

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 Moduly jazyka TECHNOL

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 nashoduje 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 interfejsu 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.

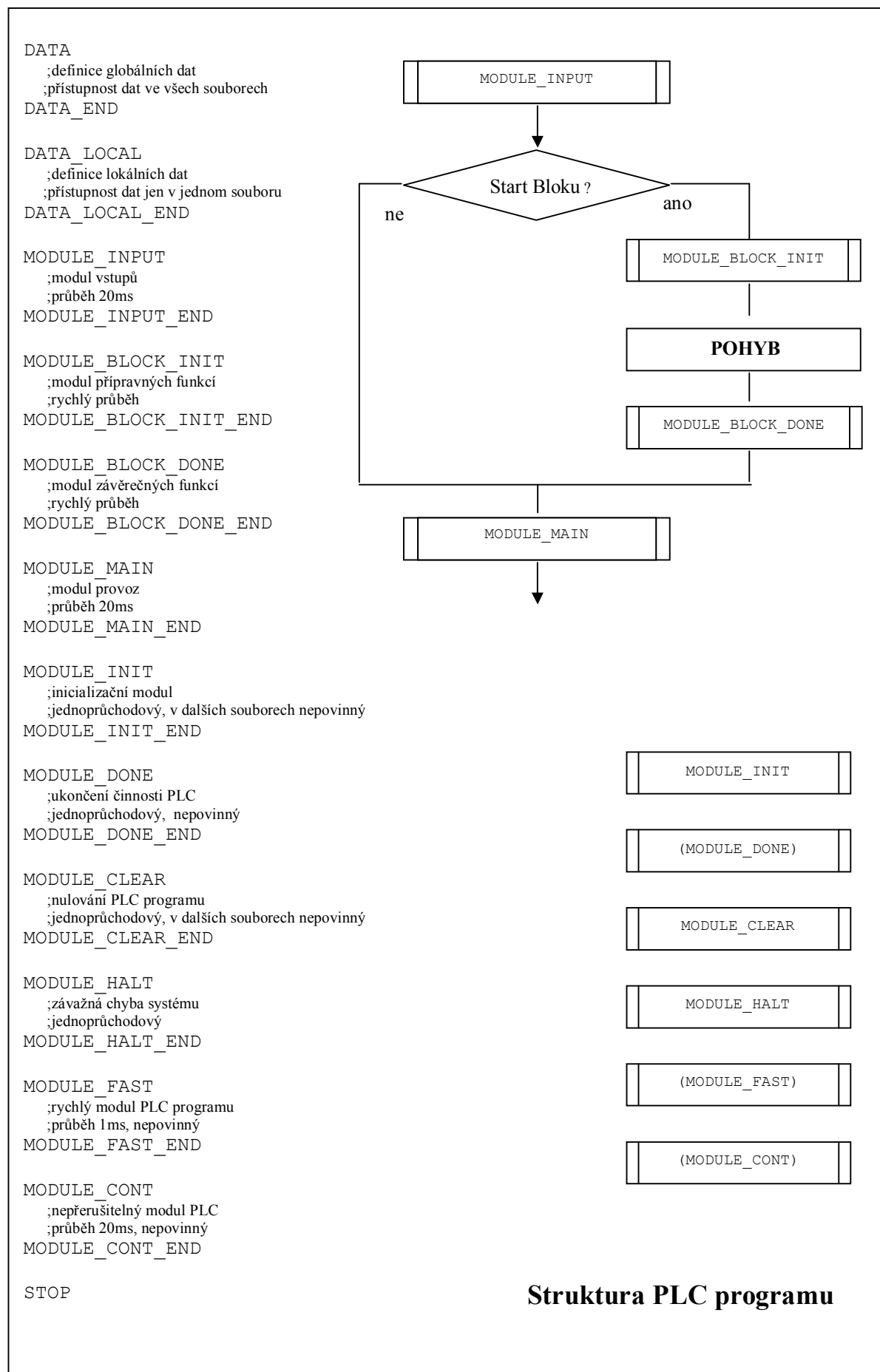
Za klíčovým slovem **DATA_END** následují další moduly, uvedené na obr. Názvy modulů jsou povinné. Na jejich pořadí nezáleží, doporučuje se však zachovat pořadí uvedené na obrázku. Některé moduly musí být uvedeny povinně a některé jsou nepovinné.

Za posledním modulem 1. souboru musí být uvedeno klíčové slovo **STOP**.

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 `MODULE_INPUT`. Za modulem `MODULE_INPUT` se program větví v závislosti na činnosti systému. Pokud není odstartován nový blok partprogramu, program neprochází modulem `MODULE_BLOCK_INIT` a `MODULE_BLOCK_DONE`, ale pokračuje modulem `MODULE_MAIN`. Pokud je odstartován nový blok, projde program výše uvedenými modulem, mezi kterými nastane, pokud je programován, také pohyb souřadnic. Moduly `MODULE_INIT`, `MODULE_CLEAR`, `MODULE_DONE` a `MODULE_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.



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

Modul globálních dat začíná klíčovým slovem **DATA** a končí klíčovým slovem **DATA_END**.
Modul je povinný v každém souboru PLC.

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.

Příklad:

```
DATA                                ;Začátek deklarace dat

      BUN1:      DS      2          ;Word
      PAM10:     DFM      , ,ALFA, , ,BETA, , ;Bitové proměnné

DATA_END                            ;Konec deklarace dat
```

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

Modul lokálních dat začíná klíčovým slovem **DATA_LOCAL** a končí klíčovým slovem **DATA_LOCAL_END**.
Modul je nepovinný. Modul může být použit v každém souboru PLC i vícekrát.

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. Lokální data se používají i pro definování “automatických” proměnných v rámci rozvoje některých instrukcí jazyka **TECHNOL**.

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	MODULE_INPUT
-------	---------------------

Modul začíná klíčovým slovem **MODULE_INPUT** a končí klíčovým slovem **MODULE_INPUT_END**.
Modul je povinný v 1. souboru PLC a v dalších souborech PLC se nesmí použít.
Modul je procházen v rastru 20ms.

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 MODULE_BLOCK_INIT

Modul začíná klíčovým slovem **MODULE_BLOCK_INIT** a končí klíčovým slovem **MODULE_BLOCK_INIT_END**.

Modul je povinný v 1. souboru PLC a v dalších souborech PLC se nesmí použít.

Modul je procházen v rychlém rastru po startu bloku (závisí jen od výkonnosti procesoru).

Modul MODULE_BLOCK_INIT se odstartuje jen po odstartování bloku. Modul má vlastnosti jako sekvenční logický celek (mechanismus). Modul může sloužit jako aktivační 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".

modul MODULE_BLOCK_DONE

Modul začíná klíčovým slovem **MODULE_BLOCK_DONE** a končí klíčovým slovem **MODULE_BLOCK_DONE_END**.

Modul je povinný v 1. souboru PLC a v dalších souborech PLC se nesmí použít.

Modul je procházen v rychlém rastru v závěrečných funkcích odstartovaného bloku (závisí jen od výkonnosti procesoru).

V modulu se obvykle řeší akce, které jsou typické jako závěrečné funkce bloků partprogramu, např. stop vřetena, vypnutí chlazení atd. 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 MODULE_MAIN

Modul začíná klíčovým slovem **MODULE_MAIN** a končí klíčovým slovem **MODULE_MAIN_END**.

Modul je povinný v každém souboru PLC.

Modul je procházen v rastru 20ms.

V modulu se obvykle řeší funkce, které musí být trvale procházeny (základní logika 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.

modul	MODULE_INIT
--------------	--------------------

Modul začíná klíčovým slovem **MODULE_INIT** a končí klíčovým slovem **MODULE_INIT_END**.

Modul je povinný v 1. souboru PLC a v dalších souborech PLC je nepovinný.

Moduly ze všech PLC souborů se prochází jednorázově při startu PLC programu.

Tento modul může být použitý ve všech souborech PLC programu. Modul slouží např. k inicializaci proměnných PLC programu a jiných akcí, které je nutné provést při startu PLC programu. Ke startu PLC programu může dojít při prvním zapnutí systému nebo na příkaz „**START PLC**“ například z ladícího programu Wintechnol. Moduly nejsou volány trvale, ale provedou se jednorázově pouze po startu PLC programu.

PLC program umístěný v tomto modulu může získat informaci, zda byl start PLC programu způsoben prvním zapnutím systému nebo to byl příkaz „**START PLC**“ například z ladícího programu Wintechnol. Informaci získá z datového DR registru za začátkem modulu.

Pokud inicializace PLC programu trvá nějakou dobu, například se čtou data z PLC tabulky a plní se sdílená paměť, tak se systému musí dát zpráva o ukončení této činnosti pomocí instrukce **MODULE_INIT_FINISHED**. Tato instrukce je nepovinná a pokud nebude použita, tak inicializace proběhne hned po průchodu modulem **MODULE_INIT**. Pokud ale v daném souboru je instrukce **MODULE_INIT_FINISHED** použita, systém čeká s pokračováním činnosti při inicializaci až do doby průchodu touto instrukcí (až potom proběhne například 1. centrální anulace). V tomto případě je vhodné v modulu **MODULE_INIT** aktivovat mechanismus, který je umístěn standardně v modulu **MODULE_MAIN** a v něm na konci po ukončení činnosti inicializace je použita instrukce **MODULE_INIT_FINISHED**. Instrukce **MODULE_INIT_FINISHED** může být použita ve všech souborech PLC programu a systém pak čeká s ukončením inicializace na všechny tyto instrukce.

Datový registr DR	Význam
0	Start PLC programu po zapnutí systému
1	Příkaz pro start PLC programu (například po stopu PLC programu)

Příklad:

```

MODULE_INIT                                     ;Začátek modulu inicializace
    EQ      CNST.1                             ;Je to na příkaz START PLC ?
    JL1     STARTPLC
    ; ....

    FL      1,MECH_INICIALIZACE                ;Start mechanismu inicializace
                                           ;na který se musí čekat
    ; ....

MODULE_MAIN                                     ;umístěno v MODULE_MAIN
    ; ....

    MECH_BEGIN MECH_INICIALIZACE              ;Mechanismus inicializace
    ; ....

    MODULE_INIT_FINISHED                      ;Konec čekání na inicializaci
    MECH_END MECH_INICIALIZACE
    ; ....

```

modul	MODULE_DONE
--------------	--------------------

Modul začíná klíčovým slovem **MODULE_DONE** a končí klíčovým slovem **MODULE_DONE_END**.

Modul je nepovinný a může se použít ve všech souborech PLC.

Moduly ze všech PLC souborů se prochází jednorůchodově při ukončení PLC programu.

Tento modul může být použitý ve všech souborech PLC programu. Modul slouží k činnosti potřebné při stopu PLC, např. k deaktivaci pohonů a k zálohování. Ke stopu PLC programu může dojít při vypínání systému nebo na příkaz „STOP PLC“ například z ladícího programu Wintechnol. Moduly nejsou volány trvale, ale provedou se jednorázově pouze při stopu PLC programu.

PLC program umístěný v tomto modulu může získat informaci, zda byl stop PLC programu způsoben vypínáním systému nebo to byl příkaz „STOP PLC“ například z ladícího programu Wintechnol. Informaci získá z datového DR registru za začátkem modulu.

Pokud závěrečné operace PLC programu trvají nějakou dobu, například se deaktivují pohony, tak se systému musí dát zpráva o ukončení této činnosti pomocí instrukce **MODULE_DONE_FINISHED**. Tato instrukce je nepovinná a pokud nebude použita, tak ke stopu dojde hned po průchodu modulem **MODULE_DONE**. Pokud ale v daném souboru je instrukce **MODULE_DONE_FINISHED** použita, systém čeká s pokračováním činnosti při ukončování až do doby průchodu touto instrukcí (až potom se vypne systém). V tomto případě je vhodné v module **MODULE_DONE** aktivovat mechanismus, který je umístěn standardně v modulu **MODULE_MAIN** a v něm na konci po ukončení činnosti je použita instrukce **MODULE_DONE_FINISHED**. Instrukce **MODULE_DONE_FINISHED** může být použita ve všech souborech PLC programu a systém pak čeká s ukončením činnosti PLC na všechny tyto instrukce.

Datový registr DR	Význam
0	Stop PLC programu při vypínání systému
1	Příkaz pro stop PLC programu (například pro načtení nového PLC programu)

Příklad:

```

MODULE_DONE                                     ;Začátek modulu ukončení
    EQ      CNST.1                               ;Je to na příkaz STOP PLC ?
    JL1     STOPPLC
    ; ....

    FL      1,MECH_DEAKTIVACE                     ;Start mechanismu ukončení
                                                ;na který se musí čekat
    ; ....

MODULE_MAIN                                     ;umístěno v MODULE_MAIN
    ; ....

    MECH_BEGIN  MECH_DEAKTIVACE                   ;Mechanismus deaktivace
    ; ....

    MODULE_DONE_FINISHED                         ;Konec čekání na deaktivaci
    MECH_END    MECH_DEAKTIVACE
    ; ....

```

modul MODULE_CLEAR

Modul začíná klíčovým slovem **MODULE_CLEAR** a končí klíčovým slovem **MODULE_CLEAR_END**.

Modul je povinný v 1. souboru PLC a v dalších souborech PLC je nepovinný.

Moduly ze všech PLC souborů se prochází jednorůchodově při nulování PLC programu.

Modul může sloužit k nulování proměnných, uvedení PLC programu do výchozího stavu. Modul se spustí pouze na příkaz pro nulování PLC (například z ladícího programu Wintechnol).

Tento modul může být použitý ve všech souborech PLC programu.

modul MODULE_HALT

Modul začíná klíčovým slovem **MODULE_HALT** a končí klíčovým slovem **MODULE_HALT_END**.

Modul je povinný v 1. souboru PLC a v dalších souborech PLC se nesmí použít.

Modul se prochází jednorůchodově při vážné chybě systému, kdy bude ukončena činnost.

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 vypnutí silové části stroje.

modul MODULE_FAST

Modul začíná klíčovým slovem **MODULE_FAST** a končí klíčovým slovem **MODULE_FAST_END**.

Modul je nepovinný a může se použít ve všech PLC souborech.

Modul je procházen v rastru interpolátoru a servosmyček (1ms).

V modulu se programují činnosti, které mají probíhat v rychlejším časovém rastru než 20 ms. Modul **MODULE_FAST** je aktivován ve stejných časových intervalech jako softwerová polohová vazba – 1ms. V modulu **MODULE_FAST** můžou být naprogramovány logické sekvenční celky. V modulu může být povoleno ladění pro sledování registrů, ale je zakázáno použít break-pointy.

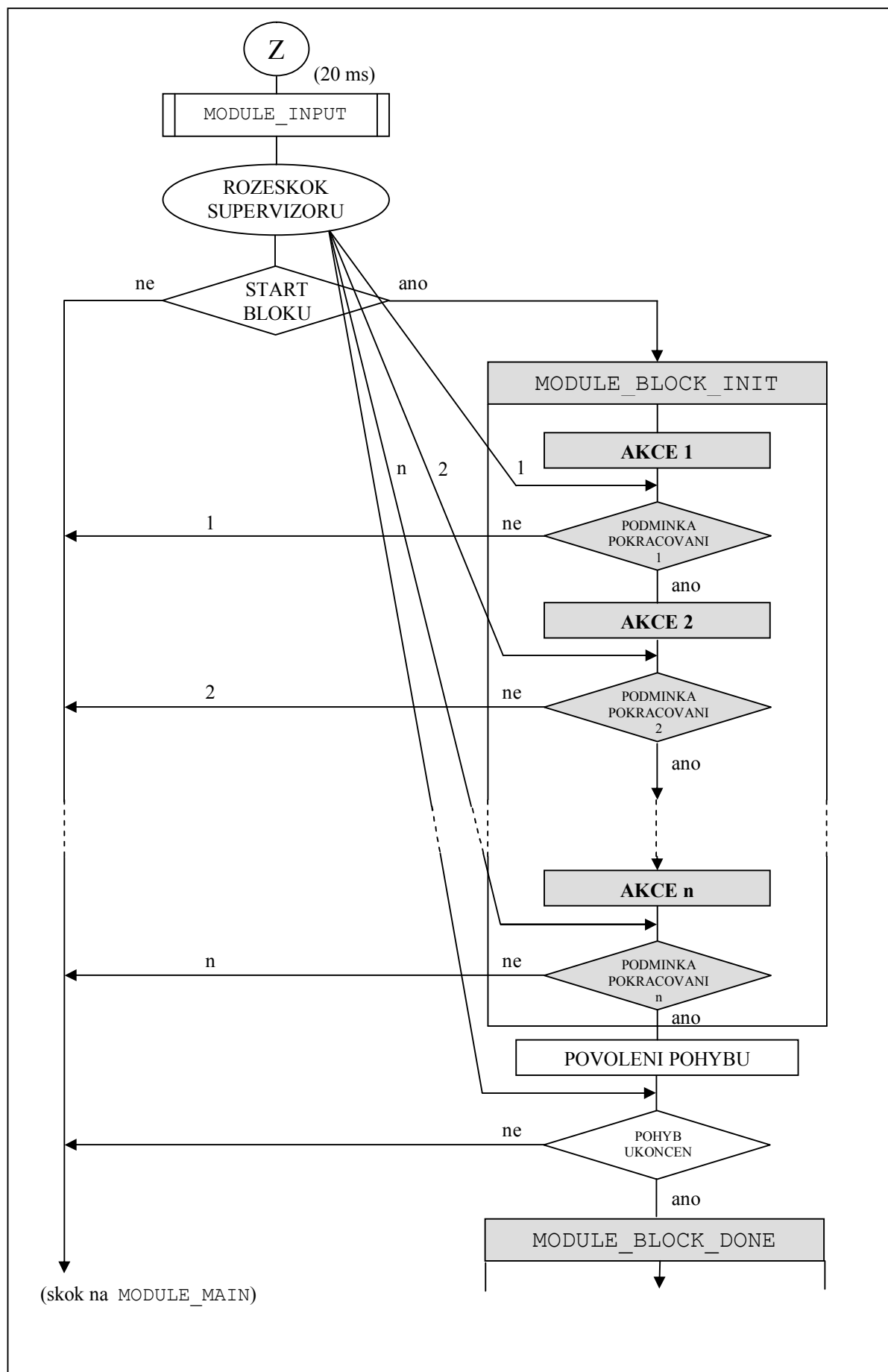
modul MODULE_CONT

Modul začíná klíčovým slovem **MODULE_CONT** a končí klíčovým slovem **MODULE_CONT_END**.

Modul je nepovinný, smí se použít v 1. souboru PLC a v dalších souborech PLC se nesmí použít.

Modul je procházen v rastru 20ms.

V modulu se programují činnosti, které nejsou přerušitelné ladícími prostředky. Tato vlastnost může být při ladění programu vhodná pro naprogramování „životně důležitých funkcí stroje“. Logika naprogramovaná v modulu **MODULE_CONT** se jeví, že probíhá paralelně s hlavními moduly PLC programu.



5.3 Řízení průchodu supervizorem interfejsu

Supervizor programovatelného interfejsu řídí průchod PLC programu následujícím způsobem. Po odstartování bloku odevzdá řízení do modulu **MODULE_BLOCK_INIT** 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 **MODULE_BLOCK_DONE**. 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 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. Návrhář PLC programu by měl dbát o to, aby v případě že se jedná o čistě pohybový blok ve kterém není programovaná žádná technologie, nedošlo ke zdržení v modulech **MODULE_BLOCK_INIT** a **MODULE_BLOCK_DONE**. Při nedodržení této podmínky by mohlo dojít k zasekávání plynulé jízdy. **Pohybový blok bez technologie musí moduly MODULE_BLOCK_INIT a MODULE_BLOCK_DONE procházet jednopřechodově.**

Modul **MODULE_MAIN** se startuje v každém taktu PLC programu (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**.

5.4 Více souborů pro psaní PLC programu

PLC program se skládá z hlavního souboru, který obsahuje všechny povinné moduly programu (MODULE_INPUT, MODULE_BLOCK_INIT, MODULE_BLOCK_DONE, MODULE_MAIN, MODULE_HALT, MODULE_CLEAR a MODULE_INIT). Kromě hlavního souboru, PLC program může být napsán v dalších samostatných souborech (v současné verzi celkem maximálně 64 souborů). Další soubory mohou mít deklarována data a jsou pokračováním modulu MODULE_MAIN z hlavního souboru.

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

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

```
DATA                                ;Globální data
    ;... deklarace globálních dat
DATA_END

MODULE_MAIN
    ;... základní logika, mechanismy
MODULE_MAIN_END
```

Další moduly jsou nepovinné:

```
DATA_LOCAL                          ;Lokální data
    ;... deklarace lokálních dat
DATA_LOCAL_END

MODULE_INIT                          ;Start PLC
    ;... inicializace dat
MODULE_INIT_END

MODULE_DONE                          ;Stop PLC
    ;... ukončovací operace
MODULE_DONE_END

MODULE_CLEAR                         ;Nulování PLC
    ;... nulování dat
MODULE_CLEAR_END
```

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

- b) 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.
- c) 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. Lokální data se používají i pro definování “automatických”

proměnných v rámci rozvoje některých instrukcí jazyka **TECHNOL**. 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**.

- d) Modul **MODULE_MAIN** 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**.
- e) Moduly **MODULE_INIT**, **MODULE_DONE** a **MODULE_CLEAR** jsou v dalších souborech nepovinné. Pokud jsou použity, jsou také pokračováním stejných modulů v předešlých souborech.
- f) Všechny soubory mohou obsahovat libovolné definice procedur **PROC_BEGIN** – **PROC_END** a také libovolná volání procedur definovaných v jiných souborech **PROC_CALL**.
- g) Ve všech souborech mohou být použity instrukce pro definici časových úseků **DFTM01**, **DFTM1**, **DFTM10**, **DFTM100** a mohou být použity v jednom souboru i vícekrát.