

Laboratorní model obráběcího stroje TRIJOINT

Bc. Lukáš Řápek

Vedoucí práce: Ing. Jan Zavřel, Ph.D.

Abstrakt

Podle skutečného obráběcího centra TRIJOINT 900H byl zkonstruován laboratorní model, který je řízen elektronikou od firmy B&R. Pro jeho řízení bylo třeba navrhnout řídicí elektroniku a ovládací prvky. Pro naprogramování řídicí elektroniky bylo třeba řešit inverzní i dopřednou kinematiku mechanismu a také naprogramovat řídicí algoritmus včetně návrhu regulátorů.

Klíčová slova

Trijoint 900H, inverzní kinematika, B&R Automation Studio, řízení, laboratorní model.

1 Úvod

Na základě kinematiky skutečného obráběcího stroje TRIJOINT 900H byl již dříve postaven laboratorní model, u kterého nyní došlo k drobné strukturální modifikaci a zprovoznění řídicím systémem. Laboratorní model je postaven v měřítku 0,43:1¹. Obráběcí centrum TRIJOINT 900H bylo vyvíjeno ve spolupráci Odboru mechaniky a mechatroniky, Fakulty strojní na ČVUT v Praze a firmy KOVOSVIT MAS, a.s. Stroj TRIJOINT 900H je horizontální obráběcí centrum, kde je použito paralelního mechanismu, který řídí pracovní pohyb vřetene. Tato konfigurace tak umožnila zvýšení tuhosti a dynamiky stroje.

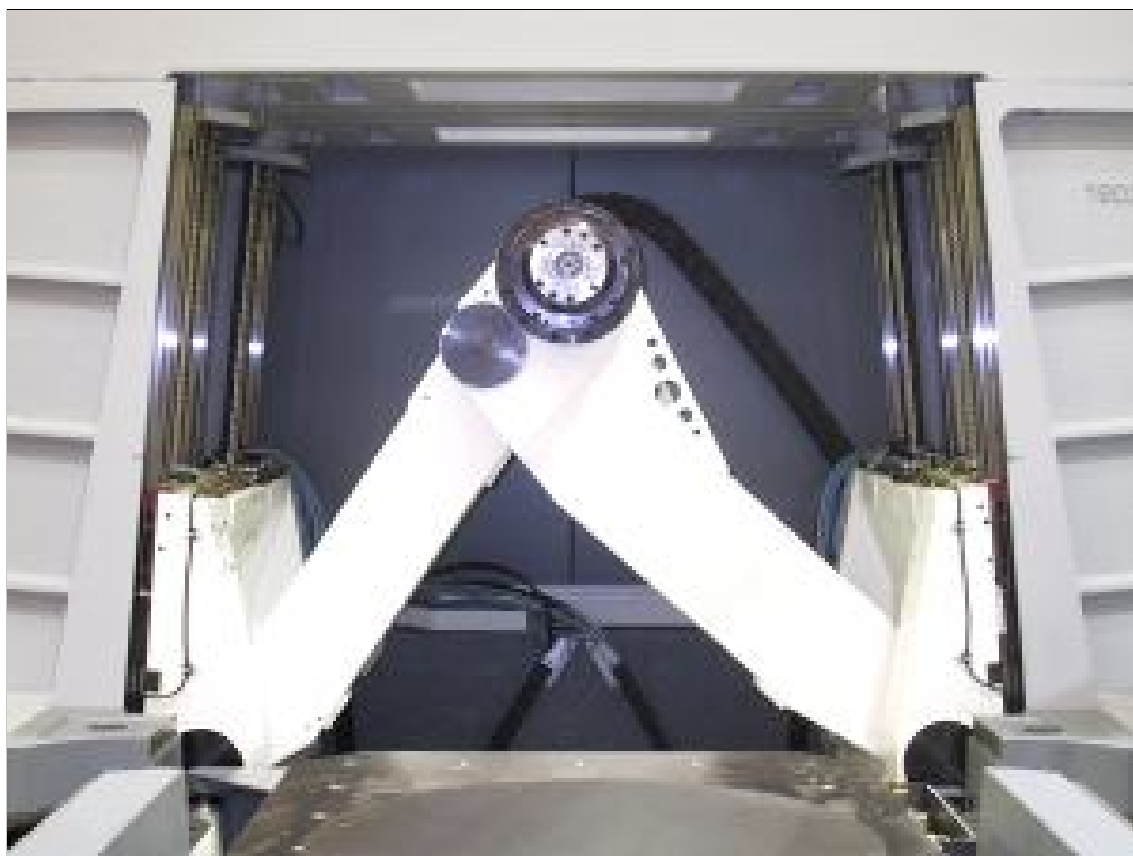


Obr. 1: Obráběcí centrum Trijoint 900H.

2 Popis laboratorního modelu

Mechanismus laboratorního modelu se skládá z rámu, dvou ramen, vedení, pohybových šroubů, pohonů, řídicího systému, ovládacího zařízení a napájení. Poloha vřetene je dána

¹Toto měřítko modelu bylo voleno a vypočteno na základě délky pohybových šroubů, které byly k dispozici při jeho stavbě.



Obr. 2: Detail kinematiky stroje.

pohybem dvou ramen, z nichž každé má na rámu stroje upevněn vlastní pohon. Pohony tak nezvyšují hmotnost pohyblivých částí, což je výraznou výhodou paralelní kinematiky.



Obr. 3: Laboratorní model obráběcího centra Trijoint 900H.

Pohony jsou propojeny přes planetové převodovky integrované v pohonu, s vedením pomocí Oldhamových spojek. Na vedení jsou umístěny pojezdy, které pohybují ramena.

Laboratorní model je sestaven z hliníkových profilů upravených na potřebné rozměry.

3 Elektronika použitá pro řízení laboratorního modelu

Mechanismus je poháněn dvojicí elektromotorů od firmy MAXON, které jsou pevně spojeny s pohybovými šrouby a tím pohybují vřetenem (v kinematice označeno bodem M). Snímání polohy je zajištěno pomocí inkrementálních snímačů, které jsou součástí motorů MAXON. Jejich nevýhodou je však, že pracují s logikou na úrovni 5 V, avšak řídicí systém je založen na napětí 24 V, je tedy třeba použít převodník napětí signálu z 5 V na 24 V. Další výraznou nevýhodou při snímání polohy je malý rozsah modulu čítače, ten je 16bitový a je schopen snímat pouze rozsah jednoho otočení rotoru, to je však možné softwarově ošetřit, jak je názorné popsáno v kapitole „Programování“.

Rozsah pohybu pojezdů po vedení je omezen koncovými snímači, které zabraňují nárazu pojezdů do rámu stroje při nesprávné manipulaci s řízením stroje. Tyto snímače jsou také využity při referencování v inicializační části programu. Pro řízení laboratorního modelu lze použít mnoho různých variant řídicích systémů od světoznámých výrobců HW. Např.:

- National Instruments
- B&R Automation
- D-Space
- Siemens

S ohledem na požadavky, jako je nízká pořizovací cena, možnost zahrnout do řídicího systému i programy z MATLABu, či Simulinku, moduly I/O, PWM, inkrementální čítač, byl zvolen řídicí systém od firmy B&R Automation.

3.1 B&R Automation

Průmyslová automatizace, která má výhodu ve velkém množství použitelných modulů, které jsou volně konfigurovatelné. Nevýhodou této volby však je, že tato průmyslová automatizace pracuje na 24V logice, avšak většina čítačů (zejména inkrementálních) pracuje na nižší úrovni napětí [2].

3.2 Použité moduly

Pro řízení laboratorního modelu, bylo třeba použít tyto moduly:

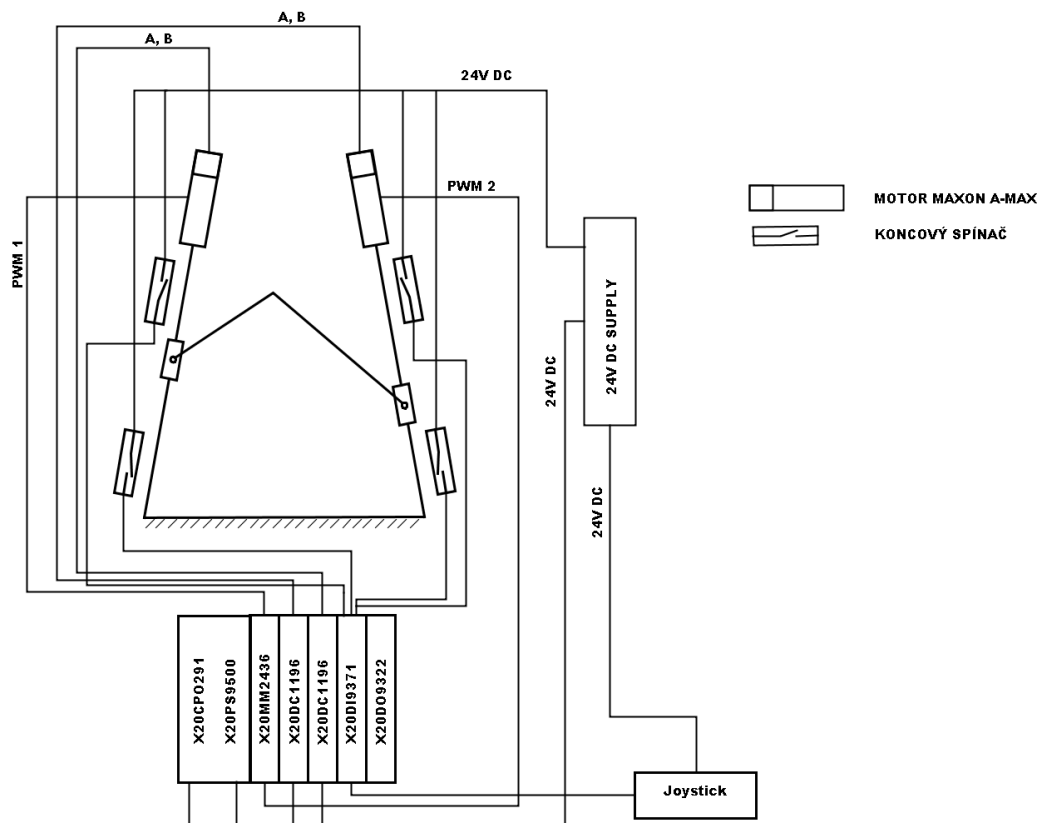
- **X20CP0291** – modul obsahující CPU od firmy Motorola taktované na frekvenci s vlastní pamětí o velikosti 3MB
- **X20PS9500** – napájecí modul pro X20CP0291, ale lze jím napájet i ostatní moduly
- **X20MM2436** – modul umožňující ovládání 2 motorů pomocí pulzně šířkové modulace (PWM), je třeba jej samostatně napájet

- **X20DC1196** – 16-bitový inkrementální čítač a převodník 5/24V DC
- **X20DI9371** – modul pro zpracování vstupních signálů, jako jsou například vstupy z ovládání, signály koncových spínačů atp.
- **X20DO9322** – modul pro digitální výstupy, např. signalizace daného stavu.

3.3 Napájecí a ovládací zařízení

Napájení je zajištěno pomocí svorkovnice, na kterou je z řízeného zdroje přivedeno napětí 24V DC, které je dle schématu zapojení dále rozváděno do systému. Schéma zapojení je blokově naznačeno na obrázku 4.

Ovládacím prvkem pro tento mechanismus byl zvolen joystick, který umožňuje pohodlné řízení pohybu vřetene.



Obr. 4: Blokové schéma zapojení elektroniky.

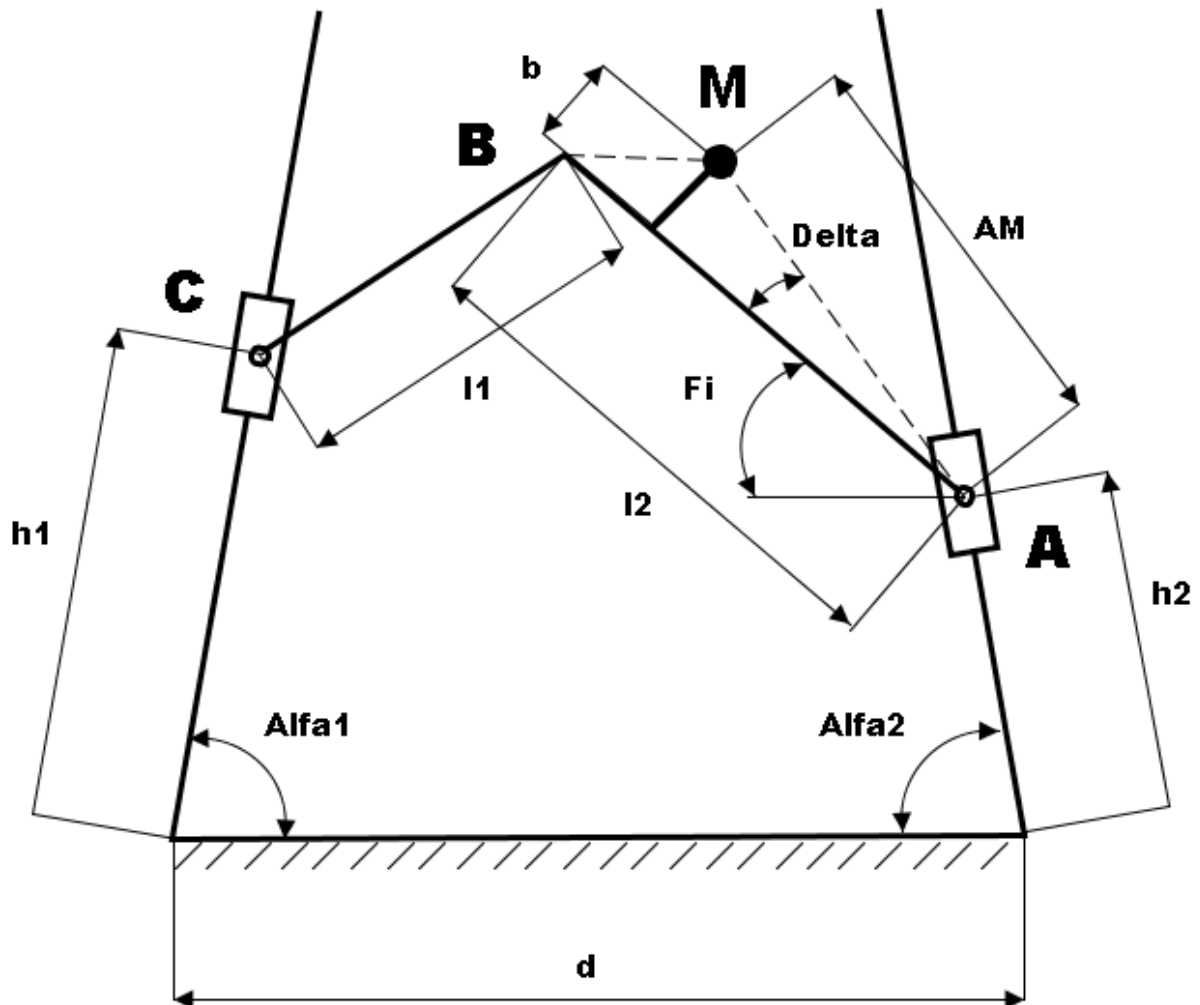
4 Kinematika

Pro popis kinematiky laboratorního modelu lze použít různé přístupy k řešení, např.:

- Lagrangeovy rovnice

- Vektorová metoda
- Maticová metoda
- Trigonometrická metoda – analytické řešení

Pro vyjádření přímé i inverzní kinematiky mechanismu byla zvolena analytická metoda a to pro její numerickou nenáročnost a snadnou programovatelnost [3].



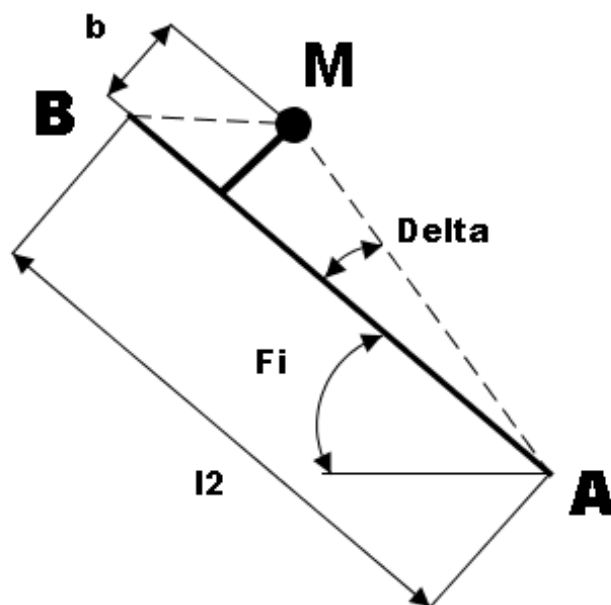
Obr. 5: Kinematické schéma mechanismu.

Pro řízení je nutné vyjádřit inverzní kinematiku mechanismu, ze které jsou získány hodnoty pojezdů h_1 a h_2 , tedy hodnoty potřebné pro požadovanou polohu bodu M tj. x_M a y_M . Pro řešení inverzní kinematiky bylo použito schéma, které je na Obr. 5.

4.1 Vyjádření bodu A

$$x_A = d - h_2 \cos \alpha_2 \quad (1)$$

$$y_A = h_2 \sin \alpha_2 \quad (2)$$



Obr. 6: Kinematické schéma pro vyjádření bodu B.

$$(x_A - x_M)^2 + (y_M - y_A)^2 = AM^2 \quad (3)$$

$$x_A^2 - 2x_Ax_M + x_M^2 + y_M^2 - 2y_M[k_Ax_A + q_A] + [k_Ax_A + q_A]^2 = AM^2 \quad (4)$$

$$x_A^2 - 2x_Ax_M + x_M^2 + y_M^2 - 2y_Mk_Ax_A - 2y_Mq_A + k_A^2x_A^2 + 2k_Ax_Aq_A + q_A^2 = AM^2 \quad (5)$$

$$x_A^2(1 + k_A^2) + x_A(2k_Aq_A - 2x_M - 2y_Mk_A) + [q_A^2 + x_M^2 + y_M^2 - 2y_Mq_A - AM^2] = 0 \quad (6)$$

Z rovnice (6) jsou získána 2 řešení x_{A1} , x_{A2} , z těchto dvou výsledků je vybrán větší a označen jako x_A . Z vypočtené proměnné x_A se pak již snadno dopočítá h_2 , jak je vyjádřeno v rovnici (7).

$$h_2 = \frac{d - x_A}{\cos \alpha_2} \quad (7)$$

4.2 Vyjádření bodu B

Pro vyjádření bodu B je použito Obr. 6

$$a^2 + b^2 = c^2 \quad (8)$$

$$AM^2 = (l_2 - a)^2 + b^2 \quad (9)$$

$$c^2 = l_2^2 + AM^2 - 2l_2AM \cos \delta_2 \quad (10)$$

$$\cos \delta_2 = \frac{-c^2 + l_2^2 + AM^2}{2l_2AM} \quad (11)$$

$$\tan \varphi_2 + \delta_2 = \frac{y_M - y_A}{x_A - x_M} \quad (12)$$

$$\varphi_2 = \arctan \left(\frac{y_M - y_A}{x_A - x_M} \right) - \delta_2 \quad (13)$$

Z rovnice (13) bylo získáno φ_2 , které bylo použito v rovnicích (14) a (15)

$$x_B = x_A - l_2 \cos \varphi_2 \quad (14)$$

$$y_B = y_A - l_2 \sin \varphi_2 \quad (15)$$

Z rovnic (14) a (15) je získána úplná poloha bodu B.

4.3 Vyjádření bodu C

Bod C (opět průnik kružnice a přímky)

$$y_C = \tan \alpha_1 x_C + q_C = k_C x_C + q_C \quad (16)$$

$$(x_B - x_C)^2 + (y_B - y_C)^2 = CB^2 = l_1^2 \quad (17)$$

$$x_B^2 - 2x_B x_C + x_C^2 + y_B^2 - 2y_B y_C + y_C^2 - l_1^2 = 0 \quad (18)$$

$$x_B^2 - 2x_B x_C + x_C^2 + y_B^2 - 2y_B [k_C x_C + q_C] + k_C^2 x_C^2 - 2k_C x_C q_C + q_C^2 - l_1^2 = 0 \quad (19)$$

$$x_C^2 (1 + k_C^2) + x_C (-2x_B - 2k_C y_B - 2k_C q_C) + (x_B^2 + y_B^2 + q_C^2 - l_1^2 - 2y_B q_C) = 0 \quad (20)$$

Z rovnice (20) z dvě možná řešení x_{C1}, x_{C2} . Ze znalosti konfigurace stroje, je vybrán menší z obou výsledků x_C . Ze souřadnic bodu C je poté dopočtena hodnota h_1

$$h_1 = \frac{x_C}{\sin \alpha_1} \quad (21)$$

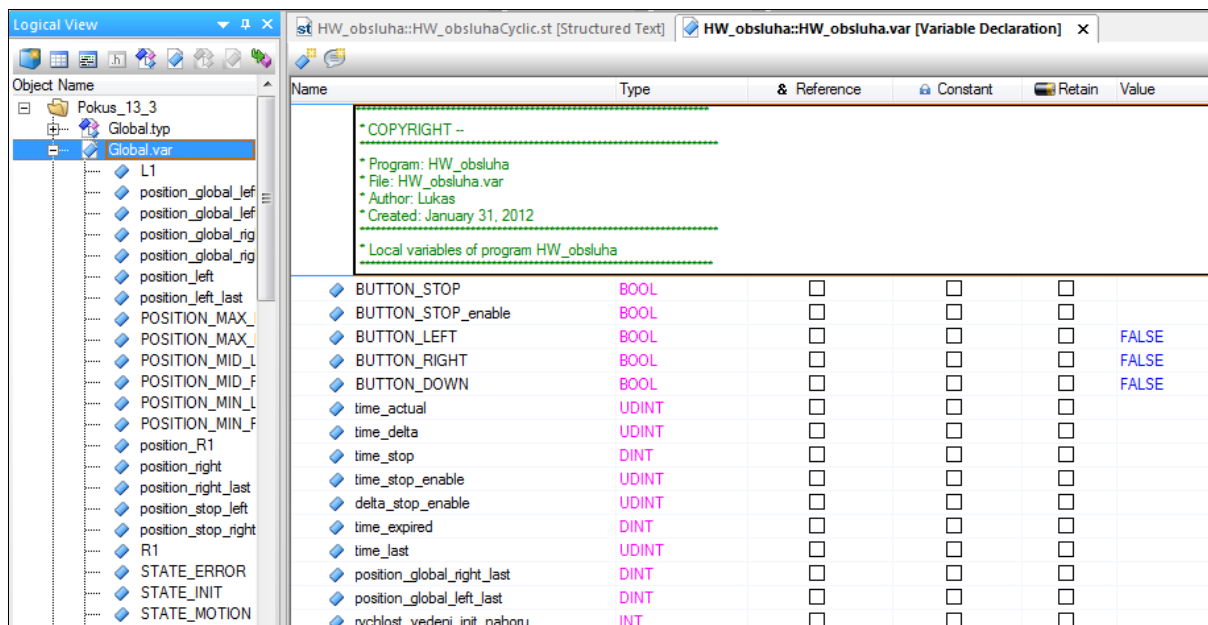
5 Řízení

Pro řízení bylo použito PI regulátoru, proto je třeba mechanismus popsat přímou a inverzní kinematikou mechanismu. Z popisu přímé kinematiky je získána okamžitá poloha bodu M a z požadované polohy bodu M je posléze počítána inverzní kinematikou hodnota h_1 a h_2 pozice pojezdů.

Pro každý pohon je použit jeden PI regulátor [4].

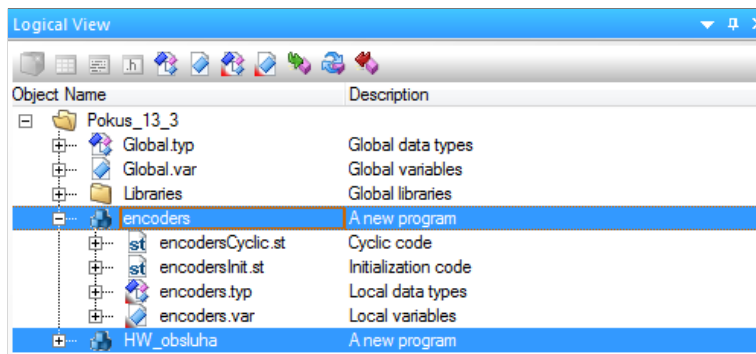
6 Programování

Společnost B&R Automation dodává ke svému HW i vývojové prostředí pod názvem Automation Studio. To umožňuje komunikaci s HW, programování v integrovaném editoru, softwarové simulace programu i HIL simulace. V editoru Automation Studia je možné programovat v řadě různých programovacích jazyků. Pro programování řídicího systému je použito strukturovaného textu. Ten má podobné syntaxe, jako programovací jazyk C. Stejně jako v jazyce C je třeba i ve strukturovaném textu třeba definovat datový typ každé proměnné.



Obr. 7: Ukázka definice proměnných v programu Automation Studio.

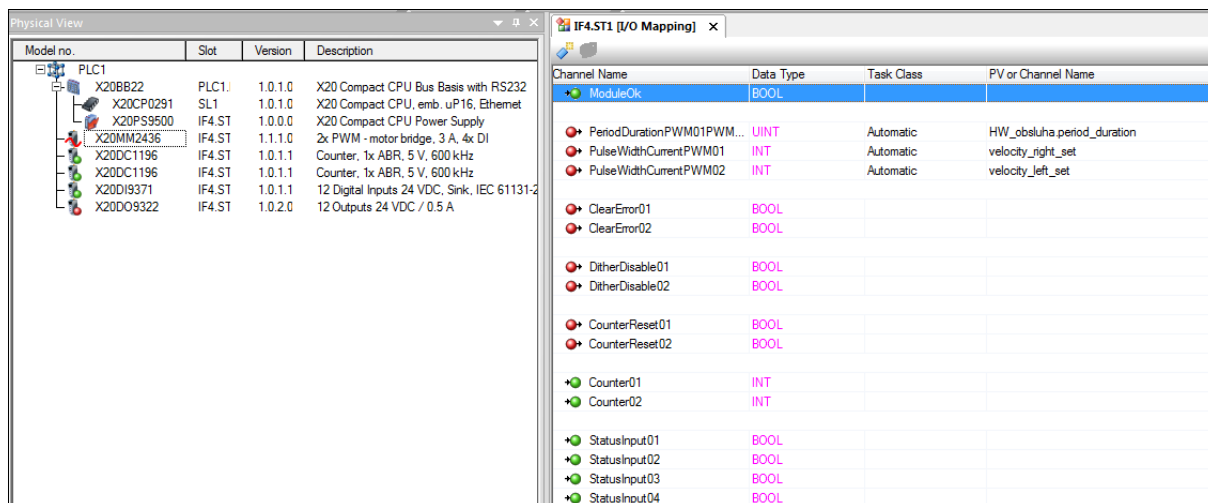
Seznam proměnných lze definovat přímo lokálně pro určitou část programu anebo v seznamu globálních proměnných, kde jsou viditelné pro všechny části programu. Celý program je rozdělen do dvou podprogramů, první z podprogramů je určený k ošetření snímání inkrementálními snímači, druhý je pak hlavní program, určený k ovládání laboratorního modelu. Každý z těchto programů běží samostatně na různém časovém kroku procesoru.



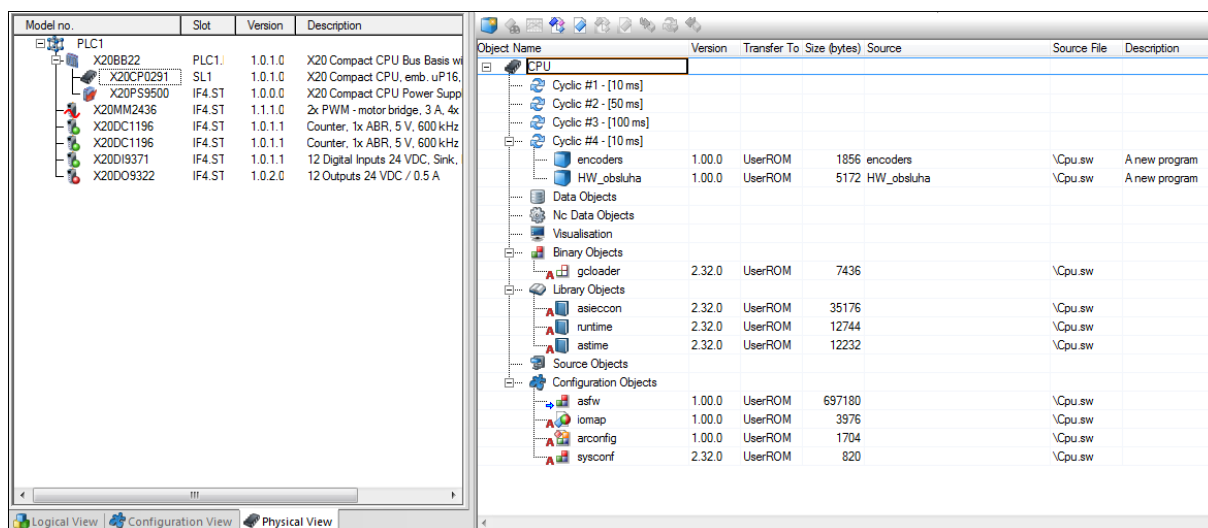
Obr. 8: Struktura dělení programu.

Další důležitou částí programu je propojení tzv. vrstvy softwarové a hardwarové, tj. přiřazení hodnot snímaných případně vysílaných HW k SW proměnným použitým v programu.

Automation Studio umožňuje programátorovi uložit napsaný kód do složek (smyček), jež se vykonávají s určitou pevně daným časovým krokem. Různé časové kroky umožňují vykonávat naprogramované operace, podle přání programátora, rychleji, či pomaleji, podle zvolené složky a její periody opakování. Typickým příkladem jsou snímače či regulátor, jejichž programy je třeba vyhodnocovat rychleji než samotný program pracující s ovládáním, které není třeba tak často vykonávat.



Obr. 9: Ukázka hardwarové vrstvy.



Obr. 10: Ukázka programovacích smyček.

6.1 Snímače

Tento program je vytvořen pro snímání dat z inkrementálních snímačů a výpočtu absolutní pozici pojezdu. Jak již bylo uvedeno dříve. Inkrementální snímače snímají jen v rozsahu jednoho otočení. Po otočení se snímače opět vynulují a snímají opět nové otočení. V inkrementálních snímačích není použit signál, který by zaznamenával přetočení o více než jednu otáčku, proto je třeba tento problém řešit softwarově.

Myšlenka řešení je založena na tom, že inkrementální čítač měří v rozsahu 2^{16} (datový typ int se znaménkem, což znamená, že rozsah je od -32768 do 32767) tj. 16-bitový čítač. Při přetečení, či podtečení tedy dohází ke změně znaménka. Vzhledem k tomu, že při měření může dojít k překmitnutí, způsobeného rychlou změnou (vysoké otáčky), o víc než jeden inkrement, je navoleno pásmo. To umožňuje tomuto případnému problému předejít.

6.2 Hlavní program

Hlavní program je realizován formou stavového automatu, jež obsahuje tyto stavy.

- **REBOOT** – stav, který nastává po chybovém stavu, systém se inicializuje
- **INIT** – systém získává údaje o maximální, minimální a absolutní poloze
- **READY** – systém je připraven
- **MOTION** – systém je řízen ovládacím panelem obsluhou
- **ERROR** – chybový stav

Než však může začít samotný řízený pohyb, je třeba zjistit maximální a minimální dosažitelnou polohu. To je třeba zjišťovat při každém zapnutí modelu, protože systém po vypnutí a opětovném zapnutí nezná svoji aktuální polohu. Z tohoto důvodu je v automatické struktuře programu zařazen jako první inicializační stav.

V případě, že nastane zakázaný stav, nebo je třeba zastavit pohyb, (stiskem červeného tlačítka) systém přeruší pohyb mechanismu. Poté je tlačítko, které umožní nové spuštění, na nastavený počet vteřin neaktivní. Po uběhnutí této doby je tlačítko opět aktivní a jeho stiskem docílíme znovu nové zprovoznění systému.

6.3 Ukázka programu

Hlavní program pro řízení se stará o přepínání mezi jednotlivými stavy a jsou zde prováděny jednotlivé subprogramy pro řízení modelu. Tato část programu zajišťuje také kontrolu tlačítka STOP, které umožňuje ukončení pohybu a zároveň přechod ze stavu ERROR do stavu INIT. Dále tato část programu kontroluje stav všech koncových spínačů a v případě sepnutí, některého z nich opět přejde program do stavu ERROR. Tlačítko STOP je naprogramováno tak, že pokud dojde k jeho zmáčknutí, je jeho funkce dočasně pozastavena po dobu definovanou na 500 ms, aby nemohlo dojít k nechtěnému dalšímu zmáčknutí. Po uplynutí této doby je tlačítko opět funkční.

```

Program: HW_obsuha
File: HW_obsuhaCyclic.st
Author: Lukas
Created: January 31, 2012
.....
Implementation of program HW_obsuha
.....

OGRAM_CYCLIC
zjisteni casoveho kroku programu
time_actual:=TIME_TO_UDINT(GetTime()); (* ms *)
time_delta:=time_actual-time_last;

//kontrola STOP tlacitka
IF (BUTTON_STOP=TRUE) AND (stav<>STATE_ERROR) AND (stav<>STATE_REBOOT) AND (BUTTON_STOP_enable=TRUE) THEN
  stav:=STATE_ERROR;
  time_stop:=TIME_TO_UDINT(GetTime()); (* ms *)
  BUTTON_STOP_enable:=FALSE;
END_IF;

//kontrola vseh 4 koncaku => STATE_ERROR
IF ((stav=STATE_READY) OR (stav=STATE_MOTION)) AND ((SWITCH_LEFT_UPPER=TRUE) OR (SWITCH_RIGHT_UPPER=TRUE) OR (SWITCH_RIGHT_DOWN=TRUE) OR (SWITCH_LEFT_DOWN=TRUE))
  THEN stav:=STATE_ERROR;
  time_stop:=TIME_TO_UDINT(GetTime()); (* ms *)
  BUTTON_STOP_enable:=FALSE;
END_IF;

//po 500 ms od REBOOTu od STOPu lze opet pouzit STOP tlacitko
delta_stop_enable:=(TIME_TO_UDINT(GetTime())-time_stop_enable);
IF (BUTTON_STOP_enable=FALSE) AND ((TIME_TO_UDINT(GetTime())-time_stop_enable)>5000) AND (stav<>STATE_ERROR) THEN      (* čekání na zprovoznění tlačítka *)
  BUTTON_STOP_enable:=TRUE;

```

Obr. 11: Ukázka části hlavního programu v jazyce B&R – „Strukturovaný text“.

7 Závěr

Model obráběcího stroje TRIJOINT 900H byl modifikován a zprovozněn na průmyslové automatizaci od firmy B&R Automation. K pohonu bylo použito pohonů od firmy MAXON. V regulaci polohy mechanismu bylo použito PI regulátorů. Laboratorní model lze ovládat joystickem.

Literatura

- [1] Kopernický V.: Stroje. STNL, Praha, 1989.
- [2] http://www.perfection-in-automation.net/cps/rde/xchg/br-automation.com/hs.xsl/company_CS_HTML.htm [online] , 15.3.2012
- [3] Stejskal V., Brousil J., Stejskal S.: Mechanika III. Vydavatelství ČVUT v Praze, 1997.
- [4] Valášek M. a kol.: Mechatronika. Vydavatelství ČVUT v Praze, 1996.