



[www.rexcontrols.cz/rex](http://www.rexcontrols.cz/rex)

---

# Ovladač SimDrv systému REX

## Uživatelská příručka

REX Controls s.r.o.

Verze 2.50.1

Plzeň

7.11.2016

# Obsah

<b>1</b>	<b>Ovladač SimDrv a systém REX</b>	<b>2</b>
1.1	Úvod . . . . .	2
1.2	Požadavky na systém . . . . .	2
1.3	Instalace ovladače . . . . .	2
<b>2</b>	<b>Zařazení ovladače do projektu aplikace</b>	<b>4</b>
2.1	Přidání ovladače SimDrv do projektu . . . . .	4
2.2	Připojení vstupů a výstupů do řídicího algoritmu . . . . .	5
<b>3</b>	<b>Konfigurace ovladače</b>	<b>7</b>
3.1	Konfigurační dialogové okno . . . . .	7
3.1.1	Globální ovládací prvky . . . . .	8
3.1.2	Zobrazení zkonfigurovaných signálů . . . . .	8
3.1.3	Konfigurace vstupů a výstupů . . . . .	9
<b>4</b>	<b>Poznámky k implementaci</b>	<b>10</b>
<b>5</b>	<b>Co dělat při problémech</b>	<b>11</b>

# Kapitola 1

## Ovladač SimDrv a systém REX

### 1.1 Úvod

Simulační ovladač byl vyvinut za účelem usnadnění ladění algoritmu systému REX na jiné platformě (obvykle osobním počítači). Ovladač nedokáže simulovat celou technologii, ale umožňuje generovat různé periodické signály a generovat signál na základě jiné hodnoty. Koncepce simulačního ovladače umožňuje k existujícímu schématu ( .mdl soubory) s odkazy do libovolného ovladače systému REX velice rychle vytvořit konfiguraci simulačního ovladače ( .rio soubor).

V této příručce je popsáno používání ovladače **SimDrv** vyvinutého firmou REX Controls , který je vhodný pro simulace na jiné než cílové platformě.

### 1.2 Požadavky na systém

Ovladač **SimDrv** je možné provozovat na všech platformách a systémech, kde je možné provozování plnohodnotného řídicího systému REX. Ovladač nelze provozovat pouze na systémech, kde funguje zjednodušená verze řídicího systému REX bez podpory ovladačů (mikroREX, JavaREX, RexLib a podobně).

### 1.3 Instalace ovladače

Ovladač **SimDrv** se instaluje buď jako součást instalace řídicího systému REX nebo samostatně po nainstalování řídicího systému. Při samostatné instalaci je pro správnou funkci ovladače nutno zvolit stejný cílový adresář jako při instalaci řídicího systému REX, např. C:\Program Files\REX Controls\REX\_<version>, kde <version> označuje verzi systému REX.

Po úspěšné instalaci se do cílového adresáře zkopírují soubory:

**SimDrv\_H.dll** – Konfigurační část ovladače **SimDrv**.

**SimDrv\_T.dll** – Cílová část ovladače **SimDrv** spouštěná exekutivou **RexCore**.

DOC\SimDrv\_CZ.pdf – Tato uživatelská příručka.

Pro jiné cílové platformy než Windows je nutno ještě doinstalovat cílovou část ovladače pro cílovou platformu. Obecně platí, že je potřeba nakopírovat příslušný soubor tam, kde jsou ostatní ovladače systému REX .

# Zařazení ovladače do projektu aplikace

## 2.1 Přidání ovladače SimDrv do projektu

The diagram illustrates the EXEC module hierarchy. It shows a tree structure starting from a root 'EXEC' box. The hierarchy is as follows:

- Modules**: Contains a yellow box labeled 'prev next' (SimDrv) which points to a white box labeled 'prev next' (MbDrv).
- Drivers**: Contains a cyan box labeled 'prev next' (SIM) which points to a white box labeled 'prev next' (MTM).
- Archives**: A list of modules including 'QTask', 'Level 0', 'Level 1', 'Level 2', and 'Level 3'.
- Task1**: A white box labeled 'prev next' that is pointed to by 'Level 0' and points to 'Task2'.
- Task2**: A white box labeled 'prev next' that is pointed to by 'Task1' and has an outgoing arrow.

Pro zařazení ovladače do projektu slouží dva zvýrazněné bloky. Nejprve je na výstup **Modules** exekutivy **EXEC** připojen blok s názvem **SimDrv** typu **MODULE**, který nemá žádné další parametry.

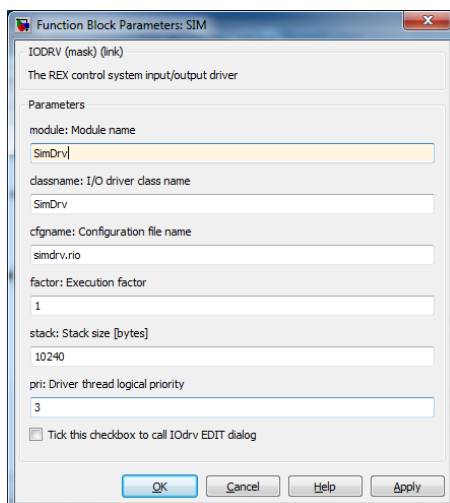
Druhý blok **SIM** typu **IODRV**, připojený na výstup **Drivers** exekutivy má dva parametry:

**classname** – jméno třídy ovladače, které se pro tento ovladač zadává **SimDrv**. Pozor, jméno rozlišuje velká a malá písmena!

**cfgname** – jméno konfiguračního souboru ovladače. Vytváření konfiguračního souboru je popsáno v kapitole 3.

Jménem tohoto bloku, na obr. 2.1 zadaným jako **SIM**, začínají názvy všech vstupních a výstupních signálů připojených k tomuto ovladači.

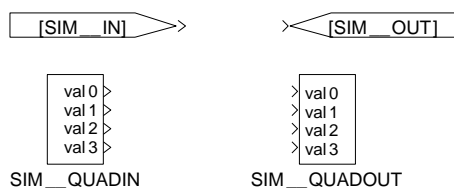
Právě popsané parametry bloku se v systému Matlab Simulink zadávají v parametrickém dialogu znázorněném na obrázku 2.2. Poslední parametr slouží k volání konfiguračního dialogu ovladače přímo z prostředí programu Matlab Simulink. Pokud při editaci parametrů je invertováno zaškrtnutí tohoto parametru, bude po stisku tlačítek **OK** nebo **Apply** zavolán konfigurační dialog ovladače **SimDrv**, popsaný v kapitole 3.



Obrázek 2.2: Konfigurace parametrů ovladače v programu Matlab Simulink

## 2.2 Připojení vstupů a výstupů do řídicího algoritmu

Vstupy a výstupy z ovladačů se připojují do souborů s příponou **.mdl** jednotlivých úloh. V hlavním souboru projektu jsou soubory úloh uvedeny pouze odkazem v blocích typu **QTASK** nebo **TASK** připojovaných na výstupy **QTask**, **Level0**, ..., **Level3** exekutivy. Pro připojení vstupů a výstupů z ovladače **SimDrv** do řídicího systému REX lze použít například bloky, znázorněné na obr. 2.3. Ovladač **SimDrv** je navržen pro jednoduché vstupy a výstupy (bloky **From** a **Goto**). Vícenásobné vstupy a výstupy podporuje jen omezeným způsobem - z výstupní hodnoty se použije jen první, všechny vstupy vícenásobného bloku mají stejnou hodnotu.



Obrázek 2.3: Příklady použití vstupně-výstupních bloků s ovladačem **SimDrv**

Blok typu **From** sloužící pro připojení jednoho vstupu má parametr **Goto tag** roven **SIM\_Measure1**, blok typu **Goto** používaný pro připojení jednoho výstupu má hodnotu parametru **Tag** rovní **SIM\_TaskOut1**. Ostatní bloky mají přímo na začátku svého jména prefix **SIM** následovaný dvěma znaky **\_** (podtržítko).

Přesněji řečeno, daný vstupně výstupní blok je považován systémem **REX** za blok připojený k ovladači **SimDrv**, pokud jeho jméno (či, v případě bloků typu **From** a **Goto**, parametry **Goto tag** a **Tag**) začíná jménem bloku typu **IODRV** popisujícího daný ovladač (na obr. 2.1 to byl právě blok **SIM**). Začátek jména vstupního nebo výstupního bloku je od zbytku jména vždy oddělen dvěma znaky **\_**. Zbytek jména vstupně výstupního bloku je uživatelským jménem signálu, zadávaným v konfiguraci ovladače, popsané v následující kapitole.

Kdyby byl např. blok **SIM** z obr. 2.1 přejmenován na **XY**, začínala by jména všech vstupně výstupních bloků připojených k ovladači **SimDrv** znaky **XY\_**. Z praktických důvodů je však rozumnější volit prefix mnemotechnicky blízký názvu ovladače. Při typickém použití to však bude prefix mnemotechnicky blízký názvu cílového ovladače (viz též dále).

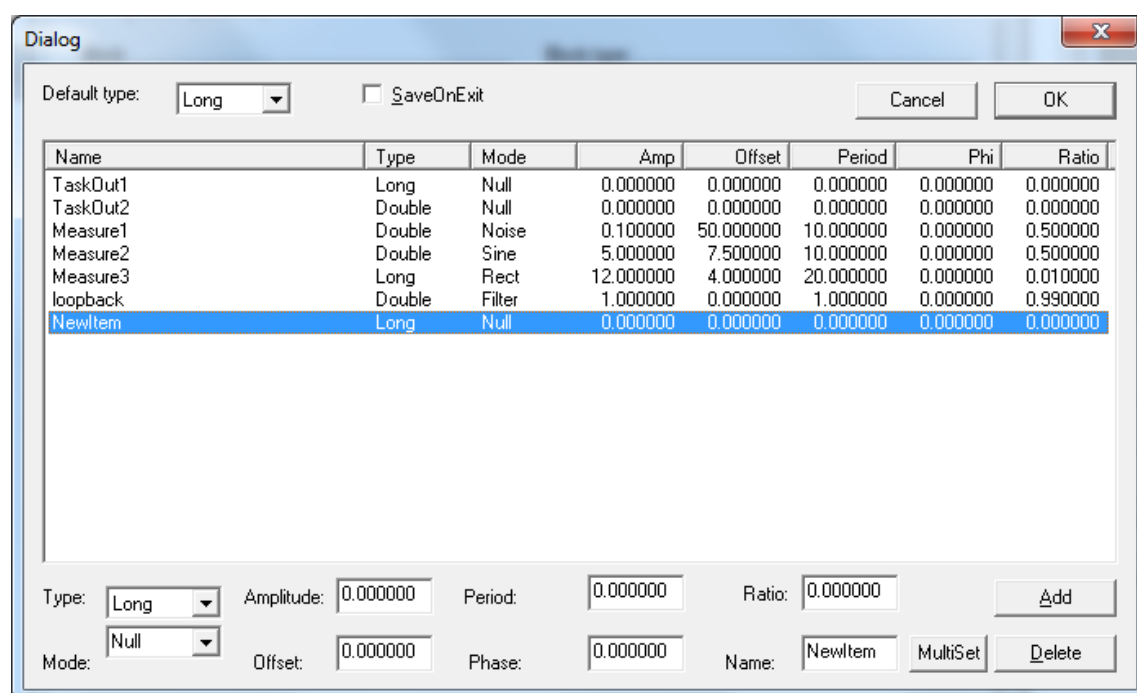
Použití bloků **From** a **Goto** pro vstup a výstup jednoho signálu do/z řídicího algoritmu umožňuje snadno přecházet ze simulační verze algoritmu testované v systému Matlab Simulink do systému reálného času **REX**. V systému Simulink je možno k blokům **From** a **Goto** přiřadit „protikusy“, kterými bude připojen simulační model procesu, po otestování může být model procesu z projektu odstraněn. Při překladu modelu nahradí díky zavedené a právě popsané konvenci systém **REX** zbylé bloky **From** a **Goto** vstupními a výstupními bloky.

## Kapitola 3

# Konfigurace ovladače

Konfigurace ovladače spočívá ve vytvoření a symbolickém pojmenování vstupně-výstupních signálů a jejich přiřazení k hodnotám poskytovaným. Obecný popis konfiguračního dialogového okna a postup při konfiguraci jednotlivých typů vstupů a výstupů je uveden v následujících sekcích této kapitoly.

### 3.1 Konfigurační dialogové okno



Obrázek 3.1: Konfigurační dialog vstupů a výstupů



Konfigurační dialogové okno, které je znázorněno na obr. 3.1 je obsaženo v souboru `SimDrv_H.dll` a aktivuje se v programu `RexDraw` stisknutím tlačítka `Special edit` v parametrickém dialogu bloku typu `IODRV` s parametry ovladače `SimDrv` (viz kap. 2).

Při vytváření konfigurace řídicího systému z editoru programu Matlab Simulink lze aktivovat konfigurační okno přímo z parametrického dialogu bloku typu `IODRV`, viz obr. 2.2.

### 3.1.1 Globální ovládací prvky

V dialogu je několik objektů. V horní části jsou to tlačítka a menu s globálním významem:

**OK** – Ukončení editace signálů a uložení konfiguračního souboru.

**Cancel** – Ukončení editace signálů bez uložení konfiguračního souboru, takže všechny provedené změny od posledního uložení konfiguračního souboru se nenávratně ztratí.

**Default type** – Simulační ovladač umožňuje automaticky přidávat položky, které jsou ve řídicím schématu, ale nejsou součástí konfigurace ovladače. Takto přidaná položka je potom typu, který vybereme v tomto políčku. Pokud vybereme **Unknown**, položka se nepřidá a při překladu je hlášena chyba.

**SaveOnExit** – Pokud je pole zaškrtnuto, tak se automaticky přidané položky uloží do konfiguračního souboru simulačního ovladače. Pokud není zaškrtnuto, tak se položky jen použijí při překladu programem `RexComp` a do konfiguračního souboru se nepřidají.

Další tlačítka ve spodní části slouží pro manipulaci se signály:

**Add** – Vložení nového signálu. Signál se přidává na konec.

**Delete** – Odstraní označené signály ze seznamu signálů.

**MultiSet** – Hromadně nastaví typ a další zadané parametry u všech označených signálů ze seznamu signálů.

### 3.1.2 Zobrazení zkonfigurovaných signálů

Ve střední části dialogového okna jsou v tabulkové formě zobrazeny zkonfigurované signály. Každý řádek odpovídá signálu, jemuž je přiřazen jeden symbolický název (viz sekci 3.1.3). V případě většího počtu řádků než se vejde do vymezeného prostoru se v pravé části objeví posouvací pruh (scroll bar), pomocí kterého lze standardním způsobem pro Windows zobrazovat požadovanou část seznamu signálů. Jednotlivé sloupce tabulky mají stejný význam jako zadávací pole při konfiguraci a jsou popsány v sekci 3.1.3.

### 3.1.3 Konfigurace vstupů a výstupů

Ostatní prvky dialogu (tj. zadávací pole ve spodní části ) z obr. 3.1 slouží pro konfiguraci vstupních a výstupních signálů. Jsou to tato políčka:

**Name** – Jméno signálu v systému REX. Musí být zadáno jednoznačně.

**Type** – Typ signálu. Význam je zřejmý. Doporučuje se přednostně používat typy:

**Bool** logický signál,  
**Long** celé číslo (32 bitů se znaménkem, tj. -2147483648 ... 2147483647)  
**Double** desetinné číslo (v rozsahu 64 bitů)

**Mode** – Režim generování signálu. Pro výstupy (tj. signály z .mdl výkresu do ovladače) musí být **Null**. Pro vstup je několik generátorů, přičemž hodnota z generátoru je násobena parametrem **Amplitude** a k výsledku je ještě přičtena hodnota parametru **Offset**, tj.  $y = Amplitude * x + Offset$ . U periodických generátorů je počátek generování posunut o hodnotu danou parametrem **Phase**. Režimy jsou následující:

**Null** hodnota 0 (po dalším zpracování tedy hodnota parametru **Offset**)  
**Noise** náhodné číslo  $0 \leq x \leq 1$   
**Rectangle** střídají se hodnoty -1 po dobu  $Ratio * Period$  a -1 po zbytek periody  
**Sawtooth** hodnota  $-1 \leq x \leq 1$  s lineárním nárůstem po dobu  $Ratio * Period$  a lineárním poklesem po  
**Sin** funkce  $y = \sin(\pi * t / Period)$   
**Filter** filtr 1. řádu ze signálu, jehož číslo je v paramtru **Period**, tj.  $x_i = Ratio * x_{i-1} + (1 - Ratio)$

**Amplitude** – Amplituda generovaného signálu.

**Offset** – Posunutí generovaného signálu (přičítá se k hodnotě)

**Period** – Perioda generátoru v sekundách

**Phase** – Relativní posunutí počátku generátoru v čase; může nabývat hodnot  $0 \leq x < 1$ .

**Ratio** – Podíl první části periody v případě typu **Sawtooth** a **Rectangle**; může nabývat hodnot  $0 \leq x < 1$

## Kapitola 4

# Poznámky k implementaci

V této kapitole jsou soustředěny poznatky, které vznikly z dosavadních zkušeností. Některé položky v konfiguraci jsou často nesprávně pochopeny, ale podrobný popis výše by zhoršoval čitelnost textu. Proto jsou tyto postřehy uvedeny ve zvláštní kapitole. Je zde také podrobně popsán způsob přenosu hodnot.

- Ovladač se typicky používá v situaci, kdy cílovou platformu nemáme k dispozici. Ve všech blocích v `IODRV` v `exec.mdl` přepíšeme parametr `modulname` a `classname` na hodnotu `SimDrv` a nastavíme jiné jméno konfiguračního souboru (parametr `cfname`). Dále stiskneme tlačítko **Configure** (konfigurační soubor neexistuje - zaklikneme vytvořit nový) a nastavíme `Default type` na `long` (lze i jiný, ale typ `long` se implicitně konvertuje na `bool` i `double` a jiné typy se vyskytují zřídka) a zaklikneme **SaveOnExit** a uložíme tlačítkem **OK**. Pak projekt přeložíme (v `RexDraw` v menu `Compile|Compile`). Nyní již lze vytvořený `exec.rex` spustit, ale na všech vstupech bude hodnota 0. Nyní je možné znovu spustit konfiguraci simulačního ovladače (tlačítko **Configure**) a nastavit u vstupů vhodnější generátory popřípadě typy hodnot (podle povahy technologie).

## Kapitola 5

# Co dělat při problémech

Nejčastější chyby jsou:

Informace budou doplněny na základě dotazů uživatelů

V případě, že daný ovladač **SimDrv** funguje v jednoduchých testovacích příkladech správně a při potřebné konfiguraci nefunguje, prosíme o zaslání informace o problému (nejlépe elektronickou cestou) na adresu dodavatele. Pro co nejrychlejší vyřešení problému by informace by měla obsahovat:

- Identifikační údaje Vaší instalace – verzi, číslo sestavení (build), datum vytvoření instalace, licenční číslo.
- Stručný a výstižný popis problému.
- Co možná nejvíc zjednodušenou konfiguraci řídicího systému REX, ve které se problém ještě vyskytuje (ve formátu souboru s příponou **.mdl**).
- Konfigurační soubor ovladače **SimDrv**.