

5

5. STRUKTURA PLC PROGRAMU

Struktura PLC programu je navržena s ohledem na co nejefektivnější návrh programu při přizpůsobení CNC systému na stroj.

5.1 Pomocné příkazy

instrukce	DATA DATA_END (START) STOP	
funkce	DATA DATA_END (START) STOP	začátek datové oblasti konec datové oblasti a začátek programu konec programu
syntax	DATA DATA_END (START) STOP	

Každý program zapsaný v jazyce PLC836 musí povinně začínat příkazem **DATA**. Program zapsaný v jazyce PLC836 musí vždy končit příkazem **STOP**. Použití instrukcí **DATA**, **DATA_END** a **STOP** je blíže popsáno v následující kapitole.

Modul **DATA** může být použitý ve všech souborech PLC programu.

5.2 Moduly jazyka PLC836

Moduly jazyka PLC836 byly vytvořeny pro zjednodušení práce při návrhu PLC programu. Zjednodušení nastává ze dvou různých pohledů na tvorbu PLC programu. V první řadě se jedná o zjednodušení navázání a synchronizace PLC automatu na CNC systém. Například modul přípravných funkcí (popsáno dále) se nashutuje jen po odstartování bloku, má vlastnosti jako sekvenční logický celek (mechanismus) a prochází jednorázově. To znamená, že všechny příkazy, které budou v tomto modulu umístěny se vykonají automaticky po startu bloku v přípravných funkcích. Další přínos spočívá v strukturalizaci PLC programu, jak už bylo popsáno dříve. Modul přípravných a závěrečných funkcí slouží jako aktivační modul jednotlivých mechanismů, které řeší dílčí procesy stroje.

Při zápisu programu programovatelného rozhraní je třeba dodržovat určitá pravidla a doporučení. Struktura programu je pevně stanovena a programátor ji musí dodržet. Program musí začínat klíčovým slovem **DATA**, za kterým programátor definuje použité proměnné a definuje jejich délku.

Začátek programového kódu je uveden klíčovým slovem **DATA_END (START)**

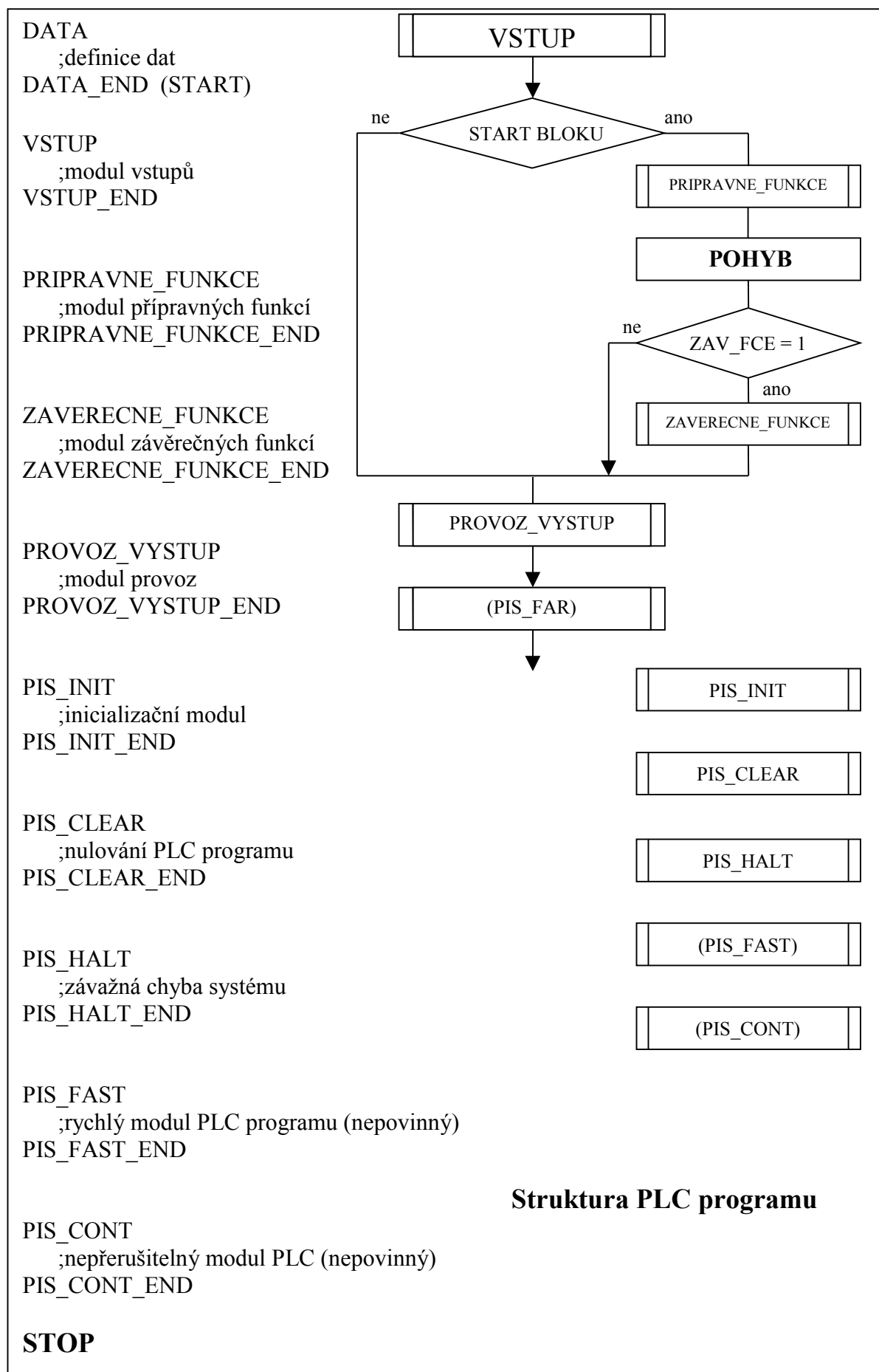
Za klíčovým slovem **DATA_END** následují další moduly, uvedené na obr. Názvy modulů jsou povinné. Na jejich pořadí nezáleží, doporučuje se však zachovat pořadí uvedené na obrázku. Není nutné využít všechny uvedené moduly, ale v případě, že modul není využit, musí být uveden alespoň jeho název (např. ZAVERECNE_FUNKCE) a ukončení (např. ZAVERECNE_FUNKCE_END).

Za posledním modulem musí být uvedeno klíčové slovo **STOP**, které ukončuje program interfejsu.

Na obrázku je uvedena struktura minimální verze programu, který se bezchybně přeloží překladačem TECHNOL. Tento program samozřejmě nevykonává žádné funkce PLC programu. Takovou prázdnou strukturu PLC programu možno nazvat *nulový PLC program*. Všechny důležité proměnné v rozhraní PLC-CNC systém jsou přednastaveny tak, že CNC systém i s nulovým PLC programem může jezdit a vykonávat všechny funkce, které nesouvisí s technologií stroje. Návrh nového PLC programu je proto vhodné postupně vytvářet ve struktuře nulového PLC programu.

Program PLC programu prochází modulem VSTUP. Za modulem MODULE_INPUT (VSTUP) se program větví v závislosti na činnosti systému. Pokud není odstartován nový blok partprogramu, program neprochází moduly MODULE_BLOCK_INIT (PRIPRAVNE_FUNKCE) a MODULE_BLOCK_DONE (ZAVERECNE_FUNKCE), ale pokračuje modulem MODULE_MAIN (PROVOZ_VYSTUP). Pokud je odstartován nový blok, projde program výše uvedenými moduly, mezi kterými nastane, pokud je programován, také pohyb souřadnic. Moduly MODULE_INIT (PIS_INIT), MODULE_CLEAR (PIS_CLEAR) a MODULE_HALT (PIS_HALT) se vykonají pouze v případě, že jsou vyvolány. V další kapitole je přesnější popis modulů.

Jednotlivé moduly se programují podle funkce, ke které jsou určeny a která vyplývá již z jejich názvu. Jak již bylo uvedeno, nemusí být moduly vůbec naplněny. V dalším textu jsou uvedeny nejčastější funkce, které se obvykle v daných modulech programují.



5.3 Popis modulů

modul	DATA
--------------	-------------

Modul globálních dat začíná klíčovým slovem **DATA** a končí klíčovým slovem **DATA_END (START)**. Každý soubor PLC programu musí povinně začínat klíčovým slovem **DATA**, za kterým následují deklarace proměnných použitých v PLC programu. Data deklarovaná v tomto modulu mají **globální** charakter, to znamená, že jsou automaticky známá a přístupná ve všech souborech PLC programu. Modul **DATA** může být použit v každém souboru s PLC programem jen jednou a to na samém začátku souboru.

modul	DATA_LOCAL
--------------	-------------------

Modul lokálních dat začíná klíčovým slovem **DATA_LOCAL** a končí klíčovým slovem **DATA_LOCAL_END** (Platí od verze 6.028).

Jedná se o nepovinný modul PLC programu pro deklarování lokálních proměnných. Data deklarovaná v tomto modulu mají **lokální** charakter, to znamená, že jsou známá a přístupná jen v souboru PLC programu, kde se modul vyskytuje. Modul **DATA_LOCAL** může být použit v každém souboru s PLC programem i vícekrát a může být přitom vnořen do jiných modulů, kromě modulu **DATA**. Lokální data se používají i pro definování “automatických” proměnných v rámci rozvoje některých instrukcí jazyka PLC836.

Data definovaná v tomto modulu jsou v této verzi neviditelná i pro ladící program WINTECHNOL. Když je potřeba pro ladění PLC programu zviditelnit lokální proměnné, dočasně přemístíme modul **DATA_LOCAL** do těla modulu **DATA**. Když je modul **DATA_LOCAL** umístěn uvnitř modulu **DATA**, který má globální charakter, zviditelní se lokální data také pro WINTECHNOL.

modul	VSTUP (MODULE_INPUT)
--------------	-----------------------------

Modul začíná klíčovým slovem **VSTUP (MODULE_INPUT)** a končí klíčovým slovem **VSTUP_END (MODULE_INPUT_END)**.

Modul se aktivuje jako první v průběhu každého PLC cyklu a nemá žádná omezení.

V tomto modulu se obvykle provádí čtení vstupních portů do deklarované paměti PLC. Ty vstupy, které přímo ovlivňují blok zpětného hlášení (například limitní a referenční spínače) se přepíší v požadované formě do bloku zpětného hlášení.

modul	PRIPRAVNE_FUNKCE (MODULE_BLOCK_INIT)
--------------	---

Modul začíná klíčovým slovem **PRIPRAVNE_FUNKCE (MODULE_BLOCK_INIT)** a končí klíčovým slovem **PRIPRAVNE_FUNKCE_END (MODULE_BLOCK_INIT_END)**.

Modul PRIPRAVNE_FUNKCE se odstartuje jen po odstartování bloku. Start bloku je při běhu programu v režimech AUT a RUP, ale také v manuálních režimech MAN a JOG a při centrální anulaci. Modul má vlastnosti jako sekvenční logický celek (mechanismus) a prochází jednorázově. Modul může sloužit jako aktivací modul jednotlivých mechanismů, které řeší dílčí procesy stroje.

V modulu se obvykle řeší akce, které jsou typické jako přípravné nebo též počáteční funkce bloků partprogramu, např. roztočení vřetena, zapnutí chlazení nebo uvolnění osy atd. Modul je v činnosti pouze při startu bloku.

Modul přípravných funkcí je logický sekvenční celek a proto se v něm můžou používat všechny **instrukce typu EX**, platné pro sekvenční celky (viz kapitola "Logické sekvenční celky"). Řízení průchodu v modulu přípravných funkcí je popsáno v kapitole "Řízení průchodu supervizorem interfejsu".

Pozn.:

Protože při dekodování přípravných funkcí se také rozpozná výskyt závěrečných funkcí, je nutno v tomto modulu při jejich výskytu nastavit příznak ZAV_FCE, čímž dojde k aktivaci modulu ZAVERECNE_FUNKCE.

modul	ZAVERECNE_FUNKCE (MODULE_BLOCK_DONE)
--------------	---

Modul začíná klíčovým slovem **ZAVERECNE_FUNKCE (MODULE_BLOCK_DONE)** a končí klíčovým slovem **ZAVERECNE_FUNKCE_END (MODULE_BLOCK_DONE_END)**.

V modulu se obvykle řeší akce, které jsou typické jako závěrečné funkce bloků partprogramu, např. stop vřetena, vypnutí chlazení atd. Podmínkou vykonání tohoto modulu je nastavení příznaku ZAV_FCE (viz poznámku u modulu PRIPRAVNE_FUNKCE).

Modul přípravných funkcí je logický sekvenční celek a proto se v něm můžou používat všechny **instrukce typu EX**, platné pro sekvenční celky (viz kapitola "Logické sekvenční celky").

modul	PROVOZ_VYSTUP (MODULE_MAIN)
--------------	------------------------------------

Modul začíná klíčovým slovem **PROVOZ_VYSTUP (MODULE_MAIN)** a končí klíčovým slovem **PROVOZ_VYSTUP_END (MODULE_MAIN_END)**.

Modul se aktivuje v průběhu každého PLC cyklu a nemá žádná omezení.

V modulu se obvykle řeší funkce, které musí být trvale sledovány, např. hlášení od stroje. Je zde vhodné umístit mechanismy. **Instrukce typu EX**, platné pro sekvenční celky (viz kapitola "Logické sekvenční celky") možno použít jen v rámci mechanismů.

Provádí se zde též vysílání výstupů. Do tohoto modulu je možné začlenit PLC programy, využívající změnové signály od systému.

Tento modul může být použitý ve všech souborech PLC programu.

modul	PIS_INIT (MODULE_INIT)
--------------	-------------------------------

Modul začíná klíčovým slovem **PIS_INIT (MODULE_INIT)** a končí klíčovým slovem **PIS_INIT_END (MODULE_INIT_END)**.

Modul slouží např. k inicializaci proměnných interfejsu a jiných akcí, které je nutné provést při zapnutí stroje. Modul tedy není volán trvale, ale pouze po zapnutí stroje. V modulu PIS_INIT je vhodné vyslat všechny výstupy v definovaném tvaru přímo na výstupní porty.

V modulu PIS_INIT musí být inicializace všech mechanismů pomocí instrukce MECH_INIT (viz kapitola "Logické sekvenční celky").

Modul PIS_INIT se neprovede jen v případě, že je nastavena první dekáda strojních konstant 89 na hodnotu 0, což znamená "Stop PLC programu po zapnutí systému". Tento režim se využívá pro ladění PLC programu (viz kapitola "Ladění PLC programu").

Tento modul může být použitý ve všech souborech PLC programu.

modul	PIS_CLEAR (MODULE_CLEAR)
--------------	---------------------------------

Modul začíná klíčovým slovem **PIS_CLEAR (MODULE_CLEAR)** a končí klíčovým slovem **PIS_CLEAR_END (MODULE_CLEAR_END)**.

Modul slouží k nulování proměnných, uvedení interfejsu do výchozího stavu. Modul se spustí pouze na základě stisknutí příslušného tlačítka pro nulování interfejsu.

V modulu PIS_CLEAR musí být inicializace všech mechanismů pomocí instrukce MECH_INIT (viz kapitola "Logické sekvenční celky").

PLC program často využívá rekonfigurovatelné proměnné například na nastavení různých časů. Tyto může obsluha měnit ve strojních konstantách, kde jsou uloženy v BCD tvaru. V PLC programu proto musí být často převod do binárního tvaru. Aby se nemusel tento převod dělat v každém cyklu PLC programu, je vhodné umístit tyto převody v modulu PIS_CLEAR.

Tento modul může být použitý ve všech souborech PLC programu.

modul	PIS_HALT (MODULE_HALT)
--------------	-------------------------------

Modul začíná klíčovým slovem **PIS_HALT (MODULE_HALT)** a končí klíčovým slovem **PIS_HALT_END (MODULE_HALT_END)**.

V modulu se programují činnosti, které se mají vykonat při závažné chybě systému předtím, než systém skončí ve stavu HALT. Doporučuje se zde programovat například nastavení výstupů, které zajistí vypnutí posuvů, případně výstupu, který globálně hlásí poruchu systému (vypnutí silové části stroje). Výstupy je nutno vyslat na výstupní porty.

modul	PIS_FAST (MODULE_FAST)
--------------	-------------------------------

Modul začíná klíčovým slovem **PIS_FAST (MODULE_FAST)** a končí klíčovým slovem **PIS_FAST_END (MODULE_FAST_END)**. Tento modul je nepovinný.

V modulu se programují činnosti, které mají probíhat v rychlejším časovém rastru než 20 ms. Modul **PIS_FAST** je aktivován ve stejných časových intervalech jako softwerová polohová vazba. Rastr pro softwerovou polohovou vazbu se nastavuje v šesté dekádě strojní konstanty 97 (viz Strojní konstanty systému CNC836). Použití modulu **PIS_FAST** předpokládá, že systém CNC836 obsahuje kartu pro rychlé vstupy **IN03** s pořadovým číslem desky vstupů 3 (adresní přepínače: 2=on, 3=off, 4=on). Jedná se o jednotku multiplexovaných vstupů, která v tomto případě nepracuje v multiplexním režimu. Pro sejmутí rychlých vstupů, kterých může být maximálně 8, se používá speciální instrukce **IN_FAST** (viz kapitola "Řízení binárních vstupů a výstupů v kazetě systému"). Aby modul **PIS_FAST** měl smysl, je nutno v něm vyslat požadované rychlé výstupy pomocí standartní instrukce pro výstup **OUTP**. V modulu **PIS_FAST** mohou být naprogramovány logické sekvenční celky.

modul	PIS_CONT (MODULE_CONT)
--------------	-------------------------------

Modul začíná klíčovým slovem **PIS_CONT (MODULE_CONT)** a končí klíčovým slovem **PIS_CONT_END (MODULE_CONT_END)**. Tento modul je nepovinný.

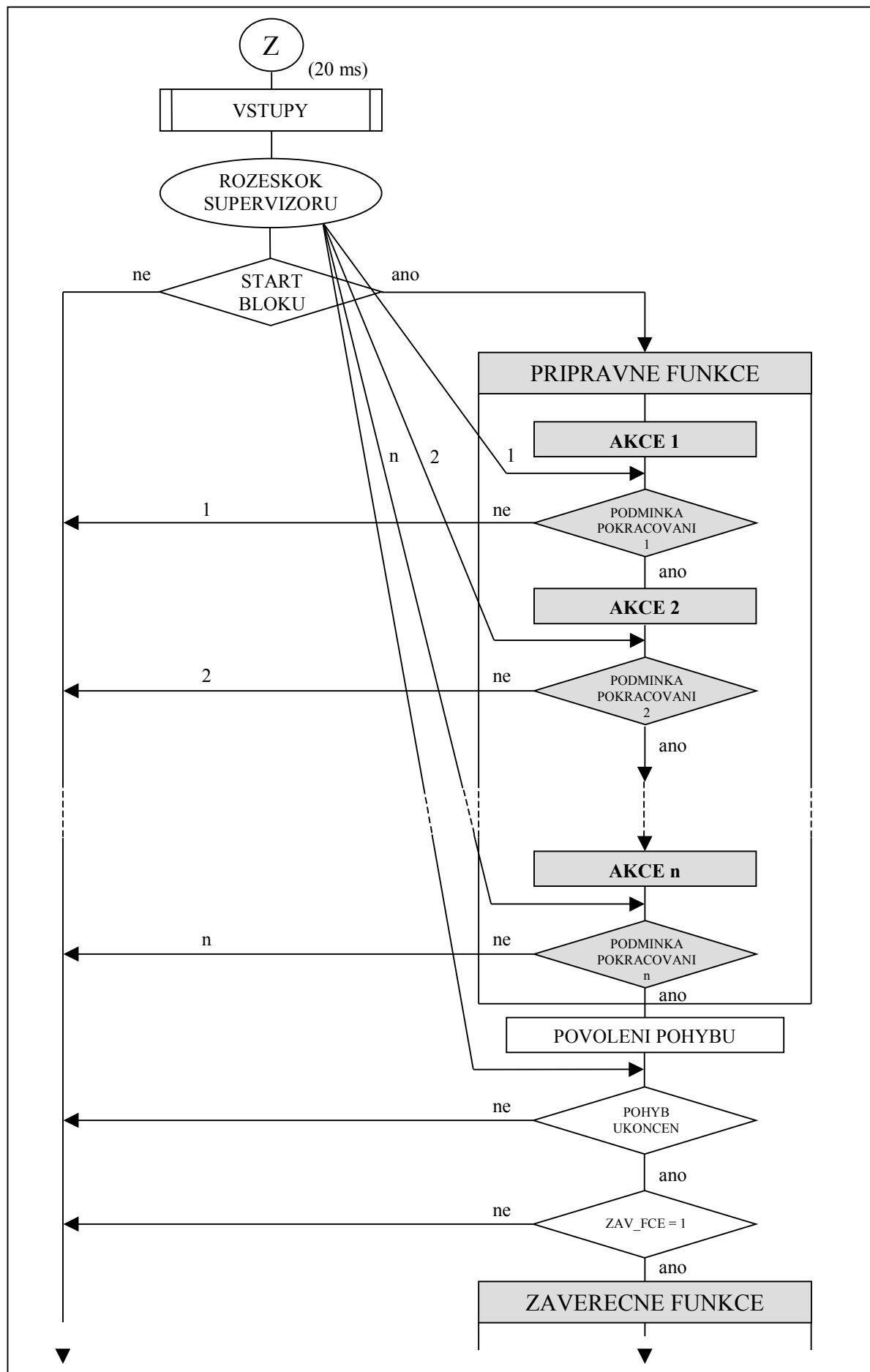
V modulu se programují činnosti, které nejsou přerušitelné ladícími prostředky. Tato vlastnost může být při ladění programu velmi výhodná. Logika naprogramovaná v modulu **PIS_CONT** se jeví, že probíhá paralelně s hlavními moduly PLC programu.

modul	PIS_FAR (MODULE_FAR)
--------------	-----------------------------

Modul začíná klíčovým slovem **PIS_FAR (MODULE_FAR)** a končí klíčovým slovem **PIS_FAR_END (MODULE_FAR_END)**. Tento modul je nepovinný.

Vydálený modul PLC programu, který logicky navazuje na modul **PROVOZ_VYSTUP**. Používá se pro delší PLC programy, kterých kódový segment přesáhne 64 kByte. Modul má všechny vlastnosti jako jsou u modulu **PROVOZ_VYSTUP**, to znamená, že je možno používat mechanismy a je možno použít instrukci **DEBUG** pro nastavení break-pointů.

Pro řadu systémů CNC8x9 DUAL se doporučuje místo modulu **PIS_FAR** napsat program do vícero souborů.



5.4 Řízení průchodu supervizorem interfejsu

Supervizor programovatelného interfejsu řídí průchod PLC programu následujícím způsobem. Po odstartování bloku odevzdá řízení do modulu **PRIPRAVNE_FUNKCE** a vykoná se úsek programu po první výskyt instrukce definice stavu, to je splnění určité podmínky. Jedná se o instrukci **typu EX** (viz kapitola "Logické sekvenční celky"). V každém dalším průchodu interfejsu se kontroluje jen splnění této poslední podmínky pokračování, to je oblast programu mezi předposlední a poslední instrukcí typu EX.

Po jejím splnění pokračuje průchod modulu přípravných funkcí po další podmínku. Podmínkami může být i provedení aktivovaných mechanismů, jak to bylo popsáno v kapitole "Logické sekvenční celky" - příklad nastartování mechanismu :

FL	1,CW	;nastavení aktivační proměnné
EX		
LDR	CW	;kontrola vykonání mechanismu
EX1		

Po vykonání celého modulu přípravných funkcí supervizor interfejsu povolí případný pohyb pro interpolátor a čeká v tomto stavu, pokud není splněna podmínka dosažení programované polohy.

Po potvrzení programované polohy supervizor odevzdá řízení modulu **ZAVERECNE_FUNKCE** , ale jen v případě nastavení bitové proměnné **ZAV_FCE**. Modul závěrečných funkcí je také logický sekvenční celek a program ním projíždí stejným způsobem, jako v modulu přípravných funkcí.

Rychlost průběhu bloku bude závislá na tom, jak se navrhne modul přípravných a závěrečných funkcí. Instrukce EX například způsobí prodlevu v průběhu o jeden cyklus PLC programu. Tyto instrukce je ale velmi vhodné použít, když má dojít k čekání na splnění určité podmínky, což je naprogramováno pomocí instrukcí EX0, EX1, TEX0 nebo TEX1. V tomto případě se instrukcí EX vlastně omezuje úsek programu, který bude procházen. (viz kapitola "Logické sekvenční celky").

Modul **PROVOZ** se startuje v každém taktu interfejsu (20 ms) a tento už není sám o sobě logickým sekvenčním celkem. V něm jsou umístěny sekvenční celky (mechanismy) pomocí příkazů **MECH_BEGIN** a **MECH_END**.

5.5 Více souborů pro psaní PLC programu

Možnost psát PLC program do více souborů je umožněna pro systémy řady CNC8x9 –DUAL.

PLC program se skládá z hlavního souboru, který obsahuje všechny povinné moduly programu (VSTUP, PRIPRAVNE_FUNKCE, ZAVERECNE_FUNKCE, PROVOZ_VYSTUP, PIS_HALT, PIS_CLEAR a PIS_INIT). Kromě hlavního souboru PLC program může být napsán v dalších samostatných souborech (ve verzi 6.001 jich může být maximálně 7). Další soubory mohou mít deklarována data a jsou pokračováním modulu PROVOZ_VYSTUP (MODULE_MAIN) z hlavního souboru.

Soubory PLC programu je možno využít pro připojování odladěných knihovních funkcí PLC programu. Soubory PLC programu musí splňovat:

- V konfiguračním souboru TECH.KNF je hlavní soubor uveden pod klíčovým slovem **FilePlc** a další soubory pod klíčovými slovy **FilePlcExt**. Tato klíčová slova jsou nepovinná.

Příklad:	FilePlc	=	I_O_MAIN	;hlavní soubor
	FilePlcExt	=	I_O_MOD2	;druhý soubor - maximálně 7 souborů
	FilePlcExt	=	NO	;nepoužito

- b) Každý z dalších souborů PLC programu musí povinně obsahovat moduly:

DATA	;Globální data
;... deklarace globálních dat	
DATA_END	;(START)
DATA_LOCAL	;Lokální data
;... deklarace lokálních dat	
DATA_LOCAL_END	
MODULE_MAIN	;(PROVOZ_VYSTUP)
;... základní logika, mechanismy	
MODULE_MAIN_END	;(PROVOZ_VYSTUP_END)

Další moduly jsou nepovinné:

MODULE_INIT	;(PIS_INIT)
;... inicializace dat	
MODULE_INIT_END	;(PIS_INIT_END)
MODULE_CLEAR	;(PIS_CLEAR)
;... nulování dat	
MODULE_CLEAR_END	;(PIS_CLEAR_END)

Všechny moduly jsou pokračováním stejných modulů z hlavního souboru PLC programu.

- c) Modul **DATA** musí být uveden jako první. Všechna data, která jsou definována v libovolném souboru PLC programu včetně hlavního souboru, mají **globální** charakter, což znamená, že jsou přístupna ve všech ostatních souborech.
- d) Modul **DATA_LOCAL** je nepovinný modul PLC programu pro deklarování **lokálních** proměnných. Data deklarovaná v tomto modulu jsou přístupná jen v souboru PLC programu, kde se modul vyskytuje. Modul **DATA_LOCAL** může být použit v každém souboru s PLC programem i vícekrát a může být přitom vnořen do jiných modulů, kromě modulu **DATA**. Lokální data se používají i pro definování “automatických” proměnných v rámci rozvoje některých instrukcí jazyka PLC836. Data definovaná v tomto modulu jsou v této verzi neviditelná i pro ladící program WINTECHNOL. Když je potřeba pro ladění PLC programu zviditelnit lokální proměnné, dočasně přemístíme modul **DATA_LOCAL** do těla modulu **DATA**. Když je modul **DATA_LOCAL** umístěn uvnitř modulu **DATA**, který má globální charakter, zviditelní se lokální data také pro WINTECHNOL.
- e) Modul **MODULE_MAIN** (**PROVOZ_VYSTUP**) je pokračováním stejného modulu z předešlých souborů PLC programu. Soubor může obsahovat mechanismy a může volat mechanismy, které jsou definovány v jiných souborech PLC programu. Také instrukce **MECH_INIT** může být použita ve všech souborech, i když tam není mechanismus definován. V modulech může být použita instrukce **DEBUG**.
- f) Moduly **MODULE_INIT** a **MODULE_CLEAR** (**PIS_INIT** a **PIS_CLEAR**) jsou v dalších souborech nepovinné. Pokud jsou použity, jsou také pokračováním stejných modulů v předešlých souborech.
- g) Všechny soubory mohou obsahovat libovolné definice procedur **PROC_BEGIN** – **PROC_END** a také libovolná volání procedur definovaných v jiných souborech **PROC_CALL**.
- h) Ve všech souborech mohou být použity instrukce pro definici časových úseků **DFTM01**, **DFTM1**, **DFTM10**, **DFM100** a mohou být použity v jednom souboru i vícekrát.