

Ovladač pro komunikaci s PLC Siemens
(modul S7Drv řídicího systému REX)

Uživatelská příručka

REX Controls s.r.o.

Verze 2.50.5
Plzeň
6.9.2017

Obsah

1	Ovladač S7Drv a systém REX	2
1.1	Úvod	2
1.2	Požadavky na systém	2
1.3	Instalace ovladače na vývojovém počítači	3
1.4	Instalace ovladače na cílovém zařízení	3
1.4.1	Zařízení s Windows	3
1.4.2	Zařízení s Windows CE	3
1.4.3	Zařízení s Linuxem	3
2	Zařazení ovladače do projektu aplikace	5
2.1	Přidání ovladače S7Drv do projektu	5
3	Konfigurace ovladače	8
3.1	Připojení vstupů a výstupů do řídicího algoritmu	8
4	Poznámky k implementaci	12
5	Co dělat při problémech	13
	Literatura	14

Kapitola 1

Ovladač S7Drv a systém REX

1.1 Úvod

V této příručce je popsáno používání ovladače **S7Drv** pro propojení programovatelných automatů (PLC) a operátorských panelů (OP) firmy Siemens¹ se systémem **REX**. Ovladač používá stejný protokol jako konfigurační prostředí **STEP7**. Je podporována pouze varianta připojení přes **Ethernet**.

Ovladač **S7Drv** využívá programový balík **Snap7** [1].

Systém **REX** může být klient (**Client**, emuluje **PG – STEP7**) nebo **Server** (emuluje **PLC**).

1.2 Požadavky na systém

Obecně lze říci, že pro použití ovladače **S7Drv** musí být dodrženy minimální požadavky nutné k provozování řídicího systému **REX**. Pro konfiguraci ovladače postačuje běžný počítač **PC** (případně v průmyslovém provedení). Provozování ovladače je možné na jakémkoliv cílovém zařízení se systémem **REX**, vyžadován je **TCP/IP stack** (tj. obvykle ethernetová karta pro připojení do **LAN**, popř. modem, **WiFi** apod.).

Aby bylo možno ovladač využívat, musí být na vývojovém (konfiguračním) počítači a na cílovém zařízení (počítači) nainstalováno programové vybavení:

Vývojový počítač

Operační systém	jeden ze systémů: Windows 7/8/10
Řídicí systém REX	verze pro operační systémy Windows

Cílové zařízení

Řídicí systém REX	verze pro zvolené cílové zařízení s jedním z podporovaných operačních systémů: Windows CE, Windows 7/8/10, Linux Debian/Xenomai/OpenWrt
-------------------	---

¹Siemens a STEP jsou registrované ochranné známky společnosti Siemens AG.

V případě, že vývojový počítač je přímo cílovým zařízením (řídící systém REX bude provozován v jedné z variant Windows), instaluje se pouze jedna kopie řídícího systému REX.

1.3 Instalace ovladače na vývojovém počítači

Ovladač `S7Drv` se instaluje jako součást řídícího systému REX. Je obsažen v instalátoru vývojových nástrojů systému REX, pro jeho nainstalování je pouze nutné ho v instalačním programu systému REX zaškrtnout. Po typické instalaci se řídící systém REX nainstaluje do cílového adresáře

`C:\Program Files\REX Controls\REX_<version>`, kde `<version>` označuje verzi systému REX.

Po úspěšné instalaci se do cílového adresáře zkopírují soubory:

`S7Drv_H.dll` – Konfigurační část ovladače `S7Drv`.

`S7Drv_T.dll` – Cílová část ovladače `S7Drv` spouštěná exekutivou `RexCore`. Tato verze se používá, pokud na cílovém zařízení běží operační systém Windows 7/8/10. Pro jinou cílovou platformu je na ni třeba nainstalovat příslušnou verzi systému REX.

`DOC\CZECH\S7Drv_CZ.pdf` – Tato uživatelská příručka.

1.4 Instalace ovladače na cílovém zařízení

1.4.1 Zařízení s Windows

Část ovladače pro cílové zařízení, která se používá pro komunikaci se zařízeními Siemens na Windows 7/8/10, je součástí instalace Vývojových nástrojů řídícího systému REX, který již byl zmíněn výše.

1.4.2 Zařízení s Windows CE

Část ovladače pro cílové zařízení, která se používá pro komunikaci se zařízeními Siemens na zařízeních s Windows CE je součástí instalačního balíčku (soubor `.cab`) runtime jádra systému REX.

1.4.3 Zařízení s Linuxem

Pokud na cílovém zařízení dosud není instalováno runtime jádro `RexCore` systému REX, nejdříve jej nainstalujte podle příručky *Začínáme se systémem REX* pro danou platformu, např. [2].

Pro umožnění komunikace s PLC Siemens z algoritmů řídícího systému REX je potřeba nainstalovat ovladač. To se provede z příkazové řádky pomocí příkazu

Debian:

```
sudo apt-get install rex-s7drv
```

OpenWrt:

```
opkg install rex-s7drvt
```

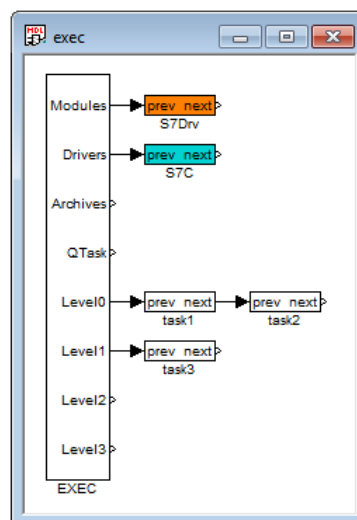
Kapitola 2

Zařazení ovladače do projektu aplikace

Zařazení ovladače do projektu aplikace spočívá v přidání ovladače do hlavního souboru projektu a v připojení vstupů a výstupů ovladače v řídicích algoritmech.

2.1 Přidání ovladače S7Drv do projektu

Přidání ovladače **S7Drv** do hlavního souboru projektu je znázorněno na obrázku 2.1. Obrázek znázorňuje přidání ovladače v režimu Client. Přidání ovladače v jiné roli má jen jiné jméno třídy (viz dále).



Obrázek 2.1: Příklad zařazení ovladače **S7Drv** do projektu aplikace

Pro zařazení ovladače do projektu slouží dva zvýrazněné bloky. Nejprve je na výstup

Modules bloku exekutivy **EXEC** připojen blok typu **MODULE** s názvem **S7Drv**, který nemá žádné další parametry.

Druhý blok **S7C** typu **IODRV**, připojený na výstup **Drivers** exekutivy má dva parametry:

classname – jméno třídy ovladače, které se pro tento ovladač zadává:

S7cDrv – pro režim klient (emulace PG)

S7sDrv – pro režim server (emulace PLC)

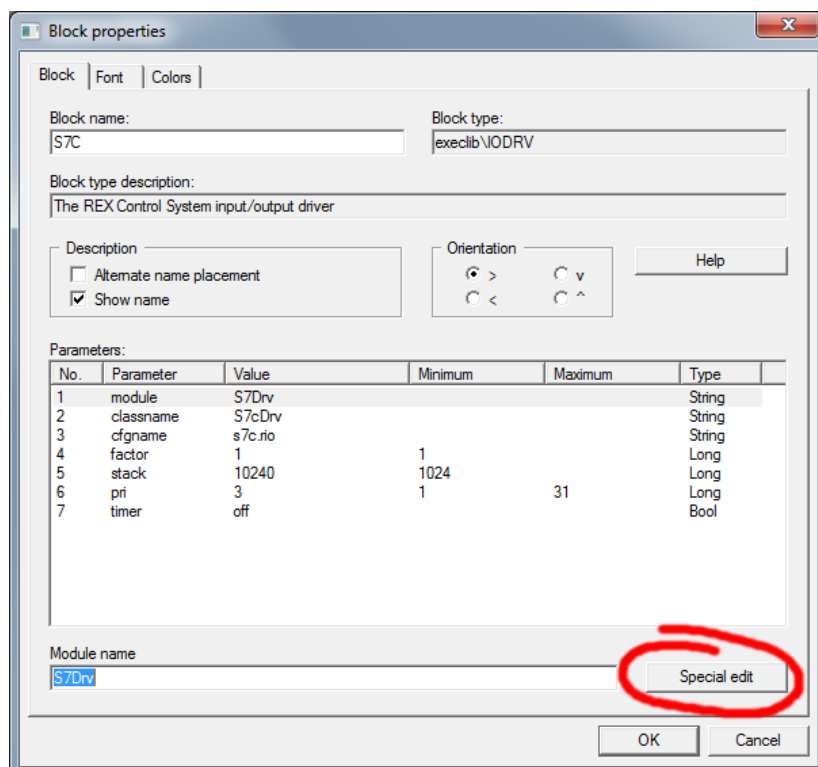
POZOR! Jméno rozlišuje velká a malá písmena!

cfgname – jméno konfiguračního souboru ovladače. Vytváření konfiguračního souboru je popsáno v kapitole 3. Doporučeno je zadávat jej ve tvaru **<jméno_třídy>.rio**, kde přípona **.rio** (REX Input/Output) byla zavedena pro tento účel.

Jménem tohoto bloku, na obr. 2.1 zadaným jako **S7C**, začínají názvy všech vstupních a výstupních signálů připojených k tomuto ovladači.

Právě popsané parametry bloku **IODRV** se konfigurují v programu **RexDraw** v dialogovém okně, jak je patrné z obrázku 2.2. Konfigurační dialog ovladače **S7Drv**, popsáný v kapitole 3, se aktivuje po stisku tlačítka **Configure**.

Ovladač **S7Drv** podporuje v režimu **Client** i úlohy běžící synchronně s komunikací. To se provede tak, že místo bloku typu **IODRV** se použije blok typu **TIODRV** (který má stejné parametry jako **IODRV**) a na jeho výstup **Tasks** připojíme blok typu **IOTASK** (má analogické parametry i význam jako blok typu **TASK**). Ovladač potom funguje tak, že nejdříve přečte všechny vstupní signály, spustí algoritmus definovaný blokem **IOTASK** a následně nastaví všechny výstupy a čeká na další periodu. Je však potřeba počítat s tím, že při přerušení komunikace se nebude **IOTASK** vykonávat, dokud nenastane tzv. *timeout* na komunikační lince, proto má využití především při periodách komunikace v řádech jednotek až desítek sekund.

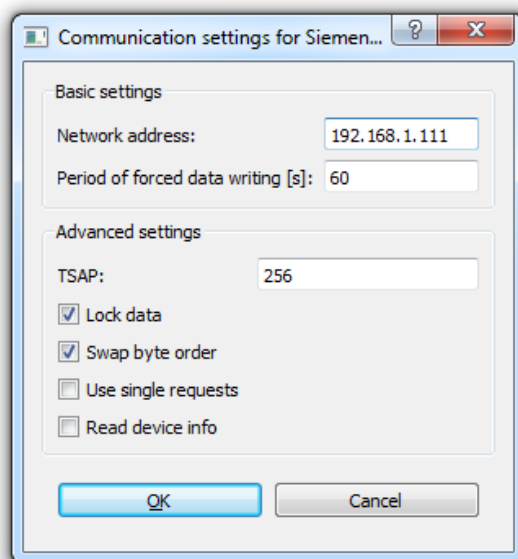


Obrázek 2.2: Vlastnosti ovladače pro zařazení do projektu

Kapitola 3

Konfigurace ovladače

Konfigurace ovladače spočívá v podstatě pouze v nastavení IP adresy zařízení, se kterým bude řídicí systém REX komunikovat. Je možné nastavit ještě několik dalších parametrů, ale implicitní hodnota ve většině případů vyhovuje.

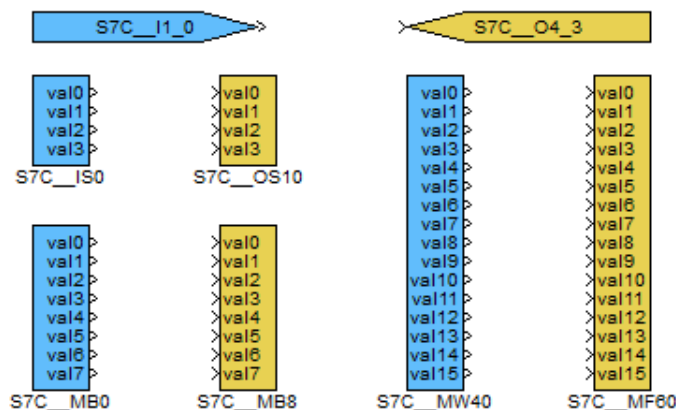


Obrázek 3.1: Konfigurační dialog ovladače

3.1 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 **S7Drv** do řídicího systému REX lze použít bloky, znázorněné na obrázku 3.2.



Obrázek 3.2: Příklady použití vstupně-výstupních bloků s ovladačem **S7Drv**

Blok typu **From** sloužící pro připojení jednoho vstupu má parametr **GotoTag** roven **S7C__<IN>**, blok typu **Goto** používaný pro připojení jednoho výstupu má tento parametr nastaven na **S7C__<OUT>**, kde **<IN>** a **<OUT>** jsou řetězce odkazující na *object dictionary* (viz dále). Všechny řetězce používané jako odkazy na data poskytovaná a přijímaná ovladačem **S7Drv** mají přímo na svém začátku prefix **S7C** povinně následovaný dvěma znaky **_** (podtržítko).

Přesněji řečeno, daný vstupně-výstupní blok je považován systémem REX za blok připojený k ovladači **S7Drv**, pokud jeho jméno (či, v případě bloků typu **From** a **Goto**, parametr **Goto tag**) začíná jménem bloku typu **IODRV** popisujícího daný ovladač. Na obr. 2.1 to byl právě blok **S7C**. Začátek jména vstupního nebo výstupního bloku je od zbytku jména vždy povinně oddělen dvěma znaky **_** (podtržítko).

Kdyby byl např. blok **S7C** z obr. 2.1 přejmenován na **XY**, začínala by jména všech vstupně-výstupních bloků připojených k ovladači **S7Drv** znaky **XY__**. Z praktických důvodů je však doporučeno volit prefix mnemotechnicky blízký názvu ovladače.

Zbytek řetězce odkazujícího na vstupní nebo výstupní data je interpretován ovladačem a má strukturu používanou ve STEP7 a doporučenou normou IEC 61131-3, pouze je místo tečky použito podtržítko, tedy:

1. **<area><type><index>**
2. **<area><index>**
3. **<area><index>_<subindex>**
4. **<area><type><index>_<subindex>**

Ve výše uvedené notaci může **<area>** nabývat následujících hodnot:

- M** – paměťová proměnná
- I** – vstup
- O** – výstup
- D** – datový blok (4. varianta formátu)
- T** – časovač (2. varianta formátu)
- C** – čítač (2. varianta formátu)

Obdobně **<type>** může být:

- B** – byte (0...255)
- W** – word (0...65535)
- D** – dword (0...4294967295)
- S** – short (-32768...32767)
- L** – long (-2147483648...2147483647)
- I** – integer, viz typ long
- F** – float (-3.4E+38...3.4E+38)

A konečně **<subindex>** jsou čísla definující objekt v *object dictionary*, jehož hodnotu čteme/zapisujeme.

Je možné číst/zapisovat další pomocné signály k danému objektu. To se provede přidáním přípony do názvu. Možnosti jsou:

- _Value** – jiný způsob zápisu pro základní hodnotu signálu
- _RE** – povolení čtení
- _WE** – povolení zápisu
- _WF** – vynucení zápisu
- _Fresh** – udává počet sekund od přijetí poslední platné hodnoty
- _Area** – umožňuje změnit kód oblasti pro signál (jen pro pokročilé uživatele)
- _Index** – umožňuje změnit index pro signál (jen pro pokročilé uživatele)
- _IndexDB** – umožňuje změnit číslo datového bloku pro signál (jen pro pokročilé uživatele)

Protože ovladač umožňuje pod jedním symbolickým jménem získávat několik vstupů či nastavovat několik výstupů, lze s výhodou používat bloky čtyřnásobných, osminásobných a šestnáctinásobných vstupů a výstupů (**INQUAD**, **OUTQUAD**, **INOCT**, **OUTOCT** a **INHEXD**, **OUTHEXD**), tak jak je znázorněno na obr. 3.2. Podrobný popis vícenásobných vstupů a výstupů lze nalézt v příručce [3]. V tomto případě je v názvu bloku odkaz na první požadovaný objekt a v následujících signálech jsou další objekty. Výhodou takového užití je zvýšení rychlosti a částečně i přehlednosti algoritmů. Podrobný popis vícenásobných vstupů a výstupů lze nalézt v příručce [3].

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.

- Pro odkazy na binární signály (např. M2.3) je v PLC zvykem (normalizováno) používat jako oddělovač tečku. To v systému REX může způsobovat potíže (některé nástroje špatně parsují jméno a pak objekt není dostupný - nelze jej číst). Proto systém REX tečku nahrazuje podtržítkem.
- Pro zařízení LOGO jsou implementována speciální označení paměťových oblastí `<area>` tak, aby byly identické s označením, které používá LOGO: `Q<index>` pro reléové výstupy, `AI<index>` pro analogové vstupy, `AM<index>` pro analogové paměťové proměnné, `AQ<index>` pro analogové výstupy.
- LOGO používá oblasti I, Q, M jen pro logické signály. Analogové signály jsou ve speciálních oblastech (area code):
 - 16 NI
 - 17 NQ (jen čtení - přepisuje algoritmus nebo nemá konfiguraci)
 - 18 AI
 - 19 AQ
 - 20 AM
 - 21 NAI
 - 22 NAQ (jen čtení - přepisuje algoritmus nebo nemá konfiguraci)

Kapitola 5

Co dělat při problémech

Jako v případě všech problémů je nejprve vhodné zkontrolovat chybové a varovné výpisy (záložka System Log v programu RexView). V případě, že je problém s komunikací nebo čtením/zápisem některých dat, bude v logu bližší popis problému.

Nejčastější zdroje problémů jsou:

- Zařízení Siemens je v režimu STOP
- Problém se sítí, konflikt IP adres
- Současný zápis do jednoho místa jak z algoritmu PLC Siemens tak z algoritmu systému REX.
- Každý komunikační standard definuje, zda se pro přenos použije little-endian nebo big-endian formát. S7COMM používá big-endian (tj. stejný jaký používají procesory Siemens). Občas se stává, že na to vývojáři zapomenou a konverzi neprovádí (problém samozřejmě vzniká, pokud procesor je little-endian, tj. například Intel nebo ARM), takže vícebytová čísla mají obrácené pořadí bytů. Pořadí bytů lze prohodit v konfiguračním dialogu ovladače.

V případě, že daný ovladač **S7Drv** 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 nejvíce zjednodušenou konfiguraci řídicího systému REX, ve které se problém ještě vyskytuje (ve formátu souboru s příponou **.mdl**).
- Konfigurační soubor ovladače **S7Drv** s příponou **.rio**.

Literatura

- [1] Davide Nardella. Snap7 - Step7 Ethernet Communication Suite. <http://snap7.sourceforge.net>, 2015.
- [2] REX Controls s.r.o.. *Začínáme se systémem REX na platformě Raspberry Pi*, 2017.
- [3] REX Controls s.r.o.. *Funkční bloky systému REX – Referenční příručka*, 2017.