



# PŘENOS INFORMACÍ POMOCÍ NC PROGRAMU U DVOUKANÁLOVÝCH OBRÁBĚCÍCH CENTER V SÉRIOVÉ VÝROBĚ

## Bakalářská práce

*Studijní program:* B2301 – Strojní inženýrství  
*Studijní obor:* 2301R000 – Strojní inženýrství  
*Autor práce:* **Radka Kovaříková**  
*Vedoucí práce:* Ing. Petr Keller, Ph.D.





## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Jméno a příjmení **Radka KOVAŘÍKOVÁ**

Studijní program **B2301 Strojní inženýrství**

Ve smyslu zákona č. 111/1998 Sb. o vysokých školách se Vám určuje bakalářská práce na téma:

### **Přenos informací pomocí NC programu u dvoukanálových obráběcích center v sériové výrobě**

#### **Zásady pro vypracování:**

(uved'te hlavní cíle bakalářské práce a doporučené metody pro vypracování)

Cílem práce je najít možnosti ovlivnit začátek zpracování daného NC programu v řídicím systému Sinumerik 840D u dvoukanálových obráběcích center např. při předchozím neočekávaném přerušení vykonávání tohoto programu. Zadání vzešlo na základě potřeby podniku Škoda Auto a.s.

Doporučené metody pro vypracování:

1. Úvod, seznamte se s problematikou vícekanálového programování a možnostmi synchronizace a navázání jednotlivých kanálů v ŘS Sinumerik 840D.
2. Zanalyzujte a navrhňte možnosti uložení informace o poslední prováděné operaci v ŘS Sinumerik 840D s přihlédnutím k vícekanálovému řízení.
3. Proveďte návrh řešení startu NC programu od poslední vykonané operace dále s cílem snížit časovou náročnost, ale i následnou zmetkovitost obrábění.
4. Zhodnot'te navržené řešení.



Forma zpracování bakalářské práce:

Průvodní zpráva v rozsahu cca 35 stran, včetně příloh.

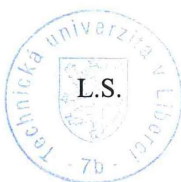
Text celé bakalářské práce včetně příloh bude v elektronické formě přiložen na CD nosiči k tištěnému svazku originálu i kopie bakalářské práce.

Seznam literatury (uved'te doporučenou odbornou literaturu):

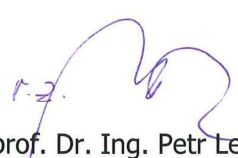
- [1] Dokumentace k ŘS Sinumerik 840D.
- [2] Keller, P. *Programování a řízení CNC strojů. Prezentace přednášek*. KVS, FS, TU v Liberci. Liberec 2005.
- [3] Články v časopisech a na internetu.

Vedoucí bakalářské práce:

Ing. Petr Keller, Ph.D.



  
Ing. Petr Zelený, Ph.D.  
vedoucí katedry

  
prof. Dr. Ing. Petr Lenfeld  
děkan

V Liberci dne 15. března 2014

Platnost zadání bakalářské práce je 15 měsíců od výše uvedeného data (v uvedeném lhůtě je třeba podat přihlášku ke SZZ).  
Termíny odevzdání bakalářské práce jsou určeny pro každý studijní rok a jsou uvedeny v harmonogramu výuky.

## Prohlášení

Byla jsem seznámena s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracovala samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 20.6.2014

Podpis: hi

## Poděkování

Jsem velice ráda, že mohu poděkovat všem, jejichž zásluhou bylo možno práci zrealizovat.

Tímto bych chtěla poděkovati Ing. Petru Kellerovi Ph.D. za odborné vedení práce, věcné připomínky, dobré rady a vstřícnost při konzultacích při vypracovávání bakalářské práce.

Zároveň bych chtěla poděkovat své rodině, která mne po celou dobu studia podporovala.

Bakalářská práce KVS – VS – 216

TÉMA BAKALÁŘSKÉ PRÁCE:

Přenos informací pomocí NC-programu u dvoukanálových obráběcích center v sériové výrobě

ANOTACE:

Cílem této práce je ovlivnit začátek zpracování NC-programu SINUMERIK 840D pomocí jeho ukončení v předešlém obráběcím cyklu.

Po přerušení obrábění dvouřetenových, dvoukanálových, jednoúčelových strojů v sériové výrobě díky vnějším vlivům (výpadek proudu, poškození nástroje, seřizování stroje, ladění NC-programu a různé pseudochyby), stroj po odstranění příčiny zastavení nechá najet jednotlivé pohybové i rotační osy do základních poloh a po opětovném spuštění stroje se NC-program zpracovává od začátku. To znamená, že místa, která byla již opracována před vynuceným zastavením, jsou obráběna znovu. Tím se však změni rozměry a povrch obráběného dílu, dochází i k poškození obráběcích nástrojů a samozřejmě se navyšuje doba prostoje obráběcího stroje.

Tato práce má za úkol navrhnout NC-program tak, aby stroj po opětovném spuštění zpracování měl informaci o naposledy použitém nástroji v předchozím zpracování a tím pokračovalo opracování obrobku obráběcím nástrojem, který bezprostředně následuje.

Pokud by k žádnému přerušení zpracování NC-programu nedošlo, pak by byl, jako naposledy zvolený nástroj s dokončeným opracováním kusu, zapsaný obráběcí nástroj, který je poslední v pořadí při obrábění dílu.

Výjimku budou tvořit závitníky, protože jejich oprava je v rychlořezném obrábění složitá, neboť při opětovném najetí závitníku do již hotového závitu se závit často deformuje.

Proto při přerušení obrábění v průběhu závitování NC program bude dále zjišťovat, který závit byl již opracován.

(Klíčová slova: NC-program, Obrábění, Obráběcí centra, přenos informací mezi kanály)

THEME OF BACHELOR THESIS THESES: Transmission of information by help of NC- programme by two-channel shaping centres in serial production

ANOTATION:

The aim of this work is beginning to affect the processing of the NC program SINUMERIK 840D through its termination in the previous machining cycle.

After the interruption of a dual-spindel , dual-channel , single-purpose machines for mass production due to external influences (power failure , damage to the tool , machine setup , tuning the NC program and various falls ) , after removal of the machine lets roll individual stop motion and rotary axes to the home position and after restarting machine NC - program processes from the start. This means that the place which has been machined prior to the forced stopping is machined again , it changes the dimensions and surface of the machined portion , it causes damage to machine tools, and of course , increases the downtime of the machine tool .

This work aims to propose an NC program to machine after restart processing had their last tool used in the previous processing and thus could move start working on the tool , which immediately follows .

Should any interrupt processing of the NC program there , then it was like the last selected tool with the finished piece written by machining machine tool, which is last in the order when machining parts .

The exception will form taps because their repair is in high speed machining of complex, because when you place again tap into the already finished the thread thread often deformed.

Thus, when interrupt processing during threading NC program will also identify which thread was already machined

(key words: NC programme, cutting operation, machining centre, transmission informations channels to channels)

## Obsah

SEZNAM POUŽITÝCH ZKRATEK A POJMŮ .....	8
1. ÚVOD.....	10
2. POPIS OBRÁBĚCÍHO STROJE.....	11
3. POPIS ŘÍDICÍHO SYSTÉMU SINUMERIK 840D.....	13
4. VOLBA ŘEŠENÍ.....	16
4.1. POUŽITÍ UŽIVATELSKÝCH PROMĚNNÝCH .....	16
4.1.1. Definice Uživatelských proměnných GUD.....	17
4.1.2. Definice Uživatelských proměnných LUD.....	17
4.1.3. Definice Uživatelských proměnných PUD.....	17
4.2. POUŽITÍ R-PARAMETRŮ .....	18
4.2.1. Definice R-parametrů.....	18
4.2.2. Použití R-parametrů.....	18
4.3. POUŽITÍ KOMBINACE UŽIVATELSKÝCH PROMĚNNÝCH A R-PARAMETRŮ .....	18
4.4. VYHODNOCENÍ JEDNOTLIVÝCH ŘEŠENÍ .....	19
5. SESTAVENÍ NC-PROGRAMU .....	20
5.1. DEFINICE GLOBÁLNĚ UŽIVATELSKÝCH PROMĚNNÝCH GUD .....	21
5.2. SYNCHRONIZACE KANÁLŮ .....	24
5.2.1. Definice programové smyčky (IF, ENDIF).....	26
5.2.2. Definice programových skoků (GOTOB, GOTOF, GOTO, GOTOC) .....	27
5.2.3. Předávání os mezi kanály (využití příkazů GET, RELEASE).....	30
5.3. ZJIŠŤOVÁNÍ STAVU OBROBENÍ PO VYNUCENÉM UKONČENÍ ZPRACOVÁNÍ NC-PROGRAMU .....	32
5.3.1. Rozhodování pomocí příkazu CASE pro podmíněné větvení programu.....	32
5.3.2. Hlášení (MSG).....	35
5.4. NAPROGRAMOVÁNÍ ZJIŠTĚNÍ NAPOSLEDY DOKONČENÉHO OBRÁBĚCÍHO NÁSTROJE DO OBRÁBĚCÍHO PROGRAMU.....	36
6. ZÁVĚR .....	37
7. ZDROJE .....	38
7.1. SEZNAM LITERATURY.....	38
7.2. ZDROJE Z INTERNETU .....	38
8. SEZNAM OBRÁZKŮ .....	39
9. PŘÍLOHY.....	40
9.1. HLAVNÍ PROGRAM PRO 1. KANÁL .....	40
9.2. HLAVNÍ PROGRAM PRO 2. KANÁL .....	50



## Seznam použitých zkratk a pojmů

Sinumerik 840D (Sinumerik 840D)

- programovací jazyk pro obráběcí stroje

NCU (NCU)

- centrální řídicí jednotka obráběcích strojů

NC-program (NC-programme)

- volně měnitelný program pro řízení pohybů obráběcího stroje

Obrobek (Workpiece)

- opracovávaný díl

Odlitek (cast)

- opracovávaný díl vyrobený odlitím

Nastavovací tabulky (Settingtables)

- standardní uživatelské obrazovky od firmy SIEMENS, které slouží k nastavení parametrů stroje (posunutí nulového bodu, nastavení parametrů obráběcího nástroje, nastavení uživatelských proměnných, atd....)

Obráběcí nástroj (Cuttingtool)

- fréza, vrták, závitník, atd....

Vřeteno (Spindle)

- rotační osa obráběcího stroje kde je uchycen obráběcí nástroj

Kanál (Chanel)

- umožňuje zpracovávat více NC-programů zároveň nezávisle na sobě

Podprogram SPF (Subprogramme)

- typ NC-program, jímž se realizuje vnořování (přecházení z jednoho programu do druhého, kde po dokončení zpracování podprogramu systém pokračuje ve zpracování hlavního programu z místa, kde byl podprogram zavolán)

MPF (MPF) - hlavní program

- NC-program, který v automatickém režimu po dokončení jednoho obráběcího cyklu umožní PLC odstartovat další.

NC\_ACT(NC\_ACT)

- soubor v NC-řízení, který obsahuje nastavovací data stroje (např. nastavení obráběcího nářadí, nulových bodů programu, uživatelská data, strojní parametry, atd...)

MMC (MMC)

- typ ovládacího panelu pro SINUMERIK 840D (viz. Obrázek č. 2).

PLC (PLC)

- program pro řízení a kontrolu logických, popř. základních matematických funkcí stroje

Externí periferie (Externalperiphery)

- stroje a zařízení, které spolupracují s obráběcím centrem (např. manipulátor obrobků, měřicí a kontrolní zařízení)

# 1. Úvod

Cílem této práce je ovlivnit začátek zpracování NC-programu SINUMERIK 840D pomocí jeho ukončení v předešlém obráběcím cyklu.

Při sériové výrobě jednoho typu dílu je po obráběcím stroji vyžadována co největší automatizace, to znamená, že je snaha minimalizovat potřebu obsluhy a seřizování tohoto stroje. Často však vznikají situace, kdy je na obráběcím stroji vlivem neočekávaných chyb ukončeno zpracování NC-programu. Tyto chyby mohou být vygenerovány například při zlomení obráběcího nástroje, výpadku proudu, při výskytu různých pseudochyb atd. ....

Po vygenerování chyby a následném odstranění její příčiny je obráběcí stroj znovu uveden do stavu, kdy lze spustit zpracování NC-programu. Po jeho spuštění se však vždy začne NC-program zpracovávat od začátku, což vede ve většině případů k poškození obráběného dílu a obráběcího nástroje vlivem opětovného obrobení těch částí obráběného dílu, které byly opravené v předešlém (přerušeném) obráběcím cyklu. Neméně důležitá je také časová prodleva, která vzniká zbytečným obráběním již opravených částí obráběného dílu. Tato časová ztráta je o to více citelná na obráběcích centrech, kde se opracovávají díly velkých a složitých tvarů s vysokou přesností. Ovlivněním místa odkud se začne NC-program znovu zpracovávat lze dosáhnout minimalizace materiálních ztrát. Toto ovlivnění lze provést pomocí informace o posledním obráběcím nástroji, který byl v předchozím obráběcím cyklu použit a část NC-programu, která ho vyvolala, byla již zpracována. S výjimkou závitníků obráběcí nástroj, jehož NC-programem, ještě zpracováno pro potřebu ovlivnění místa startu NC-programu označen nebylo, nebude jako poslední nástroj.

## 2. Popis obráběcího stroje



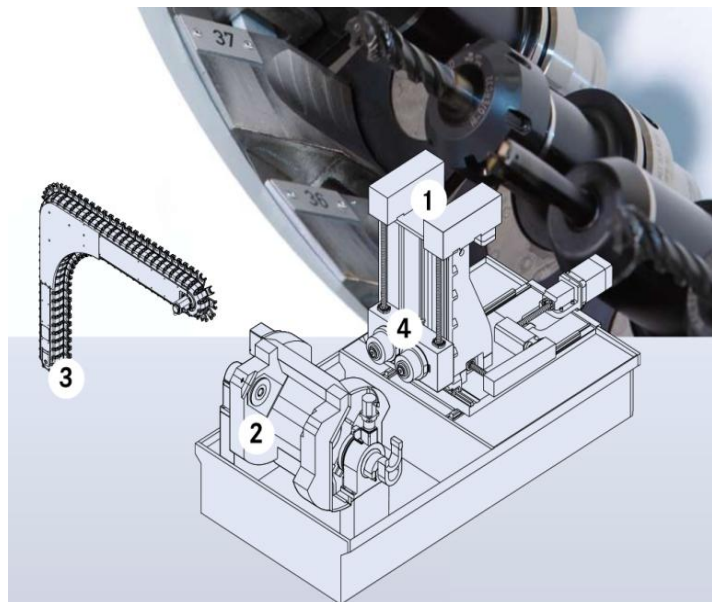
Obrázek 1 – obráběcí centrum[5]



Obrázek 2- řízení a MMC pro SINUMERIK 840D[6]

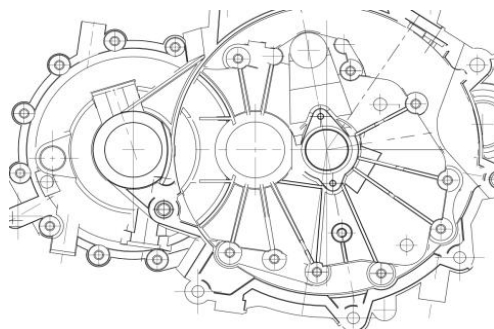
Jedná se o obráběcí centrum (obrázek č. 1) se dvěma vřeteny a dvěma kanály řízený operačním systémem SINUMERIK 840D Power line s operačním panelem (obrázek č. 2). Úpravy lze použít i pro Solution line, protože programovací jazyk Sinumerik 840D je pro oba typy řídicího systému stejný. Každé vřeteno je ovládáno zvlášť jedním kanálem. Z kanálu 1 je ovládáno vřeteno 1 a z kanálu 2 je ovládáno vřeteno 2. Každé vřeteno má tudíž své vlastní programy, jež jsou inicializovány z kanálu 1. Stroj obrábí jeden kus střídavě oběma vřeteny (obrázek č. 3 – část 4). Zásobník obráběcích nástrojů (obrázek č. 3 - část 3) je posazen nad pohony vertikálních os Y1 a Y2 (obrázek č. 3 - část 1). Horizontální pohyb (osa X) se provádí pomocí pohybu upínacího stolu, takže je osa X pro oba kanály stejná a proto musí být v NC-programu naprogramováno předávání řízení této osy spolu s řízením polohování vřeten.

Upínací systém (obrázek č. 3 - část 2) se skládá z upínacích přípravků připevněných na otočném stole (tuto strojní rotační osu pro účel této práce pojmenujeme B1).



Obrázek 3 – znázornění konstrukce obráběcího centra[5]

Obráběný díl (obrázek č. 4) je opracováván ze všech stran pomocí rotace osy B1 a lze na něj použít standartní i speciální frézy, výhrubníky, výstružníky, vrtáky a závitníky v plovoucím pouzdře.



Obrázek 4 – příklad obráběného odlitku[5]

### 3. Popis řídicího systému SINUMERIK 840D

„Sinumerik 840D power line je systém vhodný pro komplexní obráběcí úkoly. Jednotka NCU umí řídit až 31 os (až 31 vřeten) v 10 kanálech a ve standardní verzi může interpolovat až 12 os. Struktura a obsah NC programů řídicího systému Sinumerik vychází z normy DIN 66025. [4]

Tyto programy jsou sestaveny z posloupnosti bloků (vět), přičemž každý blok charakterizuje jeden krok v postupu opracovávání součásti. Do bloků jsou zapisovány příkazy (funkce) ve formě jednotlivých slov. První slovo v NC programu není u řídicího systému Sinumerik striktně předepsáno. Poslední blok v postupu opracování obrobku však musí vyjadřovat konec. Použitelných slov pro ukončení programu je ovšem několik a mezi nejužívanější je možno řadit M30, M17 nebo M2. Jednotlivá slova „NC jazyka“ se dále dělí na adresnou část a numerickou část.[4]

Adresový znak je zpravidla jedno písmeno. Numerická část slova může obsahovat znaménko plus nebo minus, číslice, desetinnou tečku a další číslice. Kladné znaménko není nutné psát.[4]

Jeden blok je tvořen minimálně jedním slovem. Blok musí obsahovat veškeré informace nezbytné pro provedení jednoho kroku pracovního postupu. V případě, že některá slova zapisovaná pro provedení kroku jsou shodná se slovy v bloku (nebo blocích) předchozích, není nutné je znovu zapisovat (tzn. Ženění používán pevný formát bloku). Délka bloku může být maximálně 512 znaků (od SW 5) a posloupnost jednotlivých slov by se z důvodu snadnější orientace měla držet normy (v dnešní době mají však normy pouze doporučující charakter). [4]

Obecný formát bloku má tedy následující strukturu:

N... G... X... Y... Z... F... S... T... D... M... H...

Adresa.....Význam“

Hodnoty adres lze také napsat pomocí uživatelských proměnných, R-parametrů, nebo jejich kombinací a výpočtů.[4]

N.....Číslo řádku (věty). Číslo řádků nemusí být programována, protože slouží jen k lepší přehlednosti NC-programu při jeho diagnostice. V případě, že řídicí systém Sinumerik vygeneruje chybové hlášení, které má návaznost na NC-program, tak v textu chyby je také uveden kanál a číslo řádku, kde je kritická instrukce.[4]

G.....Tyto funkce určují druh pohybu os. (např. G1 je lineární interpolace, G2 a G3 jsou kruhové atd....), určují, jakým způsobem stroj reaguje na zadaný posuv a otáčky (např. G94 - Lineární posuv F, G95 - Otáčkový posuv F, G96 - Zapnutí konstantní řezné rychlosti). Také se těmito funkcemi spouští některé druhy režimů a standardní cykly závitování (G33, G34 - závitování, ...). [4]

G-fce jsou dále rozlišovány podle doby působení – modální působení (jsou aktivní, dokud nedojde k jejich změně)  
- působení pouze v jedné větě (G53 – zrušení programového posunutí nulového bodu, G500 – změna posunutí nulového bodu (vynulování posunutí, pokud ve strojních datech není tato funkce nastavena jinak a deaktivace nastavitelných framů).[4]

X, Y, Z.....souřadnice osy. Hodnotu lze zadat, jak číselně, tak pomocí R-parametrů a uživatelských proměnných.

F.....hodnota posuvu (jakou rychlostí se budou osy stroje pohybovat. Posuv lze spojit s otáčkami (využívá se hlavně při závitování, nebo špičkování a soustružení), kde se otáčky mění podle posuvu.

S.....udává počet otáček u rotačních os, nebo polohu natočení rotační osy.

T.....nástroj : Identifikace se provádí přiřazením čísla (T1, T1001, ...)

D.....korekční hodnoty nástroje: Identifikace se provádí přiřazením čísla (D1, D2, ...). U každého nástroje je možné nadefinovat sadu korekcí, která usnadňuje programování a orientaci v NC-programu. Využívají se např. u vícestupňových vrtáků a fréz.

M.....doplňkové funkce, které naprogramuje výrobce stroje v PLC nebo NC-programu, a se nejčastěji používají pro různé zapínací nebo vypínací funkce stroje (zapnutí/vypnutí chlazení, upínání, pohyb pístnice atd...). M-funkce jsou také standardní naprogramované firmou Siemens, takže jsou pro všechny stroje, kde je tento řídicí systém použit, stejné (např. M30 – konec zpracování NC-programu, M17 – ukončení zpracování podprogramu a návrat do hlavního programu, M8 – zapnutí chlazení, M9 – vypnutí chlazení)

H.....pomocná funkce: Volně programovatelná funkce, kterou výrobce stroje naprogramuje dle potřeby, a pracují s hodnotami datového typu REAL. Využívá se např. pro spuštění/vypnutí kontroly zlomení nástroje, nebo aktivaci měřicího cyklu pro určitou dráhu nástroje, aktivaci nestandardního chlazení atd...

R-parametry..... Výpočtové parametry, předem definované uživatelské proměnné s označením R a typu REAL. Lze použít jak zápis s indexem pole, např. R[10], tak i zápis bez indexu pole, např. R10.[4]



## 4. Volba řešení

Pro zjištění posledního obráběcího nástroje, který byl při opracování použit před ukončením obrábění, ať už by bylo toto ukončení záměrné (dokončení opracování), nebo vynucené (chybný stav), má řídicí systém SINUMERIK 840D několik nástrojů. Lze využít Uživatelské proměnné (GUD, PUD, LUD) a R-parametry.

### 4.1. Použití uživatelských proměnných

Jelikož se jedná o dvoukanálové obráběcí centrum se samostatným řízením obou vřeten, pak musím využít uživatelské proměnné platící globálně GUD (lze s nimi pracovat v obou kanálech). Další výhodou těchto proměnných spočívá v možnosti jejich pojmenování programátorem, takže jejich jméno může napovědět k jaké skupině podmínek (vyhodnocení) tato proměnná patří. Tím lze docílit větší přehlednosti NC-programu.

Definice uživatelských proměnných se provádí pomocí slova DEF, které je klíčové pro zbytek věty a lze jím definovat vlastní uživatelské proměnné označované jako User Data.[7]

Datový typ hodnoty, která se bude do proměnné dosazovat se určuje slovy:

INT – Celočíselné hodnoty s možností požití znaménka[7]

REAL – Reálné číslo[7]

BOOL - TRUE (1) / FALSE (0)[7]

CHAR - ASCII (text)[7]

STRING[<Max. délka>] - Řetězec znaků definované délky [7]

AXIS - Identifikátor osy/vřetena [7]

FRAME - Geometrické údaje pro statickou transformaci souřadného systému[7]

### 4.1.1. Definice Uživatelských proměnných GUD

GUD jsou uživatelské proměnné, které jsou aktivní v celém NC-systému, a po vypnutí stroje nedojde k jejich vymazání. Tyto proměnné se v určitém případě mohou skládat i z lokálních a programových proměnných. Tyto proměnné se zapisí do hranatých závorek, takže konečná hodnota proměnné GUD se může měnit např. v závislosti na NC-programu, nebo kanálu ve kterém je naprogramována.[7]

Příklad definice proměnné GUD :

```
DEF NCK INT POSLTOOL1001 = 10
```

NCK určuje oblast platnosti proměnné. V tomto případě je oblast platnosti v celém NC-systému. Slovo POSLTOOL1001 je název definované proměnné a = 10 je přiřazení hodnoty 10. Tyto proměnné GUD jsou definovány v souborech s příponou DEF s rezervovanými názvy souborů: MGUD.DEF - globální data výrobce stroje, UGUD.DEF - globální data uživatele, GUD4.DEF, GUD8.DEF, GUD9.DEF - volně definovatelná data uživatele[7]

### 4.1.2. Definice Uživatelských proměnných LUD

Jedná se o lokální proměnné platící pouze v programu nebo podprogramu, ve kterém byly definovány, a které jsou definovány ve NC-programu, jenž při zpracování není hlavním NC-programem. Tyto proměnné jsou vymazány při ukončení zpracování NC-programu, nebo při resetu NC-systému.[7]

### 4.1.3. Definice Uživatelských proměnných PUD

Jsou to globální programové proměnné, které vzniknou z lokálních proměnných vlivem nastavením strojního parametru. Jsou aktivní ve všech NC-programech (všechny programové úrovně) a také tyto proměnné jsou vymazány při ukončení zpracování NC-programu, nebo při resetu NC-systému. [7]

## 4.2. Použití R-parametrů

### 4.2.1. Definice R-parametrů

R-parametry jsou výpočtové standardní definované uživatelské proměnné programovacího jazyka SINUMERIK, které jsou definované jako pole datového typu REAL. Tyto parametry lze zapisovat dvěma způsoby: R[10] a R10.[7]

V synchronních akcích jako proměnná hlavního zpracování se před R-parametr píše znak \$, např. \$R10.[7]

Tyto parametry se dají nastavit buď v tabulce R-parametrů řídicího systému SINUMERIK, nebo je lze nastavovat v NC-programu. Řídicí systém Sinumerik přednostně zpracovává nastavení z NC-programu a až po té se přepírají hodnoty z různých nastavovacích tabulek. Tato vlastnost slouží k tomu, aby bylo možné opracovávat více typů obrobků.[7]

### 4.2.2. Použití R-parametrů

Využití R-parametrů je pro dvoukanálový obráběcí stroj, kde se musí vyhodnocovat jeho aktuální stav v obou kanálech, nevhodné, protože R-parametry jsou jen kanálově specifické, takže když se například do R-parametru R12 v 1. kanále dosadí hodnota 35, tak v 2. kanálu se tento parametr nezmění. R-parametry jsou kanálově specifické

Tato vlastnost se dá změnit využitím podprogramu s přenosem a zápisem parametrů, jenž oba hlavní programy vyvolají na začátku obrábění jednotlivých nástrojů (pro synchronizaci vřeten) a vyhodnotí hodnotu R-parametru určeného k zapsání čísla již dokončeného obráběcího nástroje.

## 4.3. Použití kombinace uživatelských proměnných a R-parametrů

R-parametry by bylo možné pro tento typ stroje a vyhodnocení použít v případě, že by se zkombinovaly s uživatelskými proměnnými GUD, kde by se použily pro vyhodnocení v rámci jednoho kanálu a pro vyhodnocení při startu obráběcího stroje by byly použity GUDy do nichž by se hodnoty R-parametrů zapsaly.

#### 4.4. Vyhodnocení jednotlivých řešení

Program napsaný pomocí kombinace R-parametrů a GUDů by byl zbytečně složitý a nepřehledný, nelze totiž R-parametry pojmenovat jinak, než přiřazením čísla k adrese R.

Využití podprogramu s přenosem a zápisem R-parametrů by odstranilo nutnost definování proměnných a NC-program by byl zpracován rychleji. Avšak zásadní nevýhodou práce s R-parametry v sériové výrobě na strojích, které jsou jednoúčelové a mají být přeprogramovány je problém s identifikací volných R-parametrů. Pokud výrobce stroje nedodá seznam R-parametrů, jež nejsou v žádném z programů (NC, PLC) využity, pak uživatel stroje nemá možnost určit, který R-parametr může použít, aniž by došlo k porušení vyhodnocovacích procesů stroje.

Na základě výše popsaných výhod a nevýhod v této práci použiji k řešení výhradně globálně uživatelské proměnné GUD.

Nevýhodou tohoto řešení je, že v případě přidání nového nástroje, se musí změnit podmínky v podprogramech TOOLGUD, TOOLGUD\_2 a v hlavních obráběcích programech upravit synchronizaci, musí se přidat definice GUD pro tento nový nástroj (viz. Program pro definici) a upravit kontrola pravděpodobnosti v POSLTOOL (max. počet nástrojů). Přidání nového nástroje je tedy náročné, ale protože je tento program určen pro sériovou výrobu, tak složitost změny není směrodatná. Podstatné je, aby byl stroj co nejrychleji schopen provozu a aby byla co nejmenší tvorba neshodných dílů.

## 5. Sestavení NC-programu

Pro tvorbu NC-programu je třeba zohlednit, jak vřetena v jednotlivých kanálech pracují a kolik obrobků se obrábí současně.

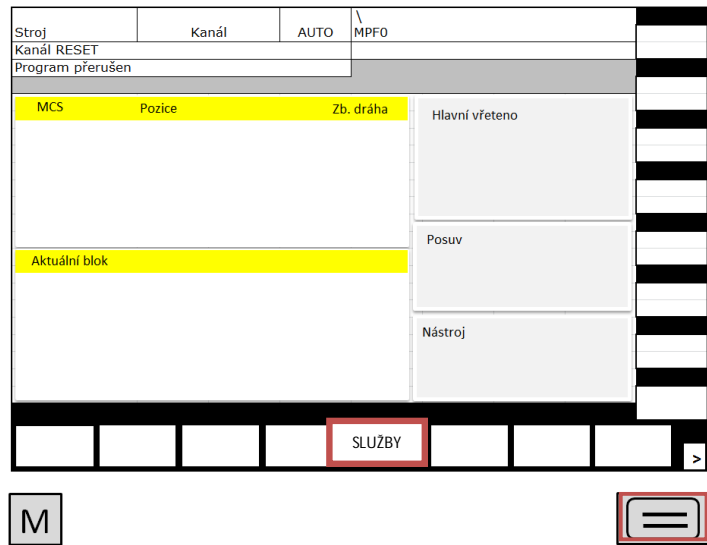
Při opracování celého obrobku jen jedním vřetenem v jednom kanálu, kde je těchto kanálů a k nim přiřazených vřeten více lze využít jen jednoho NC-programu s tím, že bude jeden kanál hlavní a ostatní budou vlečené (nastavení ve strojních parametrech). Tímto způsobem se obrábí dva obrobky najednou. Při tomto způsobu obrábění není nutné využívat skupinu uživatelských proměnných GUD, protože je opracování řízeno jen jedním kanálem a nemusím tedy přenášet informaci z jednoho kanálu do druhého. V tomto případě lze řešit zjištění posledního dokončeného obráběcího nástroje pomocí podmínek (IF, ENDIF), které díky programovým skokům (GOTO) na návěští ovlivní místo v NC-programu, kde se začne s jeho zpracováním. Pro samotné vyhodnocení posledního nástroje lze nadefinovat jen jeden GUD, který se nastaví do hodnoty nacházející se na konci každé části NC-programu určené pro daný nástroj. Každý obráběcí nástroj bude mít svou vlastní specifickou hodnotu a podle hodnoty GUDu při začátku zpracování NC-programu se podmínkami a podmíněnými skoky NC-program dostane do části programu určené pro obráběcí nástroj, který následuje za hodnotou uvedenou v GUDu.

V této práci bych se chtěla zaměřit na typ obráběcího stroje, kde jsou dvě vřetena, a každé z těchto vřeten je řízeno jedním kanálem. Obrábí se jeden díl a vřetena pracují střídavě. Ve chvíli, kdy pracuje např. 1. Kanál ve druhém probíhá výměna obráběcího nástroje. Tímto způsobem lze eliminovat časové ztráty způsobené manipulací s obráběcím nástrojem.

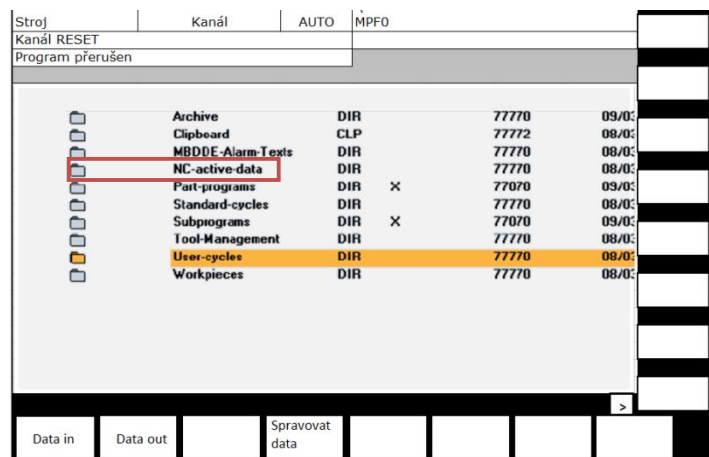
## 5.1. Definice globálně uživatelských proměnných GUD

Předávání informací o naposledy použitém nástroji se v tomto řešení bude provádět pomocí globálně uživatelských proměnných GUD, protože s těmito informacemi musí pracovat oba kanály pro zajištění synchronizace obráběcích vřeten (obrábění jednoho dílu více vřeteny).

Postup: Nejprve definuji uživatelské proměnné pro podprogram TOOLGUD, který bude obsahovat vyhodnocení GUD proměnných POSLTOOL1..., které budou nastavovány v hlavním obráběcím programu podle stavu zpracovávaného programu. Pro účely přenášení informací o nástroji využiji Globálně uživatelská data, která nadefinuji ve stroji v Systémové oblasti, kde se na horizontální liště zvolí SW tlačítko SLUŽBY (obrázek č.



Obrázek 5 - popis editace Gud proměnných



Obrázek 6 - popis editace Gud proměnných

5). Nyní se dostávám do oblasti zálohování, obnovy a definic. V seznamu adresářů vyberu název NC\_ACT (obrázek č. 6), kde pokračuji do souboru GUD4.DEF. Při editaci tohoto souboru se na panelu MMC zobrazí všechny definice GUD - dat, které stroj používá. Kurzorem najedu na místo, kam budu přidávat definice nových GUD-dat a pomocí HW-tlačítka IMPUT vytvořím nové řádky pro zápis (viz obrázek č. 7). Místo přidání nových definic je omezeno pouze syntaxí programu (např. netvořit nové řádky za M30 (standartní funkce pro ukončení zpracování NC-programu) a zachovat první cca tři řádky programu pro jeho identifikaci). Pro načtení nově definovaných proměnných je nutné jejich aktivace, která se provede stisknutím SW tlačítka AKTIVACE na vertikální liště.

Definice uživatelských proměnných GUD:

```
_NC_ACT.DIR/_N_GUD4_DEF/_N_DEF_DIR/DEF.DIR\;;;PROMENE    PRO    PROGRAM  
TOOLGUD
```

```
DEF NCK INT POSLTOOL1001  
DEF NCK INT POSLTOOL1002  
DEF NCK INT POSLTOOL1003  
DEF NCK INT POSLTOOL1004  
DEF NCK INT POSLTOOL1005  
DEF NCK INT POSLTOOL1006  
DEF NCK INT POSLTOOL1007  
DEF NCK INT POSLTOOL1008  
DEF NCK INT POSLTOOL1009  
DEF NCK INT POSLTOOL1010  
DEF NCK INT POSLTOOL  
DEF NCK INT PORADI  
DEF NCK INT POSLZAVM8  
DEF NCK INT POSLZAVM8_1  
DEF NCK INT POSLZAVM8_2  
DEF NCK INT POSLZAVM8_3  
DEF NCK INT POSLZAVM8_4  
DEF NCK INT POSLZAVM6  
DEF NCK INT POSLZAVM6_1  
DEF NCK INT POSLZAVM6_2  
DEF NCK INT POSLZAVM6_3  
DEF NCK INT POSLZAVM6_4  
M30
```

Obrázek 7 – definice uživatelských proměnných GUD

Nyní má systém definované druhy a názvy nových uživatelských proměnných a proto je lze použít v samotných NC-programech. Rozhodování, jaký nástroj byl naposledy použitý, se bude provádět v programech TOOLGUD pro 1. kanál a TOOLGUD\_2 pro 2. Kanál viz obrázek č. 8. Protože při obrábění se oba kanály střídají, nespustí obě vřetena současně, druhý kanál čeká, až bude inicializován v prvním kanále a až první nástroj T1="T1001" dokončí opracování, tak lze mít rozhodovací funkce v jednom programu, který se bude vyvolávat vždy na začátku obráběcího programu. Nejprve se vyvolá v prvním kanálu a po inicializaci druhého kanálu se vyvolá i v obráběcím programu pro druhé vřeteno.

<b>TOOLGUD.SPF:</b>	<b>TOOLGUD_2.SPF:</b>
<b>IF POSLTOOL=2 ENDIF</b>	<b>IF POSLTOOL=3 ENDIF</b>
<b>IF POSLTOOL=4 ENDIF</b>	<b>IF POSLTOOL=5 ENDIF</b>
<b>IF POSLTOOL=6 ENDIF</b>	<b>IF POSLTOOL=7 ENDIF</b>
<b>IF POSLTOOL=8 ENDIF</b>	<b>IF POSLTOOL=9 ENDIF</b>
<b>IF POSLTOOL=10 ENDIF</b>	<b>IF POSLTOOL=0 ENDIF</b>
<b>IF POSLZAVM8_1==1 POSLZAVM8=2 ENDIF</b>	<b>IF POSLZAVM6_1==1 POSLZAVM6=2 ENDIF</b>
<b>IF POSLZAVM8_2==1 POSLZAVM8=3 ENDIF</b>	<b>IF POSLZAVM6_2==1 POSLZAVM6=3 ENDIF</b>
<b>IF POSLZAVM8_3==1 POSLZAVM8=4 ENDIF</b>	<b>IF POSLZAVM6_3==1 POSLZAVM6=4 ENDIF</b>
<b>IF POSLZAVM8_4==1 POSLZAVM8=0 ENDIF M17</b>	<b>IF POSLZAVM6_4==1 POSLZAVM6=0 ENDIF M17</b>

Obrázek 8 - syntaxe podmínek pro zjištění již zpracované části NC-programu

Tento systém je možný za předpokladu, že každý nástroj bude mít svou uživatelskou proměnnou, která bude mít globální působnost (příkaz NCK v definiční větě). Dále je nutné zajistit dostatečnou RAM paměť systému pro práci s těmito proměnnými. Velikost potřebné paměti RAM závisí na počtu používaných obráběcích nástrojů.

V tomto případě názvy uživatelských proměnných sloužících ke zjištění stavu dokončení opracování nástroje pojmenují po názvu nástroje, se kterým pracuje NC-program.

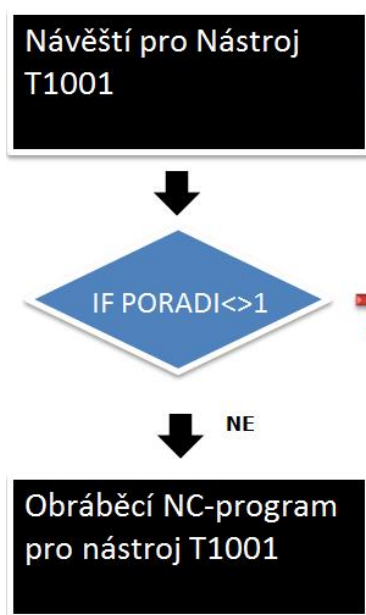
Každá uživatelská proměnná POSLTOOL100... je vyhodnocena pomocí podmínky IF, a pokud bude nastavená na hodnotu 1, pak bude jasné, že tato část NC-programu byla již úspěšně zpracována a



proto se do proměnné POSLTOOL запиše poslední číslo následujícího nástroje. Tato proměnná se pak dále vyhodnocuje v obráběcím programu a přímo určuje, kde stroj začne zpracovávat NC-program.

Proměnné POSLZAVM8\_.... a POSLZAVM6\_.... se také nastavují v obráběcím programu a týkají se závitníku pro řezání závitu o velikosti M8 a M6. Tyto závity se sice vyrábějí jedním nástrojem (jeden pro závity M8 a jeden pro závity M6), ale v případě, že dojde k přerušení zpracování NC-programu v průběhu závitování (v NC-programu je naprogramováno např. 10 závitů, ale vlivem nějaké chyby a následného ukončení zpracování NC-programu je opracováno pouze 7 děr), tak nedojde k natavení proměnné POSTOOL100..., a proto by se řezání závitů při opětovném spuštění zpracování Nc-programu opakovalo a mohlo by dojít k deformaci již vyřezaného závitu.

## 5.2. Synchronizace kanálů



Obrázek 9 – vývojový diagram pro synchronizaci kanálů pomocí podmínky IF

Protože se jedná o dvouvřetenový stroj, který obrábí jeden díl, vřetena se v opracování střídají. Je tedy nutné zajistit jejich synchronizaci, aby bylo zachováno správné pořadí technologického postupu v hlavním obráběcím programu. Tuto synchronizaci zajišťuje další globálně uživatelská proměnná typu integer (celočíselné hodnoty s možností požití znaménka), jenž jsem pro tento účel pojmenovala PORADI. Proměnná PORADI se neustále cyklicky vyhodnocuje (smýčka), dokud nenabude hodnoty povolující další zpracování NC- programu. V tomto případě se naprogramovaná smyčka provádí tak dlouho, dokud bude splněná podmínka `IF PORADI<>1 GOTOB CEKEJ_1`. Tato podmínka je naprogramována na začátku každého nástroje, kde zjišťuje stav zpracování NC-programu sousedního vřetena testováním uživatelské proměnné PORADI, která se před výměnou obráběcího nástroje na další v pořadí tohoto vřetena nastavuje na takovou hodnotu, jenž patří dalšímu obráběcímu nástroji v technologickém pořadí, to znamená v sousedním vřetenu (ná vaznost kanálů je v obrázku č. 10 znázorněna šipkou).

Část hlavního obráběcího programu pro 1. kanál HLPORG\_1.MPF, která zajišťuje synchronizaci.

#### 1. kanál

**Začátek** obrábění v 1. kanálu

```
CEKEJ_1:  
IF PORADI<>1 GOTOB CEKEJ_1  
T1="T1001"
```

```
:  
Obráběcí program  
pro nástroj T1001
```

```
:  
PORADI=2  
Výměna nástroje  
z T1001 na T1003
```

**Pokračování** obrábění v 1. Kanálu

```
CEKEJ_3:  
IF PORADI<>3 GOTOB CEKEJ_3  
T1="T1003"
```

```
:  
Obráběcí program  
pro nástroj T1003
```

```
:  
PORADI=4  
Výměna nástroje  
z T1003 na T1005
```

```
:  
M30
```

Část hlavního obráběcího programu pro 2. kanál HLPORG\_2.MPF, která zajišťuje synchronizaci.

#### 2. kanál

2. kanál čeká na dokončení zpracování NC-programu v prvním kanálu pro nástroj T1001

**Pokračování** obrábění v 2.

```
CEKEJ_2:  
IF PORADI<>2 GOTOB CEKEJ_2  
T2="T1002"
```

```
:  
Obráběcí program  
pro nástroj T1002
```

```
:  
PORADI=3  
Výměna nástroje  
z T1002 na T1004
```

**Pokračování** obrábění v 2.

```
CEKEJ_4:  
IF PORADI<>4 GOTOB CEKEJ_4  
T2="T1004"
```

```
:  
Obráběcí program  
pro nástroj T1004
```

```
:  
PORADI=5  
Výměna nástroje  
z T1004 na T1006
```

```
:  
PORADI=1  
:  
M30
```

Obrázek 10 – ukázka synchronizace kanálů (od nástroje T1001 do T1005)

Synchronizace kanálů je prováděna příkazem IF, na jehož základě buď dojde k přechodu na další programovou větu (v případě nesplnění podmínky), nebo je podmínka splněná a v tom případě pomocí příkazu GOTOB se program vrátí na návěští CEKEJ\_... , takže kanál čeká dokud se hodnota v GUD proměnné PORADI nenastaví na požadovanou hodnotu (tzn. dokud druhý kanál nedokončí opracování). Na konci obráběcího NC-programu pro druhý kanál se GUD proměnná **PORADI** nastaví na hodnotu 1, aby systém pomocí podmínky **IF PORADI<>1 GOTOB CEKEJ\_1** mohl spustit opracování dílu od začátku, protože byl NC-program zpracován až dokonce. PORADI=1 se musí zapsat na konec NC-programu pro kanál, který pojede jako poslední. Pokud by totiž končil opracování první kanál, tak pomocí podmínky pro další nástroj by nedošlo ke spuštění opracování dílu nástrojem, který by následoval po nastavení PPOADÍ=1.

Následující podkapitoly jsou věnovány definicím jednotlivých příkazů použitých pro naprogramování synchronizace kanálů.

### 5.2.1. Definice programové smyčky (IF, ENDIF)

Programová smyčka (IF) má za úkol rozhodnout jakým způsobem bude NC-program dál zpracováván. V tomto případě programové slovo vyhodnotí, zda je splněna podmínka **IF PORADI<>1 GOTOB CEKEJ\_1** (viz obrázek č. 9). Jestliže je hodnota v uživatelské proměnné různá od hodnoty 1, pak NC-program přeskočí spět na návěští CEKEJ\_1. To znamená, že smyčka probíhá tak dlouho, dokud druhé vřetenem nedokončí opracování. V tomto případě se neprogramuje ukončení podmínky ENDIF, protože podmínka není složená (neobsahuje žádné další podmínky) a je nežádoucí, aby program pokračoval dál ve zpracování, což by se stalo, pokud by podmínka nebyla splněná a následná instrukce s akcí programovaná ve větě se neprovedla (v našem případě GOTOB CEKEJ\_1).

Syntaxe: IF (příkaz pro smyčku)<podmínka>(rozhodovací část smyčky – v našem případě: PORADI<>1 GOTOB CEKEJ\_1 )ENDIF (instrukce pro ukončení platnosti smyčky a návrat na začátek smyčky)

Tuto smyčku lze rozšířit o příkaz ELSE v případě, že je několik možností jak pokračovat ve zpracování NC-programu. Tento příkaz umožňuje provést další akci, pokud není podmínka ve větě s příkazem pro smyčku splněna.[7]

Syntaxe: IF <podmínka>  
... ELSE (Další možnost v pokračování zpracování NC-programu, pokud není podmínka u smyčky splněna)...

ENDIF [7]

### 5.2.2. Definice programových skoků (GOTOB, GOTOF, GOTO, GOTOC)

Programovými skoky na návěští v tomto případě lze po vyhodnocení podmínky `IF PORADI<>2` přejít na část programu, která chronologicky nenásleduje větě, v níž je skok programován. To zde umožňuje čekání jednoho kanálu, než druhý kanál dokončí zpracování určitého úseku NC-programu.

Stejnou funkci, kterou realizují programováním skoků GOTOB a podmínkou IF lze nahradit příkazem WAITM (Marker-Nr., n,n,...), který zastaví zpracování Nc-programu. Do hodnoty Marker-Nr se dosadí číslo, podle něhož systém v NC-programech pro oba kanály vyhledá všechny naprogramované příkazy WAITM se stejnou hodnotou Marker-Nr. Následující číslo v závorce uvádí kanál, ve kterém se příkaz nachází a třetí hodnota toho příkazu udává pořadí, v jakém budou zpracovávány. [7]

Tato práce je založena na využití GUD-proměnných za použití příkazů GOTOF, GOTOB, a to i v případech, kde by se daly využít standardní funkce programovacího jazyka SINUMERIK jako je funkce WAIT, která je popsána v této podkapitole. Tato funkce, která je standardní funkcí programovacího jazyka SINUMERIK 840D, by mohla nahradit podmínky pro synchronizaci vět (viz kapitola 5.2). Já jsem funkci WAIT nepoužila, protože vzhledem ke složitosti případných úprav v tomto programu považuji za důležité, co nejvíce NC-program zpřehlednit a to znamená využít možnost pojmenování GUD dle vlastní potřeby. Oproti tomu funkce WAIT se dá rozlišit pouze čísly a jejich pořadím v následné závorce, a pakliže je nutné tuto funkci použít i v technologické části určené pro samotné opracování obrobku (např. zabránění otřesům vzniklých pohybem vedlejšího větěna při manipulaci s obráběcím nástrojem (vykládání / nakládání) právě když obráběcí větěno závituje), pak může docházet k záměně těchto funkcí, protože si jednoduše programátor poplete pořadí čísel. Toto byl můj důvod, proč jsem zvolila použití GUD-proměnných, i když by teoreticky zpracování NC-programu tím pádem mělo trvat déle - tyto proměnné systém musí složitě vyhodnocovat. Navýšení doby zpracování takto upraveného NC-programu je však zanedbatelné vzhledem k celkovému času opracování. Výhoda příkazu WAIT je v tom, že se nemusí definovat další uživatelské proměnné a tím se ušetří dynamická paměť DRAM.

Obecně pro programové skoky platí, že vložením návěští skoků (label) lze na ně skákat z jiných míst téhož programu pomocí příkazů GOTOF, GOTOB, GOTO, nebo GOTOC. Zpracování programu pak pokračuje na následné větě, která je napsána za návěštím skoku. [7]

Pomocí těchto příkazů se provádí větvení v programu. Toto větvení je možné použít jen v rámci jednoho programu. Při vnoření do dalších podprogramů funkce skoků již není aktivní. Lze jen skočit na tu část programu, kde se volá podprogram pro vnoření, nebo je napsána inicializace hlavních

programů do dalších kanálů. Jako návěští (cíl skoku – label) lze také použít čísla bloků (např. N10, ..atd.).[7]

Jestliže je ve větě programován pouze příkaz skoku, pak se tato operace provede vždy, když se tato věta zpracovává. Pokud je skok programován spolu s nějakou podmínkou, pak se skok zrealizuje, jen pokud je podmínka splněna. [7]

V případě, že je programován skok bez podmínky, pak musí být napsán na samostatném řádku, kdežto když je u skoku naprogramovaná podmínka, pak toto pravidlo neplatí (např. funkce větvení CASE, která je v tomto programu použita pro zjištění, jestli a kde byl NC-program předčasně ukončen).[7]

Věta NC-programu `IF PORADI<>2 GOTOB CEKEJ_2` znamená: Jestliže je hodnota uživatelské proměnné různá od hodnoty 2, pak se provede příkaz skoku zpět na návěští CEKEJ\_2.

Příkaz GOTOB hledá požadované návěští před programovaným skokem (směrem k začátku programu) a příkaz GOTOF prohledává program za programovaným skokem (směrem ke konci programu).[7]

Příkaz GOTO prohledává požadované návěští oběma směry. Nejprve se provede funkce GOTOF a když návěští není nalezeno, tak se provede funkce GOTOB.[7]

Pokud není návěští nalezeno, pak se přeruší zpracování NC-programu a systém vygeneruje chybové hlášení 14080 "Cíl skoku nenalezen". Tato funkce se dá potlačit použitím příkazu GOTO, kde je NC-program prohledán jako u příkazu GOTO ,ale v případě, že návěští není nalezeno, pak pokračuje zpracování NC-programu větou, která je naprogramována za příkazem skoku, jehož návěští nebylo nalezeno. [7]

Funkci GOTO a GOTO, v tomto programu nevyužívám, protože proces vyhledávání se tím prodlouží a vzhledem k požadovanému efektu (čekání na druhý kanál) je návěští pro cíl skoku programováno vždy před větou s podmínkou a programovým skokem. Návěští skoku (label) se programuje na začátek bloku a musí být programováno na samostatný řádek, kde lze ještě napsat jen číslo řádku. Návěští se musí skládat z min. dvou a max. 32 znaků. Lze ho sestavit z písmen, číslic a znaku podtržení s tím, že první dva znaky musí být buď podtržení, nebo písmena. Za jménem návěští se musí napsat dvojtečka. [7]

CEKEJ\_3: (návěští) [7]

IF PORADI<>3 GOTOB CEKEJ\_3 (podmínka IF s programovým skokem)[7]

T1="T1003" (pokračování NC-programu, pokud není podmínka splněná)[7]

### 5.2.3. Předávání os mezi kanály (využití příkazů GET, RELEASE)

Interpolace jedné nebo více os, příp. vřeten se smí uskutečňovat vždy jen v jednom kanálu. Pokud musí osa pracovat ve dvou různých kanálech střídavě (např. podavač pro výměnu palet), potom tato osa musí být napřed v aktuálním kanálu uvolněna a pak v jiném kanálu převzata. Kanály si osu vyměňují.[7] Příklad syntaxe příkazů GET a RELEASE je uvedena v obrázku č. 11.

Ukázka z obráběcího NC-programu  
pro 1. Kanál HLPORG\_1.MPF:

#### 1. kanál

*Začátek obrábění v 1. kanálu*

**Převzetí os pro obrábění**

**GET(AX1,AX5)**

:

Obráběcí program  
pro nástroj T1001

:

**Odevzdání os  
pro obrábění v 2. kanálu**

**RELEASE(AX1,AX5)**

Výměna nástroje  
z T1001 na T1003

*Pokračování obrábění  
v 1. Kanálu nástrojem T1003*

**Převzetí os  
pro obrábění v 1. kanálu**

**GET(AX1,AX5)**

:

Obráběcí program  
pro nástroj T1001

:

Ukázka z obráběcího NC-programu  
pro 2. Kanál HLPORG\_2.MPF:

#### 2. kanál

*2.kanál čeká na dokončení*

*zpracování NC-programu*

*pro nástroj T1001*

*Pokračování obrábění*

*v 2. Kanálu*

**Převzetí os  
pro obrábění v 2. Kanálu**

**GET(AX1,AX5)**

:

Obráběcí program  
pro nástroj T1002

:

**Odevzdání os  
pro obrábění v 1. kanálu**

**RELEASE(AX1,AX5)**

Výměna nástroje z T1002  
na T1004

Obrázek 11 – předávání os mezi kanály

Dosazení hodnoty do proměnné PORADI řadím až za předání osy, protože by se mohlo stát, že dojde k vynucenému ukončení NC-Programu ještě před předáním os a stroj by na základě informace z proměnné PORADI dostal pokyn k zahájení obrábění např. v 1. kanálu, ale aktivní osy by byly ve druhém.

Syntaxe:

RELEASE(název osy, název osy, ...) nebo RELEASE(S1)[7]

GET(název osy, název osy, ...) nebo GET(S2)[7]

GETD(název osy, název osy, ...) nebo GETD(S3)[7]

Pomocí příkazu GED (GET Directly) se osa přímo vyzvedne z nějakého jiného kanálu. To znamená, že k tomuto příkazu GETD nemusí být naprogramován žádný odpovídající příkaz REELASE v nějakém jiném kanálu. Znamená to ale také, že nyní musí být s oním kanálem navázána komunikace (např. značky pro čekání - v tomto případě návěští: CEKEJ\_1 až 10).[7]



### 5.3. Zjišťování stavu obrobení po vynuceném ukončení zpracování NC-programu

Vyhodnocení se provede v hlavním obráběcím programu. Hlavní obráběcí program pro 1. kanál pojmenuji HLPROG\_1.MPF a pro druhý zvolím název HLPROG\_2.MPF.

Rozhodování o startu programu na základě hodnoty v GUD proměnné POSLTOOL bude umístěno na začátek programu po počátečním nastavení stroje (kontrola najetí do základní polohy, definice lokálních proměnných, inicializace a spuštění programu v druhém kanálu, nastavení a zkontrolování vstupů pro externí periferie, naložení dílu a kontrola jeho upnutí, zapnutí dodatečných funkcí a programů (Prometec - kontrola zlomení obráběcího nástroje, atd...).

Následující podkapitoly jsou věnovány naprogramování Zjišťování stavu obrobení po vynuceném ukončení zpracování NC-programu a definicím jednotlivých příkazů, které jsou v programu použity.

#### 5.3.1. Rozhodování pomocí příkazu CASE pro podmíněné větvení programu

Funkce CASE zjišťuje nastavení různých proměnných, R-parametrů, atd... (typ: INT) a podle jejich hodnoty, nebo výsledku umožňuje skočit v programu na různá návěští (podobně jako funkce GOTO...).[7]

Syntaxe:       CASE(<název proměnné>) OF <hodnota proměnné (1. možnost)> GOTOF <cíl skoku pro 1.možnost>< hodnota proměnné (2.možnost )> GOTOF <cíl skoku pro 2. možnost> ...  
                  DEFAULT GOTOF < cíl skoku pro možnost funkce DEFAULT >[7]

Tato věta znamená, že pokud je hodnota v proměnné rovna hodnotě pro 1. možnost, pak program pomocí funkce GOTO.... (lze použít všechny druhy skoků) skočí na místo v programu označené jako cíl skoku pro 1. možnost. Pokud se hodnota proměnné nerovná hodnotě pro první možnost, pak se hodnota proměnné porovná s hodnotou pro 2. možnost. Toto porovnání se provádí tak dlouho, dokud nedojde ke shodě s jednou z možností, nebo dokud tyto možnosti nedojdou, pak se pokračuje další programovou větou (pokud není programována funkce DEFAULT) [7]

Funkce DEFAULT: Tuto funkci lze využít, když je potřeba vždy skočit na stejné místo v NC-programu, kromě některých výjimek. Tyto výjimky se pak rozepíší do funkce CASE. Pokud se nebude hodnota

proměnné shodovat s těmito výjimkami, pak program bude skákat na místo zapsané za funkcí DEFAULT.[7]

V případě, že není funkce DAFAULT použita a hodnota proměnné se neshoduje s hodnotami definovanými v příkazu CASE, pak se pokračuje následnou programovou větou za větou s příkazem CASE.[7]

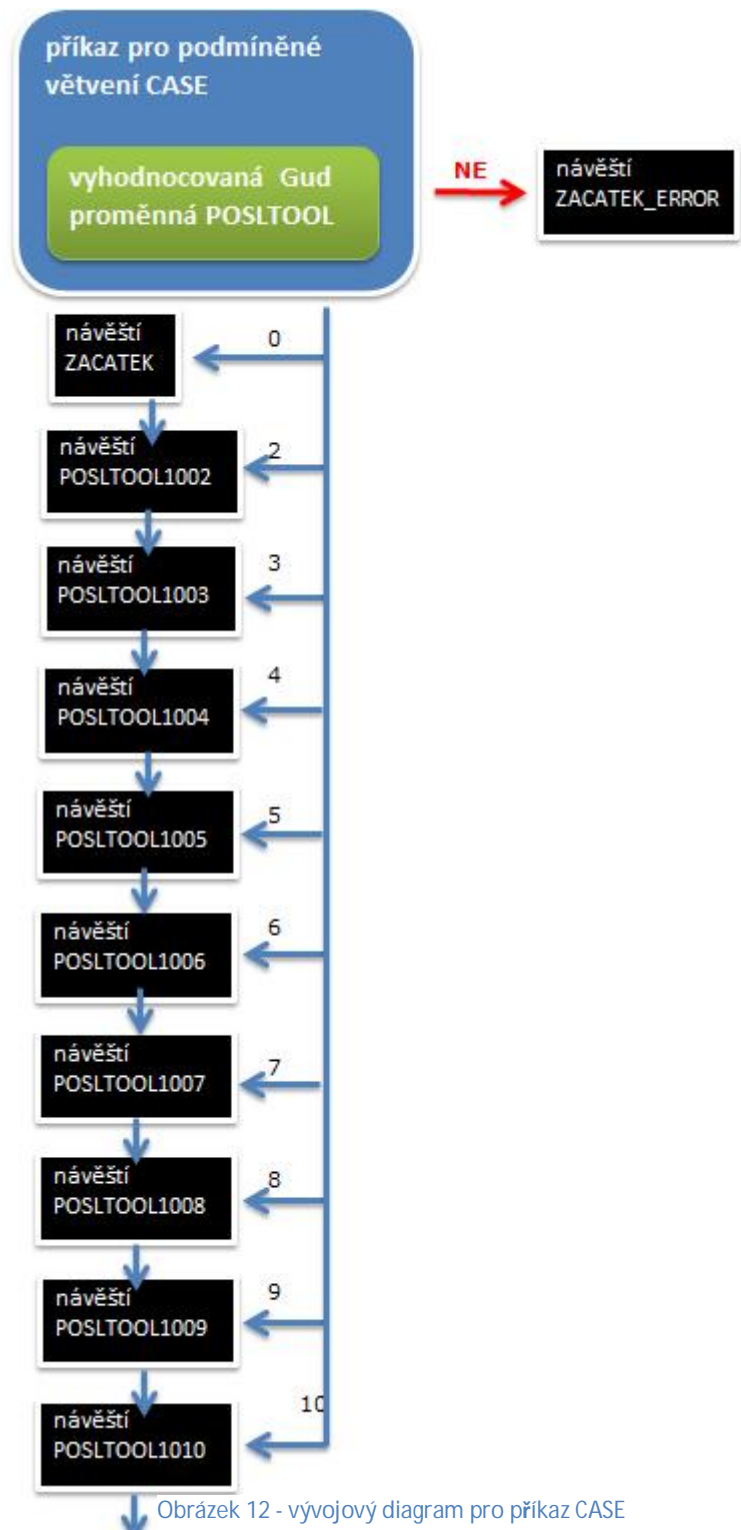
Použití příkazu CASE při rozhodování, který nástroj byl již použit (viz. obrázek č. 12).

```

CASE POSLTOOL OF 0 GOTOF
ZACATEK2 GOTOF POSLTOOL1002 3
GOTOF POSLTOOL1003 4 GOTOF
POSLTOOL1004 5 GOTOF
POSLTOOL1005 6 GOTOF
POSLTOOL1006 7 GOTOF
POSLTOOL1007 8 GOTOF
POSLTOOL1008 9 GOTOF
POSLTOOL1009 10 GOTOF
POSLTOOL1010 DEFAULT GOTOF
ZACATEK_ERROR
    
```

Význam věty: Jestliže se hodnota GUD proměnné POSITTOOL rovná 0, pak se program začne zpracovávat od návěští ZACATEK, který je umístěn hned za větvením CASE a to znamená, že obrábění dokončily všechny nástroje anebo žádný, vynucené ukončení NC programu proběhlo před dokončením obrábění nástroje "T1001".

Jestliže se hodnota GUD proměnné POSITTOOL rovná 2, pak se program začne zpracovávat od návěští POSITTOOL1002, jenž je umístěn před příkazem pro výměnu nástroje "T1002", protože obrábění dokončil jen nástroj "T1001".



Jestliže se hodnota GUD proměnné nerovná žádné hodnotě, pak se program začne zpracovávat od návěští **ZACATEK\_ERROR**, který je umístěn hned před návěštím ZACATEK a to znamená, že z nějakého důvodu byla nastavena hodnota, která neodpovídá žádné hodnotě určené v řádku programovaného větvení. Může se to stát při pokusu o přidání/smazání obráběcího nástroje, nebo nepodařené obnově dat (výměna NCU, manipulace s programy,..). Za návěštím ZACATEK\_ERROR bude umístěna M-funkce M0, což je podmíněné zastavení a lze znovu stroj spustit stisknutím hardwarového tlačítka START CYKLU na ovládacím panelu stroje. Při najetí na funkci M0 bude zároveň vygenerováno varovné hlášení povelom MSG, nebo SETAL o možnosti chyby ve volbě obráběcího nástroje. Seznámení s funkcí MSG obsahuje následná podkapitola.

### 5.3.2. Hlášení (MSG)

Povel MSG vypíše text na obrazovku stroje do pole hlášení, které je na obrázku č. 13 červeně vyznačeno.

Syntaxe:

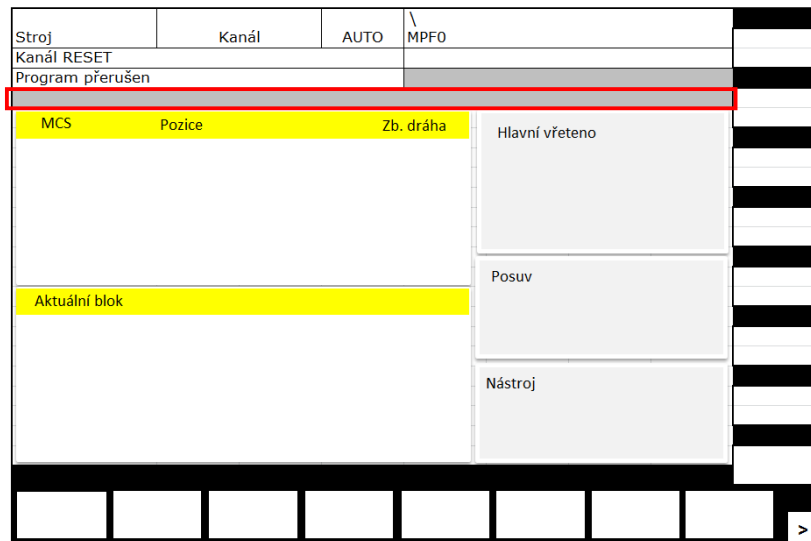
MSG("<libovolný řetězec znaků typu STRING s maximální délkou 124 znaků na dvou řádcích (2 x 62

znaků)>"[,<zde se volí, jakým způsobem má stroj reagovat

na stav, který zapříčinil skok na část programu, kde se nachází tento povel MSG. Pokud je zde nastavena 0, pak zpracování NC-programu probíhá dál, jen se na obrazovku vypíše text. Pokud je zde nastavena 1, pak se zpracování NC-programu zastaví a stroj se musí spustit znovu od začátku.>]).[7]

Do textového pole lze pomocí operátoru "<<" vypisovat proměnné, pomocí nichž lze zobrazit jejich aktuální stav. Například lze tímto způsobem operátorovi stroje dát vědět o nesprávně zadané hodnotě korekce, nebo v našem případě nesprávně zadaném číslu do proměnné POSITool.[7]

Vypsany text hlášení vyvolaného příkazem MSG je na obrazovce ovládacího panelu stroje (MMC)aktivní. [7]



Obrázek 13 – místo pro zobrazení MSG

#### 5.4. Naprogramování zjištění naposledy dokončeného obráběcího nástroje do obráběcího programu.

V této práci se zabývám možnostmi programovacího jazyka SINUMERIK 840D a syntaxi pro předávání informací z jednoho kanálu do druhého a proto samotnou technologii obrábění v navrhnutém NC-programu, který je pro ukázkou sestaven v příloze, jen naznačuji dvojtečkou. Programové příkazy jsou barevně znázorněny (skoky a jejich cíle) a text, který jen popisuje a vysvětluje jednotlivé použité příkazy a funkce je černý. Pokud by bylo potřeba podobný text naprogramovat i do NC-programu, pak by musel každý řádek, který obsahuje pouze vysvětlující text mít jako první znak středník, tím se text, který se za ním nachází, bere jako komentář a řídicí systém jej nezpracovává.

Ukázkový program je postaven pro 10 obráběcích nástrojů, které se během obrábění střídají. Počet nástrojů je variabilní (dle požadavků na geometrii a kvalitu obráběného dílu).

## 6. Závěr

Cílem této práce bylo, ovlivnit začátek zpracování NC-programu SINUMERIK 840D pomocí jeho ukončení v předešlém obráběcím cyklu. Na základě teoretického rozboru jednotlivých možností (využití GUD proměnných, R-parametrů nebo kombinaci GUD proměnných a R-parametrů) rozhodla jsem se pro použití výhradně GUD proměnných.

Výsledný NC-program zajišťuje snížení seřizovacího času po vynuceném přerušení zpracování NC-programu a tvorbu neshodných dílů vlivem opakovaného najetí obráběcího nástroje do již opracované části obrobku. NC-program také obsahuje ochranu proti nesprávnému zadání čísla posledního ukončeného nástroje (např. při opravách).

Nevýhodou výše popsaných změn je poměrně složitější přidání/odebrání obráběcího nástroje (viz. Kapitola 4.4), ale vzhledem k typu výroby (sériová) je doba editace NC-programu nepodstatná. Změny je totiž možné provést mimo obráběcí stroj a po té je do něj nahrát, nebo je lze provádět přímo na obráběcím stroji v předem připravené kopii aktuálně zpracovávaného NC-programu. Obě tyto možnosti nevyžadují narušení výrobního procesu.

NC-program, který je výsledkem této práce byl odzkoušen na tréninkových stojanech s řídicím systémem SINUMERIK od firmy SIEMENS s kladným výsledkem. NC-program splnil očekávání, protože po simulaci chyb, které by ve výrobním procesu mohly nastat a jež mají za následek přerušení chodu stroje (přerušení dodávky proudu, HW porucha pohonů a osových karet, syntaktické chyby v NC-programu na různých místech), NC-program rozpoznal poslední dokončený obráběcí nástroj a začal zpracování NC-programu po opětovném spuštění stroje dalším v pořadí.

## 7. Zdroje

### 7.1. Seznam literatury

- [1] Dokumentace k ŘS Sinumerik 840D
- [2] Keller, P. Programování a řízení CNC strojů, Presentace přednášek, KVS P5 TU v Liberci. Liberec 2005
- [3] Články v časopisech a na internetu

### 7.2. Zdroje z internetu

- [4] POLZER, Aleš. Akademie-cnc-obrabeni: akademie-cnc-obrabeni-4. [online]. s.2 [cit. 2014-05-23]. DOI: DOI. Dostupné z: [http://www.technickytydenik.cz/rubriky/serialy/akademie-cnc-obrabeni/akademie-cnc-obrabeni-4\\_8539.html](http://www.technickytydenik.cz/rubriky/serialy/akademie-cnc-obrabeni/akademie-cnc-obrabeni-4_8539.html)
- [5] Alfing: Kessler Sondermaschinen GmbH. ALFING KESSLER SONDERMASCHINEN GMBH. *ALFING KESSLER SONDERMASCHINEN GmbH* [online]. DE [cit. 2014-06-19]. Dostupné z: <http://aksde.alfing.de/de/home.html>
- [6] SIEMENS. [online]. [cit. 2014-06-19]. Dostupné z: <http://www.siemens.com/answers/cz/cz/>
- [7] SIEMENS. *SIEMENS: Základy* [online]. [cit. 2014-06-19]. Programovací příručka: 840D sl / 840DE sl / 840. Dostupné z: [http://cache.automation.siemens.com/dnl\\_iis/DE/DE0NzEwOQAA\\_28705635\\_HB/PG\\_0310\\_cs\\_cs-CZ.pdf&rct=j&frm=1&q=&esrc=s&sa=U&ei=EbeiU4WDMufi4QTUwYGYBg&ved=0CBoQFjAB&usg=AFQjCNHhA0PRrsrsyP2kLEYjYVNGuSnGZQ](http://cache.automation.siemens.com/dnl_iis/DE/DE0NzEwOQAA_28705635_HB/PG_0310_cs_cs-CZ.pdf&rct=j&frm=1&q=&esrc=s&sa=U&ei=EbeiU4WDMufi4QTUwYGYBg&ved=0CBoQFjAB&usg=AFQjCNHhA0PRrsrsyP2kLEYjYVNGuSnGZQ)

## 8. Seznam obrázků

Obrázek 1 – obráběcí centrum[5] .....	11
Obrázek 2- řízení a MMC pro SINUMERIK 840D[6].....	11
Obrázek 3 – znázornění konstrukce obráběcího centra[5] .....	11
Obrázek 4 – příklad obráběného odlitku[5] .....	12
Obrázek 5 - popis editace Gud proměnných .....	21
Obrázek 6 - popis editace Gud proměnných .....	21
Obrázek 7 – definice uživatelských proměnných GUD .....	22
Obrázek 8 - syntaxe podmínek pro zjištění již zpracované části NC-programu .....	23
Obrázek 9 – vývojový diagram pro synchronizaci kanálů pomocí podmínky IF .....	24
Obrázek 10 – ukázka synchronizace kanálů (od nástroje T1001 do T1005).....	25
Obrázek 11 – předávání os mezi kanály .....	30
Obrázek 12 - vývojový diagram pro příkaz CASE .....	33
Obrázek 13 – místo pro zobrazení MSG.....	35



## 9. Přílohy

### 9.1. Hlavní program pro 1. kanál

`% HLPROG_1_MPF`

:

; Začátek programu tvoří podmínky a funkce nutné pro zapnutí a kontrolu periférií stroje (např. ; nastavení nulových bodů, nakladače, upnutí, různé druhy chlazení, ofuků a zapnutí kontroly zlomení nástroje s jeho identifikací, atd..).

:

**GOTOF POKRAC** ; Nepodmíněný skok pro přeskočení části programu sloužící k ohlášení špatné identifikace posledního obráběcího nástroje.

Následující část programu začínající návěstím ZACATEK\_ERROR zastaví M-funkcí M0 zpracování NC-programu z důvodu splnění podmínky IF POSLTOOL>10 kontrolující, zda nebyla do proměnné POSLTOOL napsána nepravděpodobná hodnota. Tato podmínka má význam např. při opravném obrábění, kde byl obrobek již opracován, ale vlivem seřízení stroje nebo obráběcího nářadí, je nutné část obroku obrobít znovu, takže se do proměnné POSLTOOL v obrazovce UŽIVATELSKÁ DATA zapíše číslo obráběcího nástroje, který předchází nástroji, jenž chceme použít.

**ZACATEK\_ERROR:** ; Návěstí na které program přejde po zjištění, že do proměnné POSLTOOL ; bylo napsáno špatné číslo nástroje. Na toto návěstí program přejde po splnění podmínky IF POSLTOOL>10.

**MSG (" možnost chyby ve volbě obráběcího nástroje")** ; instrukce MSG vypíše text v závorce na obrazovku stroje do pole hlášení. V tomto případě upozorní obsluhu na chybně nastavený parametr POSLTOOL.

**M0** ; Pokud je v GUD proměnné POSLTOOL zadaná nepravděpodobná hodnota a v případě nesplněné podmínky ze zjišťování zda nástroj naložený ve vřetenu (připravený k obrábění) souhlasí s nastavenou GUD proměnnou , se pomocí této M-funkce 1.kanál zastaví, a protože druhý kanál pomocí synchronizace čeká na ukončení obrábění v prvním kanálu, tak je obrábění v obou kanálech pozastaveno.

**MSG (" ")** ; tento tvar instrukce pro vypsaní textu na obrazovce stroje vymaže vygenerované hlášení o chybě v GUD proměnné POSLTOOL.

**POKRAC:**

**TOOLGUD** ; Na tomto řádku je zavolán ke zpracování podprogram TOOLGUD (viz kapitola 5.1. Definice globálně uživatelských proměnných GUD), jenž zjišťuje poslední použitý obráběcí nástroj z předchozího zpracování hlavního obráběcího programu. Následuje příkaz k podmíněnému větvení, kde podle GUD proměnné POSLTOOL zjišťují, z jaké části hlavního programu se bude v tomto případě začínat obrábět.

**CASE POSLTOOL OF 0 GOTOF ZACATEK 3 GOTOF POSLTOOL1003 5 GOTOF POSLTOOL1005 7 GOTOF POSLTOOL1007 9 GOTOF POSLTOOL1009** ; Příkazem Case NC-program skáče do různých částí obráběcího programu podle hodnoty v uživatelské proměnné POSLTOOL (viz kapitola 5.3.1. Rozhodování pomocí příkazu CASE).

**ZACATEK:** ; Návěští, které slouží jako cíl skoku u větvení CASE v případě, že program byl v předchozím obráběcím cyklu zpracován celý, nebo byl přerušen hned u prvního nástroje (T1001).

**IF POSLTOOL>10** ; Tato podmínka zjišťuje, zda nebyla dosazena nepravděpodobná hodnota do uživatelské proměnné POSLTOOL. Pokud je podmínka splněná, to znamená, že hodnota v proměnné, je větší než 10, pak je jasné, že hodnota v této proměnné je špatná, protože se počítá s tím, že obráběcí program nemá více než deset obráběcích nástrojů a program přejde na další řádek, kde dojde ke skoku směrem k začátku NC-programu na návěští ZACATEK\_ERROR. V případě, že by bylo potřeba obráběcí nástroj přidat, pak by se porovnávací číslo v této podmínce muselo zvětšit. Naopak, pokud by bylo možné některý obráběcí nástroj z NC-programu odstranit (např. díky využití vícechodých nástrojů, zjednodušení obrobku atd...), pak by bylo dobré toto číslo zmenšit pro zachování zamýšlené funkce této kontroly.

**GOTOB ZACATEK\_ERROR**

**ENDIF** ; Pokud nebude podmínka IF POSLTOOL>10 splněna, pak NC-program přeskočí instrukci pro splněnou podmínku a přejde na řádek, který následuje za ukončujícím příkazem ENDIF.

**PORADI=1** ; Nastavení proměnné GUD pro synchronizaci obrábění s druhým kanálem k prvnímu nástroji "T1001"

**INIT(2,"/\_N\_WKS\_DIR/\_OBROBEK\_WPD/\_HLPROG\_2\_MPF")** ; Příkaz INIT navolí NC-program do druhého kanálu (obrábění druhým vřetenem), to znamená, že operátor při počátečním zpuštění

stroje (první zprovoznění, oprava, změna typu dílu) navolí obráběcí NC-program jen do prvního kanálu, jenž druhý kanál programově řídí.

**START(2)** ; Tento příkaz spustí zpracování NC-programu, který byl v předchozí větě navolen v druhém kanálu.

**GET(AX1,AX5)** ; Zde NC-program převezme obráběcí osy do prvního kanálu (viz kapitola 5.2.3. Předávání os mezi kanály).

; Následující část programu má za úkol zajistit synchronizaci kanálů (viz kapitola 5.2. Synchronizace kanálů)

CEKEJ\_1:

**IF PORADI<>1 GOTOB CEKEJ\_1**

**T1="T1001"** ; V tomto řádku určí obráběcí nástroj, kterým se bude díl opracovávat. Tento příkaz slouží k určení nástroje, který bude M-funkcí M6 v následné větě dosazen do vřeten. M-funkce M6 je standardní funkcí programovacího jazyka SINUMERIK pro výměnu obráběcího nástroje.

**M6**

; Následující podmínka zjišťuje, zda je ve správné kombinaci obráběcího nástroje a hodnoty v proměnné POSLTOOL. V případě, že je podmínka splněná, to znamená, že je ve vřetení upnut nástroj T1001 a současně hodnota proměnné POSLTOOL je 0, pak program přejde na větu, kde je ukončení podmínky ENDIF a pokračuje na následný řádek, což je v tomto případě samotné technologické opracování dílu. Tato podmínka je nutná pro případ, že by byl NC-program přerušen v místě, kde bylo již provedeno větvení CASE. V automatickém režimu lze zamezit možnosti spuštění zpracování NC-programu bez najetí základní polohy (to lze provést naprogramováním podmínky do PLC), ale v poloautomatickém režimu se může stát, že stroj půjde spustit z té části NC-programu, kde bylo zpracování přerušeno. Tato nastavení závisí na výrobcí stroje, resp. na naprogramování PLC. V případě, že by NC-program po přerušení pokračoval z místa, kde skončil, by se mohlo stát, že například při zlomeném závitníku kdy se musí nedoopravený díl vyndat, opravit ručně, a naložit nový, ještě neopracovaný díl, by opět došlo k poškození obráběcího nástroje, protože by nebyla pro závit připravená díra.

**IF (\$TC\_TP2[\$TC\_MPP6[9998,1]]=="T1001") AND (POSLTOOL ==0)** ; Strojní parametr \$TC\_TP2[\$TC\_MPP6[9998,1]] obsahuje číslo obráběcího nástroje, který je právě upnut ve vřetení v 1.

kanálu. To, že se má zkontrolovat upnutý nástroj ve vřetení určuje číslo 9998 a číslo kanálu udává hodnota 1.

**ELSE** ; pokud podmínka není splněna, to znamená, že se neshoduje číslo obráběcího nástroje s hodnotou POSLTOOL, pak program přeskočí na řádek se zpětným bezpodmínečným skokem GOTOB ZACATEK\_ERROR do části programu s chybovým hlášením MSG (" možnost chyby ve volbě obráběcího nástroje").

**GOTOB ZACATEK\_ERROR** ; zpětný bezpodmínečný skok

**ENDIF** ; Ukončení působení podmínky.

:

; obráběcí technologický program pro nástroj "T1001"

:

**RELEASE(AX1,AX5)** ; Po dokončení opracování dílu obráběcím nástrojem T1001 stroj pokračuje v opracování dílu druhým vřetenem a mezi tím se provádí výměna obráběcího nástroje v prvním vřeteně. Pro možnost obrábění v druhém kanále je nutné v prvním kanálu uvolnit osu X a rotační osu vřetene S (viz kapitola 5.2.3. Předávání os mezi kanály). Osy Y a Z jsou definovány pro každý kanál zvlášť jako Y1, Y2 a Z1, Z2.

**PORADI=2**; Nastavení GUD proměnné na hodnotu, která signalizuje, že obrábění prvním obráběcím nástrojem bylo dokončeno a že obrábění musí pokračovat v druhém kanále, protože v tomto kanále se čeká na hodnotu 3

**POSLTOOL1001=1** ; Nastavení GUD proměnné na hodnotu, která signalizuje, že obrábění prvním obráběcím nástrojem bylo dokončeno a že obrábění (kanál 1) v případě vynuceného zastavení může pokračovat až od druhého nástroje (T2="T1002").

;-----

; Nyní pracuje druhý kanál (T2="T1002")

;-----

**POSLTOOL1003:** Návěští pro identifikaci místa, odkud má program začít obrábět, pokud byl vynucen stop zpracování NC-programu po ukončení nástroje T2="T1002"

**T1="T1003"** ; Určení obráběcího nástroje

**M6** ; M-funkce pro výměnu obráběcího nástroje ve vřetení.

**TOOLGUD** ; Opětovné vyvolání podprogramu pro aktualizaci GUD proměnné POSLTOOL, aby bylo možné ověřit, že nástroj, který je upnutý ve vřetení 1 se shoduje s částí programu pro něj určenou

**IF (\$TC\_TP2[\$TC\_MPP6[9998,1]]=="T1003") AND (POSLTOOL ==3)** ; Jestliže je název nástroje upnutého ve vřetení 1. kanálu T1003 a současně GUD proměnná POSLTOOL má hodnotu 3, pak program pokračuje ve obrábění, pokud ale není podmínka splněná, pak program skočí na začátek programu do části ohlašující chybu identifikace nástroje.

**ELSE**

**GOTOB ZACATEK\_ERROR**

**ENDIF**

**CEKEJ\_3:** ;smyčka pro synchronizaci kanálů

**IF PORADI<>3 GOTOB CEKEJ\_3**

**GET(AX1,AX5)** ; převzetí os X a S k použití v prvním kanálu (viz kapitola 5.2.3. Předávání os mezi kanály).

:

; obráběcí program pro nástroj T1003

:

**RELEASE(AX1,AX5)** ; uvolnění os X a S k použití v druhém kanálu (viz kapitola 5.2.3. Předávání os mezi kanály).

**PORADI=4** ; Nastavení GUD proměnné na hodnotu, která signalizuje, že obrábění třetím obráběcím nástrojem bylo dokončeno a že obrábění musí pokračovat v druhém kanále, protože v tomto kanále se čeká na hodnotu 5

**POSLTOOL1003=1** ; Nastavení GUD proměnné na hodnotu, která signalizuje, že obrábění třetím obráběcím nástrojem bylo dokončeno a že obrábění (kanál 1) v případě vynuceného zastavení může pokračovat až od pátého nástroje (T1="T1005"), pokud byl nástroj T2="T1004" v druhém kanálu také dokončen.

Teoreticky bych mohla pro rozeznání naposledy dokončeného obráběcího nástroje použít Gud;proměnnou PORADI, ale pro lepší orientaci v programech raději použiji jinou proměnnou (POSLTOOL...). V těchto programech také bude rozhodování, zda se jedná o závitník a v případě, že ano, tak který závit byl naposledy dokončený.

;------

;Nyní pracuje druhý kanál (T2="T1004")

;------

**POSLTOOL1005:** ;Návěští pro identifikaci místa, odkud má program začít obrábět, pokud byl vynucen stop zpracování NC-programu po ukončení nástroje T2="T1004"

:

;Program pro nástroj T1="T1005

:

**POSLTOOL1005=1** ; Nastavení GUD proměnné na hodnotu, která signalizuje, že obrábění pátým obráběcím nástrojem bylo dokončeno a že obrábění (kanál 1) v případě vynuceného zastavení může pokračovat až od sedmého nástroje (T1="T1007"), pokud byl nástroj T2="T1006" v druhém kanálu také dokončen.

:

;------

;Nyní pracuje druhý kanál (T2="T1006")

;------

**POSLTOOL1007:** ;Návěští pro identifikaci místa, odkud má program začít obrábět, pokud byl vynucen stop zpracování NC-programu po ukončení nástroje T2="T1005"

**T1="T1007"** ;Určení obráběcího nástroje

**M6** ; M-funkce pro výměnu obráběcího nástroje ve vřetení.

**TOOLGUD** ; Opětovné vyvolání podprogramu pro aktualizaci Gud proměnné POSLTOOL, aby bylo možné ověřit, že nástroj, který je upnutý ve vřetení 1 se shoduje s částí programu pro něj určenou

`IF ($TC_TP2[$TC_MPP6[9998,1]]=="T1007") AND (POSLTOOL ==7) ;` Jestliže je název nástroje upnutého ve vřetení 1. kanálu T1007 a současně GUD proměnná POSLTOOL má hodnotu 7, pak program pokračuje ve obrábění, pokud ale není podmínka splněná, pak program skočí na začátek programu do části ohlašující chybu identifikace nástroje.

`ELSE`

`GOTOB ZACATEK_ERROR`

`ENDIF`

`CEKEJ_7: ;`smyčka pro synchronizaci kanálů

`IF PORADI<>7 GOTOB CEKEJ_7`

`GET(AX1,AX5) ;` Převzetí os z druhého kanálu pro použití v prvním kanálu

; Nástroj T1007 je závitník, který řeže závity M8. V případě vynuceného zastavení zpracování NC-programu je důležité vědět které z děr již byly opracovány a proto naprogramuji větvení CASE i v této části NC-programu, protože u závitů lze jen těžko provést strojní opravu a i v tom případě, že se v NC-programu pro závitování naprogramuje příkaz SPOSA. Tímto příkazem lze docílit, že se vřeteno natočí vždy do stejné polohy, takže by nemělo dojít k přeřezání již vytvořených závitů. V praxi však dojde k jejich deformaci, to znamená, že závit je volný a náběhová hrana je též seříznutá.

`GOTOF POKRAC_M8 ;`Nepodmíněný skok pro přeskočení části programu sloužící k ohlášení špatné identifikace již opracovaných závitů.

; Následující část NC-programu má za úkol vypsát chybové hlášení na obrazovku panelu a sdělit operátorovi stroje, že byla do proměnné POSLZAVITM8 zapsána nepravděpodobná hodnota. Pokud v této proměnné bude jiná hodnota, než ta, která se shoduje s některou z hodnot, jež jsou v ní naprogramované, pak program pomocí zpětného skoku přejde na návěští ZAVM8\_ERROR:. Tímto skokem se NC-program dostane do části, kde se pomocí MSG funkce vypíše chybové hlášení a zpracování NC-programu se zastaví na M-funkci M0. Po zkontrolování obráběného dílu a nastavení proměnné lze ve zpracování NC-programu pokračovat stisknutím strojního tlačítka CYKLUS START na ovládacím panelu stroje. Po spuštění stroje NC-program přejde na další řádek, kde prázdným příkazem MSG vymaže chybové hlášení z obrazovky a program znovu přejde na funkci větvení CASE, kde se provede opětovné rozhodování, které z děr mají být opracovány.

`ZAVM8_ERROR: ;`Návěští pro skok funkce CASE při zadání nepravděpodobné hodnoty.

MSG ("zkontrolovat stav závitů M8") ; Příkaz pro vypsání chybového hlášení na obrazovku ovládacího panelu.

M0 ; M-funkce pro zastavení zpracování NC-programu.

MSG (" ") ; Příkaz pro vymazání chybového hlášení z obrazovky ovládacího panelu.

POKRAC M8: ;Návěští pro nepodmíněný programový skok GTOF POKRAC\_M8, kterým přeskakují část programu určenou pro chybně zadanou hodnotu proměnné POSLZAVITM8.

CASE POSLZAVM8 OF 0 GOTOF ZAVM8\_1 2 GOTOF ZAVM8\_2 3 GOTOF ZAVM8\_3 4 GOTOF ZAVM8\_4  
DEFAULT GOTOB ZAVM8\_ERROR ; Programové větvení pro zjištění, který závit před přerušením NC programu byl naposledy obroben.

ZAVM8\_1: ;Návěští pro skok funkce CASE při nastavení proměnné POSLZAVITM8 na hodnotu 0, která značí, že v předchozím obráběcím cyklu byly vytvořeny všechny závity, nebo naopak ani jeden.

:

; Technologický program pro závitování díry č.1

:

POSLZAVM8\_1=1 ; Nastavení GUD proměnné na hodnotu, která signalizuje, že obrábění 1. závitu bylo dokončeno a v případě vynuceného zastavení může závitník pokračovat až od 2. závitu.

ZAVM8\_2: ;Návěští pro skok funkce CASE při nastavení proměnné POSLZAVITM8 na hodnotu 1, která značí, že v předchozím obráběcím cyklu byl vytvořen závit pouze na první díře.

:

; Technologický program pro závitování díry č. 2

:

POSLZAVM8\_2=1 ; Nastavení GUD proměnné na hodnotu, která signalizuje, že obrábění 2. závitu bylo dokončeno a v případě vynuceného zastavení může závitník pokračovat až od 3. závitu.

ZAVM8\_3: ;Návěští pro skok funkce CASE při nastavení proměnné POSLZAVITM8 na hodnotu 2, která značí, že v předchozím obráběcím cyklu byl vytvořen závit pouze na první a druhé díře.

:



; Technologický program pro závitování díry č.3

:

**POSLZAVM8\_3=1** ; Nastavení GUD proměnné na hodnotu, která signalizuje, že obrábění 3. závitů bylo dokončeno a v případě vynuceného zastavení může závitník pokračovat až od 4. závitů.

**ZAVM8\_4:** ; Návěští pro skok funkce CASE při nastavení proměnné POSLZAVITM8 na hodnotu 3, která značí, že v předchozím obráběcím cyklu byl vytvořen závit pouze na první, druhé a třetí díře.

:

; Technologický program pro závitování díry č.4

:

**POSLZAVM8\_4=1** ; Nastavení GUD proměnné na hodnotu, která signalizuje, že obrábění 4. závitů bylo dokončeno.

**RELEASE(AX1,AX5)** ; uvolnění os X a S k použití v druhém kanálu (viz kapitola 5.2.3. Předávání os mezi kanály).

**PORADI=8** ; Nastavení GUD proměnné na hodnotu, která signalizuje, že obrábění sedmým obráběcím nástrojem bylo dokončeno a obrábění musí pokračovat v druhém kanále, protože v tomto kanále se čeká na hodnotu 9

**POSLTOOL1007=1** ; Nastavení GUD proměnné na hodnotu, která signalizuje, že obrábění sedmým obráběcím nástrojem bylo dokončeno a že obrábění (kanál 1) v případě vynuceného zastavení může pokračovat až od devátého nástroje (T1="T1009").

;------

;Nyní pracuje druhý kanál (T2="T1008")

;------

; Následuje program pro nástroj T1009. Princip nastavení proměnných je stejný jako u nástrojů ;T1001, T1003, atd....

**POSLTOOL1009:** ;

Program pro nástroj T1="T1009

POSLTOOL1009=1

;------

;Nyní pracuje druhý kanál (T2="T10010") => ukončení obrábění dílu

;------

:

; pro ukončení NC-programu se zde naprogramují funkce a podmínky nutné pro vypnutí a kontrolu periférií stroje vztahované na první kanál (např. nakladače, upnutí, různé druhy chlazení, ofuků a vypnutí kontroly zlomení nástroje; s jeho identifikací, atd..).

:

M30

## 9.2. Hlavní program pro 2. kanál

NC-program pro druhý kanál je napsán stejně jako pro první kanál, jen se volí jiné obráběcí nástroje. Použité funkce a podmínky jsou popsány a vysvětlené, proto pro druhý kanál uvádím jen samotný NC-program, bez popisu.

Čísla a pořadí těchto nástrojů jsou popsány v programu pro první kanál.

```
% HLPORG_2_MPF
```

```
:
```

```
GOTOF POKRAC
```

```
ZACATEK_ERROR:
```

```
MSG (" možnosti chyby ve volbě obráběcího nástroje")
```

```
M0
```

```
MSG (" ")
```

```
POKRAC:
```

```
TOOLGUD_2
```

```
IF POSLTOOL>10
```

```
GOTOB ZACATEK_ERROR
```

```
ENDIF
```

```
CASE POSLTOOL OF 2 GOTOF POSLTOOL1002 4 GOTOF POSLTOOL1004 6 GOTOF POSLTOOL1006 8  
GOTOF POSLTOOL1008 10 GOTOF POSLTOOL1010
```

```
POSLTOOL1002:
```

```
T2="T1002"
```

```
M6
```

```
CEKEJ_2:
```

```
IF PORADI<>2 GOTOB CEKEJ_2
```

```

GET(AX1,AX5)

TOOLGUD

IF ($TC_TP2[$TC_MPP6[9998,2]]=="T1002") AND (POSLTOOL ==2)

ELSE

GOTOB ZACATEK_ERROR

ENDIF

:

;obráběcí program pro nástroj "T1002"

:

RELEASE(AX1,AX5)

PORADI=3

:

POSLTOOL1002=1

;-----

;Nyní pracuje první kanál (T1="T1003")

;-----

POSLTOOL1004:

T2="T1004"

M6

TOOLGUD_2

IF ($TC_TP2[$TC_MPP6[9998,2]]=="T1004") AND (POSLTOOL ==4)

ELSE

GOTOB ZACATEK_ERROR

```

ENDIF

CEKEJ\_4:

IF PORADI<>4 GOTOB CEKEJ\_4

GET(AX1,AX5)

:

; obráběcí program pro nástroj "T1004"

:

RELEASE(AX1,AX5)

PORADI=5

POSTOOL1004=1

;-.....

;Nyní pracuje první kanál (T1="T1005")

;-.....

POSTOOL1006:

Program pro nástroj T2="T1006

POSTOOL1006=1

;-.....

;Nyní pracuje první kanál (T1="T1007")

;-.....

POSTOOL1008:

T2="T1008"

M6

TOOLGUD\_2

```

IF ($TC_TP2[$TC_MPP6[9998,2]]=="T1008") AND (POSLTOOL ==8)

ELSE

GOTOB ZACATEK_ERROR

ENDIF

CEKEJ_8:

IF PORADI<>8 GOTOB CEKEJ_8

GET(AX1,AX5)

GOTOF POKRAC_M6

ZAVM6_ERROR:

MSG ("zkontrolovat stav závitů M6")

M0

MSG (" ")

GOTOB ZACATEK_ERROR

POKRAC M6:

CASE POSLZAVM6 OF 0 GOTOF ZAVM6_1 2 GOTOF ZAVM6_2 3 GOTOF ZAVM6_3 4 GOTOF ZAVM6_4
DEFAULT GOTOB ZAVM6_ERROR

ZAVM6_1:      ; program pro závitování 1. závitu

POSLZAVM6_1=1

ZAVM6_2:      ; program pro závitování 2. závitu

POSLZAVM6_2=1

ZAVM6_3:      ; program pro závitování 3. závitu

POSLZAVM6_3=1

ZAVM6_4:      ; program pro závitování 4. závitu

POSLZAVM6_4=1

```

RELEASE(AX1,AX5)

PORADI=9

POSTOOL1008=1

;------

;Nyní pracuje první kanál (T1="T1009")

;------

POSTOOL1010:

; Program pro nástroj T2="T1010" => konec obrábění dílu

POSTOOL1010=1

:

**PORADI=1** ; Nastavení GUD proměnné pro signalizaci, že poslední nástroj v pořadí opracování dílu byl dokončen a má se při opětovném spuštění obrábění zpravovat NC-program od začátku

; pro ukončení NC-programu se zde také naprogramují funkce a podmínky nutné pro vypnutí a kontrolu periférií stroje vztažené na druhý kanál (např. nakladače, upnutí, různé druhy chlazení, ofuků a vypnutí kontroly zlomení nástroje s jeho identifikací, atd..).

:

M30