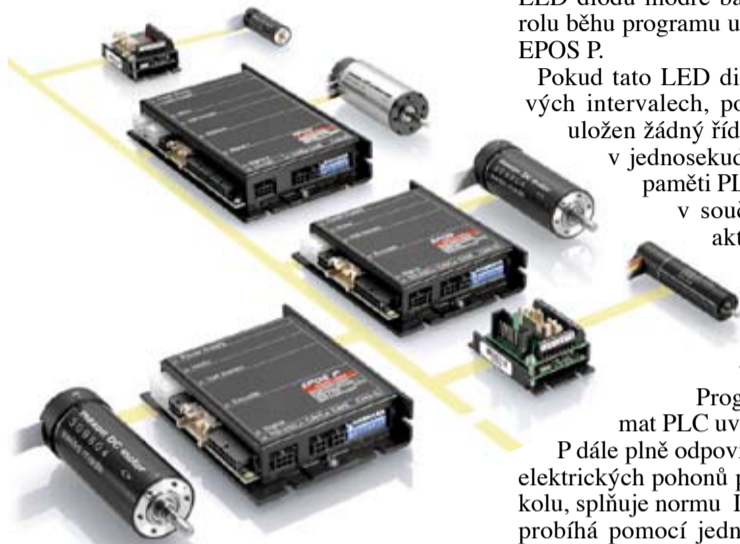


Uživatelský software

maxon řízení EPOS P



Text v tomto čísle je volným pokračováním seriálu o řídicích jednotkách stejnosměrných motorů EPOS, který vycházel v Technickém týdeníku od začátku roku 2006. Pokračování se zabývá popisem programovatelné řídicí jednotky EPOS P



EPOS P je řídicí jednotkou stejnosměrných motorů malého výkonu. Jde o výrobek švýcarské firmy maxon motor. Na rozdíl od předchozího modelu řídicí jednotky EPOS umožňuje uložit program do vnitřní paměti řídicí jednotky. Tím odpadá potřeba trvale připojeného nadřazeného počítače či PLC.

Lze tedy prohlásit, že řídicí jednotka EPOS P je kombinací řídicí jednotky EPOS a nadřazeného řídicího počítače PLC.

Řídicí jednotka EPOS P se vyrábí pouze ve výkonové verzi 24/5, což znamená, že na výstupu řídicí jednotky je možno připojit motor s maximálním odběrem 5 A při výstupním napětí 24 V.

Po hardwarové stránce je zapojení plně totožné se zapojením řídicí jednotky EPOS 24/5. Z tohoto důvodu zájemce o konkrétní informace o zapojení odkazují na první díl předešlého seriálu, kde jsme podrobně popsali jednotlivé přípojovací konektory. Kromě těchto konektorů obsahuje řídicí jednotka EPOS P navíc signální LED diodu modré barvy. Je určena pro kontrolu běhu programu uvnitř PLC řídicí jednotky EPOS P.

Pokud tato LED dioda bliká v půlsekundových intervalech, potom uvnitř paměti není uložen žádný řídicí program. Pokud bliká v jednosekundových intervalech, je do paměti PLC uložen program, avšak v současné době neběží, není aktivní. V případě, že signální LED modré barvy trvale svítí, je v paměti PLC program a v současné době je vykonáván.

Programovatelný řídicí automat PLC uvnitř řídicí jednotky EPOS

P dále plně odpovídá standardům pro řízení elektrických pohonů pomocí CANOpen protokolu, splňuje normu IEC 61131. Programování probíhá pomocí jednoho z pěti programovacích jazyků PLC. Patří mezi ně IL – Instruction list, ST – Structured Text, LD – Ladder Diagram, FBD – Function Block Diagram a SFC – Sequential function Chart.

PLC řídicí jednotky EPOS P obsahuje 32bitový ARM procesor taktovaný na 60 MHz, 1 MB paměti pro uložení programu a 1 kB energeticky nezávislé paměti určené pro uchování dat i při výpadku napájení. Výkonově dokáže PLC řídicí jednotky EPOS P zpracovat za dobu 2,5 ms 5000 řádků programu v jazyce IL.

Řídicí jednotka EPOS P je tedy schopna pracovat jako takzvaný CAN Master v síti CAN. Může řídit podřízené jednotky EPOS, popřípa-

dě i jiné prvky připojené na sběrnici CAN. Tak lze vytvářet vícemotorové pohony, a to včetně synchronizace – například polohy.

PLC řídicí jednotky EPOS P je výkonnostně dostačující pro synchronizaci polohy nejvíce čtyř pohonů s řídicími jednotkami EPOS.

Dále je vhodné zmínit další novinku ve výrobním sortimentu firmy maxon motor, a sice takzvaný pohon MCD

– Maxon Compact Drive. Jde o inteligentní pohon, který v jednom pouzdře slučuje stejnosměrný motor s elektronickou komutací o výkonu 60 W, řídicí jednotku EPOS či EPOS P a magnetorezistivní inkrementální snímač otáček. Zákazník tím pádem nemusí uvažovat o umístění řídicí jednotky v rozvaděči zařízení a o vzájemném propojení mezi motorem a řídicí jednotkou. MCD Pohon se vyrábí ve dvou verzích, jako Master version a jako Slave version. Master version obsahuje uvnitř pouzdra řídicí jednotku EPOS P, a tudíž ji lze použít jako CAN Master zařízení. Slave version obsahuje pouze řídicí jednotku EPOS. Z toho důvodu vyžaduje vždy přítomnost nadřazeného řídicího počítače.

Rozměry pohonů MCD jsou: délka 120 mm, výška 50 mm a šířka 36 mm. Tyto pohony lze doplnit rovněž planetovou převodovkou. Firma maxon motor dále dodává sadu speciálních kabelů, kterými lze tyto jednotky připojit.



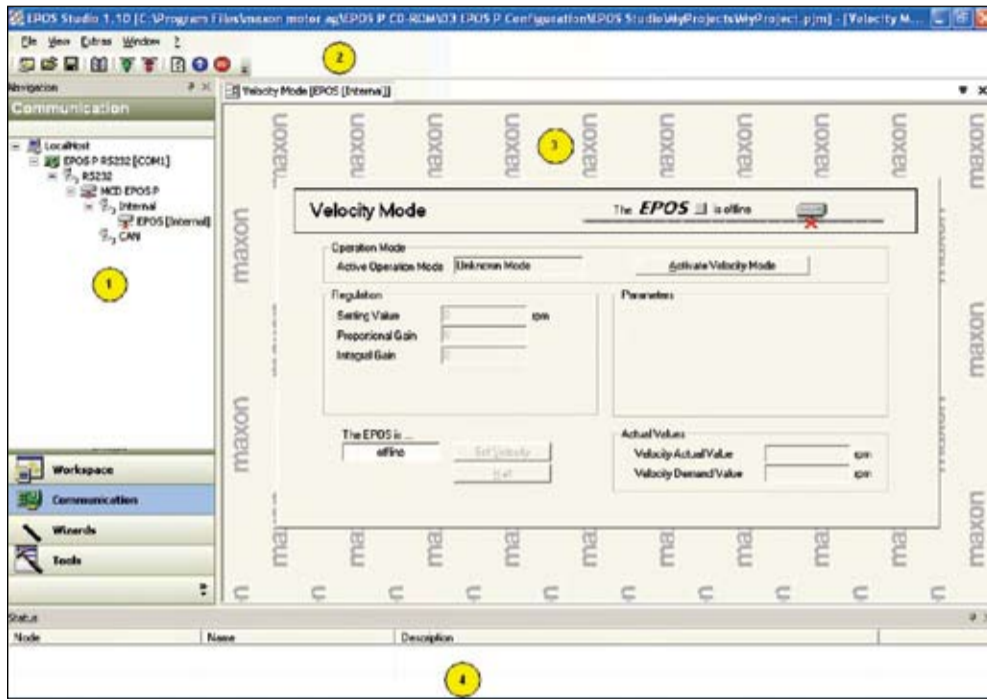
Programování řídicích jednotek EPOS P se děje prostřednictvím programu OpenPCS, který je součástí programového balíku EPOS STUDIO. Lze jej bezplatně stáhnout na internetové adrese www.maxonmotor.com v sekci services and downloads jednoduše zadáním klíčového slova EPOS P. Program EPOS STUDIO je nová konfigurační pomůcka určená k nastavení řídicích jednotek EPOS a EPOS P. Jejím popisu se bude věnovat další díl tohoto seriálu.

ING. JIŘÍ HNÍZDIL,
HNIZDIL@UZIMEX.CZ

maxon motor

driven by precision

Uživatelský software maxon řízení EPOS P



SOFTWARE EPOS STUDIO

V tomto dílu seriálu o řídicích jednotkách EPOS P se budeme zabývat popisem softwaru EPOS Studio. Jedná se o novou softwarovou pomůcku určenou pro konfiguraci řídicích jednotek EPOS, EPOS P a inteligentních pohonů MCD. Touto pomůckou nahrazuje firma maxon motor předchozí verzi softwaru zvanou EPOS USER INTERFACE, popsanou v druhém dílu seriálu o řídicích jednotkách EPOS (TT 2/2006).

Program EPOS Studio je volně ke stažení z webových stránek firmy maxon motor v sekci „Service & Download“ zadáním klíčového slova EPOS P. Velikost instalačního souboru je

105 MB. Tento program je rovněž standardně dodáván na instalačním médiu CD ROM s řídicími jednotkami EPOS P.

Obrázek zobrazuje hlavní okno programu EPOS Studio. Řídicí jednotka EPOS P je interně rozdělena do dvou modulů. Jeden z nich obsahuje programovatelný řídicí počítač PLC a druhý z nich obsahuje řídicí jednotku EPOS. Každý z těchto modulů má k dispozici různá nastavení. Zatím co u modulu s řídicí jednotkou EPOS lze provést optimalizaci regulátorů proudu, rychlosti či polohy, přepnout jednotku do určitého módu a podobně, modul s PLC umožňuje nastavit parametry komunikace s ostatními jednotkami v síti CAN, RS 232,

popřípadě je možné jej přepnout do režimu programování.

Strukturu a nastavení sítě s jednotkami EPOS, EPOS P a MCD umožňuje panel Navigation, značka č. 1. Obsahuje několik grafických tlačítek, která přepínají panel do zobrazení

- Grafické tlačítko „Workspace“ přepíná panel „Navigation“ do zobrazení použitých řídicích jednotek připojených do sítě.

- Grafické tlačítko „Communication“ přepíná panel „Navigation“ do zobrazení způsobu připojení řídicích jednotek do sítě – tj. zda je jednotka připojena prostřednictvím sítě CAN nebo RS232. Kliknutím pravého tlačítka na příslušnou jednotku je možno měnit parametry komunikace této řídicí jednotky.

- Grafické tlačítko „Wizards“ přepíná panel „Navigation“ do zobrazení průvodců nastavení řídicích jednotek. Aktivací příslušného průvodce lze spustit například automatické nastavení konstant regulátorů u řídicích jednotek EPOS nebo průvodce prvotním nastavením řídicí jednotky EPOS P.

- Grafické tlačítko „Tools“ přepíná panel „Navigation“ do zobrazení pomůcek využitelných příslušnou řídicí jednotkou. U řídicích jednotek EPOS je možnost v tomto zobrazení aktivovat příslušný mód řídicí jednotky, jako je např. rychlostní mód s profilem. U řídicích jednotek EPOS P lze v tomto zobrazení aktivovat režim programování.

Struktura sítě umožňuje uložit program EPOS Studio do souboru tzv. projektu. Uživatel provede konfiguraci sítě pouze jednou, uloží ji do souboru a při další práci si tuto konfiguraci z uloženého souboru načte.

Hlavní nabídka a panel ikon, značka 2, hlavního okna programu EPOS Studio obsahuje standardní příkazy pro práci s programem, jako je například otevření a uložení projektu, ukončení práce s programem, přidání další řídicí jednotky do projektu, nastavení zobrazení jednotlivých panelů hlavního okna. Panel ikon je pouze grafickým vyjádřením některých příkazů hlavní nabídky a slouží k ulehčení práce s programem.

Panel nastavení, značka 3, slouží k vlastnímu nastavení příslušných módů řídicích jednotek. Obrázek ukazuje strukturu tohoto panelu u řídicí jednotky EPOS přepnuté v režimu řízení rychlosti.

Okno „Status“, značka 4, slouží k zobrazení stavu řídicích jednotek, popřípadě k zobrazení chybových hlášení jednotlivých řídicích jednotek. Kliknutím pravého tlačítka myši nad tímto oknem docílíme zobrazení nabídky, určené k mazání chybových hlášení.

Program EPOS Studio je v jednotlivých nabídkách velice podobný předchozímu programu EPOS USER INTERFACE s tím, že tyto nabídky jsou logičtěji uspořádány prostřednictvím panelu „Navigation“. S ovládáním by tudíž neměl mít uživatel, který má zkušenosti s předchozím, žádné problémy.

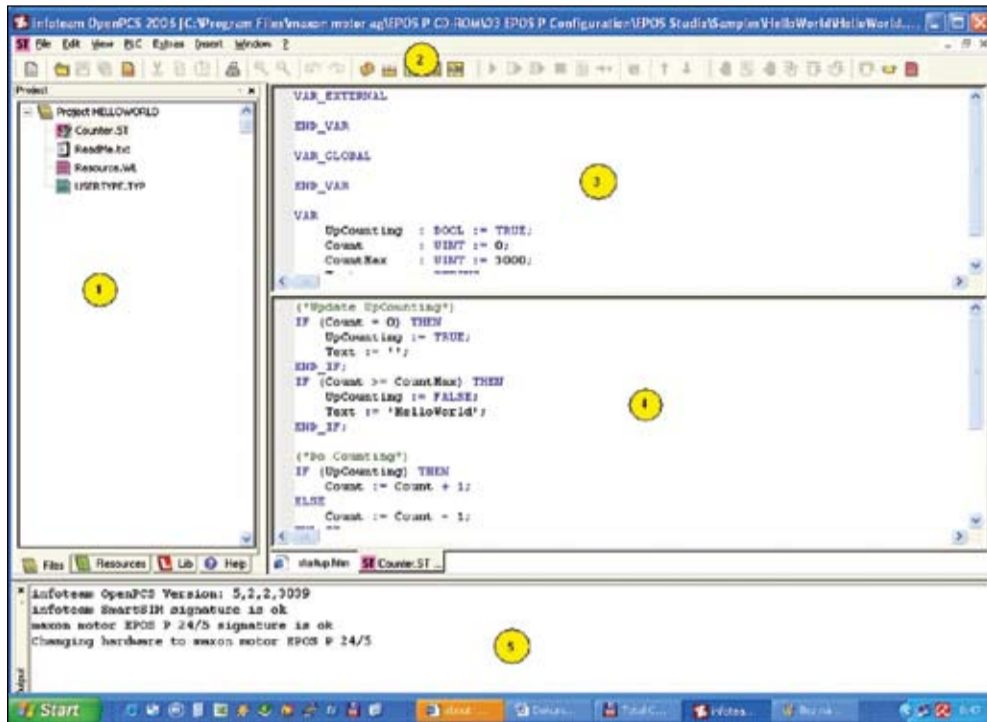
EPOS Studio je velice užitečná a komplexní pomůcka k nastavení uživatelské aplikace s řízením stejnosměrných motorů EPOS, EPOS P a MCD. V příštím dílu se budeme zabývat popisem programovacího prostředí pro řídicí jednotky EPOS P a MCD. ●

ING. JIŘÍ HNÍZDIL,
HNIZDIL@UZIMEX.CZ

maxon motor
driven by precision

Uživatelský software

maxon řízení EPOS P



VÝVOJOVÉ PROSTŘEDÍ OPENPCS

Dnešní díl seriálu o řídicích jednotkách EPOS P se zabývá popisem vývojového prostředí pro tvorbu řídicího programu řídicích jednotek EPOS P. Na úvod je třeba zmínit, že vývojové prostředí je součástí balíku EPOS STUDIO volně dostupného na internetových stránkách <http://www.maxonmotor.com> v sekci services and downloads.

Na obrázku je zobrazeno hlavní okno vývojového prostředí. Přejít na řídicí jednotku EPOS P do módu programování se děje prostřednictvím programu EPOS Studio, kliknutím pravého tlačítka myši na řídicí jednotku EPOS P a volbou „Tools“ a „IEC-61131 Programming“. Násle-

duje kliknutím levého tlačítka myši na tlačítko „Open Programming Tool“.

Vytvářený řídicí program se označuje jako projekt. Hlavní okno vývojového prostředí OpenPCS se skládá z následujících částí:

Okno projektu označené značkou 1. V tomto okně se nalézá struktura projektu. Na úvod lze prohlásit, že tak, jak je možno vytvořit projekt s nastavením řídicích jednotek v programu EPOS Studio, lze vytvořit projekt ve vývojovém prostředí OpenPCS. Projekt obsahuje nastavení řídicího programu jako celku, seznam použitých souborů, nastavení jednotlivých proměnných apod. Okno má ve

své dolní části další záložky „Files“, „Resources“, „Lib“ a „Help“.

Záložka „Files“ – okno projektu zobrazuje seznam vložených souborů do projektu.

Záložka „Resources“ – okno projektu zobrazuje seznam prostředků použitých v řídicím programu, jako jsou například funkční bloky (log. součet, součin, čítač apod.) a jejich vlastnosti.

Záložka „Lib“ – okno projektu zobrazuje seznam knihoven vložených do projektu. Knihovna je soubor funkcí použitelných při tvorbě řídicího programu. Knihovny mohou být standardní a uživatelské. Ve standardních jsou definovány všeobecné funkce, jako již zmíněný logický součet či součin, v uživatelských jsou definovány funkce pro práci s konkrétním zařízením, jako je řídicí jednotka EPOS či EPOS P.

Záložka „Help“ – okno projektu zobrazuje nápovědu k jednotlivým použitým knihovnám.

Hlavní nabídka a panel ikon, označený značkou 2, slouží k práci s vývojovým prostředím OpenPCS a obsahuje následující položky:

„File“ – příkazy pro práci se soubory jako je například založení nového souboru, uzavření tohoto souboru, kontrolu syntaxe příkazů použitých v souboru, ukončení programu a další.

„Edit“ – příkazy pro editaci zdrojového kódu jako je kopie kódu, vložení zkopírovaného kódu, vyhledání určitého kódu apod.

„View“ – volby zobrazení jednotlivých částí hlavního okna vývojového prostředí OpenPCS.

„PLC“ – příkazy pro práci s programovatelným řídicím automatem řídicí jednotky EPOS P, jako jsou například sestavení programu do

strojové podoby, spuštění či ukončení běhu programu v paměti řídicí jednotky EPOS P, reset programu apod.

„Extra“ – příkazy pro uživatelské nastavení vývojového prostředí OpenPCS, jako je například volba použité znakové sady při psaní programu, popřípadě barva písma.

„Insert“ – příkazy pro vložení funkčního bloku z příslušné knihovny či programové proměnné do řídicího programu.

„Window“ – příkazy pro práci s okny vývojového prostředí OpenPCS, jako je například rovnoměrné přeskládání oken v horizontální či vertikální rovině.

„?“ – příkazy pro práci s nápovědou a informací o licenci.

Ikony panelu ikon pak slouží jako grafická reprezentace některých příkazů z hlavní nabídky.

Okno označené značkou 3 obsahuje definice použitých proměnných.

Okno označené značkou 4 obsahuje vlastní zápis programu v kódu zvoleného programovacího jazyka.

Okno označené značkou 5 zobrazuje různá hlášení vývojového prostředí OpenPCS, jako například chyby vzniklé při kompilaci programu, informace o programovaném zařízení, informace o využití paměti a další.

V příštím díle se budeme zabývat tvorbou jednoduchého demonstračního řídicího programu v jazyce IL – Instruction List s podrobnějším vysvětlením jednotlivých funkcí vývojového prostředí OpenPCS. ●

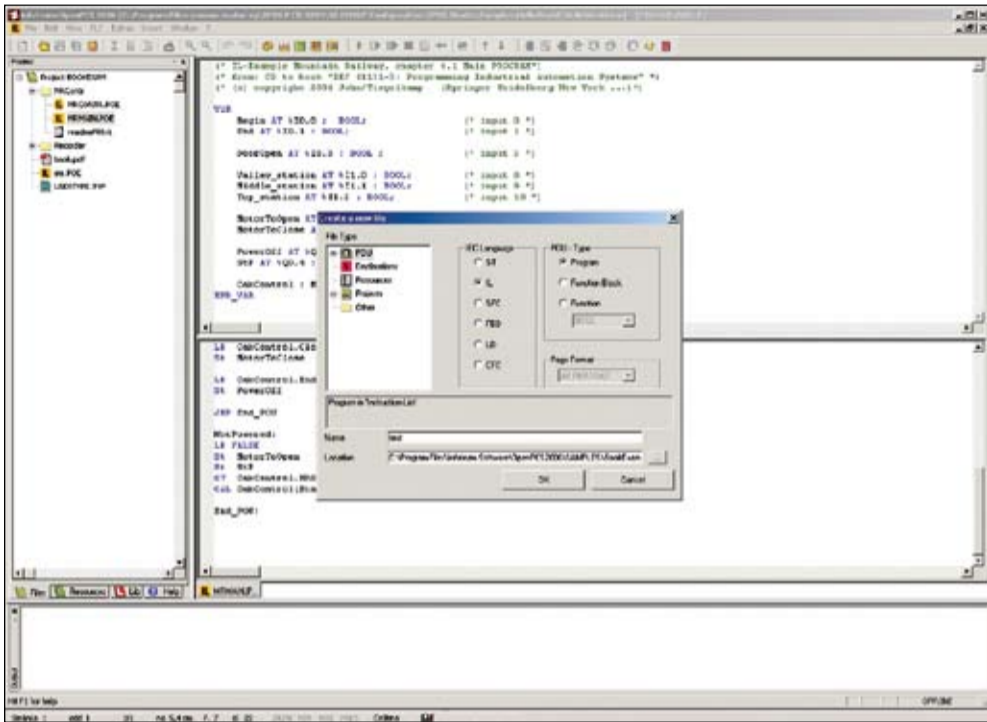
ING. JIŘÍ HNÍZDIL,
HNIZDIL@UZIMEX.CZ

maxon motor

driven by precision

Uživatelský software

maxon řízení EPOS P



ZAČÍNÁME PROGRAMOVAT, JAZYK IL

Dnešní díl seriálu se zabývá úvodem do programování spolu se stručným popisem jazyka IL. IL označuje ve zkratce slovní spojení Instruction List. Jedná se o programovací jazyk podobný assembleru odpovídající standardu programování průmyslových automatů IEC 61131-3. Při tvorbě řídicího programu je třeba funkční programové prostředí EPOS STUDIO společně s vývojovým prostředím OpenPCS. To je volně ke stažení na stránkách <http://www.maxonmotor.com>, v sekci services & downloads zadáním klíčového slova EPOS P.

Obrázek zobrazuje hlavní okno vývojového prostředí OpenPCS s otevřeným projektem v jazyce IL. V popředí je okno průvodce, kterého používáme při vytváření nového řídicího programu. Vývojové prostředí OpenPCS spustíme tak, že v prostředí programu EPOS Studio pravým klikem myši zvolíme řídicí jednotku EPOS P a dále zvolíme „Tools“ a „IEC-61131 Programming“ a „Open Programming Tool“.

Tvorba nového projektu začíná položkou „New“ v hlavní nabídce „File“. Prostřednictvím jednoduchého průvodce zvolíme tvorbu nového projektu „Projects“ a „Maxon Motor ag“ a jako

šablonu projektu zvolíme v průvodci položku „EPOS P“. Dále je nutno tento projekt pojmenovat vyplněním položky „Name“ a v položce průvodce „Location“ vybrat místo, kde bude projekt uložen na pevném disku počítače. Tím je projekt vytvořen. Do projektu můžeme zahrnout soubory s programovým kódem či deklaracemi proměnných a různé knihovny.

Na úvod bude postačující vložit do projektu pouze hlavní soubor s řídicím programem. To provedeme opět prostřednictvím hlavní nabídky Open PCS a položky „New“. Použijeme stejného průvodce jako v předchozím případě. Volbou POU - Program Unit můžeme vybrat jazyk, v kterém bude program zapsán, v našem případě tedy IL - Instruction List. Dále zvolíme „POU Type“ jako „Program“. Tvorbu hlavního programu dokončíme zadáním jména souboru a místa jeho uložení v položkách „Name“ a „Location“. Stiskneme tlačítko „OK“. Vývojové prostředí se zeptá, zdali chceme nový soubor zahrnout do projektu, a to otázkou „Do you like add jmeno programu to active resource?“. Stiskneme tlačítko Ano, popř. OK.

Nyní se otevřelo okno rozdělené na dvě části. Horní část slouží k zápisu a deklaraci (definování) proměnných řídicího programu. Do dolní

části se zapisuje vlastní řídicí program. Okno proměnných obsahuje předdefinované sekce. Sekce „VAR_EXTERNAL“ slouží pro deklaraci externích proměnných, tj. proměnných uložených v jiném kódu. Do sekce „VAR_GLOBAL“ se deklarují globální proměnné programu. Sekce „VAR“ je určena pro deklaraci lokálních proměnných. Příslušná sekce je pak ukončena zápisem „END_VAR“.

Pokud tedy budeme v řídicím programu vyžadovat proměnnou typu „BOOL“, zapíšeme do sekce „VAR“ následující kód:

```
test : BOOL :=True
```

Proměnná typu „BOOL“ je logická proměnná, nabývající pouze hodnot „TRUE“ nebo „FALSE“, neboli pravda nebo nepravda. Zápisem vznikla proměnná s názvem test a byla jí přiřazena hodnota TRUE.

V sekci proměnných můžeme rovněž definovat použití funkčních bloků z knihoven. Pro práci s řídicí jednotkou EPOS P je možno využít knihovnu funkcí „SMART Motion“. Popis funkcí v této knihovně je dostupný v dokumentu „EPOS P Programming Reference“, který je součástí instalace programového prostředí EPOS STUDIO. Využití funkce z dané knihovny v řídicím programu pak vypadá následovně. V části obsahující deklaraci proměnných definujeme novou proměnnou, např.:

```
EPOS_RESET : MC_Reset
```

Tím vznikla proměnná, která reprezentuje funkci MC_Reset. Tato funkce slouží k vymazání veškerých provozních chyb z řídicí jednotky EPOS P či jakékoli jiné řídicí jednotky EPOS na sběrnici CAN.

Většina funkcí vyžaduje zadání dalších parametrů. Parametr Axis jsou ID řídicích jednotek, kde má být funkce provedena. Parametr EXECUTE udává, zda má být funkce opravdu provedena. Jednotlivé funkce rovněž vrací parametry. Například parametr „Done“ vypovídá o tom, zda byla funkce provedena úspěšně či nikoli.

Část určená pro zápis vlastního programu obsahuje programové instrukce, které jsou spe-

cifické pro určitý jazyk. Jazyk IL, tj. Instruction List, obsahuje následující instrukce: Instrukce skoků pro řízení běhu řídicího programu, např. instrukce skoku JMP, instrukce logických operací, např. logický součet AND, instrukce matematických operací, např. odečítání SUB, instrukce pro práci s daty, např. uložení hodnoty ST a instrukce logických operátorů, např. porovnání EQ.

Změna proměnné test, kterou jsme deklarovali, by v jazyce IL mohla vypadat v programovém kódu následovně:

```
LD False
St test
```

První instrukce načte do paměťového místa označeného normou IEC 61131-3 jako CR, tj. virtuální akumulátor, hodnotu nepravda, tj. false. Druhá instrukce tuto hodnotu uloží do proměnné test.

Takto lze v jazyce IL vytvořit celý program s použitím funkcí řídicích jednotek EPOS a EPOS P. Dokončený řídicí program je nutno přeložit do strojového kódu. To lze provést pomocí položky hlavní nabídky „PLC“ volbou „Build Active Resource“. Pokud vývojové prostředí OpenPCS v zápisu řídicího programu neoznámí žádné chybové hlášení, je třeba uložit řídicí program do řídicí jednotky EPOS P. Použijeme položku hlavní nabídky PLC, volíme PC->PLC download. Řídicí program je spuštěn volbou položky „Cold Start“ hlavní nabídky PLC.

Pro demonstraci řídicího programu v jazyce IL byl vytvořen program pro simulaci pásového dopravníku. Řídicí program pracuje s řídicí jednotkou EPOS P v polohové smyčce tak, že každých 10 sekund nastaví novou polohu dopravníku. Tato poloha odpovídá změně 10 000 čtvrtin ikrementu. Vytvořený program je možno pro demonstraci stáhnout na webových stránkách firmy Uzimex Praha spol. s r.o. <http://www.uzimex.cz>

Příští díl seriálu se bude věnovat dalšímu z programovacích jazyků ST – structured text. ●

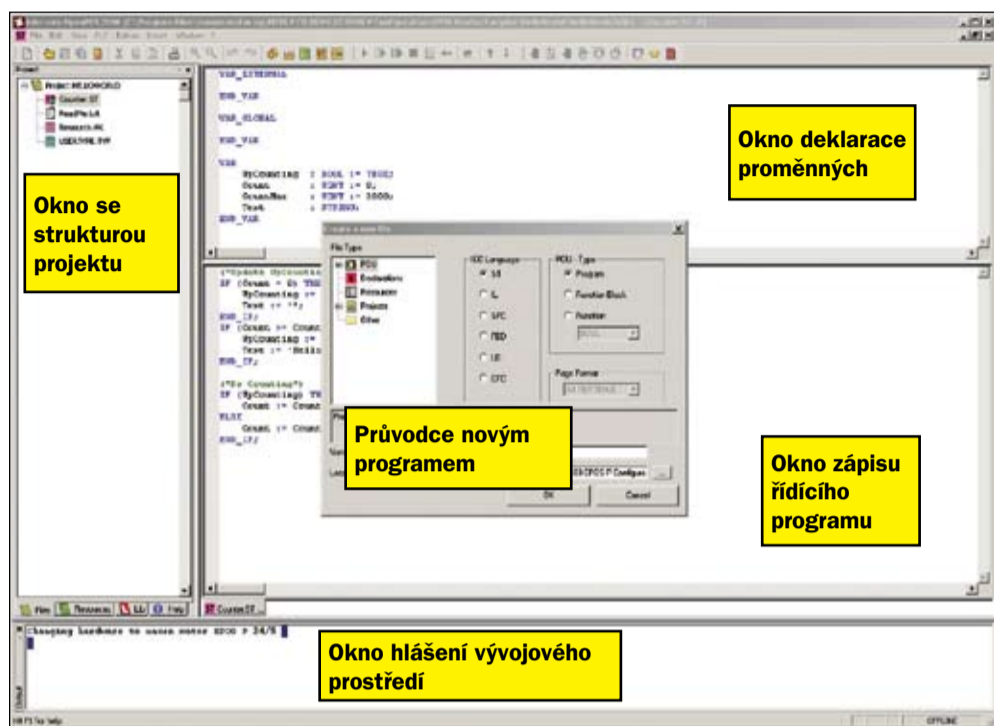
ING. JIŘÍ HNÍZDIL,
HNIZDIL@UZIMEX.CZ

maxon motor

driven by precision

Uživatelský software

maxon řízení EPOS P



PROGRAMUJEME V JAZYCE ST

Seriál se v dnešním díle zabývá úvodem do programování v jazyce ST. ST označuje ve zkratce slovní spojení Structured Text. Programovací jazyk odpovídá standardu programování průmyslových automatů IEC 61131-3. Je podobný Pascalu. Při tvorbě řídicího programu je třeba funkční programové prostředí EPOS STUDIO společně s vývojovým prostředím OpenPCS. Je volně ke stažení na stránkách <http://www.maxonmotor.com> v sekci services & downloads zadáním klíčového slova EPOS P.

Obrázek zobrazuje hlavní okno vývojového prostředí OpenPCS s otevřeným projektem v jazyce ST a dále okno průvodce s nově vytvořeným řídicím programem. Vývojové prostředí OpenPCS spustíme v prostředí programu EPOS Studio pravým klikem myši, zvolíme řídicí jednotku EPOS P a dále zvolíme „Tools“ a „IEC-61131 Programming“ a „Open Programming Tool“.

Vytvoříme nový projekt položkou „New“ v hlavní nabídce „File“ vývojového prostředí OpenPCS. Prostřednictvím jednoduchého průvodce zvolíme tvorbu nového projektu „Projects“ a „Maxon Motor ag“, a jako šablonu projektu zvolíme v průvodci položku „EPOS P“. Dále je nutno tento projekt pojmenovat vyplněním položky „Name“ a v položce průvodce „Location“ vybrat místo, kde bude projekt uložen na pevném disku počítače. Tím je projekt vytvořen. Do projektu můžeme zahrnout

soubory s programovým kódem či deklaracemi proměnných a různé knihovny.

V začátcích bude postačující vložit do projektu pouze hlavní soubor s řídicím programem. To provedeme opět prostřednictvím hlavní nabídky Open PCS a položky „New“. Použijeme stejného průvodce jako v předchozím případě. Volbou POU-Program Unit můžeme vybrat jazyk, ve kterém bude program zapsán, v našem případě tedy ST – Structured Text. Dále zvolíme „POU Type“ jako „Program“. Tvorbu hlavního programu dokončíme zadáním jména souboru a místa jeho uložení v položkách „Name“ a „Location“. Stiskneme tlačítko „OK“. Vývojové prostředí se zeptá, zda-li chceme nový soubor zahrnout do projektu, a to otázkou „Do you like add jmeno programu to active resource?“. Stiskneme tlačítko Ano, popř. OK.

Nyní se otevřelo okno. Stejně jako v předchozím případě je rozdělené na dvě části. Horní část slouží k zápisu a deklaraci, neboli definování proměnných řídicího programu. Dolní část slouží k zápisu vlastního řídicího programu. Okno proměnných obsahuje předdefinované sekce. Sekce „VAR_EXTERNAL“ slouží k deklaraci externích proměnných, tj. proměnných uložených v jiném kódu. Do sekce „VAR_GLOBAL“ se deklarují globální proměnné programu. Sekce „VAR“ je určena pro deklaraci lokálních proměnných. Příslušná sekce je pak ukončena zápisem „END_VAR“.

Pokud tedy budeme v řídicím programu vyžadovat proměnnou typu „UINT“, zapíšeme do sekce „VAR“ následující kód:

```
cislo : UINT :=77;
```

Proměnná typu „UINT“ je numerická proměnná, nabývající hodnot 0-65535 bez označení znaménka. Touto deklarací vznikla proměnná číslo a její hodnota nabývá 77. Číselná soustava, ve které je proměnná zadána, je standardně nastavená na desítkovou. Pokud je třeba zadat číslo v jiné než desítkové soustavě, deklarujeme proměnnou např. následujícím způsobem:

```
cislo : UINT :=2#1001;
```

Zápis deklaruje opět proměnnou číslo, avšak ve dvojkové soustavě.

V sekci proměnných můžeme rovněž definovat použití funkčních bloků z knihoven. Pro práci s řídicí jednotkou EPOS P využíváme knihovnu „SMART Motion“.

Část určená pro zápis vlastního programu obsahuje programové instrukce, které jsou specifické pro určitý jazyk. V úvodu jsme uvedli, že jazyk ST, tj. Structured Text, je velice podobný programovacímu jazyku Pascal. Řídicí program se skládá z jednotlivých programových řádků. Každý řádek je většinou ukončen středníkem. Při tvorbě programu je využito klíčových slov jazyka ST. Seznam těchto slov je dostupný v nápovědě k vývojovému prostředí OpenPCS. Jako příklad je možno uvést zápis podmínky:

```
If test = true then
  Test:=false;
End If;
```

Podmínka říká, že pokud je logická proměnná test pravdivá, nastaví se na hodnotu nepravda.

Změna proměnné číslo, kterou jsme deklarovali, by v jazyce ST mohla vypadat v programovém kódu následovně:

```
cislo := 66;
```

Takto lze v jazyce ST vytvořit celý program s použitím funkcí řídicích jednotek EPOS a EPOS P. Jednoduchý příklad demonstruje uvedení řídicí jednotky EPOS P do aktivního stavu, kdy je zapnutý výkonový stupeň.

Deklarace proměnných:

```
Var
Osa : Axis_Ref:=(AxisNo:=0)
Blok_Reset : MC_Reset;
Blok_Power : MC_Power;
End_Var
```

Zápis programu

```
Blok_Reset(Axis:=Osa,Execute:=True);
If Blok_Reset.Done then
  Blok_Power(Axis:=Osa,Enable:=True);
End If;
```

MC Reset je funkcí knihovny SmartMotion, která resetuje řídicí jednotku EPOS P, včetně mazání provozních chyb. Zadáme vstupní parametry funkce. Parametr Axis je id řídicí jednotky a parametrem Execute potvrdíme, že se má daná funkce opravdu provést. V případě, že daná funkce proběhne úspěšně, vrátí prostřednictvím parametru Done hodnotu True, tj. pravda. Po jejím otestování lze zavolat funkci MC_Power, která aktivuje příslušnou řídicí jednotku. MC_Power vyžaduje rovněž zadat vstupní parametry. Parametr Axis je id jednotky a True v parametru Enable potvrzuje, že funkce má být provedena.

Dokončený řídicí program je nutno přeložit do strojového kódu. To lze provést pomocí položky hlavní nabídky „PLC“ volbou „Build Active Resource“. Pokud vývojové prostředí OpenPCS v zápisu řídicího programu nenahlásí žádné chybové hlášení, uložíme řídicí program do řídicí jednotky EPOS P. Použijeme položku hlavní nabídky PLC, volíme PC->PLC download. Řídicí program je spuštěn volbou položky „Cold Start“ hlavní nabídky PLC.

Pro demonstraci řídicího programu v jazyce ST byl vytvořen program simulující posuvné dveře. Řídicí program pracuje s řídicí jednotkou EPOS P v režimu vyhledávání domovské polohy prostřednictvím koncových spínačů. Vytvořený program je možno pro demonstraci stáhnout na webových stránkách firmy Uzimex Praha spol. s r.o. <http://www.uzimex.cz>

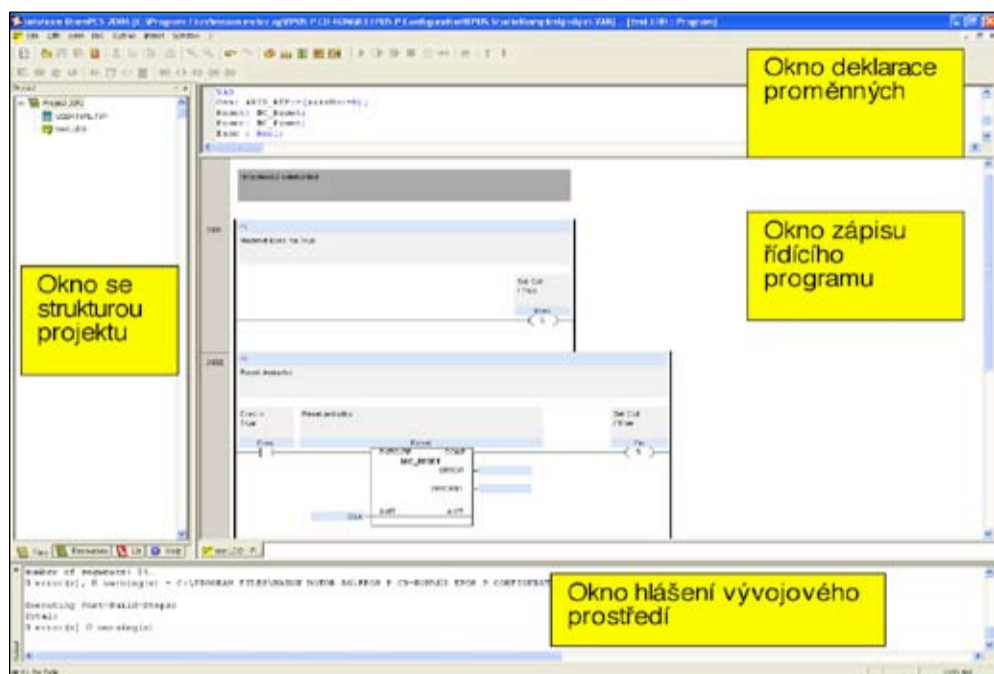
Příští díl seriálu se bude věnovat dalšímu z programovacích jazyků LD – Ladder Diagram ●

ING. JIŘÍ HNÍZDIL,
HNIZDIL@UZIMEX.CZ

maxon motor
driven by precision

Uživatelský software

maxon řízení EPOS P



PROGRAMUJEME V JAZYCE LD

Serial se v dnešním díle zabývá úvodem do programování v jazyce LD. LD označuje ve zkratce slovní spojení Ladder Diagram. Programovací jazyk odpovídá standardu programování průmyslových automatů IEC 61131-3. Jeho princip je založen na reléové logice. Při tvorbě řídicího programu je třeba funkční programové prostředí EPOS STUDIO společně s vývojovým prostředím OpenPCS. Je volně ke stažení na stránkách <http://www.maxonmotor.com> v sekci services & downloads zadáním klíčového slova EPOS P.

Obrázek zobrazuje hlavní okno vývojového prostředí OpenPCS s otevřeným projektem v jazyce LD. Vývojové prostředí OpenPCS spustíme v prostředí programu EPOS Studio pravým klikem myši, zvolíme řídicí jednotku EPOS P a dále zvolíme „Tools“ a „IEC-61131 Programming“ a „Open Programming Tool“.

Vytvoříme nový projekt položkou „New“ v hlavní nabídce „File“ vývojového prostředí OpenPCS. Prostřednictvím jednoduchého průvodce zvolíme tvorbu nového projektu „Projects“ a „Maxon Motor ag“ a jako šablonu projektu zvolíme v průvodci položku „EPOS P“. Dále je nutno tento projekt pojmenovat vyplněním položky „Name“ a v položce průvodce „Location“ vybrat místo, kde bude projekt uložen na pevném disku počítače. Tím je projekt vytvořen. Do projektu můžeme zahrnout soubory s programovým kódem či deklaracemi proměnných a různé knihovny.

V začátcích bude postačující vložit do projektu pouze hlavní soubor s řídicím programem. To provedeme opět prostřednictvím hlavní nabídky Open PCS a položky „New“. Použijeme stejného průvodce jako v předchozím případě. Volbou POU – Program Unit můžeme vybrat jazyk, v kterém bude program zapsán, v našem případě tedy LD – Ladder Diagram. Dále zvolíme „POU Type“ jako „Program“. Tvorbu hlavního programu dokončíme zadáním jména souboru a místa jeho uložení v položkách „Name“ a „Location“. Stiskneme tlačítko „OK“. Vývojové prostředí se zeptá, zdali chceme nový soubor zahrnout do projektu, a to otázkou „Do you like add jmeno programu to active resource?“. Stiskneme tlačítko Ano, popř. OK.

Nyní se otevřelo okno, stejně jako v předchozích případech je rozdělené na dvě části. Horní část slouží k zápisu a deklaraci proměnných řídicího programu. Dolní část slouží ke grafické reprezentaci vlastního řídicího programu.

Okno proměnných obsahuje předdefinované sekce. Sekce „VAR_EXTERNAL“ slouží pro deklaraci externích proměnných, tj. proměnných uložených v jiném kódu. Do sekce „VAR_GLOBAL“ se deklarují globální proměnné programu. Sekce „VAR“ je určena pro deklaraci lokálních proměnných. Příslušná sekce je pak ukončena zápisem „END_VAR“. Deklarace proměnné Exec může vypadat následovně

```
Exec : Bool;
```

Proměnná Exec byla deklarována jako proměnná typu BOOL – logická proměnná nabývající pouze dvou stavů Pravda/Nepravda.

V sekci proměnných můžeme dále definovat použití funkčních bloků z knihoven. Pro práci s řídicí jednotkou EPOS P využíváme knihovnu „SMART Motion“.

Část určená pro zápis vlastního programu obsahuje grafické schéma řídicího programu. V úvodu jsme uvedli, že jazyk LD, tj. Ladder Diagram, je graficky orientovaný programovací jazyk založený na principu reléové logiky. Relé je elektrotechnický prvek, který se skládá z cívky s jádrem a z kotvičky, která ovládá kontakty. Pokud protéká cívkou elektrický proud, jádro cívky přitáhne kotvičku. Ta mechanickým působením sepně či rozeprne kontakty a v navazujících obvodech může protékat proud. Relé může obsahovat několik nezávislých kontaktů.

V tomto duchu se nese celá filozofie programování v jazyce LD. Program je tvořen pomocí grafických symbolů, reprezentujících jednotlivé cívky a kontakty relé. Ty ovládají další funkční bloky použité v programu. Prostřednictvím dalších grafických příkazů můžeme cívky relé aktivovat či deaktivovat, popřípadě určit, že daný kontakt ovládá příslušnou cívku relé je spínací či rozpínací.

Programovací jazyk umožňuje používat přehledných textových popisů přímo k jednotlivým prvkům schématu popřípadě ke skupině prvků.

Způsob programování vysvětlíme na jednoduchém případě, kdy chceme resetovat chyby řídicí jednotky EPOS P. Reset jednotky zajišťuje funkční blok MC_Reset z knihovny Smart Motion. Tento blok vyžaduje jako vstupní parametr Axis - identifikátor osy (tj. id jednotky kterou chceme resetovat) a parametr Execute, který udává, zda li se má reset provést (true – ano/false – ne). Proto deklarujeme v sekci VAR následující proměnné

```
Var
Osa: AXIS_REF:=(AxisNo:=0);
Reset: MC_Reset;
Exec : Bool;
Fin : Bool;
END_Var
```

Nyní můžeme vytvořit vlastní řídicí program. Bude se skládat ze dvou grafických řádků. Na prvním provedeme aktivaci cívky relé označeného jako Exec, na druhém provedeme sepnutí kontaktů tohoto relé. To bude reprezentovat hodnotu True na

vstupu Execute funkčního bloku MC_Reset, který jsme deklarovali pod názvem Reset.

První grafický řádek řídicího programu bude obsahovat pouze jeden grafický příkaz Set Coil. Vložení tohoto příkazu provedeme kliknutím pravého tlačítka myši na linii grafického řádku a ze zobrazené nabídky vybereme možnost Set Coil. Pro pojmenování této cívky klikneme opět pravým tlačítkem myši na grafický symbol cívky a ze zobrazené nabídky zvolíme možnost Insert Variable. Objeví se okno se seznamem deklarovaných proměnných. Zvolíme možnost Exec.

Druhý grafický řádek přidáme do řídicího programu tak, že opět klikneme pravým tlačítkem myši na linii prvního grafického řádku a ze zobrazené nabídky vybereme možnost Insert Network. Tento řádek bude obsahovat kontakt relé Exec z prvního grafického řádku, dále funkční blok MC_Reset a cívku dalšího relé Fin, které může například aktivovat další funkční bloky. Všechny zmíněné údaje je možno na druhý grafický řádek přidat stejným způsobem jako při tvorbě řádku prvního, tj. pravé tlačítko myši a možnosti Insert Contact, Insert Function Block a Insert Set. Každému grafickému příkazu přiřadíme příslušnou proměnnou, tedy Exec, Reset a Fin. Funkční blok MC_Reset dále vyžaduje definici identifikátoru osy. To je provedeno stejně jako v ostatních případech přiřazením proměnných pomocí okna Insert Variable. Pro demonstraci je grafický zápis programu zobrazen na obrázku v úvodu.

Dokončený řídicí program je nutno přeložit do strojového kódu. To lze provést pomocí položky hlavní nabídky „PLC“ volbou „Build Active Resource“. Pokud vývojové prostředí OpenPCS v zápisu řídicího programu nenahlásí žádné chybové hlášení, uložíme řídicí program do řídicí jednotky EPOS P. Použijeme položku hlavní nabídky PLC, volíme PC->PLC download. Řídicí program je spuštěn volbou položky „Cold Start“ hlavní nabídky PLC.

Pro demonstraci řídicího programu v jazyce LD byl vytvořen program simulující pohon hvězdkového dalekohledu. Řídicí program pracuje s řídicí jednotkou EPOS P v režimu Profile Position Mode. Vytvořený program je možno pro demonstraci stáhnout na webových stránkách firmy Uzimex Praha spol. s r.o. <http://www.uzimex.cz>

Příští díl seriálu se bude věnovat popisu skutečných aplikací s řídicími jednotkami EPOS P ●

ING. JIŘÍ HNÍZDIL,
HNIZDIL@UZIMEX.CZ

maxon motor

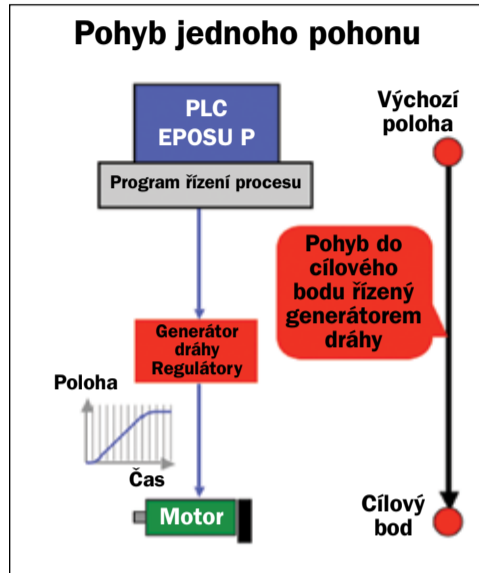
driven by precision

Jak použít maxon řízení EPOS P



POHYB JEDNOHO Pohonu

Jednoduchá aplikace je řízení pohybu jednoho pohonu. V jednotce EPOS P se po připojení na napájecí napětí spustí program, který jsme do ní předem uložili pomocí personálního počítače s klávesnicí. První smyčka programu čeká na kombinaci nul a jedniček na zvolených digitálních vstupech. Na to uvede svůj pohon do pohybu s průběhem rychlosti podle programu



se zastavením v bodě, který byl zadán do programu. Generátor dráhy a regulátory polohy, rychlosti a proudu řídí pohon podle požadavků programu. Další pohyb následuje po splnění naprogramovaných podmínek.

Poloha cílového bodu může být určena relativně od výchozí polohy. EPOS P při pohybu počítá inkrementy snímače na motoru. Cílový bod může být definován i absolutně vzhledem k základní nulové poloze poháněného zařízení. V tom případě je nutno po připojení EPOSU P na napájení nalézt základní nulovou polohu. EPOS P použije pro její nalezení některý z několika způsobů s využitím polohových nebo koncových spínačů nebo pevného dorazu. Polohu zpřesní pomocí signálu třetího kanálu inkrementálního snímače, který přichází jednou za otáčku a nejbližšího signálu kanálů A, B snímače.

ČÍM LZE OVLIVNIT POHYB DO CÍLOVÉHO BODU

EPOS P má na konektorech 6 digitálních vstupů, dva digitální výstupy a dva analogové vstupy. Dále konektor pro připojení na sběrnici CAN a konektor pro RS232. Vstupy a sběrnice

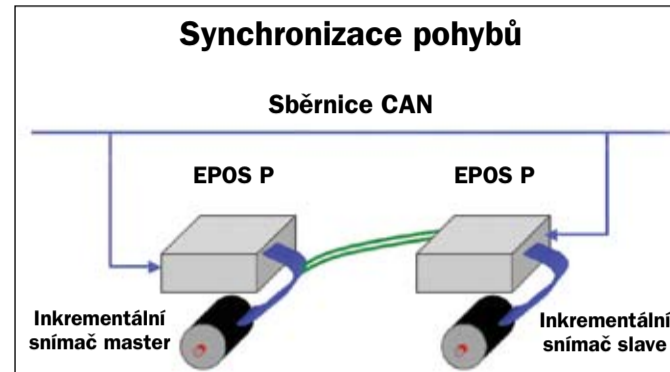
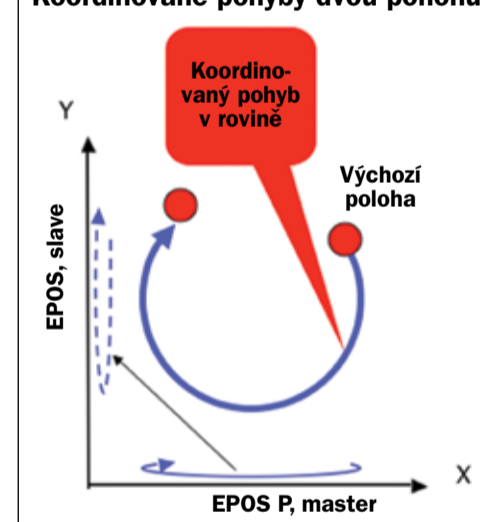
se mohou využít pro vstup parametrů k ovlivnění toku programu. Pro popsaný jednoduchý případ lze využít informace poslané po sběrnici CAN dalšími snímači na ovládaném zařízení a analogové informace, např. o teplotě nebo stavu hladiny, přivedené na analogové vstupy.

Do vnitřní paměti EPOSU P vkládáme před zahájením provozu údaje o motoru a inkrementálním snímači. Využijí se pro převod počtu inkrementů snímače na úhel nebo vzdálenost. Zadáme i přípustnou rychlost a trvale přípustný proud motoru z pohledu jeho ochrany před poškozením. Údaje o přípustném trvalém proudu a součinitel přestupu tepla z vinutí poslouží pro výpočet okamžité teploty vinutí při proměnném a pulsním zatížení.

ŘÍZENÍ DISKOVÝCH MOTORŮ EC

Kvalitní řízení polohy je podmíněno dostatečnou hustotou signálů o poloze motoru. Motory DC i válcové motory EC lze téměř vždy doplnit inkrementálním snímačem s hustotou signálu alespoň 64, u větších motorů až 1000 inkrementů na otáčku. Diskový motor obsahuje snímač se třemi I-Hallovými sondami, které postačují pro elektronické přepínání proudu s obdélkovým průběhem. 6 hran signálu během otočení přes jeden pól nedostačuje pro stabilní a přesné řízení malé rychlosti pod 1.000 ot/min na pólpar. Zastavení v požadované poloze je ovlivněno říditelností malé rychlosti. Realizace takové aplikace je podmíněna nastavením nízkých zesílení regulátorů EPOSU P. U diskového motoru proto nemůžeme očekávat vysokou přesnost zastavení. Pro aplikace s EPOSEM P se zavádí do výroby

Koordinované pohyby dvou pohonů



diskový motor s průměrem 90 mm s integrovaným inkrementálním snímačem.

ŘÍZENÍ VÍCE Pohonů NA SBĚRNICI CAN

Na sběrnici CAN, řízenou EPOSEM P, může být připojeno až 127 zařízení CAN, z toho až 32 pohonů s jednotkami EPOS. Počet pohonů s průběžně koordinovanými polohami podle požadované závislosti je reálně omezen na 3 až 4. Omezujícím faktorem je rychlost komunikace po sběrnici. Základní rychlost sběrnice CAN je 1 Mbit/s. Průběžná koordinace polohy 3 pohonů podle požadované závislosti probíhá každé 2 ms.

Koordinace poloh v průběhu pohybu není ve většině aplikací nutná. Zpravidla postačuje pouze správná koncová poloha. Rychlost komunikace je pak vyhovující i při větším počtu pohonů, neboť obvyklý pohybový příkaz ve tvaru PDO trvá 130 μ s.

LINEÁRNÍ SYNCHRONIZACE POHYBŮ NĚKOLIKA Pohonů

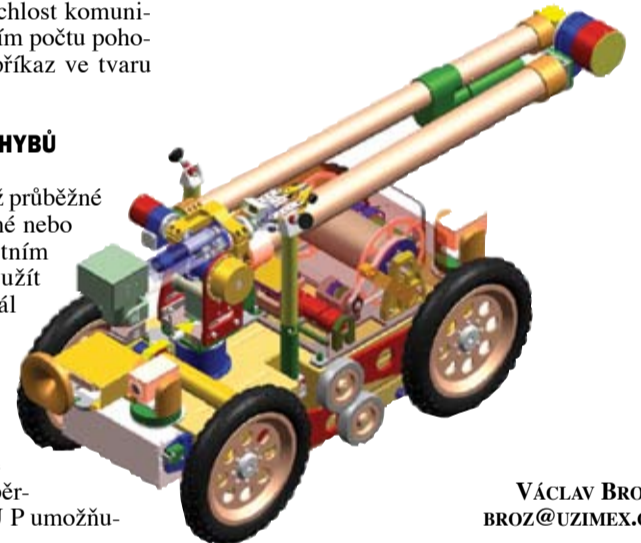
Aplikace s více pohony, jejichž průběžné polohy mají být synchronizované nebo které se mají otáčet s konstantním poměrem rychlostí, mohou využít mód Master Encoder. Signál inkrementálního snímače řídicího pohonu přivedeme na dva z digitálních vstupů podřízených jednotek. Programově zadáme přípustnou polohovou odchylku, jejíž překročení se projeví jako hlášení chyby na sběrnici CAN. Paměť RAM EPOSU P umožňu-

je uložení polohy při poruše napájení.

POJÍZDNÝ ROBOT

Čtyřkolové vozítko s manipulátorem, případně s kamerou, zastane lidskou činnost ve stísněných a nebezpečných prostorech. Vozítko tohoto typu je v programu ZTS VVU Košice. Pojízdný robot může kontrolovat stav potrubí, prohledat staticky porušenou budovu nebo nalézt

a odstranit nastraženou trhavinu. Pohony kol, zatáčení vozítka a manipulaci ramenem zajišťují komutátorové motory. Pohyby řídí nebo koordinuje EPOS P po sběrnici CAN podle dálkového ovládní joystickem. EPOS P řídí svůj motor. Ostatní motory jsou řízeny jednotkami EPOS. Analogové vstupy jednotek EPOS přijímají informace o teplotě a stavu prostředí a síle, kterou působí rameno. Na sběrnici je připojeno i ovládní kamery. Robot má bateriové napájení, pro které je důležitá vysoká účinnost motorů až 92%. Použity jsou motory typu RE40 s vysokou hustotou výkonu v zabíraném objemu a další menší motory, doplněné planetovými převodovkami a inkrementálními snímači. ●

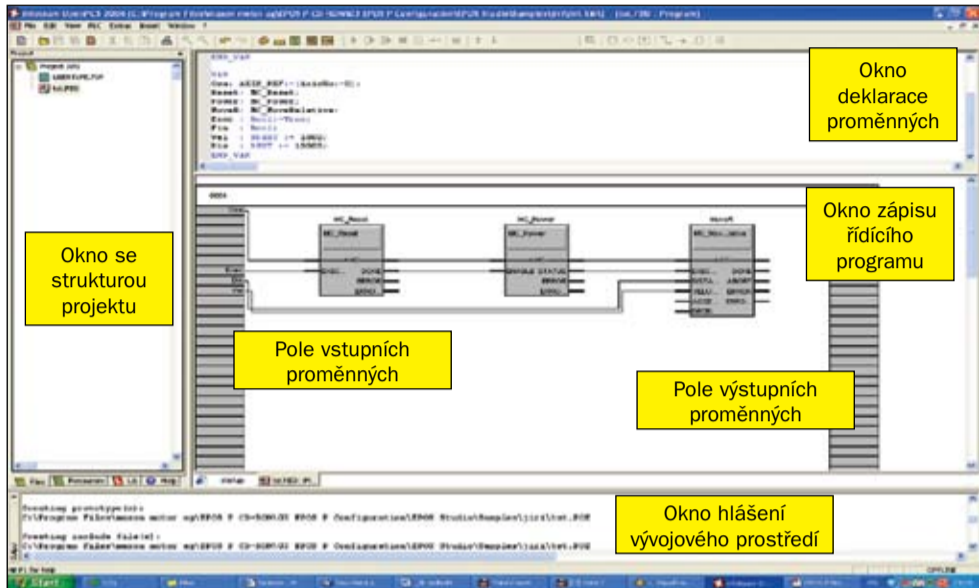


VÁCLAV BROŽ,
BROŽ@UZIMEX.CZ

maxon motor
driven by precision

Použití jazyka FBD

pro řízení EPOS P maxon



Seriál se v dnešním díle zabývá úvodem do programování v jazyce FBD. FBD označuje ve zkratce slovní spojení Function Block Diagram. Programovací jazyk odpovídá standardu programování průmyslových automatů IEC 61131-3. Jeho princip je založen na skladbě jednotlivých funkčních bloků. Při tvorbě řídicího programu je třeba funkční programové prostředí EPOS STUDIO společně s vývojovým prostředím OpenPCS. Je volně ke stažení na stránkách <http://www.maxonmotor.com> v sekci services & downloads zadáním klíčového slova EPOS P.

NOVÝ PROJEKT A PROGRAM

Obrázek zobrazuje hlavní okno vývojového prostředí OpenPCS s otevřeným projektem v jazyce FBD. Vývojové prostředí OpenPCS spustíme v prostředí programu EPOS Studio pravým klikem myši, zvolíme řídicí jednotku EPOS P a dále zvolíme „Tools“ a „IEC-61131 Programming“ a „Open Programming Tool“.

Vytvoříme nový projekt položkou „New“ v hlavní nabídce „File“ vývojového prostředí OpenPCS. Prostřednictvím jednoduchého průvodce zvolíme tvorbu nového projektu „Projects“ a „Maxon Motor ag“ a jako šablonu projektu zvolíme v průvodci položku „EPOS P“. Dále je nutno tento projekt pojmenovat vyplněním položky „Name“ a v položce průvodce „Location“ vybrat místo, kde bude projekt uložen na pevném disku počítače. Tím je projekt vytvořen. Do projektu můžeme zahrnout soubory s programovým kódem či deklaracími proměnných a různé knihovny.

V začátcích bude postačující vložit do projektu pouze hlavní soubor s řídicím programem. To

provedeme opět prostřednictvím hlavní nabídky Open PCS a položky „New“. Použijeme stejného průvodce jako v předchozím případě. Volbou POU – Program Unit můžeme vybrat jazyk, v kterém bude program zapsán, v našem případě tedy FBD – Function Block Diagram. Dále zvolíme „POU Type“ jako „Program“. Tvorbu hlavního programu dokončíme zadáním jména souboru a místa jeho uložení v položkách „Name“ a „Location“. Stiskneme tlačítko „OK“. Vývojové prostředí se zeptá, zdali chceme nový soubor zahrnout do projektu, a to otázkou „Do you like add jmeno programu to active resource?“. Stiskneme tlačítko Ano, popř. OK.

Nyní se otevřelo okno, stejně jako v předchozích případech je rozdělené na dvě části. Horní část slouží k zápisu a deklaraci proměnných řídicího programu. Dolní část slouží ke grafické reprezentaci vlastního řídicího programu.

Okno proměnných obsahuje předdefinovanou sekce. Sekce „VAR_EXTERNAL“ slouží pro deklaraci externích proměnných, tj. proměnných uložených v jiném kódu. Do sekce „VAR_GLOBAL“ se deklarují globální proměnné programu. Sekce „VAR“ je určena pro deklaraci lokálních proměnných. Příslušná sekce je pak ukončena zápisem „END_VAR“. Deklarace proměnné Exec může vypadat následovně:

```
Exec : Bool:=True;
```

Proměnná Exec byla deklarována jako proměnná typu BOOL – logická proměnná nabývající pouze dvou stavů Pravda/Nevprava.

Hodnota proměnné Exec je předdefinována jako True – Logická 1

V sekci proměnných můžeme dále definovat použití funkčních bloků z knihoven. Pro práci s řídicí jednotkou EPOS P využíváme knihovnu „SMART Motion“.

Část určená pro zápis vlastního programu obsahuje grafické schéma řídicího programu. V úvodu jsme uvedli, že jazyk FBD, tj. Function Block Diagram, je graficky orientovaný programovací jazyk založený na principu skladby jednotlivých bloků. Programovací jazyk umožňuje používat přehledných textových popisků přímo k jednotlivým prvkům schématu, popřípadě ke skupině prvků. Řídicí program se skládá z jednotlivých grafických rádků. Řádky jsou ohraničeny zleva polem pro výběr vstupních proměnných a zprava polem pro výběr výstupních proměnných.

Pro přiřazení proměnné do pole proměnných je třeba tuto proměnnou deklarovat v okně proměnných. Dále pak zvolíme položku Insert Variable kliknutím na položku pole proměnných pravým tlačítkem myši. Z otevřeného okna vybereme příslušnou proměnnou. Jednotlivé funkční bloky se do programu vkládají kliknutím pravého tlačítka myši v prostoru okna řídicího programu a následnou volbou Insert FunctionBlock. Propojení vstupů a výstupů funkčních bloků a proměnných probíhá podobným způsobem. Kliknutím levého tlačítka myši vybereme příslušný vstup a výstup či proměnnou, dále následuje kliknutí pravým tlačítkem myši a volba Insert connection.

PŘÍKLAD

Popsaný postup vysvětlíme na jednoduchém případě, kdy chceme nastavit polohu hřídele motoru relativně o 1000 qc. Nejprve provedeme Reset řídicí jednotky EPOS P funkčním blokem MC_Reset. Tento blok vyžaduje jako vstupní parametr Axis - identifikátor osy, tj. id jednotky, kterou chceme resetovat. Dále vyžaduje parametr Execute, který udává, zda li se má reset provést. Přitom true znamená ano, false znamená ne. Pak je třeba zapnout výkonový stupeň řídicí jednotky EPOS P. To zajišťuje blok MC_Power. Vstupní parametry tohoto bloku jsou Axis pro identifikátor osy a Enable pro provedení příkazu. Poslední z použitých bloků je funkční blok MC_Move_Relative, zajišťující otočení hřídele motoru. Vstupními parametry jsou parametry Axis a Exec jako u bloku MC Reset a dále Vel, Dis, Acc

a Dec pro nastavení rychlosti, polohy, zrychlení a zpomalení hřídele motoru.

V sekci VAR deklarujeme následující proměnné:

```
Var
Osa: AXIS_REF:=(AxisNo:=0);
Reset: MC_Reset;
Power: MC_Power;
MoveR: MC_Move_Relative;
Exec : Bool:=True;
Dis : UDINT:= 1000;
Vel : UINT:=1000;
END_Var
```

Kliknutím pravého tlačítka myši v prostoru okna řídicího programu vložíme funkci Insert Function-Block z knihovny Smartmotion bloky MC_Reset, MC_Power a MC_Move_Relative. Opětovným kliknutím pravého tlačítka myši na příslušný funkční blok přidáme funkci Insert Variable deklaraci funkčního bloku. Z otevřeného okna proměnných vybereme položku Reset pro funkční blok MC_Reset, Power pro MC_Power a MoveR pro MC_Move_Relative. Podobným způsobem přiřadíme ostatní proměnné Osa, Dis a Vel do pole vstupních proměnných. V konečné fázi propojíme výše popsaným způsobem příslušné vstupy, výstupy a proměnné tak jak naznačuje obrázek.

Dokončený řídicí program je nutno přeložit do strojového kódu. To lze provést pomocí položky hlavní nabídky „PLC“ volbou „Build Active Resource“. Pokud vývojové prostředí OpenPCS v zápisu řídicího programu nenahlásí žádné chybové hlášení, uložíme řídicí program do řídicí jednotky EPOS P. Použijeme položku hlavní nabídky PLC, volíme PC->PLC download. Řídicí program je spuštěn volbou položky „Cold Start“ hlavní nabídky PLC.

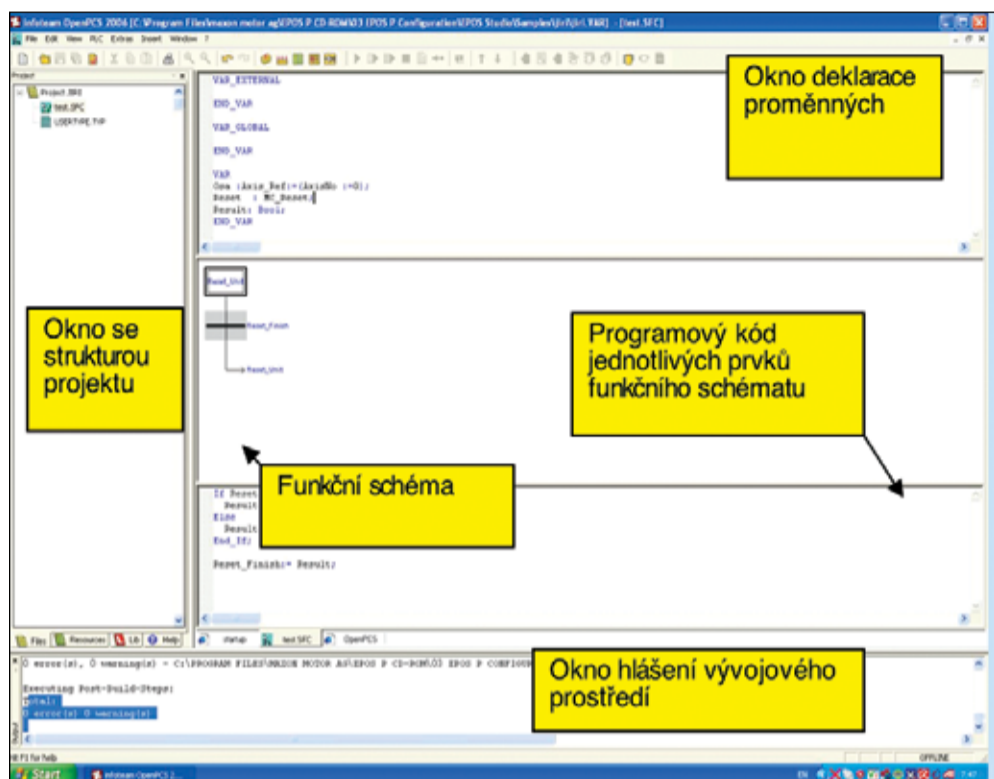
Pro demonstraci řídicího programu v jazyce FBD byl vytvořen program simulující pohon hořáku v operaci dělení materiálu. Řídicí program pracuje s řídicí jednotkou EPOS P v režimu Profile Position Mode. Vytvořený program je možno pro demonstraci stáhnout na webových stránkách firmy Uzimex Praha spol. s r.o. <http://www.uzimex.cz>

Příští díl seriálu se bude věnovat popisu programovacího jazyka SFC – Sequential Function Chart. ●

ING. JIŘÍ HNÍZDIL,
HNIZDIL@UZIMEX.CZ

maxon motor
driven by precision

Použití jazyka SFC pro řízení EPOS P maxon



Seriál se v dnešním díle zabývá úvodem do programování v jazyce SFC. SFC označuje ve zkratce slovní spojení Sequential Function Chart. Programovací jazyk odpovídá standardu programování průmyslových automatů IEC 61131-3. Jeho princip je založen na skladbě funkčního schématu doplněného příkazy jazyků ST či IL. Při tvorbě řídicího programu je třeba funkční programové prostředí EPOS STUDIO společně s vývojovým prostředím OpenPCS. Je volně ke stažení na stránkách <http://www.maxonmotor.com>, v sekci services & downloads zadáním klíčového slova EPOS P.

PROJEKT

Obrázek zobrazuje hlavní okno vývojového prostředí OpenPCS s otevřeným projektem v jazyce SFC. Vývojové prostředí OpenPCS spustíme v prostředí programu EPOS Studio pravým klikem myši, zvolíme řídicí jednotku EPOS P a dále zvolíme „Tools“ a „IEC-61131 Programming“ a „Open Programming Tool“.

Vytvoříme nový projekt položkou „New“ v hlavní nabídce „File“ vývojového prostředí

OpenPCS. Prostřednictvím jednoduchého průvodce zvolíme tvorbu nového projektu „Projects“ a „Maxon Motor ag“ a jako šablonu projektu zvolíme v průvodci položku „EPOS P“. Dále je nutno tento projekt pojmenovat vyplněním položky „Name“ a v položce průvodce „Location“ vybrat místo, kde bude projekt uložen na pevném disku počítače. Tím je projekt vytvořen. Do projektu můžeme zahrnout soubory s programovým kódem či deklaracemi proměnných a různé knihovny.

PROGRAM

V začátcích bude postačující vložit do projektu pouze hlavní soubor s řídicím programem. To provedeme opět prostřednictvím hlavní nabídky Open PCS a položky „New“. Použijeme stejného průvodce jako v předchozím případě. Volbou POU – Program Unit můžeme vybrat jazyk, v kterém bude program zapsán, v našem případě tedy SFC – Sequential Function Chart. Dále zvolíme „POU Type“ jako „Program“. Tvorbu hlavního programu dokončíme zadáním jména souboru a místa jeho uložení v položkách

„Name“ a „Location“. Stiskneme tlačítko „OK“. Vývojové prostředí se zeptá, zdali chceme nový soubor zahrnout do projektu, a to otázkou „Do you like add jmeno programu to active resource ?“. Stiskneme tlačítko Ano, popř. OK.

Nyní se otevřelo okno rozdělené na tři části. Horní část slouží k zápisu a deklaraci proměnných řídicího programu. Prostřední část slouží ke grafické reprezentaci funkční struktury řídicího programu. Dolní část slouží k zápisu kódu programu.

PROMĚNNÉ

Okno proměnných obsahuje předdefinované sekce. Sekce „VAR_EXTERNAL“ slouží k deklaraci externích proměnných, tj. proměnných uložených v jiném kódu. Do sekce „VAR_GLOBAL“ se deklarují globální proměnné programy. Sekce „VAR“ je určena pro deklaraci lokálních proměnných. Příslušná sekce je pak ukončena zápisem „END_VAR“. Deklarace proměnné Osa může vypadat následovně

```
Osa : Axis_Ref:=(AxisNo:=0);
```

Proměnná Osa byla deklarována jako proměnná typu Axis_Ref – deklarující řízenou osu. Hodnota proměnné Osa je předdefinována jako 0 – Interní osa řídicí jednotky EPOS P. Pro ostatní řízení jednotky EPOS v síti CANBUS bude tato proměnná nabývat hodnot ID přepínače příslušné řídicí jednotky.

V sekci proměnných můžeme dále definovat použití funkčních bloků z knihoven. Pro práci s řídicí jednotkou EPOS P využíváme knihovnu „SMART Motion“.

GRAFICKÝ TVAR PROGRAMU

Část s grafickou reprezentací řídicího programu obsahuje blokové schéma řídicího programu. Umožňuje používat přehledných textových popisků přímo k jednotlivým prvkům schématu, popřípadě ke skupině prvků. Řídicí program se skládá ze smyčky. Uvnitř smyčky jsou jednotlivé grafické bloky, označované jako kroky - steps, ve kterých je zapsán příslušný kód řídicího programu v jazycích ST či IL. Dále

se uvnitř funkčního schématu mohou vyskytovat skoky do jiných částí funkčního schématu - jumps. První krok řídicího programu označí programátor. Po spuštění se řídicí program rozeběhne po jednotlivých krocích se skoky do jiných částí řídicího programu. Vložení dalšího kroku se děje pomocí hlavní nabídky programu OpenPCS položkou „Insert“ a „Step/Transition“. Vložení skoku se provede podobně volbou „Jump“ – položka „Insert“ hlavní nabídky programu OpenPCS.

Část se zápisem kódu jednotlivých kroků obsahuje instrukce specifické pro daný krok, používá se jazyk Structured Text – ST nebo Instruction List – IL. Zápis se provádí kliknutím levého tlačítka myši na příslušný krok v části grafické reprezentace řídicího programu a následným zápisem kódu v části zápisu kódu.

Ostatní se řídí zvyklostmi z používání programu OpenPCS.

ULOŽENÍ DO EPOSU P

Řídicí program je nutno přeložit do strojového kódu. To lze provést pomocí položky hlavní nabídky „PLC“ volbou „Build Active Resource“. Pokud vývojové prostředí OpenPCS v zápisu řídicího programu nenahlásí žádné chybové hlášení, uložíme řídicí program do řídicí jednotky EPOS P. Použijeme položku hlavní nabídky PLC, volíme PC->PLC download. Řídicí program je spuštěn volbou položky „Cold Start“ hlavní nabídky PLC.

PŘÍKLAD

Pro demonstraci řídicího programu v jazyce SFC byl vytvořen program simulující pohon žaluzií. Řídicí program pracuje s řídicí jednotkou EPOS P v režimu Homing Mode. Vytvořený program je možno pro demonstraci stáhnout na webových stránkách firmy Uzimex Praha spol. s r.o. <http://www.uzimex.cz>

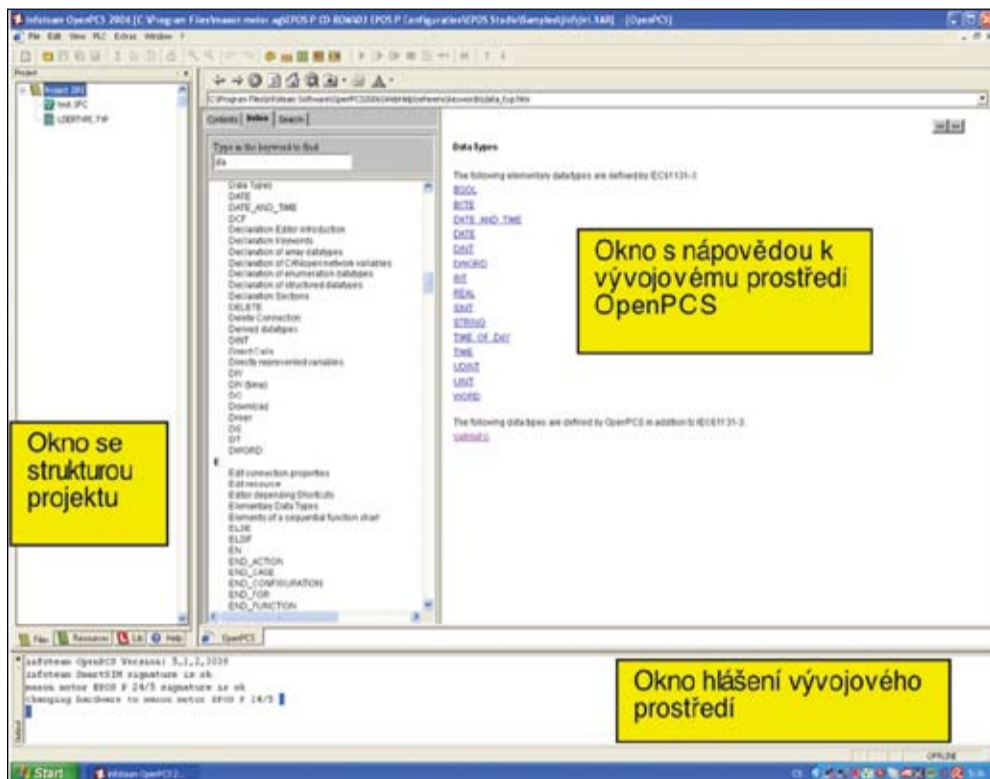
Příští díl seriálu se bude věnovat dalším postřehům při programování řídicích jednotek EPOS P.

ING. JIŘÍ HNÍZDIL
HNIZDIL@UZIMEX.CZ

maxon motor
driven by precision

Programovací postřehy

Řízení EPOS P maxon



V minulých dílech jsme se zabývali popisem programovacích jazyků určených pro tvorbu řídicího programu pro řídicí jednotku EPOS P. Dnešní díl se zabývá několika postřehy potřebnými při psaní řídicích programů.

PROMĚNNÉ

Při tvorbě řídicího programu využíváme proměnných různých datových typů. Z předchozích dílů seriálu víme, že deklarací proměnných se zabývá sekce Var řídicího programu. Proměnné mohou být dle standardu IEC 61131-3 následující:

BOOL - logická proměnná. Může nabývat hodnot true nebo false neboli logická jednička či nula.

BYTE – celočíselná proměnná. Může nabývat hodnot 0 až 255. V paměti řídicí jednotky zabírá 8 bitů.

Date and Time – strukturovaná proměnná obsahující datum a čas. V současné době není tento typ proměnných ve vývojovém prostředí OpenPCS podporován.

Date – strukturovaná proměnná obsahující datum. Není podporována vývojovým prostředím OpenPCS.

DINT – celočíselná proměnná. Nabývá hodnot od -2147483648 do 2147483647. V paměti řídicí jednotky zabírá 32 bitů.

DWORD – celočíselná proměnná. Nabývá hodnot od 0 do 4294967295. Nepracuje se znaménkem a v paměti řídicí jednotky EPOS P zabírá 32 bitů.

INT – celočíselná proměnná. Nabývá hodnot -32 768 až 32 767. V paměti zabírá 16 bitů.

REAL – proměnná pro práci s reálnými čísly. V paměti zabírá 32 bitů.

SINT – celočíselná proměnná nabývající hodnot od -127 do 127. V paměti zabírá 8 bitů.

STRING – proměnná pro práci s textovými řetězci. V paměti řídicí jednotky EPOS P zabírá proměnná místo udané počtem znaků v řetězci násobené osmi. Dále je nutno započítat 16 bitů pro řídicí znaky jako je ukončení řetězce.

Time of Day - Strukturovaná proměnná obsahující čas. V současné době není podporována vývojovým prostředím OpenPCS.

Time – strukturovaná proměnná udávající délku časového úseku, např. dobu trvání procesu. V paměti řídicí jednotky EPOS P zabírá 32 bitů.

UDINT - celočíselná proměnná nabývající hodnot 0 až 4294967295. V paměti řídicí jednotky zabírá 32 bitů.

UINT – celočíselná proměnná nabývající hodnot 0 až 65535. V paměti zabírá 16 bitů.

WORD – celočíselná proměnná identická s předchozí proměnnou UINT.

UCHOVÁNÍ DAT

Řídicí jednotka EPOS P obsahuje po hardwarové stránce energeticky nezávislou paměť o velikosti 1 KB. Do této paměti může programátor zapsat data, která při vypnutí a opětovném zapnutí řídicí jednotky EPOS P neztratí svoji hodnotu. Uložení do této paměti probíhá tak, že sekci proměnných, kterou chceme do energeticky nezávislé paměti uložit, označíme jako RETAIN. Uložení těchto proměnných do příslušné paměti se děje automaticky. Příklad následuje

```
VAR RETAIN
prvni_promenna : SINT :=3;
druha_promenna : DINT;
END_VAR
```

PŘÍMÉ ADRESOVÁNÍ

Pomocí protokolu CANOPEN a objektů PDO lze získat rychlý přístup k jednotlivým parametrům řídicí jednotky EPOS. Jako příklad parametru lze uvést informaci o aktuální poloze hřídele motoru. Při správném nastavení lze tuto hodnotu číst přímo z paměti řídicí jednotky EPOS P. K tomuto účelu slouží pomůcka označená jako Network Configuration. Spustíme ji klikem pravého tlačítka myši na ikonu řídicí jednotky EPOS P v okně Workspace programu EPOS STUDIO a následnou volbou Tools a Network Configuration. Zobrazí se tabulka, ve které definujeme požadovaný název proměnné a parametr, který se do této proměnné bude předávat, např. již zmíněná aktuální poloha hřídele motoru. Po kliknutí na tlačítko OK se uloží nastavení do paměti řídicí jednotky EPOS P a vygeneruje se soubor s deklarací proměnných, který můžeme zahrnout do projektu ve vývojovém prostředí OpenPCS. Při tvorbě řídicího programu již můžeme využívat deklarovanou proměnnou. V našem příkladě bude obsahovat vždy aktuální polohu hřídele motoru, aniž bychom k jejímu zjištění použili funkce z příslušných knihoven. Příklad deklarace proměnných, vygenerované programem EPOS STUDIO, pomůckou Network Configuration.

```
VAR
Poloha_at %0.93
END_VAR
```

NÁPOVĚDA

Zápis jednotlivých příkazů při tvorbě řídicího programu se řídí určitou syntaxí. Tato syntaxe je společně s popisem příkazů, popisem chybových hlášení a dalších dat přehledně popsána v nápovědě k vývojovému prostředí OpenPCS. Nápověda se aktivuje pomocí položky ? hlavní nabídky programu OpenPCS.

V příštím posledním díle se budeme zabývat zpracováním laboratorních setů s řídicími jednotkami EPOS P určených pro výuku na středních a vysokých školách.

ING. JIŘÍ HNÍZDIL
HNIZDIL@UZIMEX.CZ

maxon motor

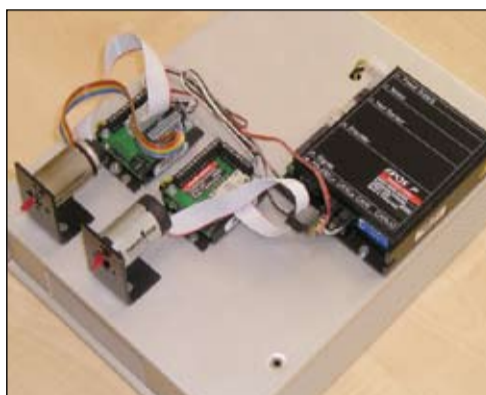
driven by precision

Aktivity ve školách

maxon řízení EPOS P



Pracovníci firmy UZIMEX PRAHA spol. s r.o. v programu pro školy vytvořili výukovou sadu řízení stejnosměrných motorů pomocí řídicích jednotek EPOS a EPOS P. Je vhodná pro střední školy a univerzity technického zaměření se specializací na pohony, kybernetiku, automatizaci, informatiku a slaboproudou elektrotechniku.



LABORATORNÍ SESTAVA EPOS P

Sada řídicích jednotek se obvykle skládá z plně programovatelné řídicí jednotky EPOS P a řídicí jednotky EPOS v provedení 24/1. K těmto jednotkám je připojen stejnosměrný motor s mechanickým komutátorem a stejnosměrný motor s elektronickou komutací. Oba motory disponují magnetorezistivními snímači polohy. Sada na obrázku obsahuje programovatelnou jednotku EPOS P bez motoru a dvě podřízené jednotky EPOS s motory. Komunikace mezi jednotkami probíhá po síti Can Bus. Funkce RS232toCan gateway zajišťuje propojení řídicích jednotek v síti CAN s osobním počítačem prostřednictvím sériové sběrnice RS232. Pro snadné uvedení laboratorní sady do chodu byl vytvořen přehledný návod.

PŘÍNOS LABORATORNÍ SESTAVY

Nespornou výhodou laboratorní sestavy je, že pracuje s malým napětím. Použité motory nejsou schopny vyvinout vysoký moment, při kterém by mohlo dojít k vážnému zranění obsluhy. Studenti si tak mohou osobně vyzkoušet v laboratorním cvičení činnost řídicích jednotek a poznávat vliv změn nastavení na chování motorů. Zadání těchto úloh může být velice rozmanité. První úloha může být uvedení řídicích jednotek do chodu prostřednictvím průvodce prvotním spuštěním. Dalšími úlohami je nastavení regulačních konstant regulátorů proudu, rychlosti a polohy automaticky programem, vliv ručně provedených změn nastavení na stabilitu regulační smyčky nebo vliv změn momentu setrvačnosti zatížení nasazením setrvačnicku na hřídel. Ve složitějších úlohách se využijí digitální vstupy a výstupy, ověří se funkce elektronické hřídele a studenti vytvoří vlastní program.

PROGRAM

Při tvorbě vlastních programů mohou studenti postupovat dvěma směry. Využít k programování dodaných dll knihoven a napsat program v některém z oblíbených programovacích jazyků Borland Delphi, Borland Builder C++, Microsoft Visual C++, Microsoft Visual Basic či LabView. Takový program nelze uložit do jednotky EPOS P, odpojit počítač a ponechat řízení EPOS P. Osobní počítač musí být trvale připojen k soustavě pohonů. Řešení není praktické v aplikacích, kde se program týká pohonů a přítomnost osobního počítače pro další funkce není nutná.

Pro provoz pohonů bez počítače lze vytvořit řídicí program v některém z jazyků standardu IEC 61131-3, jako je Structured Text nebo Instruction List a tento řídicí program uložit do paměti řídicí jednotky EPOS P. Řídicí jednotka EPOS P nahrazuje funkci osobního počítače a je schopna komunikovat na sběrnici CAN s ostatními uzly, jako jsou například řídicí jednotky EPOS, průmyslové displeje, klávesnice či snímače fyzikálních veličin. Komunikace řídicí jednotky EPOS P s těmito uzly používá protokol CanOpen.

UŽITEČNOST LABORATORNÍCH SAD

Laboratorní sady s řízením EPOS a EPOS P jsou již umístěny na pracovištích technických univerzit a středních odborných škol v České i Slovenské republice. Studenti se tak mohou

seznámit s vlastnostmi moderního řízení malých pohonů v laboratorních cvičeních elektrických pohonů a řízení. Rozsáhlejší úlohy se řeší formou diplomových, bakalářských a semestrálních prací.

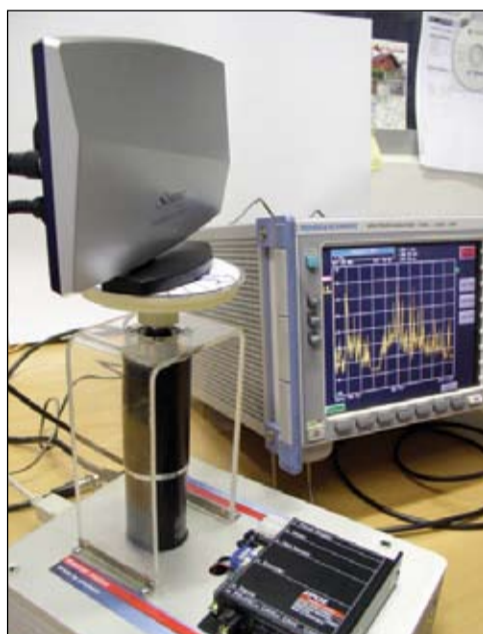
Nabyté znalosti pak mohou absolventi s výhodou uplatnit při zapojení do projektů projekčních a výrobních firem. Firmy mohou využít školní laboratorní pracoviště pro ověření funkce nebo chování jednotek EPOS v projektovaných pohonech. Absolventi budou schopni pohotově řešit malé pohony a regulační úkoly v budoucím zaměstnání.

PRAKTICKÉ APLIKACE

Praktické aplikace řízených pohonů maxon slouží už dnes pro vnitřní potřebu škol nebo představují funkční modely pro průmyslovou výrobu.

ČVUT-FEL V PRAZE

Na ČVUT-FEL v Praze ve spolupráci s firmou Rohde & Schwarz Praha s.r.o. a UZIMEX PRAHA, spol. s r.o. bylo realizováno modelové laboratorní pracoviště pro automatizaci širokopásmových měření anténních diagramů programem Span_FSx. Anténu natáčí pohon maxon s elektronicky komutovaným motorem

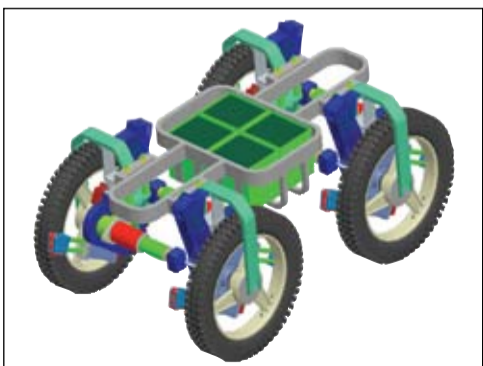


Nastavování měřené antény

EC, s inkrementálním snímačem polohy a planetovou převodovkou. Řídicí jednotka polohy EPOS řídí motor podle programu svázaného s měřením spektrální křivky. Poloha anténního systému se nastaví s přesností na 0.1°. Po nastavení a odměření kmitočtového spektra v požadované šíři se data předávají do počítače, kde se zobrazí v 2D grafu s barevným kódováním naměřené úrovně signálu.

VŠB OSTRAVA

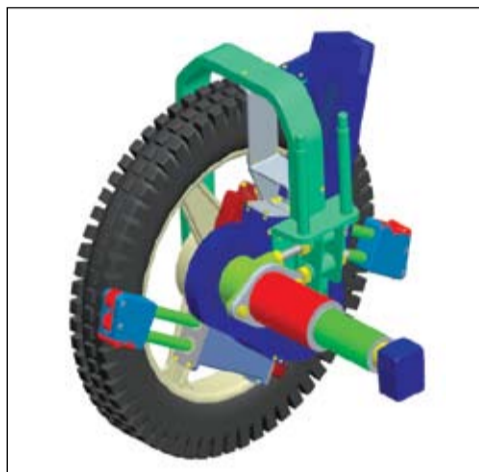
Na katedře robototechniky VŠB Ostrava je dále vyvíjen podvozek servisního zásahového robotu. Účelem robotu je monitorování budov při chemickém nebezpečí, odběr vzorků ply-



Podvozek servisního zásahového robotu

nů kapalin, stírání povrchu a další činnosti. Z požadovaných vlastností lze uvést: průchodnost dveřmi 700 mm, schopnost zdolávání překážek cca 200 mm, schopnost jízdy po schodišti. Tyto požadavky se odrazily v koncepci pohonného subsystému robotu, kde jsou použity čtyři 400wattové motory MAXON EC60 s převodovkami GP81 93:1 a inkrementálními

snímači HEDL9140 s 500 pulzy/otáčku. Motory jsou řízené jednotkami EPOS 70/10. Další čtyři menší motory MAXON včetně jednotek EPOS slouží jako pohonné jednotky přídatného mechanismu každého kola pro překonávání schodišť.



Pohony kola a mechanismu pro schodiště

VŠB OSTRAVA

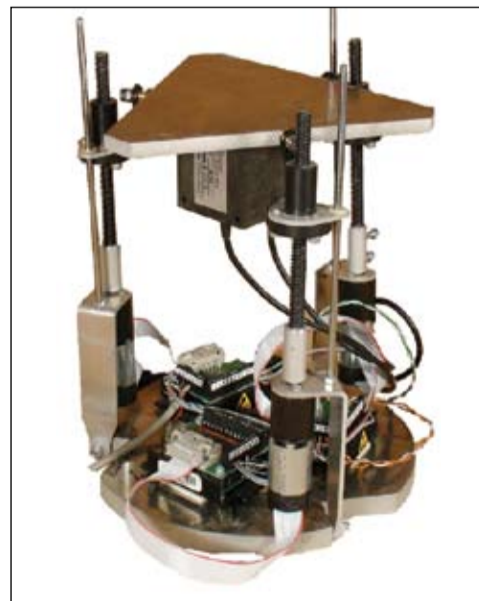
Na katedře robototechniky VŠB-TU Ostrava byl v tomto roce v rámci diplomové práce realizován model stabilizační plošiny.

Plošina je určena pro stabilizaci kamerového subsystému mobilního robotu. Obraz se bezdrátově přenáší do řídicího systému operátora. Po doplnění dalších textových a grafických informací se zobrazí v 3D helmy operátora.

Jednotlivé varianty kinematických struktur byly nejprve namodelovány v programu Pro/Engineer a ověřovány schopnosti pohybu a blokování. Vzhledem k tomu, že paralelní struktury jsou redundantní, nelze jejich chování ověřit klasickými metodami pomocí odečítání stupňů volnosti.

Pohonný subsystém vybrané varianty je ovládan trojicí stejnosměrných motorů maxon A max19 2,5W s planetovými převodovkami GP 19B a s inkrementálními snímači. První řídicí jednotka EPOS 24/1 je připojena k PC pomocí sběrnice RS-232 a tvoří bránu na sběrnici CAN s připojenými zbývajícími řídicími jednotkami EPOS 24/1. Dvojice piezorezistivních akcelerometrů je upevněna na spodní části stabilizované plošiny. Výstupní signály akcelerometrů o náklonu plošiny se přivádějí na dva analogové vstupy jedné z řídicích jednotek.

Vlastní algoritmus pro řízení jednotky byl odladěn v programu Visual Basic.net 2005 a využívá dodanou knihovnu EposCmd.DLL. Program napřed nastaví pohony do výchozí polohy najetím na pevný doraz a indikací nárustu proudu. V současné době jsou ověřovány možnosti provozu bez počítače, jednak pomocí knihovny pro mikrokontroléry řady x51/52, jednak přechodem na jednotku EPOS P.



Realizovaný model stabilizační plošiny

TU V LIBERCI

Řízení otáčení clony světlíku v nové moderní budově Informačního centra Technické univerzity v Liberci.

Místnost je osvětlována jednak umělým osvětlením, jednak slunečním světlem, které shora prochází světlíkem. Optimální osvětlení za dne se dosahuje nastavováním odrazné clony ve světlíku podle polohy slunce a intenzity jeho záření. Otáčení clony zajišťuje pohon maxon. Motor RE40 s převodovkou GP52C a inkrementálním snímačem MR je řízen pomocí řídicí



Informační centrum TU v Liberci

jednotky EPOS P. Systém budovy vyhodnocuje informaci o poloze Slunce a údaj posílá do jednotky EPOS P.

Pohon byl navržen tak, aby mohl pracovat v automatickém i manuálním režimu. Manuální režim umožňuje přes dálkové ovládání natočit clonu podle požadavků osob v místnosti.



Regulace osvětlení clonou světlíku

JAK ZAČÍT

Pokud Vás zaujala nabídka sady pro školy a chtěli byste ji nasadit ve výuce, můžete pro další informace kontaktovat některou z poboček firmy UZIMEX PRAHA spol. s r.o. v Praze, Brně nebo v Liberci. Firma UZIMEX PRAHA, spol. s r.o. dodává komponenty laboratorních sad pro vzdělávání za výhodné ceny. Kontaktní informace jednotlivých poboček získáte na internetu na adrese www.uzimex.cz

ING. JIŘÍ HNÍZDIL, HNIZDIL@UZIMEX.CZ
ING. VÁCLAV BROŽ, BROZ@UZIMEX.CZ

maxon motor
driven by precision