

20. Vizualizace a simulace (PROMOTIC, FLUIDSIM, ML)

20. Vizualizace a simulace (PROMOTIC, FLUIDSIM, ML)

- základní pojmy a zásady vizualizace a simulace řízení pracovního procesu
- vizualizační a simulační software (Promotic) – základní vlastnosti, základní části – editory projektu a jeho úrovně
- základní objekty a proměnné, komunikační ovladače
- obrazy a ovládání – vlastnosti, dynamické vazby, programování ve VBscriptu,
- postup při tvorbě aplikace
- vizualizační a simulační software elektrických, elektropneumatických a pneumatických systémů (Fluidsim) – základní úrovně a části editoru
- základní objekty a ovladače, vazby
- postup při tvorbě aplikace

FestoFluidsim

1) Postup při tvorbě aplikace

V simulační programu Festo tvoříme odspodu, jako první je vždy zdroj (stlačený vzduch), pro školní potřeby není třeba připojovat jednotku pro úpravu vzduchu (v programu), na stolech je, ale v praxi nutné. Odstranění nečistot ze vzduchu, odstranění mastnoty atd. Potom následují ovládací prvky, to jsou například tlačítka, koncové spínače (koncový spínač svojí vnitřní stavbou je totožný jako ovládací tlačítko, jen je rozdíl v tom, že je ovládán pneomotorem). Po ovládacích prvcích, následují prvky logické logický součin (AND) a logický součet (OR) logika zapojení logický členů záleží na příslušném zadání. Po té máme prvky řídicí, což jsou rozvaděče 5/2 nebo 3/2, jsou to vlastně bistabilní klopné obvody a nebo v případě rozvaděče 3/2 monostabilní.

-bistabilní klopný obvod, má 2 stabilní stavy, v našem případě rozvaděč ovládáme řídicími vstupy 14 a 12, kdy překlápíme rozvaděč a ten vzduch rozvádí buď na vyjetí nebo na zajetí pneumotoru.

-monostabilní klopný obvod, má 1 stabilní stav, v našem případě tlačítko, vyšlu impuls, ale vrátím se zpět do původního (stabilního) stavu

nakonec dáváme pneumotory (jednočinné, dvojčinné) a škrtící ventily, škrtící ventily dáváme tak, abychom škrtily výstup, nechceme ubírat sílu (tlak) vháněný do válce, ale pouze regulovat jeho rychlost.

2) Základní objekty

Jak již bylo zmíněno – základní objekty(součástky) pro tvorbu aplikace jsou následující, pneumotory, rozvaděče, řídicí a logické členy, ovládací členy, zdroj stlačeného vzduchu, jednotka pro úpravu vzduchu, je taky důležité, aby každá dokumentace obsahovala seznam součástek a také krokový diagram. Podle krokového diagramu zkontroluji, zda je daná úloha funkční či nikoli. Vidíme průběh/y všech pneumotorů => odhalení možného hazardu. Plno dalších součástek pro tvorbu elektropneumatiky. Časovač – máme 2 druhy:

a) V klidové poloze uzavřen – pokud na tento časovač přivádím stálý tlak (jinak by nejel) tak mi začne odpočítávat od Xsekund až po 0. Po skončení odpočítávání se čítač překlápí a vyšle impuls.

b) V klidové poloze otevřen – na tento časovač přivádím opět stálý tlak. Po uplynutí námi nastavené doby, se čítač uzavře.

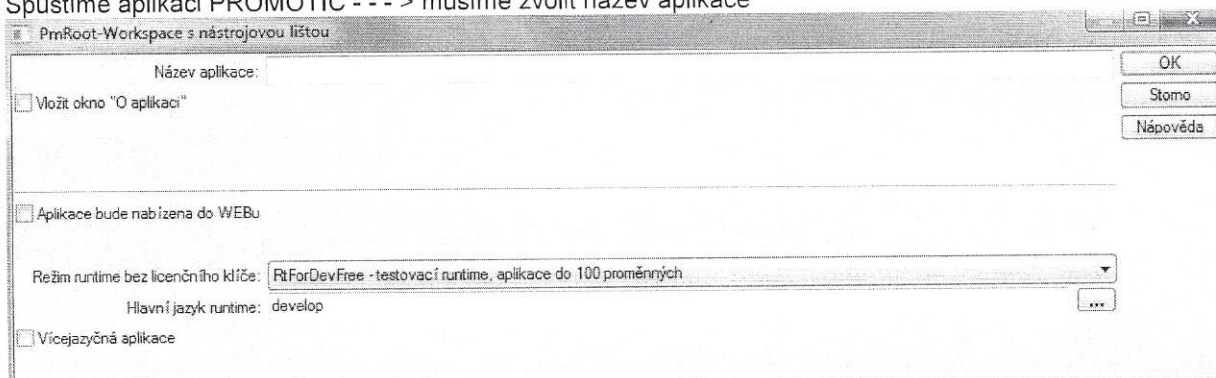
Tvorba logických členů pomocí tlačítek:

Logický součin – AND – 2 nebo 3 nebo N tlačítek v sérii, výstup z jednoho tlačítkem jde na přívod vzduchu toho dalšího

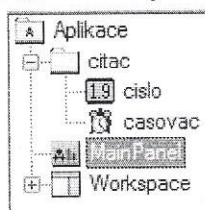
Logický součet – OR – nelze realizovat – vzduch jde také cestou menšího odporu, kdybychom zapojili 2 tlačítka paralelně, vzduch nepůjde do řídicího členu, ale vrátí se na na výstup druhého tlačítka.

Promotic

Spustíme aplikaci PROMOTIC - - - > musíme zvolit název aplikace



Tvorba statických objektů:



PmFolder – složka (v našem případě s názvem „citac“), do které ukládám všechny statické objekty

PmNumber – objekt, do kterého ukládám hodnoty

PmPanel – uživatelská grafika, kde provádím vizualizaci, například křížovatky, změna barev semaforů je závislá na čase apod.

Casovac – místo pro tvorbu programu

```
1 Dim ocislo
2 Set ocislo=PMe.Pm("/citac/cislo")
3
4
5 if ocislo.value < 5 then
6   ocislo.value = ocislo.value + 1
7 else
8   ocislo.value = ocislo.value + 1
9 end if
10
```

jednoduchý program pro čítání do nekonečna, propojení grafických objektů se statickými.. barva potrubí se nám bude měnit na základě hodnoty ve statickém objektu cislo, do proměnné ocislo, hodnotu jen ukládám

DIM – definování proměnných

SET – nastavení odkazu na objekt

Program pro počítání do 5 a opětovné opakování cyklu – v mojem případě změna barvy

```
1 Dim ocislo
2 Set ocislo=PMe.Pm("/citac/cislo")
3
4
5 if ocislo.value < 5 then
6   ocislo.value = ocislo.value + 1
7 else
8   ocislo.value = 0
9 end if
```


V promotocu máme možnost, změnit periodu se kterou bude program pracovat, ale jsme omezení rychlostí programu minimální periodu, kterou můžeme nastavit, je 0.5s, při menší periodě program neodpovídá.

Nejprve si zvolíme grafický objekt, který chceme měnit, na výběr máme mnoho různých možností, od geometrický tvarů, až po výměníky, čerpadlo, zásobníky apodobně. Budeme měnit barvu závislou na čase, každou vteřinu bude jiná barva, proto nejprve klikneme na prvek pravým tlačítkem myši a vybereme možnost „editovat aktuální prvek“

MatLab

MATLAB je interaktivní programové prostředí

MATLAB umožňuje počítání s maticemi, vykreslování 2D i 3D grafů funkcí, implementaci algoritmů, počítačovou simulaci, analýzu a prezentaci dat i vytváření aplikací

V MATLABu nemají po deklaraci určený datový typ a mění datový typ během své existence

M-File

```
function [ x ] = linearniRovnice(a,b,c)
% linearniRovnice - funkce pro výpočet lineární rovnice
% parametry a,b,c zastupují hodnoty z rovnice ax+b=c

if a==0
    if b==c
        x = 'Rovnice má nekonečně mnoho řešení';
    else
        x = 'Rovnice nemá řešení';
    end;
else
    x = (c-b)/a;
end;
end
```

Soubory s koncovkou *.m, takzvané m-files, obsahují definice funkcí

Sami si můžete m-files vytvářet a tak přidávat do MATLABu nové funkce a rozšiřovat tak jeho využití. Název funkce v m-filu musí být stejný jako název souboru. U funkce si nastavíte vstupní parametry a návratové hodnoty, následný komentář, slouží jako nápověda, popisující k čemu daná funkce slouží.

MATLAB podporuje funkce s různým počtem vstupních i výstupních parametrů. Veškerý text na řádku za znakem '%' je brán jako komentář. Středníky v MATLABu neslouží k ukončení řádku nebo příkazu, ale k zabránění výpisu na obrazovku. Kdybychom středníky nepoužili, na obrazovku by se vypsaly všechny kroky a jejich mezivýsledky, které funkce provádí.

Simulink

využívá Matlab a jeho funkce k simulaci dynamických systémů.

ovládání Simulinku je jednodušší než MatLabu, ale pokročilejší funkce nelze provádět bez znalosti jazyka Matlab

Po spuštění je vytvořeno okno pro tvorbu nového modelu a okno obsahující základní nabídku otevírání knihoven zdrojů signálů, základních spojitých, diskrétních a nelineárních bloků a bloků pro zobrazování a ukládání signálů.