

# Ovladač systému REXYGEN pro OPC UA (modul `OpcUaDrv`)

## Uživatelská příručka

REX Controls s.r.o.

Verze 2.50.12  
Plzeň  
23.9.2022

# Obsah

<b>1</b>	<b>OpcUaDrv modul systému REXYGEN</b>	<b>2</b>
1.1	Úvod . . . . .	2
1.2	Instalace ovladače na cílovém zařízení . . . . .	2
1.2.1	Zařízení s operačním systémem Windows . . . . .	2
1.2.2	Zařízení s operačním systémem Linux . . . . .	2
1.3	Alternativy . . . . .	3
<b>2</b>	<b>Zavedení ovladače do projektu</b>	<b>4</b>
2.1	Přidání OpCuaDrv ovladače . . . . .	4
2.2	Konfigurační dialog OpCuaDrv ovladače . . . . .	5
2.3	OPC UA Klient . . . . .	5
2.4	OPC UA Server . . . . .	6
<b>3</b>	<b>Zapojení vstupů a výstupů a použití funkčních bloků ovladače v řídicím algoritmu</b>	<b>9</b>
3.1	Signály přímých vstupů a výstupů ovladače . . . . .	9
3.2	Funkční bloky ovladače . . . . .	9
3.2.1	OPC UA Klient . . . . .	9
3.2.2	OPC UA Server . . . . .	10
<b>4</b>	<b>Příklady</b>	<b>11</b>
<b>5</b>	<b>Odstranění problémů</b>	<b>12</b>
	<b>Literatura</b>	<b>13</b>

# Kapitola 1

## OpcUaDrv modul systému REXYGEN

### 1.1 Úvod

V tomto manuálu je popsán **OpcUaDrv** modul pro komunikaci přes protokol OPC UA v rámci systému REXYGEN. Ovladač byl vyvinut společností REX Controls.

OPC UA je otevřený komunikační protokol pro průmyslovou automatizaci. Na rozdíl od předchozích OPC protokolů je OPC UA multiplatformní protokol, který může být použit jako webová služba a krom standardních funkcí jako jsou například události a přístup k datům nabízí i pokročilé funkce jako je diagnostika, vzdálené volání metod a různé úrovně zabezpečení a autentizace. OPC UA se stává preferovaným komunikačním rozhraním mnoha zařízení různých výrobců.

OPC UA protokol obecně není vhodný pro komunikaci striktně v reálném čase, ale implementace ovladače je optimalizována tak, aby poskytovala co nejnižší latence. Mezi hlavní oblasti využití jsou rozhraní člověk-stroj a propojení různých zařízení v heterogenním prostředí. Více informací naleznete ve specifikaci OPC UA protokolu [1] (pouze anglicky).

### 1.2 Instalace ovladače na cílovém zařízení

#### 1.2.1 Zařízení s operačním systémem Windows

Část ovladače pro cílové zařízení, která slouží pro běh **OpcUaDrv** on Windows 7/8/10 je součástí instalace vývojových nástrojů systému REXYGEN.

#### 1.2.2 Zařízení s operačním systémem Linux

Pokud není na cílovém zařízení nainstalován **RexCore** runtime modul, pak je nutné ho nainstalovat podle kroků popsaných v návodu REXYGEN [2]. Instalace obsahuje všechny dostupné ovladače včetně **OpcUaDrv**.

Pro instalaci ovladače odděleně může být z terminálu použit příkaz  
`sudo apt-get install rex-opcuadrvt`

### 1.3 Alternativy

Kromě `OpcUaDrv` existuje také samostatná aplikace `OPC UA server` pro REXYGEN. Tato aplikace je připojena k systému REXYGEN a vystavuje všechny datové signály definované v algoritmu REXYGEN jako uzly protokolu OPC UA. Další více informací viz `OPC UA server - Referenční manuál` [3].

V porovnání s `OPC UA serverem` pro REXYGEN umožňuje ovladač nejen přístup k vybraným datovým signálům systému REXYGEN v roli serveru, ale i vyčítat data z jiných zařízení v roli klienta.

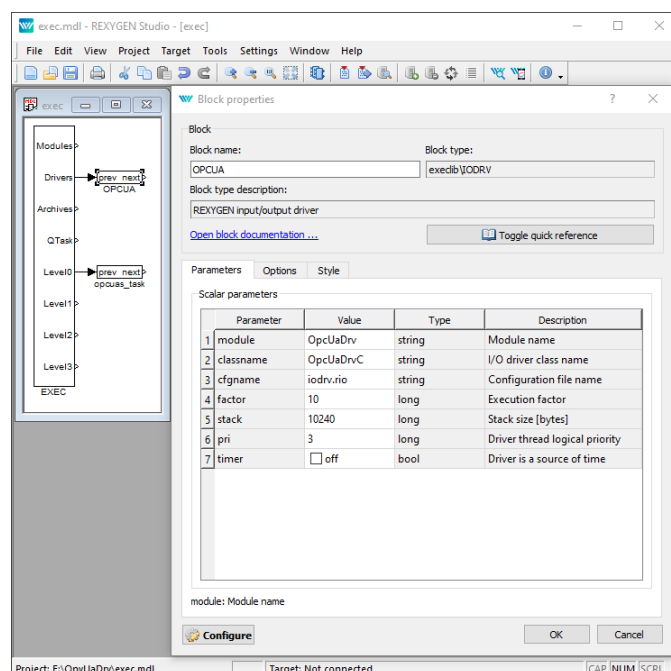
## Kapitola 2

# Zavedení ovladače do projektu

Ovladač je zaveden do projektu v momentě, kdy je ovladač přidán do hlavního souboru a vstupy a výstupy jsou zapojeny do řídicího algoritmu.

### 2.1 Přidání OpcUaDrv ovladače

Hlavní soubor projektu se zavedeným ovladačem je zobrazen na obrázku 2.1.



Obrázek 2.1: Příklad hlavního souboru projektu se zavedeným ovladačem OpcUaDrv.

Pro zavedení ovladače musí být do projektu přidán blok typu IODRV a musí být napojen

na výstup **Drivers** bloku **EXEC**. Jméno tohoto bloku (**OPCUA**, viz obr. 2.1) musí být zároveň prefixem všech vstupních a výstupních signálů, které ovladač poskytuje.

Mezi nejvýznamnější parametry bloku **IODRV** patří:

- **module** – jméno modulu, na který je ovladač napojen, v tomto případě **OpcUaDrv**
- **classname** – třída ovladače, která definuje roli cílového zařízení:
  - OpcUaDrvC** – driver vystupuje jako OPC UA Klient
  - OpcUaDrvS** – driver vystupuje jako OPC UA Server
- **cfgname** – jméno konfiguračního souboru, např. **opcua\_cfg.rio**
- **factor** – násobek parametru **tick** bloku **EXEC** definující periodu spouštění ovladače

Výše zmíněné parametry bloku **IODRV** jsou konfigurovány v programu **REXYGEN Studio**. Konfigurační dialog je také zobrazen na obrázku 2.1.

Tlačítko **Configure** otevírá konfigurační dialog **OpcUaDrv** ovladače, který je popsán v kapitole 2.2.

## 2.2 Konfigurační dialog OpcUaDrv ovladače

Konfigurační dialog se zobrazí z **REXYGEN Studio** kliknutím na tlačítko **Configure** v parametrickém dialogu bloku **IODRV** (přejmenovanému na **OPCUA**, viz kapitola 2.1).

## 2.3 OPC UA Klient

Každá instance ovladače může obsluhovat více spojení se servery. Konfigurační dialog ovladače ve verzi OPC UA Klient je zobrazen na obrázku 2.2 a má podobu tabulky jednotlivých spojení s tlačítky pro přidání, editaci a mazání spojení. Konfigurační dialog editace spojení je zobrazen na obrázku 2.3.

Parametry spojení:

**Alias** – povinné, *Alias* musí být použit ve jménu vlajek a bloků, které patří do kontextu tohoto spojení.

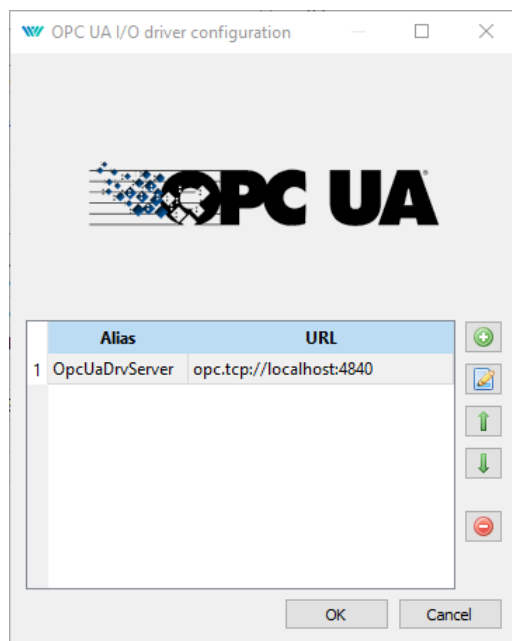
**URL** – adresa serveru

**Reconnection timeout** – počet sekund mezi jednotlivými pokusy o obnovu spojení

**Read/Write timeout** – počet sekund na dokončení operací čtení a zápisu

**Authentication** – sekce pro konfiguraci autentizace

**Security** – sekce pro konfiguraci zabezpečení



Obrázek 2.2: Ovladač ve módu OPC UA Klient – tabulka spojení

**Namespace definitions** – sekce pro definice jmenných prostorů tak, jak jsou definované na serveru

- indexy jmenných prostorů jsou obsaženy v parametru **NodeId** OPC UA bloků
- V průběhu inicializace spojení jsou jmenné prostory z tabulky resolvovány vůči jmenným prostorům serveru a indexy na straně klienta jsou převedeny na skutečné indexy na straně serveru.

## 2.4 OPC UA Server

Konfigurační dialog pro OPC UA Server je zobrazen na obrázku 2.4.

Parametry serveru:

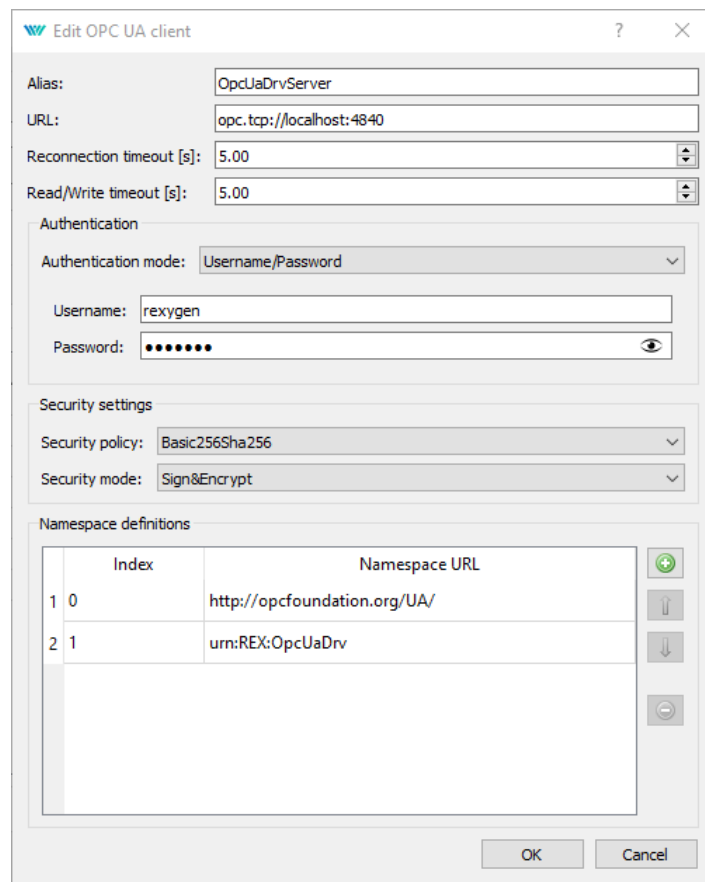
**Port** – port serveru

**Application URI** – identifikátor aplikace

- Identifikátor aplikace musí být stejný jako identifikátor definovaný v certifikátu používaným serverem.
- Pokud je používán vlastní certifikát, identifikátor aplikace musí být adekvátně nastaven. V opačném případě by parametr měl zůstat prázdný.

**Namespace** – jmenný prostor, který bude oužit pro všechny uzly serveru

**Certificate path** – cesta na cílovém zařízení k vlastnímu certifikátu, Pro použití standardního certifikátu by měl parametr zůstat prázdný.



Obrázek 2.3: Ovladač ve módu OPC UA Klient – konfigurace spojení

**Private key path** – cesta na cílovém zařízení k vlastnímu soukromému klíči certifikátu,  
Pro použití standardního certifikátu by měl parametr zůstat prázdný.

**Authentication mode** – konfigurace autentizace

- *Anonymous* – bez autentizace
- *Username/Password* – autentizace uživatelským jménem a heslem

**Username** – uživatelské jméno

**Password** – heslo

**Security policy** – zásady bezpečnosti pro autentizaci



OPC UA I/O driver configuration

**OPC UA**

Port: 4840

Application URI:

Namespace: urn:REX:OpcUaDrv

**Security**

Certificate path:

Private key path:

**Authentication**

Authentication mode: Username/Password

Username: rexygen

Password: •••••

Security policy: Basic256Sha256

OK Cancel

Obrázek 2.4: Konfigurace ovladače v módu OPC UA Server

## Kapitola 3

# Zapojení vstupů a výstupů a použití funkčních bloků ovladače v řídicím algoritmu

Vstupy a výstupy ovladače musí být zapojeny uvnitř schématu konkrétních úloh konfigurace (v `.mdl` souborech). Samostatné úlohy (bloky `QTASK` nebo `TASK`) jsou napojeny na výstupy `QTask`, `Level0`, ..., `Level3` hlavního bloku `EXEC`.

### 3.1 Signály přímých vstupů a výstupů ovladače

Použití vstupů a výstupů ovladače `OpcUaDrv` je znázorněno na obrázku 3.1.

První blok typu `From` umožňující uživateli číst informaci o stavu spojení má `Goto tag` nastavený na `OPCUAC__OpcUaServer_ConnectionStatus`.

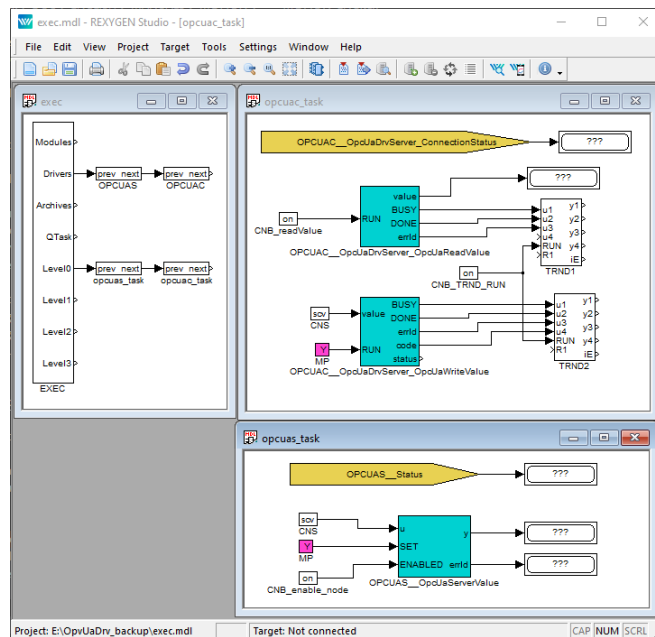
Další blok typu `From` umožňující uživateli číst stav serveru má `Goto tag` nastavený na `OPCUAS__Status`. Bloky vždy musí mít jméno ovladače jako prefix následovaný dvěma `_` (podtržítka) znaky na začátku tagu. Pokud blok náleží k ovladači v módu OPC UA Klient, po jménu ovladače s podtržítka musí následovat alias spojení definovaném v konfiguračním dialogu a dalším podtržítka.

### 3.2 Funkční bloky ovladače

#### 3.2.1 OPC UA Klient

Ovladač v módu OPC UA Klient je zodpovědný za udržování spojení se serverem. Pro čtení hodnoty OPC UA uzlu přes OPC UA protokol slouží blok `OpcUaReadValue`. Pro zápis hodnoty slouží blok `OpcUaWriteValue`. Příklad použití funkčních bloků ovladače je znázorněn na obrázku 3.1.

Funkční bloky musí mít jméno ovladače jako prefix následovaný dvěma `_` (podtržítka) znaky na začátku tagu a po jménu ovladače s podtržítka musí následovat alias spojení



Obrázek 3.1: Ukázka použití vstupních a výstupních vlajek a bloků `OpUaDrv` ovladače

definovaném v konfiguračním dialogu a dalším podtržítkem. Pro více informací o blocích `OpUaReadValue` a `OpUaWriteValue` lze nahlédnout do referenční příručky bloků [4].

### 3.2.2 OPC UA Server

Ovladač v módu OPC UA Server je zodpovědný za obsluhu veškeré komunikace s připojenými klienty. Pro definování a vystavení OPC UA uzlu slouží blok `OpUaServerValue`. Příklad použití funkčních bloků ovladače je znázorněn na obrázku 3.1. Funkční bloky musí mít jméno ovladače jako prefix následovaný dvěma \_ (podtržítka) znaky na začátku tagu. Pro více informací o bloku `OpUaServerValue` lze nahlédnout do referenční příručky bloků [4].

## Kapitola 4

# Příklady

Pro snadný start lze využít následující příklady jako referenci a jejich modifikováním lze dosáhnout cíleného chování dané aplikace.

- 0408-01 OPC UA Communication/OPC UA Data Exchange – Příklad demonstruje komunikaci mezi ovladačem v módu OPC UA Klient a ovladačem v módu OPC UA Server.

## Kapitola 5

# Odstranění problémů

V případě, že diagnostické nástroje systému REXYGEN (např. REXYGEN Diagnostics) reportují neočekávané či chybové hodnoty vstupů a výstupů, je vhodné vyzkoušet požadované funkce mimo systém REXYGEN. Existuje velké množství volně dostupných nástrojů, které mohou být použity pro monitorování OPC UA komunikace jako například [UaExpert](#).

V případě, že dané vstupy a výstupy fungují v kombinaci s jinými softwarovými prostředky a nefungují v kombinaci se systémem REXYGEN, prosím, reportujte nám tento problém. Preferovaný způsob je zaslání e-mailu na adresu [support@rexygen.com](mailto:support@rexygen.com). Připojte prosím tyto informace v popis problému, abychom byli schopni problém co nejdříve vyřešit:

- Identifikace systému REXYGEN, který je používán. Stačí jednoduše provést export z programu REXYGEN Studio volbou v menu Target → Licensing... → Export.
- Krátký a výstižný popis problému.
- Konfigurační soubory systému REXYGEN (.mdl a .rio files) redukováné na svou nejjednodušší podobu, která stále demonstuje problematické chování.

# Literatura

- [1] OPC Foundation. OPC Unified Architecture Specification, 2020.
- [2] REX Controls s.r.o.. *Začínáme s řídicím systémem REXYGEN*, 2020. [→](#).
- [3] REX Controls s.r.o.. *OPC UA server pro systém REXYGEN – Referenční manuál*, 2019. [→](#).
- [4] REX Controls s.r.o.. *Funkční bloky systému REXYGEN – Referenční příručka*, 2020. [→](#).