Ovladač `HlsDrv` systému REX
Uživatelská příručka

REX Controls s.r.o.

Verze 2.50.3
Plzeň
28.2.2017

Obsah

Kapitola 1

Ovladač `HlsDrv` a systém REX

1.1 Úvod

V této příručce je popsáno používání ovladače `HlsDrv` pro připojení technických prostředků firmy Hilscher. Tato firma vyrábí komunikační karty netX pro různé (všechny obvyklé) průmyslové protokoly (například EtherCAT, EtherNet/IP, PROFINET, POWERLINK, SERCOS, Modbus TCP, Modbus RTU, CANopen, CC-Link, DeviceNet, PROFIBUS), přičemž ve všech případech je stejné rozhraní pro navazující aplikaci. Tyto komunikační karty se dodávají s různým rozhraním (PCI, PCIexpress, miniPCI), takže je lze použít téměř v každém počítači. Ovladač umožňuje získávat vstupy a nastavovat výstupy z tzv. „process image“. Ovladač byl vyvinut firmou REX Controls.

1.2 Požadavky na systém

Obecně lze říci, že pro použití ovladače `HlsDrv` musí být dodrženy minimální požadavky nutné k provozování řídicího systému REX. Pro konfiguraci ovladače postačuje běžný počítač PC (případně v průmyslovém provedení). Pro provozování ovladače na cílovém zařízení je potřeba do počítače zapojit komunikační kartu `netX`. Protože konfigurační software je dodáván pouze pro OS Windows, je potřeba pro prvotní nastavení (a pro jeho změny) buď `netX` kartu vložit do vývojového počítače, kde je OS Windows nebo nabootovat Windows na cílové zařízení.

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 Vista/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: Linux Debian/XENOMAI/openWRT, Windows Vista/7/8/10.

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

1.3 Instalace ovladače

Pro operační systém Windows se ovladač `HlsDrv` instaluje jako součást instalace řídicího systému REX. Pro nainstalování ovladače je nutné v instalačním programu systému REX zaškrtnout volbu `Ovladač pro komunikační karty Hilscher`. Po typické instalaci se řídicí 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:

`HlsDrv_H.dll` – Konfigurační část ovladače `HlsDrv`.

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

`DOC\HlsDrv_CZ.pdf` – Tato uživatelská příručka.

Pro operační systém Linux je potřeba nainstalovat balíček `rex-hlsdrvt`. Pro variantu Linux/XENOMAI též balíček `kmod-xuio`. Dále je nutné označit komunikační kartu, aby ji jádro Linuxu ignorovalo a mohl ji využít subsystém XENOMAI. To se provede buď tak, že se smaže modul `uio_netx.ko` (a popř. `uio_cif.ko`) nebo se zakáže v souboru `/etc/modules` (je potřeba do souboru přidat řádku `blacklist <modulname>`).

Dále je nutné nainstalovat konfigurační a ovládací program komunikační desky. Ten je na CD dodaném společně s komunikační deskou. Tento program by měl být na cílovém zařízení, ale je k dispozici jen pro operační systém Windows, takže je možné jej nainstalovat na vývojový počítač (podrobnosti viz 3.1).

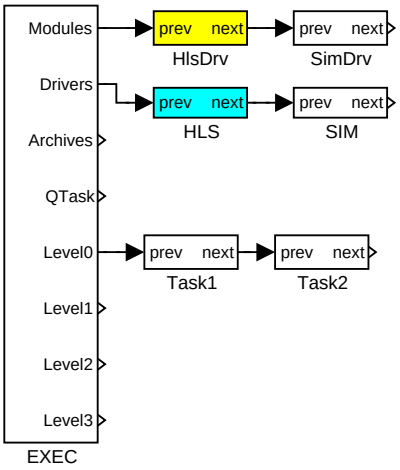
Kapitola 2

Zařazení ovladače do projektu aplikace

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

2.1 Přidání ovladače `HlsDrv` do projektu

Přidání ovladače `HlsDrv` do hlavního souboru projektu je znázorněno na obr. 2.1.



Obrázek 2.1: Příklad zařazení ovladače `HlsDrv` 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 `HlsDrv`, který nemá žádné další parametry.

Druhý blok `HLS` typu `IODRV`, připojený na výstup `Drivers` exekutivy má parametry:

`modul` – jméno třídy ovladače, které se pro tento ovladač zadává: `HlsDrv`

`classname` – jméno třídy ovladače, které se pro tento ovladač zadává: `HlsDrv` POZOR! Jméno rozlišuje velká a malá písmena!

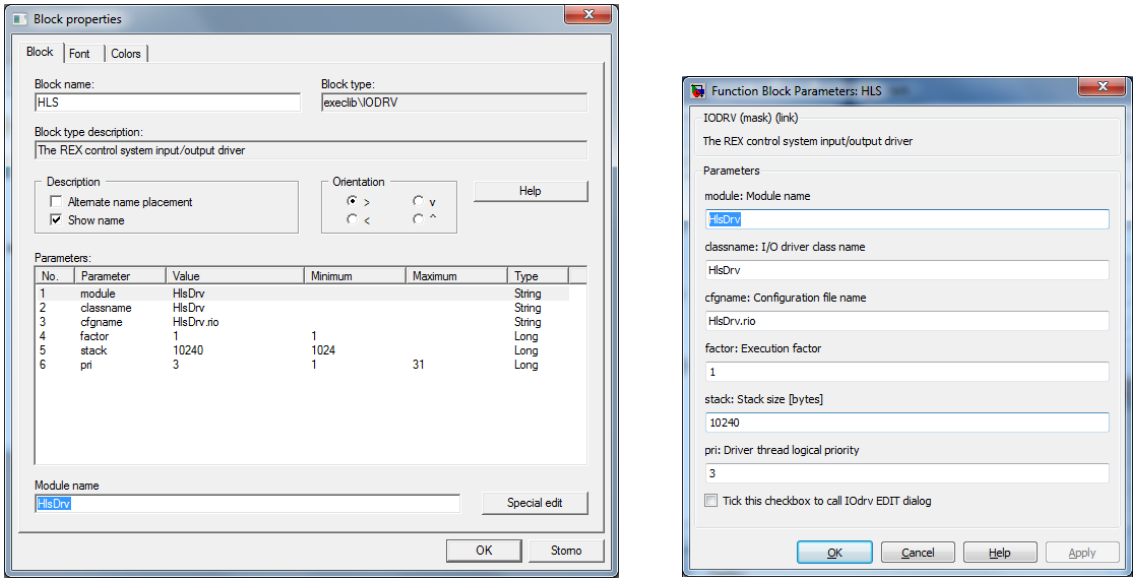
`cfgname` – jméno konfiguračního souboru ovladače. Vytváření konfiguračního souboru je popsáno v kapitole 3. Doporučeno je zadávat jej ve tvaru `<jméno_třídy>.rio`, kde přípona `.rio` (Rex Input Output) byla zavedena pro tento účel.

Další parametry slouží pro speciální případy a vyhovuje původní nastavení.

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

Ovladač `HlsDrv` podporuje 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 přečte všechny „process image“, spustí algoritmus definovaný blokem `IOTASK` nastaví všechny „process image“ a čeká na další periodu.

Právě popsané parametry bloku `IODRV` se konfigurují v programu RexDraw v dialogovém okně, jak je patrno z obr. 2.2 a). Konfigurační dialog ovladače `HlsDrv`, popsaný v kapitole 3, se aktivuje po stisku tlačítka `Special Edit`.



a) b)

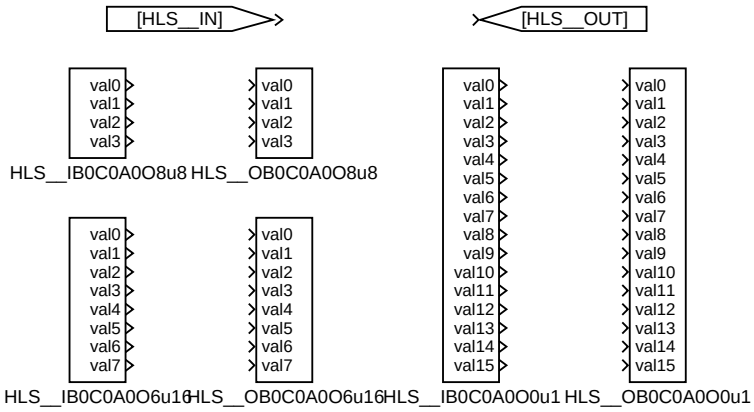
Obrázek 2.2: Konfigurace parametrů ovladače

V programovém systému Matlab Simulink se parametry bloku `IODRV` zadávají v parametrickém dialogu znázorněném na obrázku 2.2 b). 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 `HlsDrv`, popsaný v kap. 3.

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` nebo `IOTASK` připojovaných na výstupy `QTask`, `Level0`,..., `Level3` exekutivy. Pro připojení vstupů a výstupů z ovladače `HlsDrv` do řídicího systému REX lze použít bloky, znázorněné na obr. **??**.



Obrázek 2.3: Příklady použití vstupně-výstupních bloků s ovladačem `HlsDrv`

Blok typu `From` sloužící pro připojení jednoho vstupu má parametr `Goto tag` roven `HLS__IN`, blok typu `Goto` používaný pro připojení jednoho výstupu má hodnotu parametru `Tag` (v programu RexDraw parametru `GotoTag`) rovnu `HLS__OUT`. Ostatní bloky mají přímo na začátku svého jména prefix `HLS` následovaný dvěma znaky `_` (podtržítko).

Přesněji řečeno, daný vstupně-výstupní blok je považován systémem REX za blok připojený k ovladači `HlsDrv`, 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 `HLS`). Začátek jména vstupního nebo výstupního bloku je od zbytku jména vždy oddělen dvěma znaky `_`.

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

Zbytek jména vstupně-výstupního bloku má následující strukturu:

`<I|O>B<boardID>C<channelID>A<areaID>O<offset><value type>`

což představuje čtení nebo zápis hodnoty typu `<value type>` z desky číslo `<boardID>`, jejího kanálu `<channelID>` a oblasti paměti/„process image“ číslo `<areaID>`. Hodnota se nečte ze začátku paměťové oblasti, ale její začátek je posunut o `<offset>` bajtů (v

případě logické hodnoty `<offset>` bitů). Typ hodnoty může být:

- `u1` logická hodnota
- `u8` celé číslo bez znaménka o velikosti 8 bitů (tj. může nabývat hodnot 0 až 255)
- `u16` celé číslo bez znaménka o velikosti 16 bitů (tj. může nabývat hodnot 0 až 65535)
- `i16` celé číslo se znaménkem o velikosti 16 bitů (tj. může nabývat hodnot -32768 až 32767)
- `u32` celé číslo bez znaménka o velikosti 32 bitů (tj. může nabývat hodnot 0 až 4294967295)
- `i32` celé číslo se znaménkem o velikosti 32 bitů (tj. může nabývat hodnot -2147483648 až 2147483647)
- `f32` desetinné číslo o velikosti 32bitů (dle standardu IEEE 754)
- `f64` desetinné číslo o velikosti 64bitů (dle standardu IEEE 754)

Dále lze použít speciální diagnostické a řídící signály:

`<I|O>B<boardID>C<channelID>A<areaID>STATUS` – Čtení stavu oblasti.

`<I|O>B<boardID>C<channelID>A<areaID>ERROR` – Čtení chyby oblasti.

`<I|O>B<boardID>C<channelID>A<areaID>FRESH` – Vrací počet sekund (desetinné číslo) od poslední aktualizace dat v oblasti.

`<I|O>B<boardID>C<channelID>STATUS` – Čtení stavu kanálu. Při zápisu jde o zápis řídícího slova.

`<I|O>B<boardID>C<channelID>A<areaID>ERROR` – Čtení chyby kanálu.

`<I|O>B<boardID>STATUS` – Čtení stavu desky (jejího systémového kanálu).

`<I|O>B<boardID>ERROR` – Čtení chyby desky (jejího systémového kanálu).

Tyto signály poskytuje rozhraní API dodané firmou Hilscher s komunikační kartou. Jejich význam není úplně zřejmý a v některých případech závisí na komunikačním protokolu. Signály slouží hlavně pro ladění ovladače. Význam bude postupně doplňován.

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.

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`), viz obr. **??**. V tomto případě je v názvu bloku odkaz na první požadovaný

objekt a v následujících signálech je posunutý offset, tak aby tvořili souvislou oblast (tj. pro typ u8 má každý další signál offset o 1 větší, pro typ i32 má každý další signál offset o 4 větší, atp.). Výhodou takového užití je zvýšení rychlosti a částečně i přehlednosti algoritmů. Přechod od simulační verze je však v takovém případě trochu pracnější. Podrobný popis vícenásobných vstupů a výstupů lze nalézt v příručce [1].

Kapitola 3

Konfigurace ovladače

Konfigurace ovladače má dvě fáze. Nejprve je potřeba nakonfigurovat vlastní komunikační desku. To se provádí pomocí speciálního konfiguračního programu dodaného firmou Hilscher.

Druhá fáze je vytvoření konfiguračního souboru ovladače `HlsDrv`. Obecný popis konfiguračního dialogového okna a postup při konfiguraci je uveden v následujících sekcích této kapitoly.

3.1 Konfigurace komunikační desky

Tento program funguje pouze v operačním systému Windows a vyžaduje mít konfigurovanou komunikační kartu namontovanou v tomto počítači. Je možné mít v konfiguračním počítači jinou kartu stejného typu, ale je potřeba od obou zjistit sériové číslo (viz dále).

Podrobný popis tohoto konfiguračního programu je v dokumentaci firmy Hilscher a není zde uveden pro velký rozsah. Důležité je, že výsledkem této konfigurace jsou soubory *.nxd a *.nxf. Pokud konfigurační program není na cílovém zařízení, je potřeba tyto soubory na cílové zařízení nakopírovat a to ve stejné adresářové struktuře (tj. konfigurace kanálu `<channelID>` je v adresáři `<netX basedir>/<board type>_<board serial number>/Channel<channelID>/`). Je také nutné nakopírovat bootloader (tj. soubor `NETX100-BSL.bin` a/nebo `NETX50-BSL.bin` podle typu desky - nejlépe je nakopírovat oba a pak ovladač funguje s oběma typy) do adresáře `<netX basedir>/`. Soubor bootloaderu se při změně konfigurace nemění, takže jej stačí nakopírovat jen jednou. Soubory *.nxd se musí kopírovat při každé změně konfigurace. Soubory *.nxf jen při změně typu protokolu.

3.2 Konfigurační dialogové okno

Zatím není implementováno. Pouze vygeneruje výchozí konfigurační soubor. Dále je nutné editovat přímo *.rio soubor v textovém editoru (viz 4).

Kapitola 4

Formát konfiguračního souboru

Soubor *.rio je textový, takže jej lze v případě potřeby prohlížet i upravovat v libovolném textovém editoru pracujícím s prostým textem (například Notepad). Struktura souboru je zřejmá z následujícího příkladu:

```
Hilscher {
  BaseDir   "/var/lib/rexcore/netx/"
  BoardID         "/dev/uio0"
  Mode            0x106
  Timeout         0
}
```

Platí, že parametry, jejichž název začíná znakem # jsou ignorovány a lze je tedy využít jako komentář. V názvech parametrů i sekcí se rozlišují velká a malá písmena.

Význam jednotlivých parametrů je následující:

`BaseDir` – Adresář s konfiguračními soubory. Pokud se parametr neuvede, je užito /var/lib/rexcore/netx/ Ve Windows se parametr nepoužívá.

`BoardID` – Název zařízení, na které je namapována komunikační deska. Ve Windows se nepoužívá. V Linuxu to bude velmi pravděpodobně `\dev\uio0` (popř. jiné číslo na konci); pro XENOMAI `rtuio0`.

`Mode` – Upravuje některé vlastnosti ovladače. Každý bit představuje/zapíná určitou vlastnost, přičemž:

`bit 0`	sync mode	IOTASK je synchronizován s příchodem dat pro boardID=0, channelID=0, areaID=0
`bit 1`	swap bytes	prohazuje pořadí bajtů ve vícebajtových datech (word, long, ...)
`bit 2`	lockData	synchronizace semaforem (lze pro urychlení vypnout, pokud všechny vstupy a výstupy do tohoto ovladače vedou jen z jemu přidruženému IOTASKu)
`bit 8`	toolkit driver	slouží pro ladící účely (0=ovladač komunikační karty je v kernelu, 1=ovladač komunikační karty je toolkit přisestavený k `HlsDrv`)
`bit 9`	poll mode	slouží pro ladící účely (vypíná použití interruptu pro synchronizaci ovladače `HlsDrv` s komunikační deskou)

Kapitola 5

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.

- `boardID`, `channelID`, `areaID` se číslují od 0. Filozofie je taková, že `boardID` je vlastní netX komunikační karta (ačkoliv typicky je použita jen jedna, ovladač podporuje několik komunikačních karet v jednom počítači). Pořadí jednotlivých komunikačních karet určuje operační systém, lze se orientovat podle sériového čísla, které ovladač vypisuje při inicializaci. Každá komunikační karta může mít několik portů/konektorů (v nejobecnějším případě každý s jiným protokolem), přičemž každému odpovídá jedna komunikační síť a `channelID`. Na každém konektoru/komunikační síti/`channelID` je možné komunikovat s několika zařízeními. Každému zařízení odpovídá jedna oblast paměti/`areaID` (resp. jedna pro čtení a jedna pro zápis).

- netX karty s ethernetem mají často dva porty/ethernetové konektory, ale jen jeden channelID. Zařízení se pak chová jako dvouportový switch, resp. využívá se to k eliminování potřeby switchů při budování ethernetové sítě. Některé protokoly (např. EtherCAT slave) explicitně vyžadují dva porty/konektory (každý má trochu jiné použití) nebo se dají použít pro vytvoření redundantní sítě.

- První znak ve vlajce ovladač nikde nepoužívá, ale doporučuje se používat `I` pro vstupy a `O` pro výstupy. Znak je nutný, protože vlajka musí být v rámci tasku unikátní a nebylo by možné přistupovat na stejnou pozici ve vstupní a výstupní oblasti.

- Kromě přenosu „process image“ z jednotlivých stanic, se kterými se komunikuje umožňuje karta `netX` posílat a přijímat zprávy (funkce xChannelGetPacket() a xChannelPutPacket()). Formát těchto zpráv však závisí na použitém komunikačním protokolu (tj. je potřeba znát podrobně 7. vrstvu příslušného protokolu), navíc dokumentace dodané firmou Hilscher je k tomu málo, takže toto není podpořeno. Je pouze možné přijaté zprávy/packety vypisovat do logu v binární podobě.

- Pokud konfigurační programy firmy Hilscher nejsou instalovány na cílovém zařízení, je po změně konfigurace komunikační desky (tj. po nakopírování souborů *.nxd na cílovém zařízení) nutné cílové zařízení resetovat. Nová konfigurace se totiž aplikuje až po resetu komunikační desky.
- Pokud je potřeba nechat nepoužívané soubory *.nxd a *.nxf v adresáři odkud je ovladač načítá, je potřeba změnit jejich příponu, protože ovladač načte vždy všechny soubory s uvedenou příponou.

Kapitola 6

Co dělat při problémech

Nejčastější chyby jsou:

Ovladač `HlsDrv` je tak nový, že ještě nemá uživatele a tím ani nejčastější chyby.

V případě, že daný ovladač `HlsDrv` 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 `HlsDrv`.

Literatura

[1] REX Controls s.r.o.. *Funkční bloky systému REX – Referenční příručka*, 2016.

Referenční číslo dokumentace: 7855