

Vložené řídicí systémy Řídicí systém REX

Pavel Balda
ZČU v Plzni, FAV, KKY

Osnova přednášky

- n Knihovna funkčních bloků RexLib
- n Řídicí systém REX
 - n Struktura ŘS REX
 - n Provozování ŘS REX, diagnostika, vizualizace
 - n Konfigurace ŘS REX
- n Sekvenční řízení v ŘS REX – blok ATMT
- n Univerzálně programovatelný blok REXLANG

2

Řídicí systém REX

- n Cíle návrhu
 - n Kompatibilita s globálně rozšířeným programovým systémem **Matlab-Simulink**
 - n Kvalitní knihovna funkčních bloků (blockset) splňující požadavky na využití v průmyslu (spolehlivost algoritmů, ošetření mezních stavů ...)
 - n Standardní komunikace s s nadřazeným systémem prostřednictvím **OPC** (OLE for Process Control)
 - n Přenositelnost na různé HW i SW platformy (různá cílová zařízení a operační systémy).
 - n Definované rozhraní pro ovladače vstupně-výstupních zařízení
 - n Snadná změna řídicích algoritmů změnou konfiguračního souboru

3

Knihovna funkčních bloků RexLib (1/2)

- n Algoritmy ŘS REX jsou soustředěny do knihovny funkčních bloků **RexLib**, obsahující více než 130 různých FB
- n Bloky jsou (pro větší přehlednost) rozděleny do podknihoven:
 - n **ExecLib** – konfigurace exekutivy reálného času systému REX
 - n **InOutLib** – bloky vstupů a výstupů systému REX
 - n **MathLib** – bloky pro matematické operace
 - n **AnaLib** – bloky pro zpracování analogových signálů
 - n **GenLib** – bloky pro generování signálů
 - n **RegLib** – bloky pro regulaci
 - n **LogicLib** – bloky pro logické řízení
 - n **ArLib** – bloky pro archivaci dat
 - n **SpecLib** – speciální bloky

4

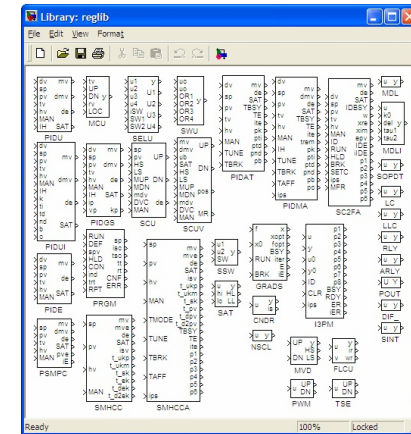
Knihovna funkčních bloků RexLib (2/2)

- n Knihovna RexLib je přenositelná na libovolnou platformu s překladačem jazyka C/C++
- n Existující verze knihovny:
 - n RexLib pro systém Matlab/Simulink – umožňuje simulaci řídicích algoritmů před nasazením do aplikace
 - n Pro všechny platformy systému REX – Windows, Windows CE, Phar Lap ETS,
 - n Platformy různých mikroprocesorů, mikrokontrolérů a DSP, kde je nazývána Micro RexLib (viz jednu z dalších přednášek)
 - n dsPIC firmy Microchip
 - n ARM7 firmy Atmel
 - n Další mohou přibývat dle potřeby

5

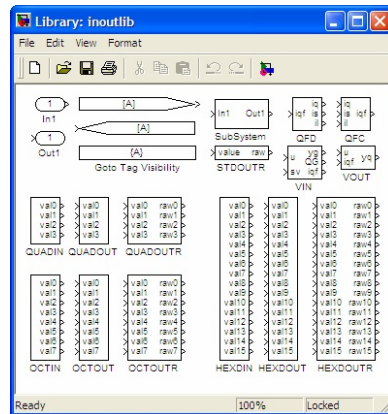
Knihovna bloků pro regulaci

- n Knihovna **RegLib**
 - n Regulátory, včetně pokročilých regulátorů s automatickým nastavováním parametrů
 - n Modely procesů,
 - n Zadávací jednotka
 - n Dynamické kompenzátory
 - n Bloky pro přepínání regulační struktury
 - n A mnoho dalších ...



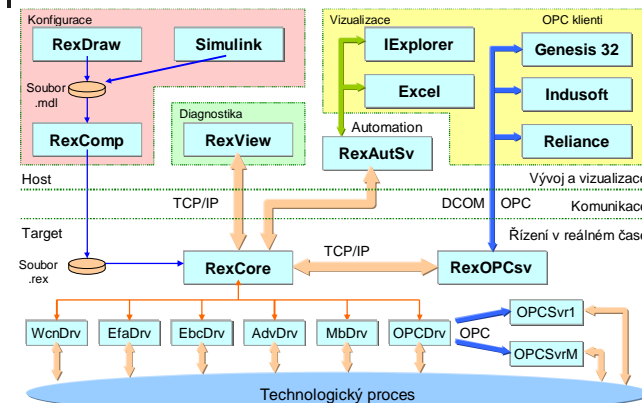
Knihovna vstupně-výstupních bloků

- n Knihovna **InOutLib**
 - n Bloky komunikující se vstupně-výstupními ovladači systému REX
 - n Propojovací bloky pro přenos signálů mezi úlohami a „neviditelnými“ spojovacími čarami
 - n Bloky pro zpracování kvality vstupních a výstupních signálů



7

Struktura ŘS REX



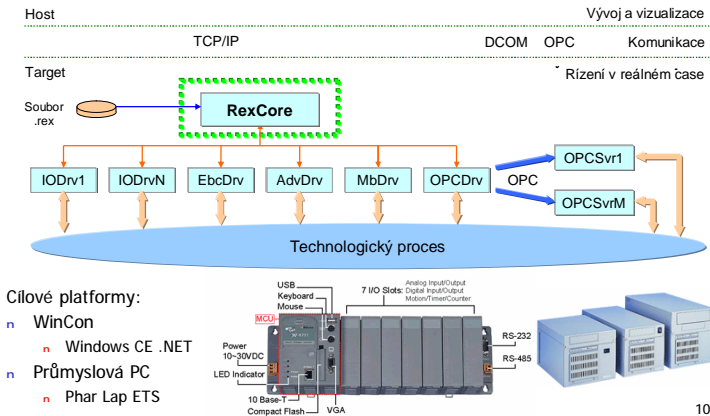
8

Provozování ŘS REX, diagnostika

- n Jádru systému REX – program **RexCore**
 - n Subsystémy jádra
 - n Subsystém reálného času
 - n Vstupně-výstupní subsystém
- n Diagnostický program **RexView**
 - n Spouštění aplikací ŘS
 - n Pracovní prostor bloků
 - n Běh úloh subsystému reálného času
 - n Trendy
- n Alternativní spouštění konfigurací – programy **RexRun** a **DDShell**

9

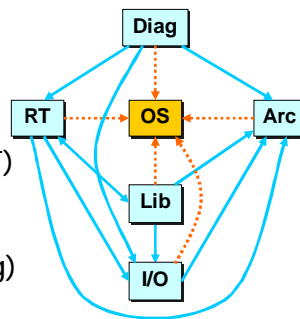
Jádru systému REX – program RexCore



10

Subsystémy jádra RexCore

- n Algoritmický subsystém – knihovna funkčních bloků RexLib (Lib) – viz výše
- n Subsystém reálného času (RT)
- n Vstupně-výstupní subsystém (I/O)
- n Diagnostický subsystém (Diag)
- n Archivační subsystém (Arc)



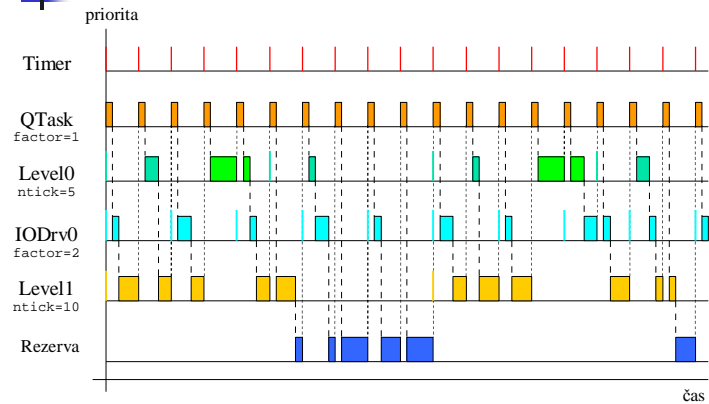
11

Subsystém reálného času

- n Určuje periodické spouštění jednotlivých úloh operačního systému reálného času
- n **Úlohy OS:**
 - n Výpočetní úlohy (Quick task, Tasks)
 - n Úlohy některých vstupně-výstupních ovladačů
- n **Deterministický plánovač (Scheduler)**
 - n Technika „Rate Monotonic Scheduling“
 - n 32 uživatelsky přiřaditelných logických priorit
 - n Nejvyšší prioritou 0, nejnižší 31
 - n Priority jsou přednastaveny, změna jen po důkladném uvážení ! Špatné nastavení může zhoršit chování systému

12

Plánovač (Scheduler) – Příklad



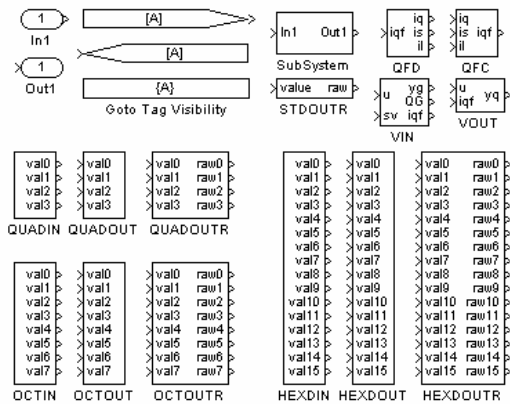
13

Vstupně-výstupní subsystém

- n Vazby mezi jádrem a vstupně/výstupními zařízeními z/do procesu
- n Interface pro ovladače vstupně/výstupních zařízení (IO drivers)
- n Interface mezi ovladači a vstupně výstupními funkčními bloky z knihovny **InOutLib**
 - n Vstupní bloky: **STDIN, QUADIN, OCTIN, HEXDIN**
 - n Výstupní bloky: **STDOUT, QUADOUT, OCTOUT, HEXDOUT**
 - n Výstupní bloky s indikací zapsané hodnoty: **STDOUTR, QUADOUTR, OCTOUTR, HEXDOUTR**
 - n Bloky pro zpracování kvality signálu: **QFD, QFC, VIN, VOUT**

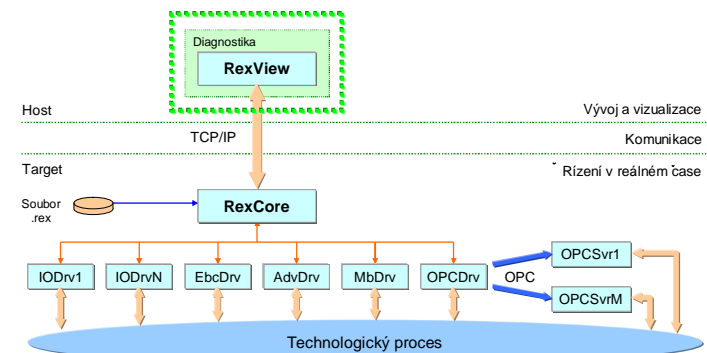
14

Knihovna InOutLib – I/O subsystém



15

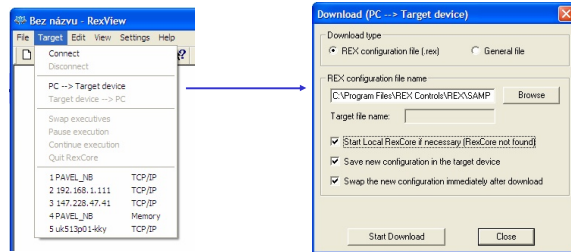
Diagnostický program RexView



16

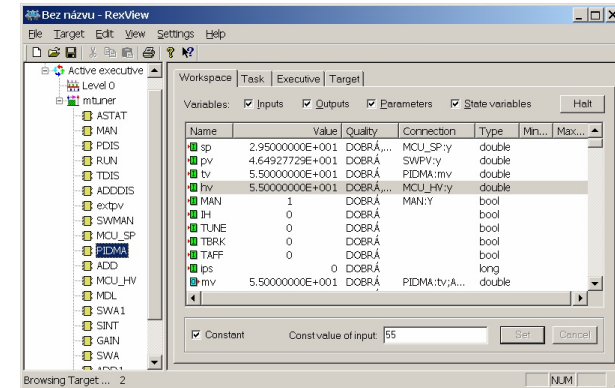
RexView – práce s cílovým zařízením

- Menu **Target** – umožňuje ovládat běh exekutivy RexCore
 - Navázání spojení s běžící exekutivou, odpojení se od běžící exekutivy
 - Posílání konfigurací a souborů do/z běžící exekutivy
 - Řízení běhu exekutivy: pozastavení, pokračování, ukončení, záměna posledních dvou konfigurací (Swap)
- Příklad: posílání konfigurace do programu RexCore



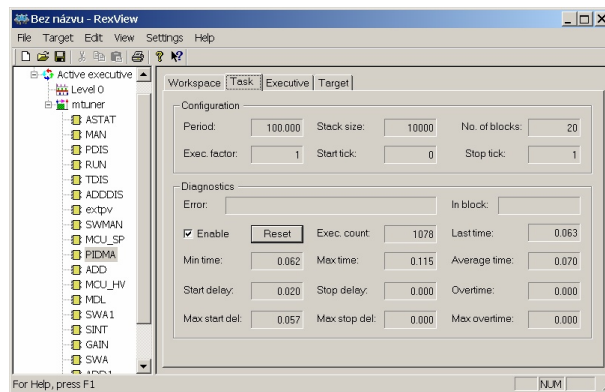
17

RexView – pracovní prostor bloku



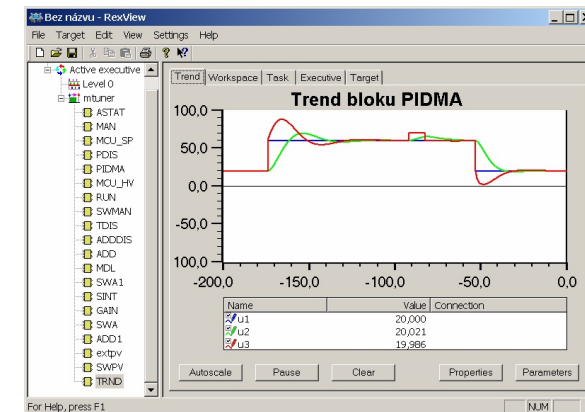
18

RexView – diagnostika tasku



19

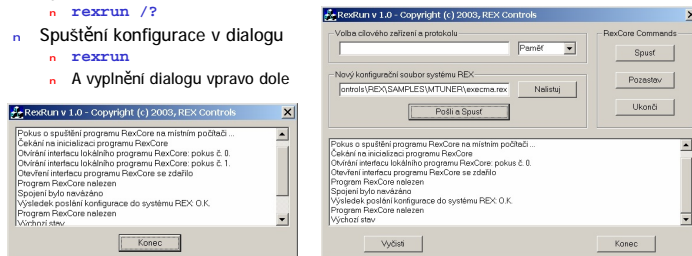
RexView – trendy



20

Spouštění aplikací programem RexRun

- n **RexRun** – program pro změnu konfigurace programu **RexCore** za běhu
- n Spuštění konfigurace **execma.rex** v řádkové verzi (výsledek vlevo dole)
 - n **rexrun execma**
- n Ukončení běhu konfigurace a odstavení RexCore v řádkové verzi
 - n **rexrun /q**
- n Výpis všech parametrů z příkazového řádku v řádkové verzi
 - n **rexrun /?**
- n Spuštění konfigurace v dialogu
 - n **rexrun**
 - n A vyplnění dialogu vpravo dole



21

Diagnostický program DDSShell

- n **DDShell** – řádkově orientovaný program pro nízkourovňové ladění konfigurací, snadno přenositelný na jiné platformy
- n Příklad: Trvalý zápis konfigurací do cílového zařízení

- n **dCdnlId(<n>,<f>)**
- n První parametr **<n>** je jméno konfigurace v uvozovkách
- n Druhý parametr **<f>** jsou příznaky

- n 1 ... ulož
- n 2 ... ihned spusť
- n 3 ... oboji

```

G:\Program Files\REX Controls\REX\DDShell.exe
dEC(sName) get executive configuration parameters
dED(sName) get executive diagnostic variables
dSAF(FC,TC,FL,TL,FID,UID) Set archive filter
dRAF Reset archive filter
dAC(sName) get archive configuration parameters
dARF(sName[.date],time) Read first archive items
dARN(sName) Read next archive items
dTRF(sName) Read first trend items
dTRN(sName) Read next trend items
dCdnlId(sFileName[,wFlags]) download configuration file to the target
dCUpId(sFileName) upload configuration file from the target
dFDnlId(sHostName,sTgtFName) download file to the target file
dFUpId(sTgtFName,sHostFName) upload file from the target file
dEPause stops the active executive
dERun starts the active executive
- Print more help (V/N) ? n
sz dler
REX Version 1.25 Build 461, Date 2005-09-12
Device: PC, OS: Windows 2000
PerMin[us]: 20000, PerMax[us]: 2147483647, TickRes[us]: 10000
Supported Archive flags: 0x01
> dCdnlId "exec.rex" 3
dCdnlId succeeded, 4538 bytes sent.
Reconnection to the target succeeded.
sz
    
```

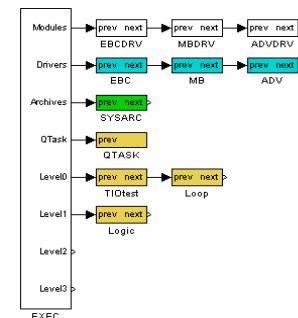
22

Konfigurace ŘS REX

- n Algoritmy ŘS REX se konfiguruji pomocí souborů **.mdl**
 - n Algoritmy lze navrhovat v programu **RexDraw** (součást instalace ŘS REX)
 - n Druhým možností je využít Simulink s nainstalovanou knihovnou funkčních bloků **RexLib**. Simulink navíc umožňuje detailní simulaci navrhovaného ŘS
- n Ovladače ŘS REX mají vlastní konfigurátory, které se volají z hlavních souborů projektu
 - n V programu RexDraw pomocí tlačítka "Special Edit"
 - n V Simulinku po změně zaškrtnutí položky "Toggle ..." a stisku Apply
- n Rozdíly oproti Simulinku
 - n Hlavní soubor projektu aplikace
 - n Překlad konfigurace programem RexComp

23

Hlavní soubor projektu - příklad



Moduly jádra RexCore

Vstupně-výstupní subsystém

Archivní subsystém

Subsystém reálného času

- n Bloky pro konfiguraci hlavního souboru projektu jsou v podknihovně ExecLib

24

Konfigurace exekutivy reálného času

Simulink

RexDraw

The image shows two windows from the RexCore software. On the left is the 'Block Parameters: EXEC' dialog, which has a tree view on the left with levels Level0 through Level3, and a list of parameters on the right including 'target' (PC-Windows), 'tick[s]', 'rtick0', 'rtick1', 'rtick2', 'rtick3', 'rtick4', 'rtick5', 'rtick6', 'rtick7', 'rtick8', 'rtick9', 'rtick10', 'rtick11', 'rtick12', 'rtick13', 'rtick14', 'rtick15', 'rtick16', 'rtick17', 'rtick18', 'rtick19', 'rtick20', 'rtick21', 'rtick22', 'rtick23', 'rtick24', 'rtick25', 'rtick26', 'rtick27', 'rtick28', 'rtick29', 'rtick30', 'rtick31', 'rtick32', 'rtick33', 'rtick34', 'rtick35', 'rtick36', 'rtick37', 'rtick38', 'rtick39', 'rtick40', 'rtick41', 'rtick42', 'rtick43', 'rtick44', 'rtick45', 'rtick46', 'rtick47', 'rtick48', 'rtick49', 'rtick50', 'rtick51', 'rtick52', 'rtick53', 'rtick54', 'rtick55', 'rtick56', 'rtick57', 'rtick58', 'rtick59', 'rtick60', 'rtick61', 'rtick62', 'rtick63', 'rtick64', 'rtick65', 'rtick66', 'rtick67', 'rtick68', 'rtick69', 'rtick70', 'rtick71', 'rtick72', 'rtick73', 'rtick74', 'rtick75', 'rtick76', 'rtick77', 'rtick78', 'rtick79', 'rtick80', 'rtick81', 'rtick82', 'rtick83', 'rtick84', 'rtick85', 'rtick86', 'rtick87', 'rtick88', 'rtick89', 'rtick90', 'rtick91', 'rtick92', 'rtick93', 'rtick94', 'rtick95', 'rtick96', 'rtick97', 'rtick98', 'rtick99', 'rtick100'. On the right is the 'Block properties' dialog for the EXEC block, showing 'Block name: EXEC', 'Block type: execib/EXEC', and a table of parameters.

No.	Param.	Value	Minimum	Maximum	Type
1	target	PC-Windows			String
2	tick	0.05			Double
3	rtick0	4			Long
4	rtick1	50			Long
5	rtick2	100			Long
6	rtick3	1000			Long
7	rtick4	5			Long

25

Konfigurace ovladačů

n Příklad zařazení ovladače EbcDrv do programu RexCore

n Pozor třídy rozlišují malá a velká písmena!

Simulink

RexDraw

The image shows two windows from the RexCore software. On the left is the 'Block Parameters: EBC' dialog, showing 'Block name: EBC', 'Block type: execib/EBC', and a table of parameters. On the right is the 'Block properties' dialog for the EBC block, showing 'Block name: EBC', 'Block type: execib/EBC', and a table of parameters.

No.	Param.	Value	Minimum	Maximum	Type
1	module	EbcDrv			String
2	classname	EbcDrv			String
3	classname	EbcDrv			String
4	factor	1			Long
5	stack	10240			Long
6	pr	3			Long

26

Konfigurační editor ovladačů

n Příklad editoru ovladače EbcDrv volaného z předchozího dialogu

n Pro Simulink změnou zaškrtnutí "Toggle the check box ..." a stiskem tlačítka Apply nebo OK

n Pro RexDraw stiskem tlačítka "Special Edit"

The image shows the 'REX EbcDrv - Configuration editor' dialog. It has a 'Network Specification' section with 'Communication protocol' set to 'UDP/IP'. Below is a table with columns: Name, DFmt, Station, Module, Chnl, Cnt, Range.

Name	DFmt	Station	Module	Chnl	Cnt	Range
AL_PV	DWI	258: TestEBC	1: T1F-08AD-1	0	1	+/-20mA
AQ_LV	DWV	258: TestEBC	2: T1F-16DA-2	0	1	+/-10V
<<end>>						

At the bottom, there are fields for 'Station' (258: TestEBC), 'Data Format' (DWI-DWord In), 'Channel' (0), and buttons for 'Insert', 'Delete', 'Copy', 'OK', and 'Cancel'.

27

Konfigurace archivů

n Simulink nemá analogii k archivům, přesto je lze konfigurovat

Simulink

RexDraw

The image shows two windows from the RexCore software. On the left is the 'Block Parameters: ARC' dialog, showing 'Block name: ARC', 'Block type: execib/ARC', and a table of parameters. On the right is the 'Block properties' dialog for the ARC block, showing 'Block name: ARC', 'Block type: execib/ARC', and a table of parameters.

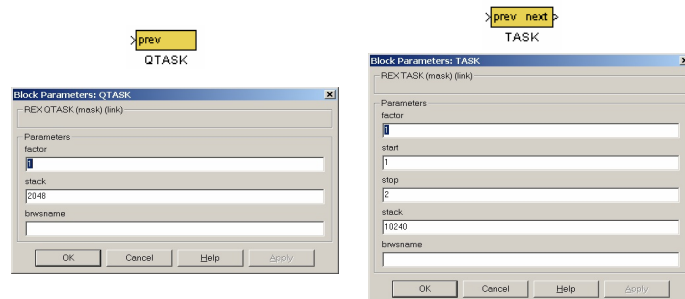
No.	Param.	Value	Minimum	Maximum	Type
1	atype	1			Long
2	esize	102400			Long
3	rmarks	720			Long

Below the dialog windows is a diagram titled 'Knihovna ArcLib' showing a network of blocks. The blocks are arranged in a grid-like structure with arrows indicating connections between them. The blocks include: U lac, ALB, HA, men LA, ALNI, ALN, HHA, HA, LA, LLA, ALN, ALN, y1, y2, y3, y4, delta E, ACD, RUN, R1, TRND, IE.

28

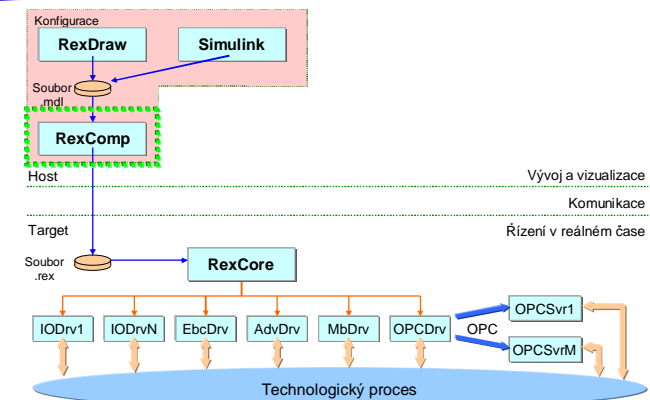
Konfigurace úloh exekutivy

- Úlohy QTASK a TASK jsou spouštěny danou periodou vzorkování. Název bloku určuje jméno souboru .mdl s řídicím algoritmem
 - QTASK – velmi rychlá úloha s periodou $factor * tick$ exekutivy
 - TASK – úloha s periodou $factor * ntick <i> * tick, <i>$ je číslo úrovně (level)



29

Překladač konfigurací RexComp



30

Fáze překladač

- Nalezení bloku **EXEC** v hlavním souboru projektu
- Nalezení a kontrola ostatních objektů
- Přidání modulů, ovladačů, archivů a úloh do konfigurace
- Alokace paměti a nastavení parametrů bloků
- Kontrola a propojení řídicích úloh
- Kontrola správnosti celé konfigurace
- Uložení přeloženého souboru s příponou **.rex** na disk

31

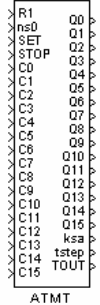
FB v IEC 61131-3 a v REXu

- FB v REXu inspirovány normou IEC 61131-3
 - Důraz kladen na ošetření mezních stavů, např. ošetření unášení integrační složky u regulátorů
 - FB, které mohou selhat mají **chybový výstup**, který je v případě selhání nastaven a lze jej testovat následnými bloky ve schématu
 - Bloky pracují pouze v diskretním čase, spojitě bloky jsou diskretizovány
- Na rozdíl od IEC 61131-3 jsou v systému REX implementována následující rozšíření:
 - Knihovna funkčních bloků **RexLib** ŘS REX obsahuje celou řadu velmi pokročilých bloků
 - Výstupy schématu se nenastavují až na konci programu, ale v závislosti na použitém vstupně-výstupním ovladači mohou být poslány do technologie okamžitě po vygenerování, čímž se minimalizuje parazitní zpoždění způsobené exekucí v PLC

32

SFC a blok ATMT

(1/3)



- n Blok **ATMT** realizuje logický automat mající až 16 stavů $S0 \dots S15$ indikovaných binárními výstupy $Q0 \dots Q15$ a 16 podmínek přechodu $C0 \dots C15$
- n Další vstupy:
 - n **R1** – Reset převádí automat do stavu **0**, má přednost před vstupem **SET**
 - n **ns0** – přednastavený stav, do kterého je automat převeden náběžnou hranou vstupu **SET**
 - n **STOP** – hodnota **1** blokuje činnost automatu v daném stavu, výstup **tstep** nenarůstá
- n Další výstupy:
 - n **ksa** – celočíselná reprezentace stavu
 - n **tstep** – čítač času v daném stavu. Při přechodu do jiného stavu se nuluje
 - n **TOUT** – příznak překročení časového limitu pro aktuální stav

33

SFC a blok ATMT

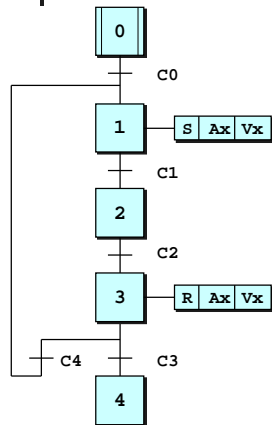
(2/3)

- n Chování bloku **ATMT** je určeno tabulkou přechodů ve tvaru:
 - n $S1 \ C1 \ NS1$
 - n $S2 \ C2 \ NS2$
 - n ...
 - n $Sn \ Cn \ NSn$
- n kde každý řádek tabulky vyjadřuje jednu podmínku přechodu
- n Např. prvý řádek $S1 \ C1 \ NS1$ znamená: Je-li aktuální stav $S1$ a současně je splněna podmínka $C1$, přejdi do nového stavu $NS1$
- n Dalším parametrem je pole časů (timeoutů) v jednotlivých stavech
 - n $TO1 \ TO2 \dots \ TOm$
 - n Setrvává-li automat ve stavu Si delší dobu než TOi , je nastaven výstup **TOUT** na **1** (při přechodu do nového stavu je **TOUT** nastaven na **0**)

34

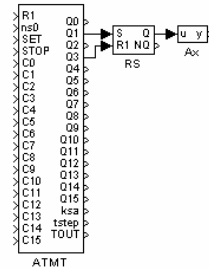
SFC a blok ATMT

(3/3)



- n Příklad: SFC na obr. Vlevo odpovídá následující tabulce přechodů bloku ATMT na obr. vpravo:

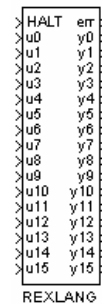
	S	C	NS
0	0	0	1
1	1	1	2
2	2	2	3
3	3	3	4
4	3	4	1



35

ST a blok REXLANG

(1/2)



- n Blok **REXLANG** je volně programovatelný blok ŘS REX v jazyku téměř kompatibilním s jazykem C.
 - n Uživatel vytvoří zdrojový soubor s algoritmem vlastního bloku a jméno souboru uvede jako parametr bloku
 - n Systém REX pak soubor přeloží a pokud překlad skončí úspěšně, blok periodicky spouští v úloze, kam je zařazen
- n Oproti jazyku C má zejména následující omezení
 - n Navržený algoritmus pracuje pouze s jednoduchými typy **long** a **double** (**int**, **short** a **bool** se zpracovávají jako **long**, typ **float** se zpracovává jako **double**)
 - n Nejsou implementovány ukazatele pomocí operátoru *****, jsou však implementována jednorozměrná indexovatelná pole
 - n Nejsou implementovány standardní knihovny, je však definována většina funkcí z **math.h**
 - n Algoritmus se skládá z funkcí **init(void)**, **main(void)**, **exit(void)** a volitelně též **parchange(void)**

36

ST a blok REXLANG

(2/2)

```
n Příklad jednoduchého číselového filtru
double input(0) vstup;
//promenna 'vstup' odpovida u0
double output(0) vystup;
//promenna 'vystup' odpovida y0
double stav[20], param[20];
const long count=20;
int init(void)
{
    long i; const double a=0.95;
    param[0]=0.2;param[5]=0.2;
    param[10]=0.2;param[12]=0.2;
    param[15]=0.2;
    for(i=0;i<count;i++)
    {
        param[i]=param[i]+exp(-i*a)/a;
        Trace(1,param[i]);
    }
    return 0;
}

int main(void)
{
    long i;
    double soucet=0.0;
    for(i=0;i<count-1;i++)
        stav[i]=stav[i+1];
    if(fabs(vstup)>1)
        stav[count-1]=(vstup>0)? 1 : -1;
    else
        stav[count-1]=vstup;
    for(i=0;i<count;i++)
    {
        soucet+=stav[i]*param[count-1-i];
        Suspend(0.1);
    }
    vystup=soucet;
    return 0;
}

int exit(void){return 0;}
```

37

Jak získat ŘS REX

- n ŘS REX a knihovna FB RexLib pro Matlab-Simulink včetně dokumentace jsou dostupné na

www.rexcontrols.cz

- n ŘS REX pracuje v demonstrační verzi dvě hodiny, pak se sám ukončí
- n Z knihovny RexLib pro Matlab-Simulink (pro Windows) vyžadují licenční klíč pouze některé pokročilé bloky
- n Pro oba systémy je zdarma dostupná studentská licence pracující po dobu 1 roku

38