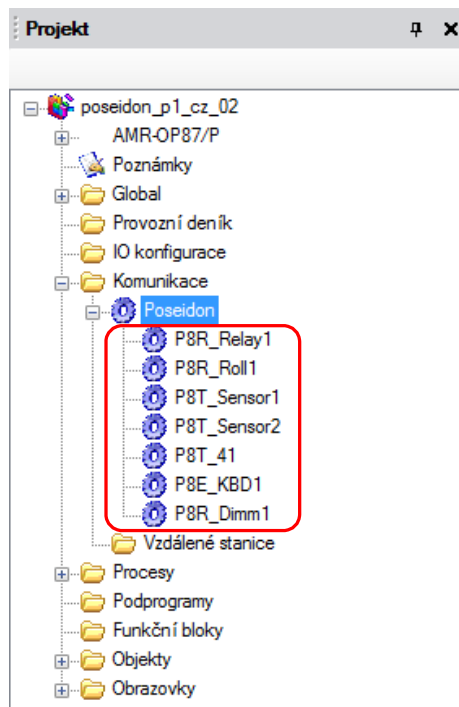
V rámci ukázkové aplikace bude probíhat komunikace s periferiemi:

- ◆ P8 R 2 N/K – dvoukanálový přijímač,
- ◆ P8 R R I – roletový vestavný přijímač,
- ◆ P8 R DALI N – vestavný přijímač s DALI výstupem,
- ◆ P8 T Temp/RH – vysílač teploty a vlhkosti,
- ◆ P8 T 4 – nástěnný čtyřkanálový vysílač.

Na základě informací, uvedených v nápovědě k části EsiDet vývojového prostředí **DetStudio**, je nutné do projektu vložit objekty:

- ◆ **P8R\_Relay** pro ovládání dvoukanálového přijímače,
- ◆ **P8R\_Roll** pro ovládání roletového přijímače,
- ◆ **P8R\_Dimm** pro vestavný přijímač s DALI výstupem,
- ◆ **P8T\_Sensor** pro vysílač teploty a vlhkosti (pro každou hodnotu jeden objekt),
- ◆ **P8T\_4** pro nástěnný čtyřkanálový vysílač.

Roletový přijímač se bude ovládat také pomocí tlačítek emulovaných panelem. Pro emulaci tlačítek je nutné vložit také objekt **P8E\_KBD**.



Obr. 5 – Definice periferií sítě Poseidon

Kliknutím pravým tlačítkem myši na jednotlivé objekty lze v panelu „Vlastnosti“ provést nastavení komunikace.

Objektům **P8R\_Relay** a **P8R\_Dimm** je nutné nastavit počet kanálů (v závislosti na způsobu párování – viz kapitola 4.3 „Vytvoření vazby mezi stanicí AMiT a periferií“), který je fyzicky na periferiích P8 R 2 N/K a P8 R DALI N. Počet kanálů je dán vlastností **ChannelCount**. Pokud bude vlastnost obsahovat jiný počet kanálů, nebude komunikace funkční.

### **Poznámka**

Objektům **P8R\_xxx** lze mimo jiné nastavit vlastnost **FailureThreshold**, pomocí které lze nastavit počet nedoručených zpráv před hlášením chyby komunikace s periferií.

Objektům **P8T\_xxx** (které aktivně zasílají data bez zásahu uživatele) lze mimo jiné nastavit vlastnost **TimeOut**, která udává maximální dobu čekání na zprávu od periferie.

Obě výše zmíněné vlastnosti lze ponechat ve výchozím stavu.

Po vložení objektů, lze pracovat s jejich vlastnostmi jak na obrazovkách, tak v procesech.

## **4.2.1 Vlastnosti objektů na obrazovkách**

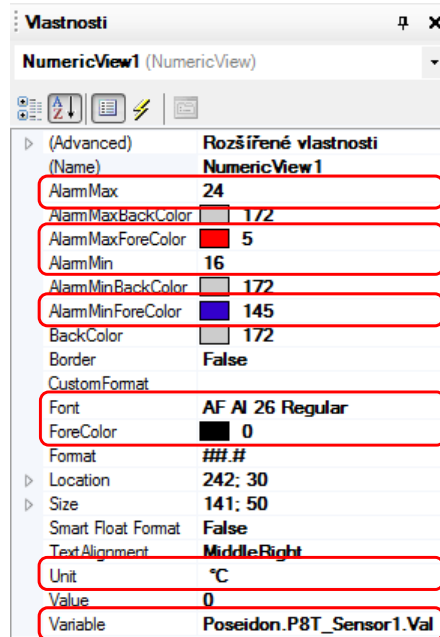
Pokud je požadavek, aby panel data nezpracovával na regulační úrovni, ale prováděl pouze zobrazení obdržených data na obrazovkách, případně zasílal povely (např. jako reakci na stisk tlačítka). Není třeba nic programovat na úrovni procesů. Pouze se využije možností programování a parametrizace obrazovek.

### **Zobrazení analogových hodnot**

Při požadavku na zobrazení obdržené analogové hodnoty od vysílače lze využít prvek **NumericView**. Typickým příkladem může být požadavek na zobrazení teploty, vlhkosti, či aktuální polohy rolet nebo žaluzií.

Po vložení prvku **NumericView** na obrazovku lze jednoduše za jeho vlastnost **Variable** dosadit požadovanou vlastnost (např. vlastnost **Value** objektu **P8T\_Sensor1** pro zobrazení teploty nebo

vlastnost `ActualPosition` objektu `P8R_Ro11` pro zobrazení aktuální polohy žaluzií či rolet). Ostatní parametry prvku `NumericView` lze nastavit dle požadavků programátora na vzhled.

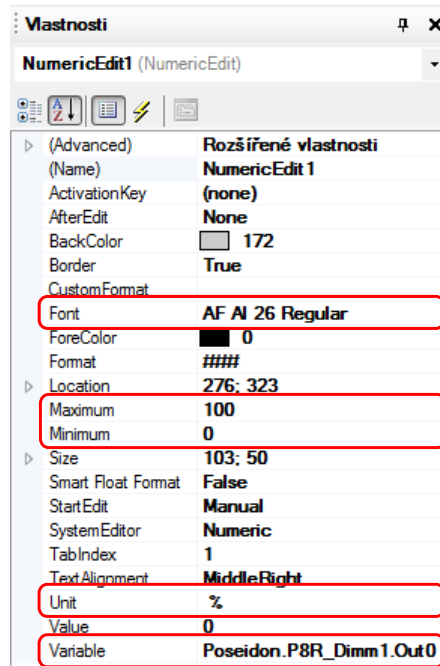


Obr. 6 – Nastavení prvku `NumericView`

### Nastavení analogových hodnot

Při požadavku na nastavení požadované analogové hodnoty lze využít prvek `NumericEdit`. Typickým příkladem může být požadavek na nastavení úrovně osvětlení či aktuální polohy rolet nebo žaluzií.

Po vložení prvku `NumericEdit` na obrazovku lze jednoduše za jeho vlastnost `Variable` dosadit požadovanou vlastnost (např. vlastnost `out0` objektu `P8R_Dimm1` pro nastavení požadované intenzity osvětlení prostřednictvím DALI). Ostatní parametry prvku `NumericEdit` lze nastavit dle požadavků programátora na vzhled.

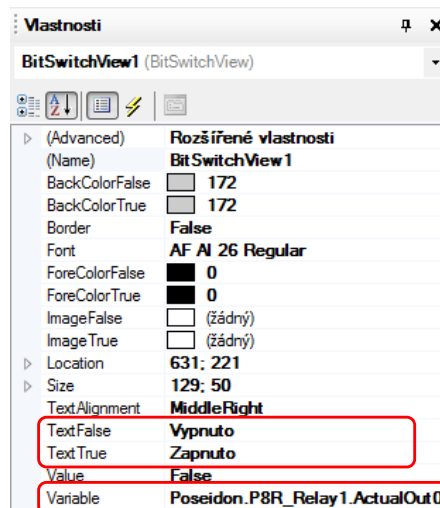


Obr. 7 – Nastavení prvku NumericEdit

### Zobrazení binárních hodnot

Při požadavku na zobrazení obdržené binární hodnoty od periferie lze využít prvek `BitSwitchView`. Typickým příkladem může být požadavek na zobrazení stavu relé či informaci o pohybu v místnosti od PIR.

Po vložení prvku `BitSwitchView` na obrazovku lze jednoduše za jeho vlastnost `Variable` dosadit požadovanou vlastnost (např. vlastnost `ActualOut0` objektu `P8R_Relay1` pro zobrazení stavu relé 1). Ostatní parametry prvku `BitSwitchView` lze nastavit dle požadavků programátora na vzhled.



Obr. 8 – Nastavení prvku BitSwitchView

## Nastavení binárních hodnot

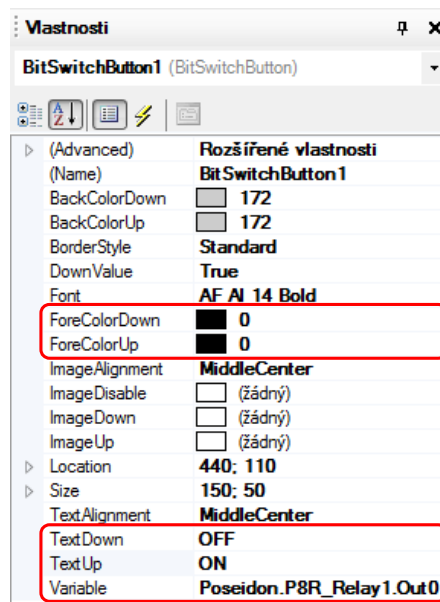
Nastavení binárních hodnot lze rozdělit do dvou skupin:

- ♦ povelování,
- ♦ emulace stisku tlačítka.

### Povelování

Povelování používá pro komunikaci objekt **P8R\_xxx**, který je v rámci prostředí **DetStudio** přímo určen pro ovládání zvolené periferie (např. objekt **P8R\_Relay** je přímo určen pro periferii P8 R 2 N/K).

Při požadavku na nastavení požadovaného stavu binárního výstupu z obrazovek lze využít např. prvek **BitSwitchButton**. Po vložení prvku **BitSwitchButton** na obrazovku lze jednoduše za jeho vlastnost **Variable** dosadit požadovanou vlastnost (např. vlastnost **Out0** objektu **P8R\_Relay1** pro nastavení požadovaného stavu relé 1 na periferii P8 R 2 N/K). Ostatní parametry prvku **BitSwitchButton** lze nastavit dle požadavků programátora na vzhled.



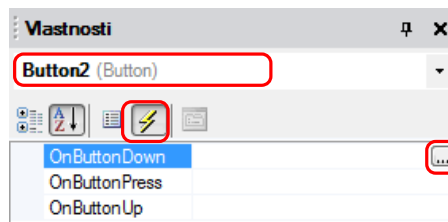
Obr. 9 – Nastavení prvku **BitSwitchButton**

### Emulace stisku tlačítka

Emulace stisku tlačítka používá pro komunikaci objekt **P8E\_KBD**, který simuluje stisk bezdrátového tlačítka. Stanice AMiT se tedy v případě využití objektu **P8E\_KBD** chová stejně jako samostatné bezdrátové tlačítko. Funkci emulace bezdrátového tlačítka lze využít např. pro ovládání žaluzií či rolet pomocí grafických tlačítek umístěných na displeji stanice AMiT.

Při práci s objektem **P8E\_KBD** nelze na obrazovkách uživatelsky přívětivě využít prvek **BitSwitchButton**. Objekt **P8E\_KBD** vyžaduje (po nastavení požadovaného chování virtuálního tlačítka) nastavení příznaku, který odešle odpovídající zprávu do sítě Poseidon. Funkci lze na obrazovkách v prostředí **DetStudio** naprogramovat pouze pomocí prvků, které umožňují tzv. skriptování na obrazovkách (prvek **BitSwitchButton** využití skriptu neumožňuje). Lze však využít např. prvky **Button** či **ToggleButton**.

Po vložení prvku **Button** na obrazovku lze využít jeho událost „OnButtonDown“. Na seznam událostí prvku **Button** se lze přepnout pomocí tlačítka se žlutým bleskem v panelu „Vlastnosti“ (při vybraném prvku **Button**).



Obr. 10 – Události prvku **Button**

Kliknutím na tlačítko „...“ dojde k vytvoření vybrané události a přepnutí do skriptovací části obrazovky.

Do nadefinovaných událostí lze vložit požadovaný kód.

```

event Button1_OnButtonDown()
    Poseidon.P8E_KBD1.OFF1 = true; //Informace o stisku levého dolního tlačítka
    Poseidon.P8E_KBD1.ON1 = false; //Zrušení informace o stisku levého horního
tlačítka
    Poseidon.P8E_KBD1.Transmit = true; //Pokyn pro vyslání informace o stisku
tlačítka
end;

event Button2_OnButtonDown()
    Poseidon.P8E_KBD1.OFF1 = false; //Zrušení informace o stisku levého dolního
tlačítka
    Poseidon.P8E_KBD1.ON1 = true; //Informace o stisku levého dolního tlačítka
    Poseidon.P8E_KBD1.Transmit = true; //Pokyn pro vyslání informace o stisku
tlačítka
end;

```

Reakce na stisk emulovaného tlačítka je dána periferií, která má s emulovanými tlačítky vytvořenu vazbu. V případě vazby na P8 R R I (ovládání rolet či žaluzií) tak může dojít např. k otevření či zavření žaluzií. Funkce je dána nastavením v SW Poseidon Asistent (viz kapitola 4.3.1 „SW Poseidon Asistent“).

## 4.2.2 Vlastnosti objektů v procesech

Využití vlastností komunikačních objektů Poseidon v procesech je na stejném principu jako práce s jinými komunikačními objekty či proměnnými.

### Zpracování analogové hodnoty od vysílače a zaslání přijímači

Při požadavku na zpracování měřené teploty či vlhkosti lze pouze zavolat vlastnost `Val` objektů `P8T_Sensor1` (měřená teplota) a `P8T_Sensor2` (měřená vlhkost). Veličina, která je přenášena jednotlivými objekty `P8T_Sensorx` je dána typem bezdrátového vysílače a počtem veličin, které do sítě poskytuje (viz sekce „Párování periferií typu vysílač“ v kapitole 4.3.2 „Párovací režim“).

Pokud by měla stanice spínat bezdrátové relé v závislosti na měřené teplotě, je možné funkcionalitu naprogramovat pomocí modulu `Hyst` následovně:

```
Hyst1(In = Poseidon.P8T_Sensor1.Val, Out => Poseidon.P8R_Relay1.Out1);
```

Výše uvedeným kódem dojde k sepnutí relé 2 na periférii P8 R 2 N/K v případě, že teplota měřená bezdrátovým vysílačem teploty překročí limit (nadefinovaný vlastností `Limit` bloku `Hyst1`) navýšený o polovinu hodnoty zadané ve vlastnosti `Hysteresis` bloku `Hyst1`.

Pokud je požadavek na získání skutečného stavu výstupu, lze využít vlastnost `ATSO` nebo `OutputsRead`. Vlastnost `ATSO` nabývá hodnoty `True` v případě, kdy periférie zaslala aktuální stavy svých výstupů. Vlastnost `OutputsRead` nabývá hodnoty `True` v případě, že jsou dotazováním



získány všechny stavy výstupů dané periferie (viz návod k části EsiDet vývojového prostředí **DetStudio**).

Aktuální stavy výstupů jsou uloženy do vlastností **ActualOutx**.

Využití vlastnosti **OutputsRead** a **ActualOutx** je např. v případech, kdy je periferie v síti Poseidon ovládána z více míst najednou. Typicky bezdrátovým tlačítkem a stanicí AMiT.

### **Zpracování stisku bezdrátového tlačítka**

Tlačítkové vysílače zasílají při stisku tlačítka informaci o jeho stisku a při uvolnění tlačítka informaci o jeho uvolnění. Pokud uživatel stiskne tlačítko na krátkou dobu, jednoduše se může stát, že by stanice AMiT stisk nezaznamenala. Z toho důvodu je doporučeno vložit zpracování stisku tlačítek do procesu s periodou 100 ms a méně. Navíc je často použito řešení, kdy je jedno tlačítko na periférii jako zapínací a druhé jako vypínací. I toto musí být v rámci kódu ve stanici ošetřeno.

Po vložení objektu **P8T\_4** do projektu by pak výsledný kód pro zpracování tlačítek pro zapnutí a vypnutí mohl vypadat následovně.

```
RS1 (
    Set = Poseidon.P8T_41.ON1,
    Reset = Poseidon.P8T_41.OFF1,
    Output => @Status
);
```

Alias **@Status** bude mít hodnotu **True** nebo **False** v závislosti na tom, zda uživatel stiskne levé horní nebo levé dolní tlačítko na čtyřtlačítkovém vysílači.

### **Ovládání žaluzií**

Stanice AMiT umožňuje ovládání žaluzií dvěma způsoby:

- ♦ emulací tlačítka (viz sekce „Emulace stisku tlačítka“ v kapitole 4.2.1 „Vlastnosti objektů na obrazovkách“),
- ♦ povelováním (zasláním požadované hodnoty otevření / náklonu žaluzií).

Zaslání požadované hodnoty otevření / náklonu žaluzií z procesů lze využít např. v případě, kdy je požadavek, aby žaluzie reagovaly na aktuální polohu slunce. Polohu slunce lze zjistit pomocí objektu **Astro** (viz návod k části EsiDet prostředí **DetStudio**). Závislost otevření či náklonu žaluzií na poloze slunce je plně dána uživatelským algoritmem.

### **4.2.3 Využití obrazovkových prvků v kombinaci s procesy**

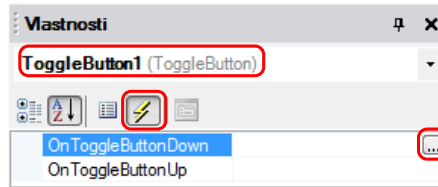
V některých případech je vhodné doplnit k ovládání na obrazovkách také další funkce. Typickým případem může být změna stavu ovládacího prvku na obrazovce (např. přepínacího tlačítka) v závislosti na skutečném stavu výstupu periferie (např. stavu relé).

#### **Stav ovládacího prvku dle skutečného stavu výstupu periferie**

Přepínání dle skutečného stavu lze využít např. v případech, kdy je periferie ovládána z více míst (bezdrátová tlačítka, PIR detektor, stanice AMiT, atd.). Při řešení změny stavu ovládacího prvku na obrazovce v závislosti na aktuálním stavu výstupu periferie (vlastnost **ActualOutx**) nelze jednoduše využít prvek **BitSwitchButton**, použitý v sekci „Nastavení binárních hodnot“ kapitoly 4.2.1 „Vlastnosti objektů na obrazovkách“. Je nutné použít např. prvek **ToggleButton** či **SelectButton** (a jim podobné), které lze ovládat také ze skriptu obrazovky. Dále je nutné naprogramovat funkci, která po určité době od povelu zjistí aktuální stav výstupu, dle kterého případně upraví stav tlačítka na obrazovce. Pro danou problematiku je možné využít jeden blok **TimerOff** a dva aliasy.

## Ovládání binárního výstupu se zjištěním skutečného stavu

Pro ovládání binárního výstupu lze využít prvek `ToggleButton`. Po vložení prvku na obrazovku lze využít jeho události „`OnToggleButtonDown`“ a „`OnToggleButtonUp`“. Na seznam událostí prvku `ToggleButton` se lze přepnout pomocí tlačítka se žlutým bleskem v panelu „Vlastnosti“ (při vybraném prvku `ToggleButton`).



Obr. 11 – Události prvku `ToggleButton`

Kliknutím na tlačítko „...“ dojde k vytvoření vybrané události a přepnutí do skriptovací části obrazovky.

Do nadefinovaných událostí lze vložit požadovaný kód. V případě ukázkové aplikace se nastaví příznak (je třeba nadefinovat jako alias `@Change` v procesu) o požadavku na změnu stavu a požadovaný stav relé (příznak o požadavku na změnu stavu bude sloužit pro spuštění časovače bloku `TimerOff`).

```
event ToggleButton1_OnToggleButtonDown()
    Process1.@Change = true; //Nastavení příznaku o změně
    Poseidon.P8R_Relay1.Out0 = true; //Nastavení požadovaného stavu relé
end;
event ToggleButton1_OnToggleButtonUp()
    Process1.@Change = true; //Nastavení příznaku o změně
    Poseidon.P8R_Relay1.Out0 = false; //Nastavení požadovaného stavu relé
end;
```

Dále je nutné v periodickém procesu nadefinovat blok `TimerOff`, který bude zpoždovat sestupnou hranu aliasu `@Change`. S každou změnou stavu ze strany panelu se tak nastaví příznak, který bude vzápětí v procesu nastaven na hodnotu `False`. Po dobu zpoždění sestupné hrany aliasu `@Change` (délku lze nastavit pomocí vlastnosti `Delay`) bude nastaven alias `@P8R2NK_CH1` dle požadovaného stavu relé. Jakmile dojde k ukončení časovače, bude mít alias `@P8R2NK_CH1` stav dle skutečného stavu relé.

```
TimerOff1(Input = @Change); //Zpoždění příznaku o změně
If TimerOff1.Out then
    @Change = false; //Zrušení příznaku o změně
    @P8R2NK_CH1 = Poseidon.P8R_Relay1.Out0; //Alias dle požadovaného stavu relé
Else
    @P8R2NK_CH1 = Poseidon.P8R_Relay1.ActualOut0; //Alias dle skutečného stavu relé
    If Poseidon.P8R_Relay1.ActualOut0 != Poseidon.P8R_Relay1.Out0 then
        //Synchronizace aktuálního stavu s požadavkem ze strany panelu
        Poseidon.P8R_Relay1.Out0 = Poseidon.P8R_Relay1.ActualOut0;
    EndIf;
EndIf;
```

Alias `@P8R2NK_CH1` bude sloužit pro účely zobrazení stavu tlačítka na obrazovce. Po změně požadovaného stavu relé bude tlačítko zobrazovat požadovaný stav relé (po dobu `Delay` bloku `TimerOff`). Po uplynutí nastavené doby se bude stav tlačítka zobrazovat dle aktuálního stavu relé. Funkcionalitu řeší následující kód, který je nutné vložit do události „`OnRefresh`“ obrazovky, kde se nachází ovládací tlačítko.

```
ToggleButton1.Checked = Process1.@P8R2NK_CH1;
```

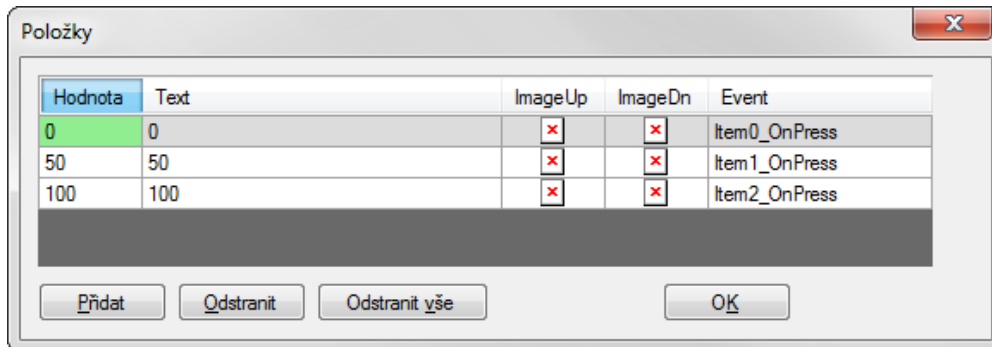
Výše uvedený kód řeší i situaci, kdy by došlo ke změně stavu relé např. z bezdrátového tlačítka. Tlačítko na panelu automaticky změní svůj stav, dle nastaveného stavu relé ze strany bezdrátového tlačítka.

### Ovládání analogového výstupu pomocí tlačítek se zjištěním skutečného stavu

Pro ovládání analogového výstupu pomocí tlačítek lze využít prvek `SelectButton`.

Po vložení prvku na obrazovku lze nadefinovat:

- ◆ počet tlačítek,
- ◆ hodnotu, dle které se tlačítka přepínají,
- ◆ obrázky, které se mají zobrazovat při stisku či uvolnění tlačítka (nejsou vyžadovány).



Obr. 12 – Nastavení tlačítek prvku `SelectButton`

Po potvrzení nadefinovaných tlačítek pomocí tlačítka „**OK**“ vznikne pro každé tlačítko samostatná událost „`SelectButton1_ItemX_OnPress()`“. Dále je postup obdobný jako v případě ovládání relé pomocí tlačítka `ToggleButton`.

Do nadefinovaných událostí lze vložit požadovaný kód. V případě ukázkové aplikace se nastaví příznak (je třeba nadefinovat jako alias `@ChangeDali` v procesu) o požadavku na změnu stavu a požadovaný stav relé (příznak o požadavku na změnu stavu bude sloužit pro spuštění časovače bloku `TimerOff`).

```

event SelectButton1_Item0_OnPress()
    Process1.@ChangeDali = true;
    Poseidon.P8R_Dimm1.Out0 = 0.0;
end;

event SelectButton1_Item1_OnPress()
    Process1.@ChangeDali = true;
    Poseidon.P8R_Dimm1.Out0 = 50.0;
end;

event SelectButton1_Item2_OnPress()
    Process1.@ChangeDali = true;
    Poseidon.P8R_Dimm1.Out0 = 100.0;
end;

```

Dále je nutné v periodickém procesu nadefinovat blok `TimerOff`, který bude zpoždovat sestupnou hranu aliasu `@ChangeDali`. S každou změnou stavu ze strany panelu se tak nastaví příznak, který bude vzápětí v procesu nastaven na hodnotu `False`. Po dobu zpoždění sestupné hrany aliasu `@ChangeDali` (délku lze nastavit pomocí vlastnosti `Delay`) bude nastavena hodnota proměnné `ActualOut` dle požadované hodnoty intenzity osvětlení. Jakmile dojde k ukončení časovače, bude hodnota proměnné `ActualOut` nastavena dle skutečné hodnoty intenzity osvětlení.

```

TimerOff2 (Input = @ChangeDali); //Zpoždění příznaku o změně
If TimerOff2.Out then
    @ChangeDali = false; //Zrušení příznaku o změně
    ActualOut = Real_To_Int(Poseidon.P8R_Dimm1.Out0);
Else
    ActualOut = Real_To_Int(Poseidon.P8R_Dimm1.ActualOut0);
EndIf;

```

Hodnota `ActualOut` bude sloužit pro účely zobrazení stavu tlačítek na obrazovce. Po změně požadované hodnoty intenzity osvětlení budou mít tlačítka stav dle požadované intenzity osvětlení (po dobu `Delay` bloku `TimerOff`). Po uplynutí nastavené doby budou mít tlačítka stav dle aktuální intenzity osvětlení. Funkcionalitu řeší následující kód, který je nutné vložit do události „OnRefresh“ obrazovky, kde se nachází prvek `SelectButton`.

```
SelectButton1.Value = Process1.ActualOut;
```

Výše uvedený kód řeší i situaci, kdy by došlo ke změně požadované hodnoty intenzity osvětlení z jiné bezdrátové periferie. Stav tlačítek na panelu se automaticky změní, dle nastavené intenzity osvětlení z jiné bezdrátové periferie (pouze v případě, kdy nastavená intenzita osvětlení přesně odpovídá hodnotě, zadané v prvku `SelectButton`).

### 4.3 Vytvoření vazby mezi stanicí AMiT a periferiemi

Jak již bylo uvedeno, každá periferie systému Poseidon má své jedinečné ID, které je dáno výrobcem. Pro správnou obsluhu periferie stanicí AMiT musí být toto ID vyplněno v patřičném objektu a do paměti přijímače musí být uložena vazba se stanicí AMiT.

Vazbu se stanicí AMiT lze do periferie uložit dvěma způsoby:

- ◆ pomocí SW Poseidon Asistent od firmy ENIKA.CZ,
- ◆ párovacím režimem.

#### 4.3.1 SW Poseidon Asistent

SW Poseidon Asistent umožňuje mimo vyčtení ID periférií a vytvoření vazeb také rozšířené nastavení periférií. Jakmile jsou vytvořeny vazby prostřednictvím Poseidon Asistent, je nutné vyčtená ID zadat také do vlastností ID odpovídajících objektů `P8x_xxx` v projektu DetStudio.

PC s nainstalovaným SW Poseidon Asistent lze do bezdrátové sítě připojit prostřednictvím:

- ◆ USB vysílač pro konfiguraci P8 TR USB od firmy ENIKA.CZ,
- ◆ stanice firmy AMiT, která má rozhraní Ethernet a rozhraní Poseidon.

Z výše uvedeného plyne, že **jakoukoliv stanicí s rozhraním Ethernet a Poseidon z produkce firmy AMiT lze použít** nejen jako bránu pro komunikaci s periferiemi sítě Poseidon, ale **také jako bránu pro konfiguraci periférií v síti Poseidon**.

Aby byla možná komunikace SW Poseidon Asistent prostřednictvím stanice AMiT, musí být ve stanici zaveden program s objektem `Poseidon`, který má nastavenou vlastnost `AsistentEnable` na `True`.

#### **Pozor**

*Jakmile se ke stanici AMiT připojí PC prostřednictvím SW Poseidon Asistent, vypne se komunikace s periferiemi v síti Poseidon, nadefinovaná v programu, který běží ve stanici. Tato bude funkční až po ukončení komunikace se SW Poseidon Asistent.*

Stanice bez ethernetového rozhraní (např. **AMR-OP70RHP**) funkci brány pro SW Poseidon Asistent neumožňují.

Více informací o použití SW Poseidon Asistent lze získat na webových stránkách [www.enika.cz](http://www.enika.cz).

Jestliže není možné použít SW Poseidon Asistent, lze vazbu mezi stanicí AMiT a periferiemi vytvořit dle níže uvedeného postupu.

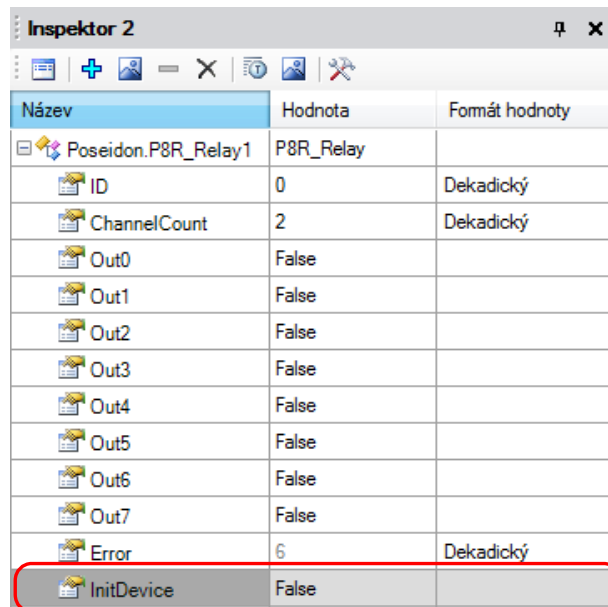
### 4.3.2 Párovací režim

#### Párování periferií typu přijímač

Ukázku párování lze popsat na vložených objektech **P8R\_Realy** a **P8R\_Ro11**. Před samotným párováním je nutné, aby byla ve stanici AMiT zavedena vygenerovaná aplikace s vybranými objekty. Přijímače nesmí mít vazbu se stanicí AMiT již nadefinovanou a v daný okamžik párování musí být viditelné v síti Poseidon (viz manuál pro jednotlivé periferie firmy ENIKA.CZ).

Postup párování se provede následujícími kroky.

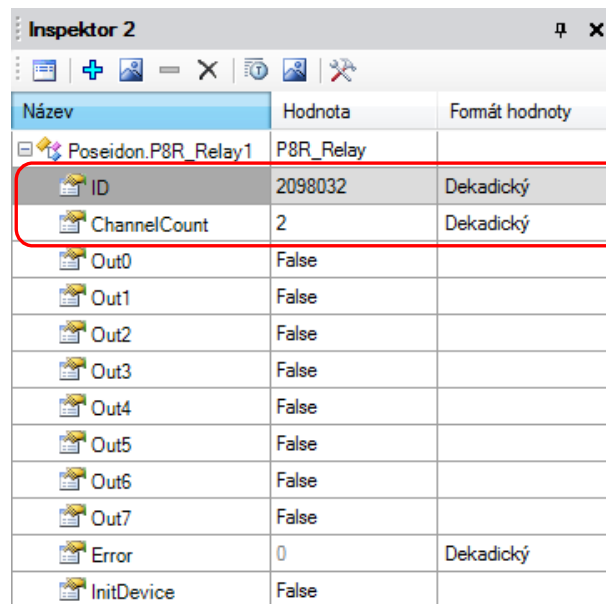
- ♦ Stanici AMiT i přijímač připojit k napájecímu napětí.
- ♦ V projektu DetStudia otevřít pomocí menu „Ladění/Inspektor 1“ panel „Inspektor 1“. Do otevřeného okna „Inspektor 1“ vložit (pomocí tlačítka „+“) objekt **P8R\_Relay1**.



Název	Hodnota	Formát hodnoty
Poseidon.P8R_Relay1	P8R_Relay	
ID	0	Dekadický
ChannelCount	2	Dekadický
Out0	False	
Out1	False	
Out2	False	
Out3	False	
Out4	False	
Out5	False	
Out6	False	
Out7	False	
Error	6	Dekadický
InitDevice	False	

Obr. 13 – Objekt **P8R\_Relay1** a některé jeho vlastnosti v okně „Inspektor 1“

- ♦ V následujícím kroku na přijímači **1x** stisknout a uvolnit programovací tlačítko (viz manuál pro jednotlivé periferie firmy ENIKA.CZ), tím se přijímač uvede do programovacího režimu.
- ♦ Po uvolnění programovacího tlačítka na přijímači nastavit v okně „Inspektor 1“ vlastnost **InitDevice** objektu **P8R\_Relay1** na hodnotu „1“ (True). Tím se spustí ze strany stanice AMiT párovací sekvence a v případě, že párování bylo úspěšné, zobrazí se ve vlastnosti **ID** vyčtená hodnota ID přijímače. Také se automaticky načte hodnota odpovídající fyzickému počtu reléových výstupů do vlastnosti **ChannelCount**, viz následující obrázek.



Název	Hodnota	Formát hodnoty
Poseidon.P8R_Relay1	P8R_Relay	
ID	2098032	Dekadický
ChannelCount	2	Dekadický
Out0	False	
Out1	False	
Out2	False	
Out3	False	
Out4	False	
Out5	False	
Out6	False	
Out7	False	
Error	0	Dekadický
InitDevice	False	

Obr. 14 – Vyčtené ID přijímače po úspěšném párování

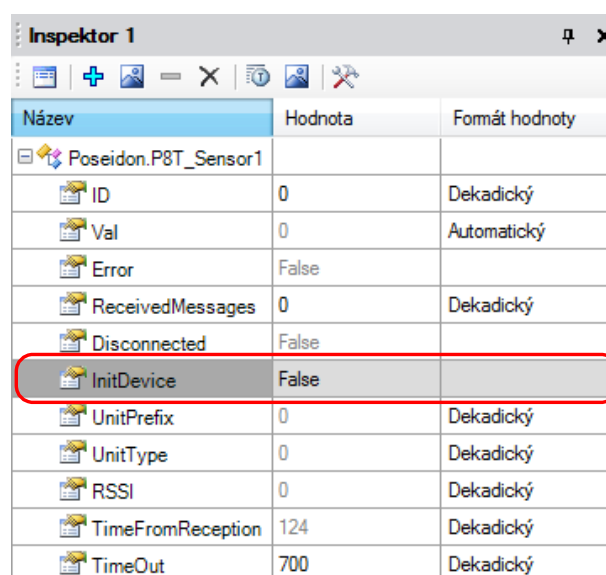
Nyní lze s přijímačem komunikovat v síti Poseidon, tzn., lze sledovat/měnit stav jeho reléových výstupů, např. přímo v okně „Inspektor 1“ případně lze stav měnit pomocí obrazovek či pomocí algoritmů zapsaných v periodických procesech a inspektorem sledovat změny.

### Párování periferií typu vysílač

Ukázku párování lze popsat na vložených objektech **P8T\_Sensor**. Před samotným párováním je nutné, aby byla ve stanici AMiT zavedena vygenerovaná aplikace s vybranými objekty.

Postup párování se provede následujícími kroky.

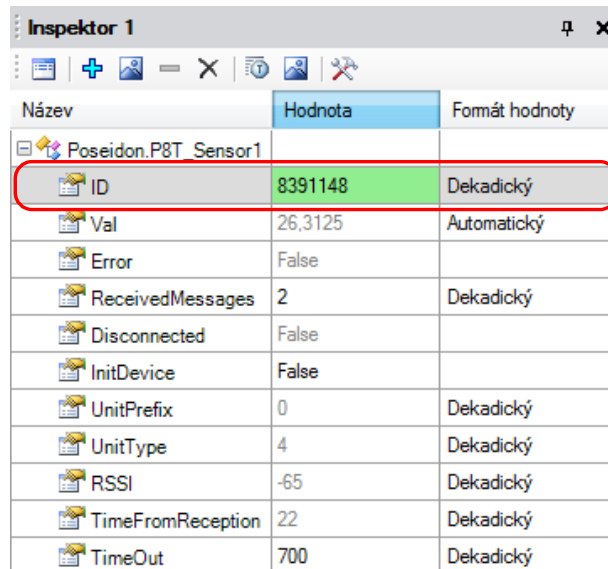
- ♦ Stanici AMiT připojit k napájecímu napětí. Vysílače jsou zpravidla napájeni baterií, která musí být ve vysílači vložena a mít dostatečnou úroveň napětí.
- ♦ V projektu DetStudia otevřít pomocí menu „Ladění/Inspektor 1“ panel „Inspektor 1“. Do otevřeného okna „Inspektor 1“ vložit (pomocí tlačítka „+“) objekt **P8T\_Sensor1**.



Název	Hodnota	Formát hodnoty
Poseidon.P8T_Sensor1		
ID	0	Dekadický
Val	0	Automatický
Error	False	
ReceivedMessages	0	Dekadický
Disconnected	False	
InitDevice	False	
UnitPrefix	0	Dekadický
UnitType	0	Dekadický
RSSI	0	Dekadický
TimeFromReception	124	Dekadický
TimeOut	700	Dekadický

Obr. 15 – Objekt **P8T\_Sensor1** a jeho vlastnosti v okně „Inspektor 1“

- ♦ V okně „Inspektor 1“ nastavit vlastnost `InitDevice` objektu `P8T_41` na hodnotu „1“ (True). Tím se očekává příchozí párovací sekvence, kterou vyšle vysílač.
- ♦ Aktivovat párovací sekvenci se ze strany vysílače (viz dokumentace k použitému vysílači, která je dostupná na [www.enika.cz](http://www.enika.cz)). Pro vysílač teploty a vlhkosti párovací sekvence aktivuje stiskem spodního tlačítka na vysílači (na vysílači dojde k rozsvícení dvou LED) a následně stiskem horního tlačítka (dojde k odeslání párovací sekvence a LED zablikají).
- ♦ Úspěšné párování se opět projeví vyplněním vlastnosti `ID` hodnotou ID přijímače, viz následující obrázek.

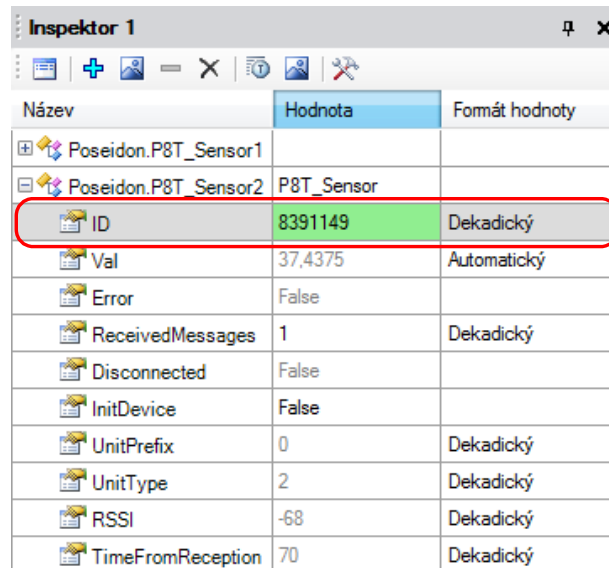


Název	Hodnota	Formát hodnoty
Poseidon.P8T_Sensor1		
ID	8391148	Dekadický
Val	26,3125	Automatický
Error	False	
ReceivedMessages	2	Dekadický
Disconnected	False	
InitDevice	False	
UnitPrefix	0	Dekadický
UnitType	4	Dekadický
RSSI	-65	Dekadický
TimeFromReception	22	Dekadický
TimeOut	700	Dekadický

Obr. 16 – Vyplněné ID vysílače po úspěšném párování

Nyní lze s vysílačem komunikovat v síti Poseidon, např. lze sledovat v okně „Inspektor 1“ hodnoty měřené teploty.

Pokud je požadavek na párování s periferií, která poskytuje do sítě Poseidon více informací, je nutné vložit do projektu tolik komunikačních objektů, kolik informací periferie sítě Poseidon poskytuje. Každému objektu, který reprezentuje další veličinu z periferie, je nutné nastavit adresu o jedničku vyšší. Pokud je tedy k dispozici periferie P8 T TempRh, která poskytuje nejen informaci o teplotě, ale také o vlhkosti musí se vlhkost nadefinovat pomocí dalšího objektu `P8T_Sensor`, kterému je nutné zadat adresu o jedničku vyšší (v našem případě 8191149).



Název	Hodnota	Formát hodnoty
Poseidon.P8T_Sensor1		
Poseidon.P8T_Sensor2	P8T_Sensor	
ID	8391149	Dekadický
Val	37,4375	Automatický
Error	False	
ReceivedMessages	1	Dekadický
Disconnected	False	
InitDevice	False	
UnitPrefix	0	Dekadický
UnitType	2	Dekadický
RSSI	-68	Dekadický
TimeFromReception	70	Dekadický

Obr. 17 – Vyplněné ID vysílače pro další veličinu

### Poznámka

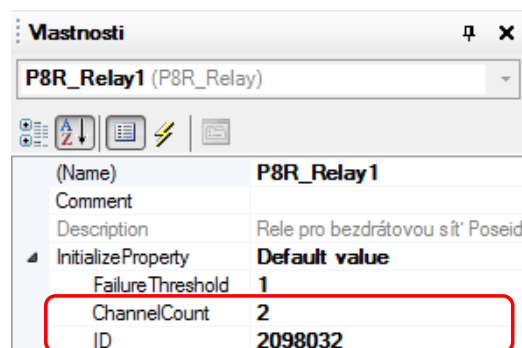
Pokud jsou známa ID vysílačů, lze je zadat do vlastnosti ID přímo v panelu „Vlastnosti“ již při vytváření projektu. Není pak nutné provádět párovací sekvenci manuálně. Stanice AMiT bude požadavky od periferie s daným ID přijímat automaticky.

Párovací režim lze naprogramovat s využitím výše uvedených vlastností přímo na obrazovky panelu. Pro úspěšné spojení stanice AMiT s periferiemi v síti Poseidon pak nebude nutné využívat vývojové prostředí DetStudio – sekvence párování proběhne prostřednictvím obrazovek panelu. Ukázku vytvořené obrazovky lze nalézt na obrazovce „Screen2“ v příkladu poseidon\_p1\_cz\_02.dsox, který je součástí této aplikační poznámky.

## 4.4 Přehrání aplikace

Jestliže je nutné v uživatelské aplikaci provést nějaké změny a požaduje se, aby stávající nastavené parametry objektů (ID periferie, počet kanálů periferie apod.) byly zachovány po přehrání aplikace, je možno využít některého z následujících postupů.

- ♦ ID, počet kanálů apod. vyplnit do příslušného parametru v okně „Vlastnosti“ u jednotlivých objektů. Viz následující obrázek znázorňující okno „Vlastnosti“ objektu P8R\_Relay1.



Vlastnosti	
P8R_Relay1 (P8R_Relay)	
(Name)	P8R_Relay1
Comment	
Description	Rele pro bezdrátovou síť Poseid
InitializeProperty	Default value
Failure Threshold	1
ChannelCount	2
ID	2098032

Obr. 18 – Okno vlastnosti objektu P8R\_Relay1



- ♦ Pomocí správce dat provést zálohu dat stávající aplikace ze stanice AMiT a po přehraní aplikace data obnovit. Více informací lze nalézt v nápovědě **EsiDet** v kapitole „Správce dat“
- ♦ Pomocí zatržení volby „Provádět zálohu a obnovu proměnných ve stanici“. Více informací o této volbě lze získat v nápovědě **EsiDet** v sekci „Přenos programu“ kapitoly „/Obsah/Zavedení aplikace“.

## 5 Diagnostika komunikace

Stanice AMiT automaticky hlídá vnitřními funkcemi stav fyzického komunikačního modulu bezdrátové sítě 868 MHz. Pokud by došlo k zahlcení nebo jinému nestandardnímu stavu, komunikační modul se automaticky resetuje.

### 5.1 Síla signálu

Objekty, které v rámci projektu obsluhují přijímače i vysílače, disponují vlastností **RSSI**. Hodnota této vlastnosti udává sílu RF signálu přijaté zprávy od daného vysílače nebo dotazovaného přijímače.

Hodnotu signálu lze rozdělit následovně:

- ♦ velmi dobrý > -50 dBm,
- ♦ použitelný -80 dBm až -50 dBm,
- ♦ špatný < -80 dBm.

### 5.2 Stav komunikace s periferiemi

Stav komunikace s obsluhovanou periferií systému Poseidon lze vyhodnocovat pomocí vlastností **Error** a **Disconnected**.

#### Error – chybová hlášení přijímačů

Hodnota	Význam
0	Bez chyby.
1	Neočekávaný typ povelu v odpovědi.
2	Při inicializaci – typ přijímače nedopovídá typu objektu.
3	Stav výstupu se neshoduje s požadovaným.
4	Bez odpovědi.
5	Neočekávané číslo kanálu v odpovědi.
6	Byla detekována změna přiřazeného ID z aplikace.

#### Error – chybová hlášení vysílačů

Hodnota	Význam
0	Baterie v pořádku.
1	Baterie je vybitá.

#### Disconnected – chybová hlášení vysílačů

Hodnota	Význam
0	Bez chyby.
1	Vysílač neposlal data po dobu delší než 10 min.

Vlastnost **Disconnected** je dostupná pouze u objektů obsluhující periferie typu vysílač, kromě objektů **P8T\_4** a **P8T\_4x8**.

## 6 Seznam podporovaných periférií

Tabulka níže obsahuje k datu vytvoření této aplikační poznámky typy periférií od firmy ENIKA.CZ, které lze v síti Poseidon obsluhovat stanicemi AMiT prostřednictvím uvedených objektů.

### Podporované periferie

Periferie	Objekt	Počet objektů pro obsluhu periferie	Poznámka
<b>Přijímače spínací</b>	P8R_Relay	1	Max. 8 kanálů.
P8 R 1 xxx *)			
P8 R 2 xxx			
P8 R 4 xxx			
P8 R 8 xxx			
<b>Přijímače stmívací</b>	P8R_Dimm	1	Max. 4 kanály.
P8 R D I xxx *)			
P8 R 01-10 N			
P8 R DALI xxx *)			
<b>Přijímače roletové</b>			
P8 R R xxx *)	P8R_Roll	1	–
P8 R 4R S	P8R_Roll4	1	–
<b>Přijímače pro řízení termohlavic</b>	P8R_Relay		
P8 R Valve N 24V		1	–
<b>Regulátory osvětlení **)</b>	P8T_Contact	1	Detekce přítomnosti.
	P8T_Percent	1	Výstup regulátoru osvětlení.
	P8T_Sensor	1	Intenzita okolního osvětlení.
P8 TR PSMR16 (HR)			
P8 TR PS HB			
P8 TR PS LR xxx *)			
<b>Vysílače s tlačítky *)</b>	P8T_4	1	Max. 4 tlačítka.
P8 T 2 xxx			
P8 T 3 xxx			
P8 T 4 xxx			
P8 T 4a xxx			
<b>Vícekanálové vysílače s tlačítky</b>	P8T_4x8	4	Jeden objekt pro skupinu A až D.
P8 T 4x8a xxx *)			
<b>Vysílače stavu kontaktů</b>	P8T_Contact	2	Jeden objekt na kontakt.
P8 T(R) 2C I			
<b>Vysílače stavu kontaktů a HDO</b>	P8T_Contact	2	Jeden objekt na kontakt. Lze použít pouze v režimu „Jednotlivý vysílač“, viz manuál k periferii.
P8 TR 2C DIN			
P8 TR 2U DIN			
<b>Pohybové spínače</b>	P8T_Contact	1	–
P8 T PS W			
P8 T PSMR16/A xxx *)			
<b>Vysílače teploty</b>	P8T_Sensor	1	–
P8 T Temp xxx			
<b>Vysílače teploty a vlhkosti</b>	P8T_Sensor	2	Jeden objekt pro jednu veličinu.
P8 T Temp/RH xxx *)			
<b>Vysílače CO<sub>2</sub>, teploty a vlhkosti</b>	P8TContact	1	Ovládání vestavěného relé.
	P8T_Sensor	3	Jeden objekt pro jednu veličinu.
P8 T CO2 xxx *)			

Periferie	Objekt	Počet objektů pro obsluhu periferie	Poznámka
Vysílače zaplavení	P8T_Contact	1	Detekce zaplavení.
	P8T_Sensor	1	Měření externí teploty
P8 T AQ			

*\*) Znaky xxx zastupují další specifikaci periferie, např. provedení (vestavné, do pohledu, ...) nebo design rámečku apod.*

*\*\*\*) U kombinovaných periferií, jako je např. regulátor osvětlení, není nutnou podmínkou použít pro zpracování zpráv všechny uvedené objekty (P8T\_Contact, P8T\_Percent, P8T\_Sensor). Bude-li žádoucí zpracovávat např. jen informaci o intenzitě osvětlení, použije se pouze objekt P8T\_Sensor.*

## 7 Univerzální aplikace

---

Ve spolupráci s firmou ENIKA.CZ byla vytvořena aplikace pro komunikační jednotku, která zabezpečuje zpracování předdefinovaného počtu periferií typu vysílač a přijímač. Aplikaci lze stáhnout na stránkách [amitautomation.cz](http://amitautomation.cz) v sekci „**ŘEŠENÍ TYPOVÁ ŘEŠENÍ APLIKACÍ**“. Jedná se o typové řešení s názvem „AMR-CP24 – bezdrátová síť Poseidon“.

Počet předdefinovaných periferií je následovný:

- ◆ 40× až 8 kanálový reléový přijímač,
- ◆ 40× roletový přijímač,
- ◆ 40× stmívací přijímač (lze použít stmívač vestavný, 1 V až 10 V nebo DALI),
- ◆ 25× až 4 tlačítkový vysílač,
- ◆ 25× vysílač stavu kontaktu,
- ◆ 40× vysílač teploty nebo vlhkosti,
- ◆ 20× emulace stisku tlačítka jednoho 4 tlačítkového vysílače.

Konfigurace skutečného počtu periferií v dané technologii se provádí pomocí SW Poseidon Asistent. Konfigurace jednotlivých periferií jsou do komunikační jednotky uloženy prostřednictvím protokolu MODBUS TCP/IP. Pomocí tohoto protokolu lze také vyčítat stavy a povelovat nakonfigurované periferie. Seznam registrů protokolu MODBUS TCP/IP je součástí SW Poseidon Asistent. Více informací lze nalézt na stránkách [www.enika.cz](http://www.enika.cz).

Pro čtení stavů a povelování výstupů periferií pomocí nadřazeného systému AMiT je v aplikaci připravena komunikace **DB-Net/IP**. Seznam WIDů pro obsluhu jednotlivých periferií je obsažen v dokumentaci k výše uvedenému typovému řešení „AMR-CP24 – bezdrátová síť Poseidon“.

## 8 Technická podpora

---

Veškeré informace ohledně začlenění stanic AMiT do sítě Poseidon, Vám poskytne oddělení technické podpory firmy AMiT. Technickou podporu můžete kontaktovat nejlépe prostřednictvím emailu na adrese **support@amit.cz**.

## 9 Upozornění

---

AMiT, spol. s r. o. poskytuje informace v tomto dokumentu, tak jak jsou, nepřijímá žádné záruky, pokud se týče obsahu tohoto dokumentu a vyhrazuje si právo měnit obsah dokumentu bez závazku tyto změny oznámit jakékoli osobě či organizaci.

Tento dokument může být kopírován a rozšiřován za následujících podmínek:

1. Celý text musí být kopírován bez úprav a se zahrnutím všech stránek.
2. Všechny kopie musí obsahovat označení autorského práva společnosti AMiT, spol. s r. o. a veškerá další upozornění v dokumentu uvedená.
3. Tento dokument nesmí být distribuován za účelem dosažení zisku.

V publikaci použité názvy produktů, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.