

Databázový ovladač pro PostgreSQL systému REXYGEN
(Ovladač PqDrv)
Uživatelská příručka

REX Controls s.r.o.

Verze 3.0
Plzeň
15.12.2023

Obsah

1	Ovladač PqDrv a systém REXYGEN	2
1.1	Úvod	2
1.2	Systémové požadavky	2
1.3	Instalace ovladače na vývojovém počítači	2
1.4	Instalace ovladače na cílovém zařízení	3
1.4.1	Windows zařízení	3
1.4.2	Linux stroje	3
2	Zahrnutí ovladače do projektu	4
2.1	Přidání ovladače PqDrv	4
2.2	Propojení vstupů a výstupů v řídicím algoritmu	5
3	Konfigurace I/O ovladače	7
3.1	Záložka Připojení	7
3.2	Záložka Archivy	7
3.3	Záložka Grupy	9
3.4	Speciální signály	11
4	Podrobnosti implementace	14
5	Řešení problémů	17
	Literatura	18

Kapitola 1

Ovladač PqDrv a systém REXYGEN

1.1 Úvod

Tento manuál popisuje ovladač PqDrv pro připojení k PostgreSQL databázi. Ovladač umožňuje jak čtení, tak zápis dat. Dále je možné exportovat archivy systému REXYGEN do databáze, tj. alarmy, události a trendy.

1.2 Systémové požadavky

Ovladač PqDrv lze použít na cílových zařízeních s operačními systémy Windows a Linux. Pro komunikaci je vyžadován TCP/IP stack (Ethernetová karta, USB WiFi dongle, atd.).

Pro použití ovladače musí mít vývojový počítač a cílový počítač nainstalován následující software:

Vývojový počítač

Operační systém

Vývojové nástroje

jeden z následujících: Windows 10/11, GNU/Linux
verze vývojových nástrojů systému REXYGEN pro
odpovídající operační systém

Cílové zařízení

Systém REXYGEN

IO ovladač

jádro pro odpovídající operační systém

verze pro odpovídající operační systém

1.3 Instalace ovladače na vývojovém počítači

Ovladač PqDrv je součástí instalačního balíčku Vývojových nástrojů systému REXYGEN. Je nutné vybrat příslušný balíček v instalačním programu. Systém REXYGEN se obvykle instaluje do složky

`C:\Program Files\REX Controls\REXYGEN <verze>`.

Následující soubory jsou zkopírovány do instalační složky:

Bin\PqDrv_H.dll – Konfigurační část ovladače PqDrv.

Bin\PqDrv_T.dll – Část ovladače PqDrv pro cílové zařízení, která je volána modulem RexCore.

Doc\PDF\CZECH\PqDrv_CZ.pdf – This user manual.

1.4 Instalace ovladače na cílovém zařízení

1.4.1 Windows zařízení

Cílová část ovladače určená pro připojení k databázi na systému Windows 10/11 je součástí Vývojářských nástrojů systému REXYGEN, jak bylo zmíněno výše.

1.4.2 Linux stroje

Pokud na vašem cílovém zařízení ještě není nainstalován runtime modul RexCore, nainstalujte jej nejdříve podle průvodce pro začátečníky systému REXYGEN [1]. Instalace zahrnuje veškeré potřebné ovladače, včetně PqDrv.

Pokud chcete nainstalovat PqDrv samostatně, můžete tak učinit z příkazového řádku pomocí následujícího příkazu:

```
sudo apt-get install rex-pqdrvt
```

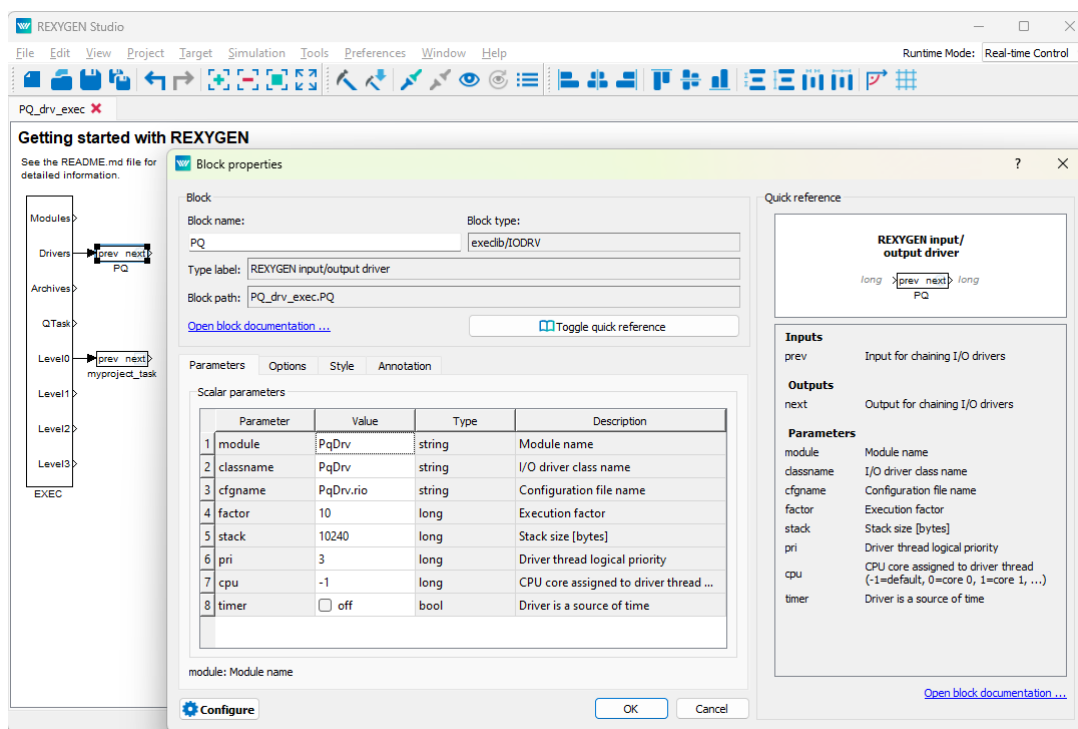
Kapitola 2

Zahrnutí ovladače do projektu

Ovladač je do projektu zahrnut v okamžiku, kdy je přidán do hlavního souboru projektu a vstupy a výstupy jsou propojeny v řídicích algoritmech.

2.1 Přidání ovladače PqDrv

Hlavní soubor projektu se zahrnutým ovladačem PqDrv je zobrazen na Obrázku 2.1.



Obrázek 2.1: Příklad hlavního souboru projektu se zahrnutým ovladačem PqDrv

K zahrnutí ovladače do projektu je třeba přidat jediný blok. Blok typu `IODRV` přejmenovaný na `PQ` a připojený k výstupu `Drivers` hlavního bloku `EXEC`. Název tohoto bloku (`PQ`, viz Obr. 2.1) je prefixem všech vstupních a výstupních signálů poskytovaných tímto ovladačem. Čtyři nejdůležitější parametry jsou:

`module` – název modulu propojeného s ovladačem, v tomto případě `PqDrv`

`classname` – třída ovladače, v tomto konkrétním případě `PqDrv`

`cfgname` – název konfiguračního souboru ovladače, např. `Pqdrv.rio`.

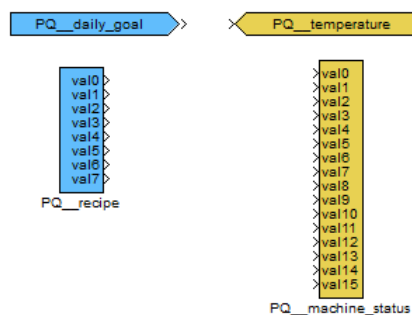
`factor` – násobek parametru `tick` bloku `EXEC`, který definuje periodu provádění úloh ovladače

Výše uvedené parametry funkčního bloku `IODRV` jsou konfigurovány v programu `REXYGEN Studio`. Konfigurační dialog je rovněž zobrazen na Obr. 2.1.

Tlačítko `Configure` otevře konfigurační dialog ovladače `PqDrv`, který je popsán v kapitole 3.

2.2 Propojení vstupů a výstupů v řídicím algoritmu

Vstupy a výstupy ovladače musí být propojeny s jednotlivými úlohami (`.mdl` soubory). Jednotlivé úlohy (bloky `QTASK` nebo `TASK`) jsou propojeny s výstupy `QTask`, `Level0`, ..., `Level13` hlavního bloku `EXEC`. Pro výměnu dat mezi řídicím algoritmem a ovladačem `PqDrv` použijte uvnitř jednotlivých tasků bloky znázorněné na Obr. 2.2.



Obrázek 2.2: Příklad vstupních a výstupních vlajek ovladače `PqDrv`

Blok `From` umožňující uživateli číst jeden vstupní signál má nastavený `Goto tag` na `PQ__<IN>`. Blok `Goto` umožňující uživateli nastavit jeden výstupní signál má nastavený `Goto tag` na `PQ__<OUT>`, kde `<IN>` a `<OUT>` jsou řetězce odkazující se na položky definované v konfiguračním souboru `*.rio`. Bloky vždy mají předponu `PQ` na začátku tagu, následovanou dvěma podtržítka `__`. Bloky s více vstupy/výstupy mají tuto předponu přímo ve svém názvu.

Použití bloků s více vstupy/výstupy se doporučuje, pokud je rychlost výměny dat (vzorkovací frekvence) prioritou. Podrobnosti o blocích INOCT, OUTOCT, INHEXD, OUTHEXD najdete v manuálu pro referenci funkčních bloků [2].

Každé jméno I/O musí být jedinečné ve všech skupinách. Sekce **Group** proto obsahuje volitelný parametr **Name** (viz kapitola 3 pro podrobnosti o sekci **Group**). I/O signál je odkazován jako `<group_name>_<item_name>` v tomto případě.

Instalace systému REXYGEN zahrnuje knihovnu příkladů, kde se mimo jiné sekce 0404-02_PqDrv věnuje použití PqDrv. Příklad 0404-02-00_IO_Flags obsahuje knihovnu použitelných vstupů a výstupů.

Kapitola 3

Konfigurace I/O ovladače

Tato kapitola popisuje konfiguraci jednotlivých vstupních a výstupních signálů a jejich symbolické pojmenování. Signály jsou mapovány na příslušnou databázi.

Konfigurační dialog je součástí souboru `PqDrv_H.dll`. Lze ho aktivovat z REXYGEN Studio stisknutím tlačítka **Configure** v dialogu parametrů bloku `IODRV` (viz kapitola 2). Výsledná konfigurace je uložena v souboru `*.rio` jako standardní pro ostatní REXYGEN ovladače. Dialog je rozdělen do tří záložek, které jsou podrobně popsány níže.

3.1 Záložka Připojení

První záložka je věnována konfiguraci parametrů připojení pro navázání spojení s databází. Tato záložka je znázorněna na Obrázku 3.1. Níže naleznete vysvětlení jednotlivých parametrů:

Server host – Odkazuje na adresu serveru, kde je hostována databáze.

Server port – Číslo portu použité pro připojení k serveru databáze.

Database – Určuje název databáze, ke které se chcete připojit.

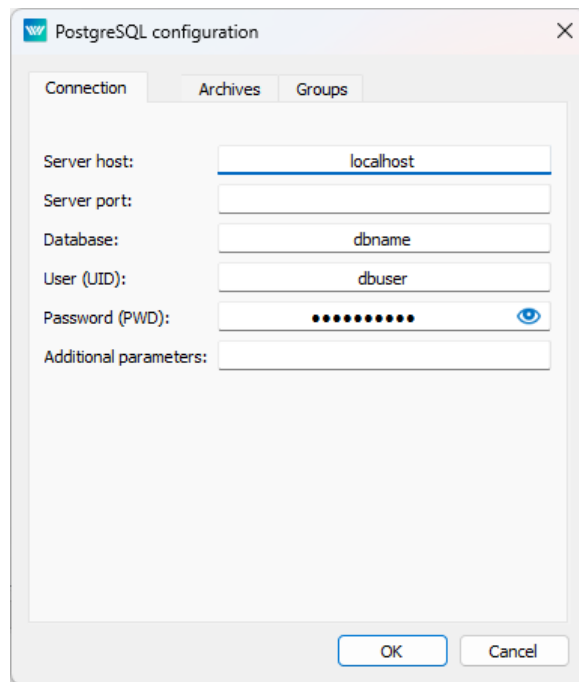
User (UID) – Uživatelské jméno (User Identifier) spojené s účtem databáze.

Password (PWD) – Příslušné heslo pro uživatele databáze.

Additional parameters – Můžete použít k zahrnutí jakýchkoli dalších parametrů nebo nastavení potřebných pro konkrétní připojení k databázi.

3.2 Záložka Archivy

Tato záložka může být použita k nastavení ukládání archivů z RexCore do databáze. Zobrazení záložky je vidět na Obrázku 3.2, a význam položek je následující:



Obrázek 3.1: Konfigurační dialog ovladače PgDrv – Podrobnosti o připojení k databázi

Mode – Definiuje strukturu exportu dat do databáze. Možnosti jsou:

- 0 Nic se neexportuje (používá se k dočasnému zakázání položky).
- 1 Jsou exportovány pouze alarmy a události (filtrováno dodatečnými parametry). Tabulka v databázi musí obsahovat následující sloupce: **Čas**, **ID alarmu**, **Kód**, **Úroveň**, **Hodnota**.
- 2 Jsou exportovány pouze trendy, tedy data uložená blokem TRND. Data jsou filtrována dodatečnými parametry. Tabulka v databázi musí obsahovat následující sloupce: **Čas**, **ID skupiny**, **Hodnota1**, **Hodnota2**, ...
- 3 Jsou exportovány pouze trendy, ale na rozdíl od předchozího má parametr **SQL** význam kompletního SQL dotazu, do kterého jsou vloženy hodnoty. Lze použít následující zástupné symboly: ?G = hodnota parametru **GlobalId** v souboru *.rio; ?S = hodnota parametru **GlobalString** v souboru *.rio; ?Y = rok; ?M = měsíc; ?D = den v měsíci; ?T = časové razítko (čas a datum položky archivu); ?N = nanosekundy (v sekundách) časového razítka; ?C = kód; ?L = úroveň; ?I = ID položky; ?1 = 1. položka; ?2 = 2. položka; ... Obyčejný otazník má význam *další položka* v následujícím pořadí: 1. hodnota, 2. hodnota, ...). Další parametr **SQL2** má význam SQL dotazu použitého pro detekci poslední zaznamenané položky v databázi, který je použit pro automatické načítání dat archivovaných, když nebyla k dispozici databáze.
- 4 Jsou exportovány pouze alarmy a parametr **SQL** má význam kompletního SQL dotazu. Zástupné symboly a parametr **SQL** jsou stejné jako v předchozím režimu.

Archive ID – Číslo archivu, ze kterého se mají číst data. Archivy jsou číslovány od 1 podle pořadí v konfiguraci výkonného systému REXYGEN (ARCHIVE bloky připojené k bloku EXEC).

Item ID ranges – Rozsah ID (parametr **id** původního bloku) pro export z archivu do databáze (v sekci ARCHIVE). Musí být sudý počet položek, kde liché položky určují začátek intervalu a sudé položky jeho konec. Například "100,100,104,109" znamená ID 100, 104, 105, ..., 109. Položky musí být seřazeny vzestupně.

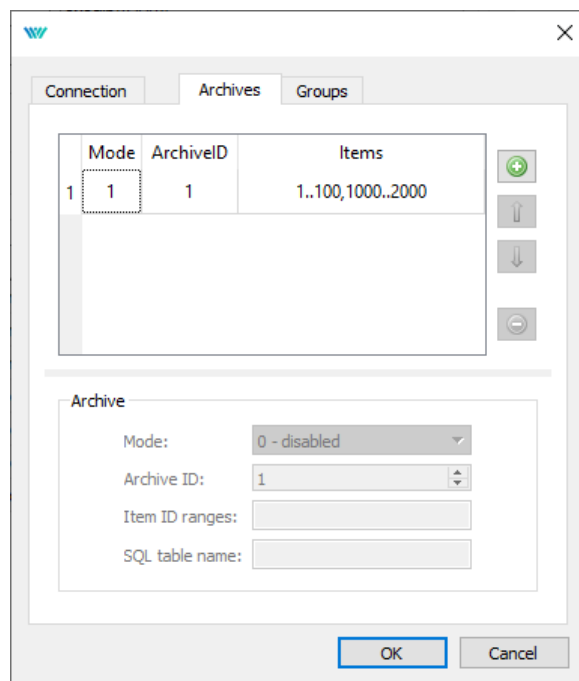
SQL table name – Název tabulky SQL nebo celý SQL příkaz, jak je definováno parametrem **Režim**. V některých případech musí být použita notace <název_databáze>.<název_tabulky> pro odkazování na tabulky databáze.

3.3 Záložka Grupy

Poslední záložka se používá k přímému čtení a zápisu dat do a z algoritmu. Záložka Grup je znázorněna na Obrázku 3.3 a je rozdělena do tří částí:

- **Group table** – Tato část zobrazuje všechny připravené datové skupiny. Význam sloupců je následující:

Name – Název skupiny pro čtení/zápis položek.



Obrázek 3.2: Konfigurační dialog ovladače PgDrv – Nastavení sekce archivů

Mode – Dostupné režimy jsou popsány v tabulce níže.

Period [s] – Perioda v sekundách pro generování SQL dotazu.

- **Group settings** – Tato část zahrnuje nastavení pro aktuálně vybranou skupinu. V této části naleznete tři výše uvedené položky a navíc další položku, buď **SQL table name** (jméno tabulky) nebo **SQL query** (SQL dotaz), v závislosti na vybraném **Mode** (režimu). V některých případech musí být použita notace `<název_databáze>.<název_tabulky>` pro odkazování na tabulky databáze.

- **Item table** – Tato tabulka je zobrazena pouze při výběru skupiny a zobrazuje čtecí a zápisové položky pro tuto skupinu. Význam sloupců je následující:

Name – Název čtecí/zápisové položky.

Type – Možné typy jsou: `bool`, `int`, `double`, `string`, `large`.

Dostupné režimy skupin

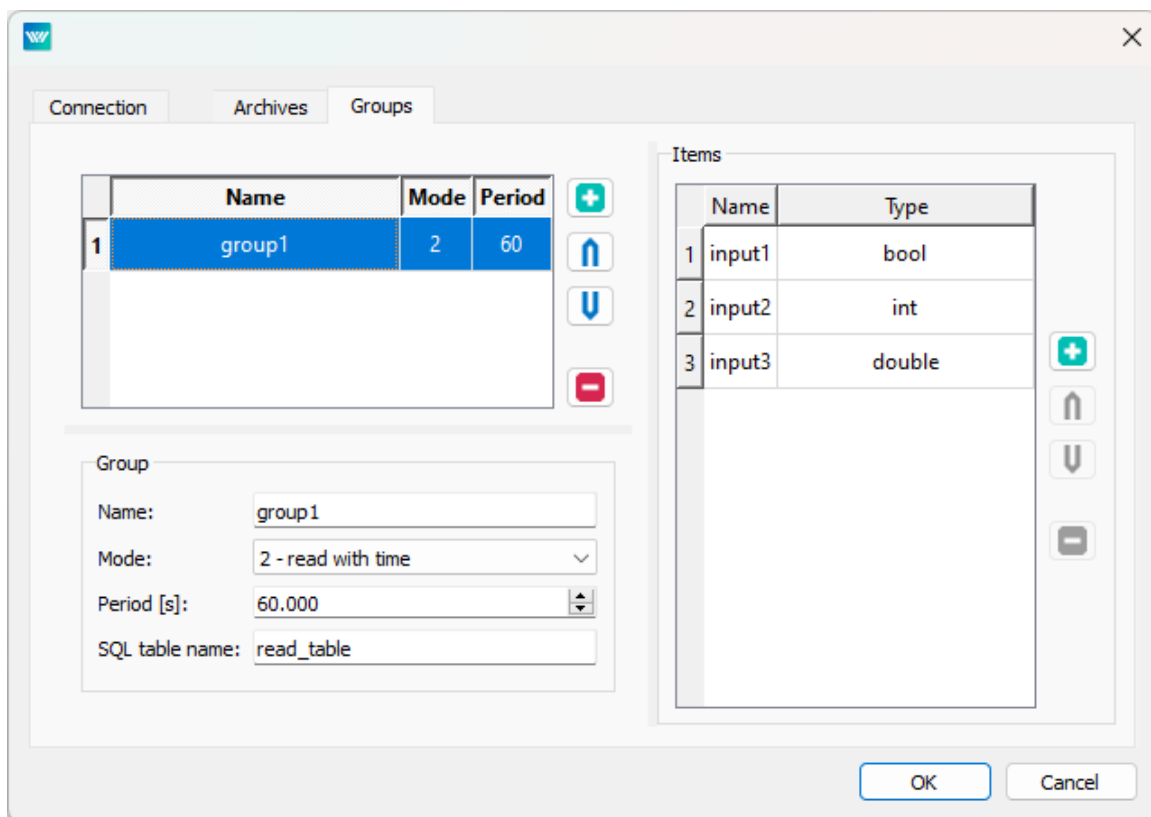
0	Nic se nečte (používá se k dočasnému zakázání položky).
1	Předpokládá se, že tabulka je seřazena podle sloupce ID. Řádek s nejvyšším ID je dodáván odpovídajícím vstupním vlnkám v úloze. Sloupce a položky/vlnky musí mít stejný název.

- 2 Předpokládá se, že tabulka je seřazena podle sloupce **Time** (**ID** je vedlejší klíč). Je vybrán řádek s nejvyšším časem, který není umístěn v budoucnosti, a výsledná data jsou dodávána odpovídajícím vstupním vlajkám v úloze. Sloupce a položky/vlajky musí mít stejný název. Tento režim umožňuje použití předgenerované posloupnosti dat.
- 3 Je proveden SQL dotaz z parametru **SQL**, vstupy jsou aktualizovány prvním řádkem odpovědi (1. sloupec odpovídá 1. položce, ...). Je možné použít **?T** v SQL dotazu, který je nahrazen aktuálním časem. Je také možné použít **?1** v SQL dotazu, který je nahrazen hodnotou 1. položky, **?2** , který je nahrazen hodnotou 2. položky,...
- 128 Nic se nezapisuje (používá se k dočasnému zakázání položky).
- 129 Hodnoty z odpovídajících vlajek v úlohách jsou zapsány do databáze. Sloupce a položky/vlajky musí mít stejný název.
- 130 Podobné jako výše, pouze zde existuje ještě jeden sloupec s názvem **Time**, který obsahuje aktuální čas (včetně data) **REXYGEN** runtime jádra v UTC (nebo jiný čas definovaný parametrem **Možnosti**).
- 131 Je proveden SQL dotaz z parametru **SQL**. Můžete použít následující zástupce: **?G** = hodnota parametru **GlobalId** v *.rio souboru; **?S** = hodnota parametru **GlobalString** v *.rio souboru; **?Y** = aktuální rok; **?M** = aktuální měsíc; **?D** = aktuální den v měsíci; **?T** = aktuální čas (včetně data); **?I** = položkaID; **?1** = 1. položka; **?2** = 2. položka; ... Obyčejný otazník znamená *další položku* v následujícím pořadí: 1. hodnota, 2. hodnota, ...

3.4 Speciální signály

Pro každý I/O signál existují další pomocné signály. Ty lze získat přidáním následujících řetězců k referenci signálu:

- _Status** – stavový kód - výsledek posledního dotazu. Kódy jsou: 0 ... poslední požadavek byl úspěšný 1 ... poslední požadavek selhal 2 ... ještě nebyl proveden žádný požadavek 3 ... poslední požadavek vrátil prázdnou sadu dat 101 ... poslední požadavek selhal kvůli odpojení databáze 102 ... spojení s databází není navázáno, požadavek není vyžadován
- _Disable** – Pokud je **True**, čtecí/zápisové operace pro celou skupinu jsou zakázány.
- _Trigger** – Vzestupná hrana spouští provedení čtecích/zápisových operací.
- _Age** – Počet sekund od posledního přístupu k databázi (čtení/zápis).
- _Fresh** – Stejně jako **_Age**
- _AgeDb** – Počet sekund od posledního přístupu k databázi (čtení/zápis), na rozdíl od předchozího je stáří definováno položkou v databázi.
- _FreshDb** – Stejně jako **_AgeDb**



Obrázek 3.3: Konfigurační dialog ovladače PgDrv – Konfigurace sekce Grup

_Period – Obsahuje nebo nastavuje parametr **Period**, tedy periodu vykonávání SQL dotazu, jak je definováno pro každou skupinu.

_Done – Označuje, že poslední požadavek byl úspěšně dokončen (hodnota se vypne po čtení, pokud je **Trigger** vypnutý).

_Error – Indikuje, že poslední požadavek selhal (hodnota se vypne po čtení, pokud je **Trigger** vypnutý).

_Empty – Označuje, že poslední požadavek vrátil prázdnou sadu dat (hodnota se vypne po čtení, pokud je **Trigger** vypnutý).

Tyto atributy jsou atributy skupiny, protože celá skupina je zapsána/čtena jedním SQL příkazem. Proto lze na atribut odkazovat řetězcem reference signálu `PQ__<název_skupiny>_<název_atributu>`, kde `<název_skupiny>` je hodnota volitelného parametru **Name** ve části **Group**.

Existují také další pomocné globální signály:

Connect – připojení k databázovému serveru.

Connected – stav připojení k databázovému serveru.

Reset – resetování čtení archivu (všech archivačních skupin).

Resetting – indikuje, že resetování skončilo.

GlobalId – nastaví nebo získá hodnotu parametru **GlobalID** v souboru *.rio.

GlobalString – nastaví nebo získá hodnotu parametru **GlobalString** v souboru *.rio.

Pro každého pracovníka archivu (spojeného se sekci **Archiv** v konfiguračním souboru) existují také další pomocné signály. To vyžaduje volitelný parametr **Name** v sekci **Archive**. Ty lze získat pomocí řetězce reference signálu **PQ_<název_archivu>_<název_volby>**, kde **<název_volby>** je:

_Count – počet úspěšně zapsaných položek archivu do databáze

_ErrorCount – počet neúspěšných položek archivu (přeskočených a nezapsaných do databáze)

_LastErrorStr – časové razítko poslední zpracované položky archivu (zapsané do databáze nebo přeskočené)

Kapitola 4

Podrobnosti implementace

Další informace o použití a implementaci ovladače `PqDrv` v systému REXYGEN jsou shromážděny v této kapitole.

- Parametr `Items` v sekci `Archive` je seznam čísel, kde liché položky znamenají *od* a sudé *do*. Např. `Items "2, 5, 10, 15"` exportuje položky s ID 2 až 5 a 10 až 15. Počet položek musí být sudý, i když se exportují položky pouze s jedním ID. Položky v parametru `Items` musí být seřazeny vzestupně.
- Po navázání spojení s databází jsou data konfigurovaná sekcemi `Archive` automaticky načtena do databáze od posledního uloženého záznamu. Tato funkce není aktivní, pokud je tabulka databáze prázdná. V takovém případě se načítání dat spustí pouze od aktuálního času. Pokud chcete načíst stará data z archivů do prázdné databáze, vložte umělý záznam s časovým razítkem před bod, ze kterého chcete začít. Načítání starých dat je omezeno na 100 záznamů na období spuštění ovladače, aby se předešlo problémům s výkonem. Při nastavování doby spuštění toto prosím vezměte na vědomí.
- Přestože většina databázových systémů nerozlišuje velká a malá písmena, řídicí systém REXYGEN je rozlišuje. Proto je ovladač `PqDrv` také citlivý na velká a malá písmena ve vlajkách I/O (vlajky odpovídají názvům sloupců v databázi).
- Všechny hodnoty zapsané nebo načtené z databáze jsou desetinná čísla (typ `double`). Sloupce databáze mohou být jiného typu, protože SQL dotazy jsou textové. Volitelný parametr `Type` může být použit, kde `i` znamená, že hodnota je zpracovávána jako typ `long`, `b` označuje typ `bool`, `s` označuje typ `string` a `r` označuje typ `real`. Například `Type "rrisb"` znamená, že první a druhá hodnota jsou reálná čísla, třetí hodnota je celé číslo, čtvrtá hodnota je řetězec a pátá hodnota je boolean.
- Vlajky musí být jedinečné v celém projektu, protože neobsahují identifikátor `Group`. V případě duplicity se zpracovává pouze první výskyt.
- V sekci `Group` je možné definovat (volitelný) parametr `Name`. V tomto případě musí být vlajky ve tvaru `<group_name>_<item_name>`.

- Ovladač vyžaduje uživatelské jméno a heslo pro přihlášení do databáze. Obojí je uloženo v souboru *.rio jako čistý text stejně jako všechny ostatní parametry. Proto se rozhodně doporučuje používat dedikované přihlašovací jméno s velmi omezenými oprávněními.
- Všechna časová razítka (např. náhrada za ?T v SQL dotazech) jsou rozšířena do řetězce ve tvaru <year>-<month>-<day> <hour>:<minute>:<second>.<mikrosecond>. Je použita časová zóna UTC (pokud není definována jiná časová zóna pomocí parametru `Options`).
- Pro časová razítka se doporučuje používat SQL datový typ s alespoň mikrosekundovým rozlišením¹.
- Současná implementace omezuje SQL řetězce na 1023 znaků (po expanzi otazníků). Parametr `Items` (jak v sekci `Archive`, tak v sekci `Group`) je omezen na 64 hodnot.
- Sloupec `Code` v exportu alarmu je celé číslo, kde nejnižších 5 bitů je třída alarmu a vyšších 3 bity je podtyp alarmu. Třídy jsou:

0	Systémový alarm
1	Boolovský alarm
2	Alarm pro Byte hodnotu
3	Alarm pro short hodnotu (znaménkové celé číslo 16 bitů)
4	Alarm pro Long hodnotu (znaménkové celé číslo 32 bitů)
5	Alarm pro Word hodnotu (bez znaménka 16 bitů)
6	Alarm pro DWord hodnotu (bez znaménka 32 bitů)
7	Alarm pro Float hodnotu
8	Alarm pro Double hodnotu
10	Alarm pro Large hodnotu (znaménkové celé číslo 64 bitů)
12	Alarm pro String hodnotu
13 .. 16	Není použito
17	Událost skupiny hodnot typu Boolean
18	Událost skupiny hodnot typu Byte
19	Událost skupiny hodnot typu Short
20	Událost skupiny hodnot typu Long
21	Událost skupiny hodnot typu Word
22	Událost skupiny hodnot typu Dword
23	Událost skupiny hodnot typu Float
24	Událost skupiny hodnot typu Double
26	Událost skupiny hodnot typu Large
31	Potvrzení alarmu

 Podtypy pro systémové alarmy jsou:

0	Označení data (není exportováno)
1	Výkonná událost
2	Archivační událost

¹<https://www.postgresql.org/docs/current/static/datatype-datetime.html>

Úroveň v tomto případě označuje událost. Výkonné události jsou následující:

- 0 Reset systému
- 1 Začátek stahování
- 2 Konec stahování
- 3 Stahování se nezdařilo
- 4 Zastavení výkonného programu
- 5 Spuštění výkonného programu
- 6 Výměna výkonného programu
- 7 Nastavení času

a archivní události:

- 0 Vymazání archivu (nyní nepoužíváno)
- 1 Uložení obnovy archivu (nyní nepoužíváno)
- 2 Normální obnova archivu (nyní nepoužíváno)
- 3 Chyba kontrolního součtu (nyní nepoužíváno)
- 4 Chyba integrity (nyní nepoužíváno)
- 5 Změna velikosti (nyní nepoužíváno)
- 6 Překročení limitu (pouze diskové archivy)
- 7 Přetečení bufferu

Podtypy pro booleovské alarmy jsou následující:

- 0 Sestupná hrana (tedy připojená booleovská proměnná byla změněna z hodnoty vysoká/pravda/1 na nízkou/nepravda/0)
- 1 Vzestupná hrana (tedy připojená booleovská proměnná byla změněna z hodnoty nízká/nepravda/0 na vysokou/pravda/1)

Podtypy pro číselné alarmy jsou následující:

- 0 Nízký alarm
- 1 Vysoký alarm
- 2 2. nízký alarm
- 3 2. vysoký alarm

Úroveň 0 označuje konec podmínek alarmu. Alarmy s úrovní od 128 do 255 neoznačují konec podmínek alarmu tímto speciálním událostním alarmem. Potvrzení alarmu by mělo mít stejný podtyp a úroveň jako potvrzený alarm. Neaktivní alarmy jsou však považovány za potvrzené potvrzením kteréhokoli podtypu. První úroveň alarmu je také považováno za potvrzení potvrzením druhé úrovně.

Kapitola 5

Řešení problémů

Nejdříve je vhodné prozkoumat knihovnu příkladů, zejména sekci 0404-02_PqDrv, která se týká použití PqDrv.

Stejně jako v případě jakéhokoli jiného problému se doporučuje prohlížet chybové a ladící informace (sekce **System Log** v REXYGEN Studio). Neúspěšné připojení k databázi a/nebo špatně nakonfigurované SQL dotazy jsou uvedeny v protokolu. Nejčastější problémy zahrnují:

- Požadované tabulky nejsou k dispozici v databázi. Databáze může rozlišovat velikost písmen.
- Nesoulad ve jménech sloupců. Databáze může rozlišovat velikost písmen.
- Ačkoli ovladač PqDrv používá velmi jednoduchou syntaxi SQL, existují některé rozdíly mezi jednotlivými databázovými systémy.
- Zejména při ruční definici SQL dotazů je nutné důkladně ověřit syntaxi.
- Duplicitní název položky v parametru **Items**, což má za následek nedostupnost položky.

V případě, že daný vstup nebo výstup funguje s jinými softwarovými nástroji a nefunguje v REXYGEN, prosím oznamte nám problém. E-mail je preferován, kontaktujte nás na support@rexygen.com. Za účelem co nejrychlejšího zpracování vaší žádosti nás prosím informujte o následujících informacích:

- Identifikace REXYGEN systému, který používáte. Jednoduše jej exportujte do souboru pomocí REXYGEN Studio (Target → Licensing → Export).
- Stručný a přesný popis vašeho problému.
- Konfigurační soubory REXYGEN (.mdl soubory) zredukované na nejjednodušší případ, který stále demonstruje problémové chování.

Literatura

- [1] REX Controls s.r.o.. *Začínáme se systémem REXYGEN na Debian*, 2020. [→](#).
- [2] REX Controls s.r.o.. *Funkční bloky systému REXYGEN – Referenční příručka*, 2020. [→](#).