

Návrh aplikací průmyslového řídicího systému REX

Pavel Balda, Miloš Schlegel, Milan Štětina

Příspěvek popisuje strukturu a možnosti nového průmyslového řídicího systému REX. Systém byl navržen s důrazem na možnost úplné simulace řídicích algoritmů v prostředí Matlab-Simulink. Po simulačním ověření lze řídicí algoritmy přeložit do binárních konfiguračních souborů, které pomocí diagnostického protokolu založeného na standardu TCP/IP je možno poslat přímo do cílových zařízení a podle nich zahájit řízení bez nutnosti odstavení zařízení. Ekvivalentní chování simulace a řízení v reálném čase zaručuje rozsáhlá knihovna funkčních bloků ve verzích jak pro Simulink, tak i pro každou cílovou platformu. Systém REX nevyužívá Real Time Workshop firmy The Mathworks.

Úvod

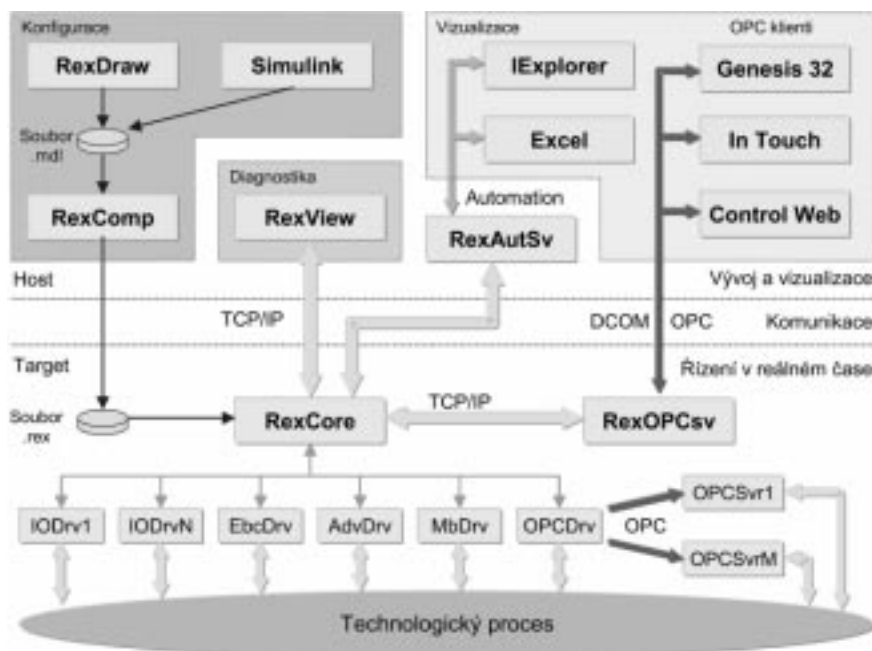
Již delší dobu lze pozorovat tlak ze strany zákazníků na neustálé zkracování doby určené pro instalaci a uvedení do provozu řídicích systémů technologických procesů. Nasazované řídicí systémy jsou však stále složitější a rozsáhlejší. Tato skutečnost nutí dodavatele, aby nasazovali řešení co možná nejlépe ověřená v předchozích aplikacích. Pokud jde o inovační řešení řídicího systému, dobu nutnou k uvedení do provozu lze podstatně zkrátit předchozí detailní simulací. Pravděpodobně nejrozsáhlejší nástroj pro vývoj a testování nových algoritmů v oblasti řízení procesů je programový systém Matlab-Simulink, který postupně proniká z oblasti akademického výzkumu na univerzitách i do inženýrské praxe.

Cílem tohoto příspěvku je stručně popsat vytváření aplikací nového řídicího systému REX s důrazem na možnost téměř úplné předchozí simulace v systému Simulink. Řídicí systém REX je otevřený a škálovatelný systém, vhodný pro vnořené (embedded) řízení, přenositelný na různé platformy s překladací jazyka C a C++, a to od jednoúčelových řídicích desek s jednoduchou exekutivou reálného času, až po procesní stanice se standardními operačními systémy (Windows CE, Windows NT/2000/XP, VxWorks apod.). Kompatibilita řídicího systému REX s programovým balíkem Simulink byla jednou ze základních myšlenek návrhu systému REX. Následující sekce popisují:

- strukturu systému REX,
- běh systému REX a diagnostiku,
- konfiguraci systému REX,
- využití standardu OPC (OLE for Process Control).

1. Struktura řídicího systému REX

Struktura řídicího systému REX a jeho vazby na okolní prostředí jsou schematicky znázorněny na obr. 1. V horní části obrázku jsou obsaženy komponenty vývojového



Obr.1 Struktura řídicího systému REX

prostředí (host), které je současně i prostředím pro vizualizaci a operátorské ovládání řídicího systému. Všechny současné vývojové nástroje systému REX jsou určeny pro operační systémy Windows 95/98/ME/NT/2000/XP. Pro vizualizaci se používají standardní vizualizační nástroje (některé jsou uvedeny přímo v obrázku). Vizualizační nástroje jsou napojeny na řídicí systém buď přes rozhraní OPC, využívající služby COM (Component object model) a DCOM (Distributed COM), nebo přes Automation (OLE automation) a skripty (VBScript, Jscript, Visual Basic).

Spodní část obrázku naznačuje strukturu cílového prostředí (target), které realizuje vlastní řízení v reálném čase. Cílovým prostředím mohou být (stejně jako u vývojového prostředí) operační systémy Windows 95/98/ME/NT/2000/XP, ale i např. Windows CE, VxWorks nebo Real Time Linux. V případě operačních systémů Windows může být cílové prostředí totožné s vývojovým, dokonce i na jednom počítači.

Spojovacím článkem mezi prostředím „host“ a „target“ je komunikační vrstva (na obr. 1 uprostřed). Nejčastěji používaným protokolem pro komunikaci je standard TCP/IP, nad nímž je vybudován vlastní diagnostický protokol systému REX.

2. Běh systému REX a diagnostika

Program RexCore, trvale provozovaný na cílové platformě, tvoří jádro řídicího systému REX. Aby bylo možno sledovat činnost řídicího systému na vývojové platformě (zejména v procesu uvádění do provozu), jsou třeba ještě další nástroje. Částečně lze provoz systému sledovat i v již zmíněných vizualizačních nástrojích, detailní pohled však poskytuje program RexView.

2.1 RexCore – jádro řídicího systému

Jádro řídicího systému RexCore je komplexní program provádějící paralelně různé činnosti obvyklé v řídicích systémech.

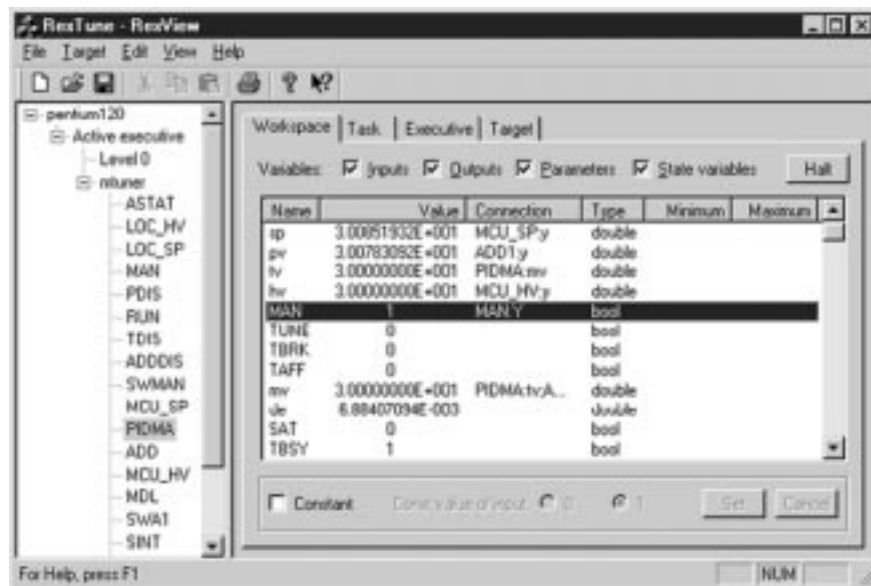
Jednotlivé činnosti jsou vykonávány na základě priorit v režimu preemptivního multitaskingu pomocí jednotlivých subsystémů jádra. Jádro obsahuje následující subsystémy:

- **Subsystém reálného času** – stará se o spuštění jednotlivých úloh a v nich vložených funkčních bloků, řídí spuštění ovladačů, získává a poskytuje diagnostické informace o časování úloh, ovladačů a o vytížení systému.
- **Vstupně-výstupní subsystém** – poskytuje rozhraní pro ovladače technických prostředků, pro získávání vstupů z procesu a pro nastavování výstupů.
- **Algoritmický subsystém** – obsahuje algoritmy funkčních bloků, které jsou volány z úloh subsystému reálného času.
- **Diagnostický subsystém** – poskytuje diagnostické informace o běhu řídicího systému, umožňuje download a ladění aplikací.
- **Archivační subsystém** – slouží pro archivaci událostí, alarmů a historických trendů veličin řídicího systému.

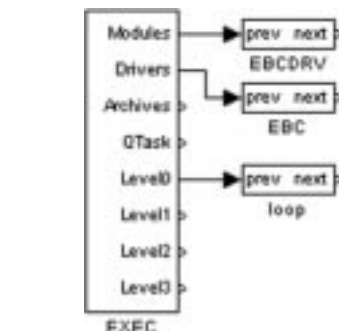
2.2 RexView – diagnostický nástroj

Program RexView umožňuje sledovat, co se děje v jádře řídicího systému REX při jeho běhu. Proto je velmi důležitým nástrojem při uvádění řídicího systému do provozu i v případě vzniku nějakých problémů již během rutinního provozu. Program poskytuje detailní, hierarchicky uspořádané informace o všech subsystémech jádra. Komunikace pomocí protokolu TCP/IP umožňuje připojit se k běžícímu jádru na lokálním počítači, v lokální síti i ve vzdálené síti (např. Přes internet). Na obr. 2 je patrná ukázka jednoho typu snímku programu RexView, podrobný popis celého programu je uveden v příručce [2].

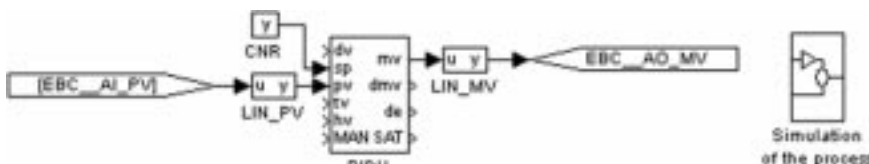
Levou část snímku tvoří hierarchie objektů v řídicím systému uspořádaná podle jednotlivých subsystémů. V této stromové struktuře je vybrán blok PIDMA, realizující



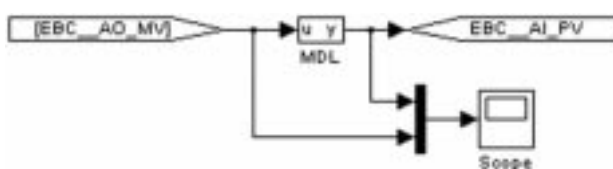
Obr.2 Ukázka obrazovky programu RexView



Obr.3 Hlavní soubor příkladu jednoduché regulační smyčky exec.mdl



Obr.4 Vlastní konfigurace příkladu jednoduché regulační smyčky loop.mdl



Obr.5 Simulační subsystém Simulation of the process z konfigurace loop.mdl

cí PID regulátor s vestavěným autotunem [6], který je v pořadí jedenáctým blokem úlohy mtuner subsystému reálného času. (Bloky dané úlohy ve stromové struktuře jsou zobrazeny v tom pořadí v jakém se v systému REX spouštějí.)

V pravé části snímku je pro blok PIDMA vybrána záložka Workspace s proměnnými pracovního prostoru bloku. V horní části záložka obsahuje zaškrtnuté políčko pro volbu druhů proměnných pracovního prostoru, které se zobrazují (vstupy, výstupy, parametry a stavy). Ve střední části je vidět seznam vybraných proměnných bloku s jejich periodicky aktualizovanými hodnotami a připojením vstupů a výstupů k ostatním blokům. Spodní část umožňuje

zadávat hodnoty parametrů bloků a simulovat hodnotu vybraných vstupů. V tomto příkladě je vybrán vstup MAN rovný 1 (tj. Regulátor je v manuálním režimu). Pokud bychom chtěli pro ladicí účely přepnout regulátor „natvrdo“ do automatického režimu, stačí zaškrtnout políčko Constant, v políčku Const value of input zvolit hodnotu 0 a stisknout tlačítko Set.

Ostatní záložky pravé části snímku odpovídají vlastnostem nadřazených objektů vybraného objektu (zde bloku PIDMA):

- **Task** – zobrazuje diagnostiku úlohy mtuner (měřené okamžité, průměrné, minimální a maximální doby trvání jednoho cyklu úlohy, indikace výpočetních chyb aj.).
- **Executive** – zobrazuje identifikační údaje a diagnostiku běžící exekutivy reálného času Active executive (datum vytvoření, nahrání a posledního spuštění konfigurace, obsazení paměti aj.).
- **Target** – identifikuje cílové zařízení (zde Pentium 120), se kterým program RexView právě komunikuje (typ cílového zařízení, název a verzi operačního systému, verzi a datum sestavení systému REX apod.).

Pro ostatní objekty jádra řídicího systému REX existují další záložky, které se zobrazí při vybrání objektu daného typu, např. Pro moduly, ovladače, trendy a archivy.

3. Konfigurace systému REX

Systém REX se konfiguruje formou vytváření funkčních schémat složených z funkčních bloků z rozsáhlé knihovny bloků [4]. Knihovna bloků existuje ve verzích pro systém Simulink i každou cílovou platformu (target) řídicího systému REX. To umožňuje kreslit funkční schémata přímo v grafickém editoru zabudovaném do systému Simulink. Druhou možností je použít vlastní grafický editor RexDraw.

Simulink i RexDraw ukládají funkční schémata do souborů s příponou .mdl (model). Tyto textové soubory jsou příliš rozsáhlé pro přenos na některá cílová zařízení, např. Pro paměťově omezená zařízení vnořených

(embedded) řízení. Proto jsou soubory. mdl překládány do binárního formátu. rex překladačem RexComp (sekce 3.2).

Konfigurace systému REX bude demonstrována na příkladě jednoduché regulační smyčky, která je znázorněna na obr. 3, 4 a 5. Vzájemné vazby všech tří obrázků a důvody právě takového zkonfigurování jsou vysvětleny ve zbytku této kapitoly.

3.1 RexDraw – grafický editor funkčních schémat

Program RexDraw umožňuje navrhovat funkční schémata řídicího systému REX velmi podobným způsobem jako se konfiguruje ve vestavěném editoru systému Simulink. Oba programy generují soubory s příponou. mdl, v možnostech konfigurace však existují určité rozdíly.

Především editor RexDraw umožňuje vytvářet soubory. mdl složené jen z bloků rozsáhlé knihovny systému REX [4]. Všechny bloky uvedené knihovny pracují v diskrétním čase, i když jejich velká část je diskretizována pro danou periodu vzorkování. Tato skutečnost odpovídá nastavení parametrů Solver options v Simulinku na Type: Fixed step a discrete (no continuous states).

Zatímco v systému Simulink je celá konfigurace tvořena jediným souborem, který může obsahovat více subsystémů (v našem případě soubor loop.mdl z obr. 4 a subsystém Simulation of the process z obr. 5), v systému REX musí být konfigurace tvořena alespoň dvěma soubory, z nichž právě jeden je tzv. Hlavním souborem projektu (v našem případě soubor exec.mdl z obr. 3).

Hlavní soubor projektu specifikuje konfiguraci jednotlivých subsystémů jádra řídicího systému RexCore, popsaných v 3.1. V uvedeném příkladě z obr. 3 je situace velmi jednoduchá, exekutivu reálného času RexCore vyjadřuje blok EXEC, ostatní bloky mají následující význam:

- EBCDRV – je jedním z modulů řídicího systému REX. Konkrétně je v tomto modulu implementován ovladač stanic s rozhraním EBC (Ethernet Base Controller), to ovšem systému REX říká až následující blok.

(Rozhraním EBC mohou být vybaveny např. Moduly řady Terminator I/O nebo automaty řady DL-205, jejichž distributorem je firma Automationdirect.com. Konkrétně moduly řady Terminator I/O jsou použity pro měření analogového vstupu regulované veličiny a nastavování analogového výstupu akční veličiny regulační smyčky.)

- EBC – je blokem vstupně výstupního ovladače, jehož jedním parametrem je jméno modulu, implementujícího daný ovladač, v tomto případě je to EBCDRV. Dalším parametrem je jméno konfiguračního souboru ovladače, který kromě

vlastních parametrů vstupně-výstupního zařízení, obsahuje hlavně informace odkud se mají zjišťovat hodnoty vstupů a kam se mají nastavovat hodnoty výstupů řídicího systému. Z hlediska projektování řídicího systému je nejdůležitější jméno bloku samotného (zde EBC), které určuje prefix názvů všech vstupů a výstupů, které budou k tomuto ovladači připojeny, blíže viz 3.2.

- Loop – je po přidání přípony. mdl názvem souboru, ve kterém je zkonfigurována řídicí úloha, která je zde zařazena do řídicí úrovně 0 (výstup Level0 bloku EXEC). Je to právě úloha z obr. 4.

Právě popsané tři bloky mají vstupy prev (předchozí) a next (následující), které umožňují do řady za sebou přidávat další bloky, reprezentující v konfiguraci další moduly, ovladače i řídicí úlohy. Stejně pravidlo platí i pro v tomto příkladu nepoužité archivy, které se kreslí připojením na výstup Archives bloku EXEC, a úlohy z dalších řídicích úrovní (Level1, Level2 a Level3). Jedinou výjimkou je (opět nepoužitý) blok Qtask pro velmi rychlou řídicí úlohu, který smí být v systému REX nejvýše jeden.

3.2 RexComp – překladač konfigurací

Na základě hlavního souboru projektu aplikace ve formátu. mdl program RexComp generuje binární konfigurační soubor. rex řídicího systému REX. Připomeňme, že příklad konfigurace hlavního souboru projektu exec.mdl je znázorněn na obr. 3. Překlad konfigurace se skládá zhruba řečeno z následujících kroků:

1. Nalezení bloku exekutivy reálného času EXEC v hlavním souboru projektu, kontrola parametrů exekutivy.
2. Nalezení a kontrola všech objektů řídicího systému, které jsou nakresleny v hlavním souboru projektu.
3. Přidání jednotlivých objektů do konfigurace: modulů, ovladačů, archivů, rychlé úlohy a úloh jednotlivých použitých výpočetních úrovní.
4. Alokace paměti a nastavení parametrů bloků z konfigurace.
5. Kontrola a propojení řídicích úloh.
6. Kontrola správnosti celé konfigurace.
7. Uložení přeloženého souboru s příponou. rex na disk.

Při svém spuštění vypisuje překladač informace o překládaných souborech a případně i výskyt chyb překladu. V každém z uvedených kroků může být detekována závažná chyba, která ukončí překlad konfigurace a zabrání vytvoření výsledného binárního souboru.

Pro efektivní přechod od simulace v systému Simulink k řízení v reálném čase jsou v překladači RexComp implementovány následující dva klíčové rysy:

1. Všechny subsystémy (v souborech. mdl), jejichž název začíná slovem Simulation, jsou vypuštěny (tj. nepřekládají se).
2. Všechny bloky From a Goto jejichž značka má tvar <prefix>__<name> jsou nahrazeny standardními vstupními a výstupními bloky systému REX, které odkazují na blok ovladače s názvem <prefix>, v němž hledají symbolický název vstupu nebo výstupu <name>. Dva znaky _ (podtržítka) za sebou slouží jako oddělovač specifikující, že jde o vstup nebo výstup ovladače.

Vrátíme-li se k našemu příkladu, podle pravidla 1 z obr. 4 je vypuštěn subsystém Simulation of the process. Podle pravidla 2 je blok From se značkou EBC__AI_PV nahrazen standardním vstupem systému REX, který se odkazuje na signál AI_PV ovladače, zkonfigurovaného v hlavním souboru projektu (obr. 3) blokem s názvem EBC.

Podobně blok Goto se značkou EBC__AO_MV odkazuje na signál AO_MV téhož ovladače.

Při dodržení uvedených pravidel lze právě popsaným postupem přejít od simulace řídicího systému v systému Simulink ke konfiguraci systému reálného času REX, a to bez jakékoliv změny konfiguračních souborů.

4. Využití standardu OPC (OLE for Process Control)

OPC je relativně nový, ale dnes už velmi rozšířený standard pro výměnu dat mezi prostředky pro řízení procesů, zejména mezi procesními stanicemi a nadřazenou vrstvou vizualizace. Současná verze systému REX podporuje specifikaci OPC Data Access [5] dvěma způsoby, popsanými v následujících podkapitolách.

4.1 RexOPCsv – OPC server systému REX

Pro komunikaci zejména s nadřazeným systémem byl vytvořen OPC Data Access server RexOPCsv, znázorněný ve spodní části (target) obr. 1. Nadřazeným systémem může být jakýkoliv systém, který je klientem OPC Data Access verze 2.0. Takových systémů jsou dnes na trhu desítky, několik z nich je přímo naznačeno na obr. 1.

Uvedená struktura je použitelná tehdy, podporuje-li cílové prostředí standard DCOM, tj. zejména pro operační systémy Windows. Pro platformy podporující pouze TCP/IP lze server spouštět až v prostředí host. Tuto strukturu lze výhodně použít i v rozsáhlých sítích (internet), kde vstupní brány do lokálních sítí jsou zabezpečeny proti neoprávněnému přístupu (firewall), a tím vzniká problém s průchodností paketů DCOM.

4.2 OPCDrv – ovladač pro komunikaci s OPC servery

Druhý způsob využití standardu OPC představuje ovladač OPCDrv. Na rozdíl od serveru RexOPCsv je tento ovladač klientem OPC Data Access 2.0, a tak umožňuje systému REX číst (nastavovat) vstupy (výstupy) libovolných zařízení, pro která existují OPC Data Access servery. Takových zařízení s příslušnými servery jsou dnes na trhu stovky od mnoha firem.

Závěr

Článek se snažil představit nový řídicí systém REX čtenářům, kteří jsou seznámeni s používáním rozsáhlého simulačního systému Matlab-Simulink. Při návrhu řídicího systému REX byl kladen důraz na možnost věrné simulace aplikace řídicího systému v prostředí Matlab-Simulink ještě před jejím nasazením. Z tohoto důvodu je konfigurace systému REX do značné míry kompatibilní s konfigurací systému Simulink (sekce 3). Za určitých podmínek, popsanych v sekci 3.2 lze dokonce přejít (či během procesu uvádění do provozu i opakovaně přecházet) od simulace v systému Simulink k řízení v reálném čase v systému REX bez jediné změny konfigurace. Více informací o řídicím systému REX a jeho dalším vývoji lze nalézt na www.rexcontrols.cz.

Literatura

- [1] The Mathworks 2001. Using Simulink, Version 4.1.
- [2] Řídicí systém REX – uživatelská příručka. REX Controls, s. r. o., Plzeň 2002.
- [3] Funkční bloky systému REX. REX Controls, s. r. o., Plzeň 2002.
- [4] SCHLEGEL, M., BALDA, P., ŠTĚTINA, M.: Knihovna C MEX bloků pro průmyslovou regulaci s aplikačními příklady. Konference Matlab 2001, Humusoft, s. r. o.
- [5] OPC Foundation 2000. OLE for Process Control, Data Access Custom Interface Standard. Version 2.04.
- [6] SCHLEGEL, M., BALDA, P., ŠTĚTINA, M.: PID autotuner pro průmyslové použití. Automatizace energetických procesů '02, Zlín, Česká republika.

Ing. Pavel Balda
doc. Ing. Miloš Schlegel, CSc.
Ing. Milan Štětina

40

Západočeská univerzita v Plzni
Univerzitní 8, 306 14 Plzeň, ČR
Tel.: ++420/377 63 25 19
e-mail: balda@rexcontrols.com