

Dialogové obrazovky

Popis:

V systému jsou zabudovány nové obrazovky.

Tyto obrazovky slouží ke změnám parametrů NC programu (soubor .PAR) a předávání informací PLC.

1 Parametry NC programu (soubor .PAR)

Představme si na chvíli, že máme stroj, který vyrábí z plechu jen kolečka a čtverečky různých rozměrů. V takovém případě není vhodné použít klasický NC program, ale je lepší použít NC program parametrický. Zde bude figurovat jako parametr zda se obrábí kolečko nebo čtverec a poloměr kružnice nebo délka strany čtverce. Běžně by se tyto parametry zadaly ručně do tabulky parametrů. Nyní je zde nový prostředek, jak tyto parametry zadat. Za pomoci textového editoru a pomocí pár řádek textu (popis v následujících kapitolách) lze vytvořit dialog, který uživatele nechá graficky zadat parametry a sám se postará o to, aby se tyto parametry uložily do tabulky parametrů. Uživatel už pak jen spustí program a ten bude vyrábět kolečka nebo čtverečky přesně tak, je před tím byly zadány v dialogu.

- 1 Je zapotřebí vyrobit jednoduchou obrazovku, kde se zadají nějaké parametry jako tloušťka materiálu a podobně. Výroba PLC obrazovky by pro takto jednoduchý případ byla zbytečně složitá a zdoluhavá. Potom lze použít nové dialogové obrazovky, které lze rychle vytvořit. Tyto obrazovky mohou vytvářet nejen parametry programů, ale především mohou zapisovat hodnoty do položek STCH v PLC. PLC se tedy ihned dozví, že uživatel změnil tloušťku materiálů z 10 na 5 mm a může tomu přizpůsobit např. Rychlost.

Výhodou těchto obrazovek je jejich snadné vytvoření a jednoduchost zabudování do systému (tyto obrazovky je možné např. spouštět z libovolného tlačítka na panelu systému nebo v kterýkoliv okamžik z PLC).

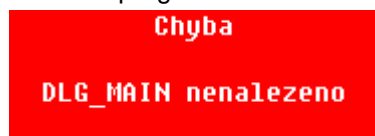
Dalším přínosem obrazovek je to, že mají v sobě zabudovaný prostředek k vypočítávání výrazů. Vzpomeňme jakým způsobem se na systému například vypočítává průměr kružnice z poloměru. Bylo k tomu zapotřebí nastavit parametr R5 a spustit funkci G26. Velmi nepřehledné. Nyní bude vše vypočítáno ještě před tím, než se NC program spustí. Zápisem „2*R20“ dostaneme dvojnásobek parametru R20, ve kterém může být například poloměr kružnice a máme průměr kružnice. Je to mnohem přehlednější.

Zároveň s příchodem obrazovek je v systému zabudovaná i nová funkce. Je to načtení části (nebo všech) parametrů z tabulky parametrů (soubor .PAR). Nyní je možné počítat až se 700 parametry R. Do těchto parametrů můžeme uložit např. souřadnice koncových bodů NC programu. Program pak už bude jezdit jen na dané R parametry a uvnitř programu není zapotřebí vypočítávat nic.

Jejich největší výhodou je to, že existuje jejich verze pod Windows. Na vytvoření obrazovky nyní nepotřebujete systém. Stačí libovolný počítač se systémem Windows. Obrazovku si tak přesně rozvrhne i bez přítomnosti systému, vyzkoušíte a odladíte a pak teprve přenesete na systém.

Vytvoření první obrazovky s verzí pod Windows

Z webových stránek www.mefi.cz stáhneme program [Script Editor](#). Po rozbalení uložíme např. do adresáře „Nyní kdekoli (např. na ploše) vytvoříme textový soubor „TAB_KOCT.SCT“. Na tento soubor ůtkneme pravým tlačítkem myši a zvolíme „Otevřít v programu > Zvolit program...“. Stisknout „Procházet“ a najdeme náš Editor Scriptu který jsem před chvílí uložili (nejčastěji do Program files) a zvolíme „Script Editor.exe“. Zaškrtneme „k otevření souborů tohoto typu...“ a potvrdíme OK. Od této chvíle se jakýkoliv dialog pro systém otevře a zobrazí sám po kliknutí myši. Otevřel se program a v něm červený obdélník



Vlevo dole vidíme čtyři tlačítka



Tlačítko Otevřít slouží k otevření jiné dialogové obrazovky (dále jen scriptu), Upravit otevře textový editor a v něm script pro právě otevřenou obrazovku, Restart restartuje obrazovku, např. po nějaké úpravě a tlačítko Parametry otevře naposledy vytvořenou tabulku parametrů.

Začněme tlačítkem Upravit. Zobrazil se nám prázdný dokument. Napišme sem

DIALOG "Nastavení parametru" "DLG_MAIN"

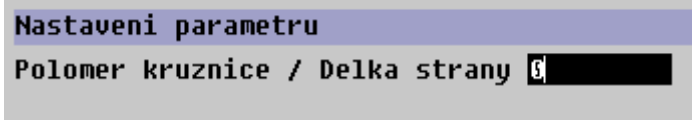
Zavřít dokument uložit a stisknout Restart. Nyní se nám zobrazila dialogová obrazovka s nadpisem „Nastavení parametru“. Je celá prázdná, proto do ní vložíme nějaké položky pro zadání velikosti poloměru nebo strany čtverce opět stiskneme Upravit a na druhý řádek napíšeme

REAL "Polomer kruznice / Delka strany" R10

Uložíme a stiskneme Restart.

Teď už obrazovka dostává smysl. Má nějaký nadpis a nabízí nám možnost změnit nějaký rozměr. Při stisku

jakéhokoliv tlačítka ovšem nereaguje. Je to proto, že jsme obrazovce nesdělili, s jakou tabulkou parametrů má pracovat



vedle scriptu TAB_KOCT.SCT vytvoříme textový soubor TAB_KOCT.PAR (to je tabulka parametrů). Vložíme do něj text:

```
$PAR
```

```
10: 00001.000
```

```
*
```

Upravíme script tak, že na první řádku napíšeme

```
REKFILE "$PAR" "TAB_KOCT.PAR" C96
```

Nyní máme ve scriptu:

```
REKFILE "$PAR" "TAB_KOCT.PAR" C96
```

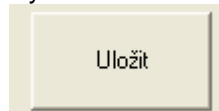
```
DIALOG "Nastavení parametru" "DLG_MAIN"
```

```
REAL "Polomer kruznice / Delka strany" R10
```

Script uložíme a restartujeme. Ve tmavém obdélníku je nyní



Nyní můžeme do této položky psát libovolné číslo. Napíšeme např. 1234.56 a stiskneme



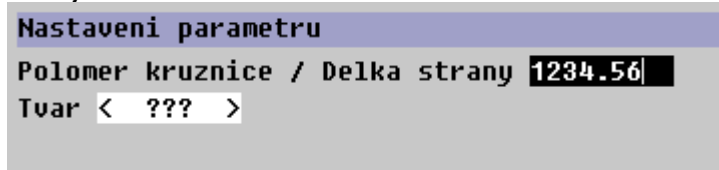
po té stiskneme tlačítko Parametry a vidíme, že text který jsme před tím napsali do do tabulky parametrů se změnil z 00001.000 na +01234.560.

Ted' už vše dává smysl. Máme tabulku parametrů a dialog, který s tabulkou spolupracuje. Do tmavé buňky napíšeme hodnotu a ta se projeví v tabulce. Tuto tabulku si později načte NC program a v parametru R10 bude mít uloženou hodnotu poloměru nebo strany.

Přidejme do Dialogu další položku. Pod řádek REAL napíšeme řádek

```
LIST "Tvar" R11 V73"Kolečko" V78"Ctverec"
```

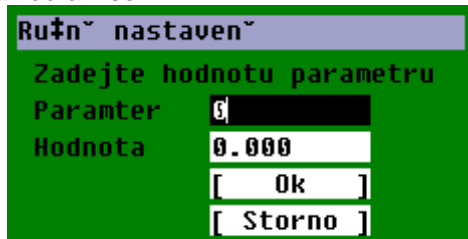
restartujeme obrazovku a vidíme



Vedle textu Tvar jsou tři otazníky. Stiskneme



a zobrazí se



do první řádky zadáme 11, to je číslo parametru R, který chceme upravit. Vzpomeňme si, že jsme do tabulky parametru napsali jen parametr R10 (10: 00001.000), který ovládá buňka „Poloměr kružnice/...“ R11 ovládá buňka „Tvar“. V tabulce parametrů ale R11 není a tak jej přidáme ručně. Nyní stiskneme Ok (tlačítko zeleného dialogu).

Tím jsme přidali parametr R11 do tabulky parametrů a dialog s ním může pracovat.

Pokud je zapotřebí, můžeme kdykoliv měnit hodnotu parametru i z ručního zadání tím, že do prvního řádku zadáme číslo parametru (jako tomu bylo teď) a do druhého řádku přímo zadáváme jeho hodnotu. Potvrdíme Ok a parametr je nastaven. To je pro případ, že parametr není zobrazen v dialogu.

My ale máme parametr zobrazený v dialogu. Nyní po ručním zadání položky je aktivní a můžeme ji libovolně měnit

šipkami vlevo a vpravo. Pokud měníme vidíme, že se mění jen Mezi Kolečko a Čtverec. To je správně, protože na řádce LIST "Tvar" R11 V73"Kolečko" V78"Čtverec" nebyly žádné jiné možnosti nadefinovány. Pokud uložíme, do tabulky parametrů se dostane do R11 jen 78 nebo 73, což je velmi výhodné. Z NC programu můžeme totiž rovnou skočit na příslušnou řádku bez použití porovnávacích operací v G26.

N10 GR11 L30

N20

;ctverec

N0 G73 L40

N30

;kolečko

N0 G73 L40

N40

;konec

také jsme mohli přiřadit parametru R11 hodnotu návěští a v NC programu by to vypadalo takto

N10 G73 LR11.

Možností je několik. Ta první je lepší, pokud rozhodujete mezi dvěma možnostmi a ta druhá se hodí pro rozhodnutí mezi více než dvěma možnostmi.

Podívejme se nyní na další možnost dialogu
na poslední řádek napíšme:

TEXT "První program"

V dialogu se objeví text „První program“

BITMAP "POZADI.BMP" X1 Y1

Na pozadí dialogu se zobrazí obrázek, načtený ze souboru „“. Soubor BMP musí být maximálně 256-ti barevný. Pro zmenšení objemu a zrychlení vykreslování se doporučuje komprese metodou RLE

INT "Celé číslo " R12

Stejně jako položka „REAL "Polomer kruznice / Delka strany" R10“ i tato položka umožňuje zápis čísla. Do takto vytvořené buňky lze ale zapsat pouze čísla celá (bez desetinné tečky)

HEX "Hexadecimální číslo " R13

Podobný REAL a INT, ale vytvoří buňku, do které lze zadat hexadecimální číslo. Toto číslo se pak převede na dekadické a uloží do R13

Více dialogů

Ze zobrazeného dialogu je také možné zobrazit i jiný dialog. To se využívá především, pokud se buňky pro zadání parametrů nevejdou na jednu obrazovku.

Dále se toho využívá, pokud chceme vytvořit sadu dialogů, které se uživatele zeptají jednotlivě nebo po menších skupinkách parametrů na různé rozměry. V každém dialogu bude dotaz na jeden nebo dva rozměry, případně se k němu připojí nějaký komentář a obrázek a každý dialog bude mít v sobě odkaz na další dialog. To způsobí sled dialogů například jako při instalování programu. Nejprve se instalátor zeptá na cestu kam se bude instalovat, po tom se zeptá jestli si přejeme plnou nebo uživatelskou instalaci, pak jazyk, atd. Právě podobný sled lze vytvořit i zde. Nejprve se zeptá na bezpečnou vzdálenost, pak na otáčky, na tloušťku atd. Na konci dá uživatel uložit a parametry jsou uloženy.

Dalším příkladem více dialogu navzájem spouštěných je třeba to, když se požaduje udělat dialogy např. pro obrábění kapsy a ostrůvku. V takovém případě není vhodné z jednoho dialogu zobrazit parametry pro oba. Lepší je pro každý zvlášť vyrobit jiný dialog a z dalšího (hlavního) dialogu tyto dva dialogy zobrazovat.

Více dialogů se hodí i při vytváření nápovědy. Do dialogu vložíte buňku REAL, která umožní zadat desetinné číslo.

Jaký má toto číslo význam potom můžete napsat do jiného dialogu, který zobrazíte, pokud uživatel stiskne Nápověda.

Ukažme si to na příkladu:

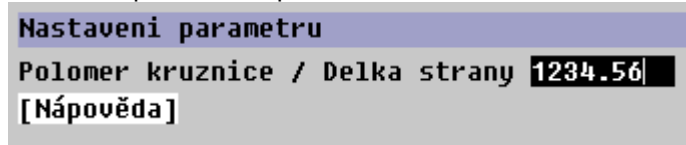
Zkopírujme script TAB_KOCT.SCT, který jsme vytvořili v první části návodu do TAB_DLG.SCT

ponechme jen první tři řádky a ostatní vymažeme. Zbyla nám jen tabulka parametrů, jméno hlavního dialog a buňka pro zadání poměru.

Na další řádek napíšeme:

```
SUB "Nápověda" "DLG_HELP"
```

Uložíme a spustíme script, zobrazí se toto



Přidalo se tlačítko [Nápověda]. Pokud je označeno a stiskneme Enter, otevře se nový dialog.

Tento dialog zatím není nadefinovaný, takže pokud se o to pokusíme, program nám vypíše chybu.

Pojďme jej nadefinovat, celý script pak bude vypadat takto:

```
REKFILE "$PAR" "TAB_KOCT.PAR" C96
```

```
DIALOG "Nastavení parametru" "DLG_MAIN"
```

```
REAL "Polomer kruznice / Delka strany" R10
```

```
SUB "Nápověda" "DLG_HELP"
```

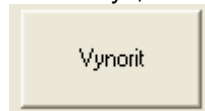
```
DIALOG "Informace" "DLG_HELP"
```

```
TEXT "R11 slouží k ..."
```

```
REAL R11 X20 Y0
```

Co je zde nového? Přibyla další položka DIALOG. Vše co v ní je se zobrazí, pokud uživatel stiskne [Nápověda] z hlavního dialogu. Prvním parametrem položky dialog je jeho nadpis, druhým parametrem je identifikátor dialogu. Ten slouží k rozpoznání při spouštění z příkazu SUB. Podívejme se, že SUB a druhý DIALOG mají stejný identifikátor (DLG_HELP). Proto se po stisku SUB (Nápověda) zobrazí právě DIALOG se stejným identifikátorem jako má SUB. Jen doplním, že dialog, který se zobrazí jako první, musí mít vždy identifikátor DLG_MAIN. Bez tohoto dialogu nelze obrazovku spustit. Další důležitou informací je, že text ve tlačítku u SUB nemusí být stejný jako nadpis dialogu. Podívejme se co ještě je zde nového. Poslední položka REAL neobsahuje žádný text a přibýly jí položky X a Y. Pokud položka neobsahuje text, zobrazí se buňka bez textu. X a Y slouží k umístění v dialogu. Tyto rozměry jsou zadávány na šířku a výšku jednoho znaku. U všech ostatních položek dialogu lze zadat X a Y. Jen u položky TEXT je podmínkou text v uvozovkách. Ostatní jej mít nemusí.

Vraťme se na začátek. Jak vytvořit sled dialogu jako je tomu například u instalátorů aplikací? Pokud do každého dialogu umístím buňku SUB, která bude ukazovat na další dialog, dostanu sled dialogů. Pokud bych se chtěl vrátit o úroveň výš, stisknu tlačítko



Jak rozdělit dialogy na čtverec a kruh? Do hlavního dialogu se vloží dvě položky typu SUB které budou ukazovat na dva nové další dialogy. V hlavním dialogu se tedy objeví dvě tlačítka a při stisku každého z nich se zobrazí jiný dialog. V každém z dialogů budou buňky, do kterých uživatel zadá rozměry. V jednom případě bude zadávat např. poloměr a ve druhém stranu. Oba Tvary od sebe budou graficky odděleny.

Jak ale rozpoznat, že uživatel otevřel ten či onen dialog. Na to existuje položka INIT

```
INIT R20 I1234
```

Těchto položek INIT je možné naskládat do každého dialogu libovolný počet. Ve chvíli, kdy uživatel otevře dialog, a ten obsahuje INIT, nastaví se R (v našem případě R20) na hodnotu I(1234). To je jen ukázkový příklad, když ale nastavíme do R20 návěští v NC programu, tak jako tomu bylo na začátku návodu, můžeme z NC programu skočit na toto návěští a podle toho obrábět kruh nebo čtverec. Číslo návěští se nastaví hned, jakmile uživatel stiskne Enter např. na tlačítko [Čtverec]. Ostatní parametry vyplní v dialogu a je možné obrábět.

Co když máte vaše skripty příliš velké, anebo chcete, aby každý dialog používal jinou tabulku parametrů? Potom se vyplatí pro každý dialog zvlášť (nebo pro malou skupinu dialogů) vytvořit vlastní script. Jak to udělat? Ukažme si to na příkladu. Mějme čtverec a kruh. Vytvoříme samostatný script pro oba:

čtverec:

```
REKFILE "$PAR" "TAB_CTVR.PAR" C96
```

```
DIALOG "Čtverec" "DLG_MAIN"
```

```
REAL "Delka strany" R10
```

kruh:

```
REKFILE "$PAR" "TAB_KRUH.PAR" C96
```

```
DIALOG "Kruh" "DLG_MAIN"
```

REAL "Polomer" R10

Nyní máme dva scripty. Jeden se stará o čtverec a druhý o kruh. K nim vyrobíte dva NC programy pro každý tvar zvlášť a do hlavičky každého z nich dáte příslušný soubor s tabulkou parametrů. Jak ale Vyrobit hlavní script, který nám dá na výběr mezi Kruhem a Čtvercem? Takto:

DIALOG "Kruh" "DLG_MAIN"

SCRIPT "Ctverec" "TAB_CTVR.SCT" R10

SCRIPT "Kruh" "TAB_KRUH.SCT" R10

Příkaz SCRIPT je velmi podobný příkazu SUB. Rozdíl je ten, že na místo identifikátoru se udává jméno souboru scriptu, který chceme otevřít. V prvních uvozovkách tedy zadáme text, který bude zobrazen na tlačítku a druhé uvozovky jsou na jméno souboru. Při stisku tlačítka se pak otevře dialog s identifikátorem DLG_MAIN ze zadaného scriptu.

Pozor na jednu důležitou věc. Script, který obsahuje REKFILE (tabulka parametrů), nemůže obsahovat příkaz SCRIPT.

Výpočty výrazů

Jednou z dalších výhod těchto obrazovek je to, že před uložením tabulky parametrů mohou přepočítat parametry nějakým výrazem. Např. máte zadanou délku a úhel úsečky, s tím ale systém pracovat neumí a tak je zapotřebí přepočítat tyto hodnoty na souřadnice koncového bodu. Jednou z možností je udělat to v NC programu, kde je zapotřebí nastavit R5 a spustit G26. To je velmi nepřehledné a nepraktické. Představme si, že souřadnice koncového bodu lze spočítat ještě před tím, než bude NC program uložen. V každém z nadefinovaných dialogů lze zavést libovolný počet výrazů. To znamená, že každý dialog může vypočítat libovolný vzorec a jeho výsledek uložit do parametru R. Tímto způsobem lze např. předpočítat všechny koncové a středové souřadnice celého programu a v NC programu bude už jen dráha na daný parametr např. N0 G1 XR10 YR11. V programu už potom není třeba nic vypočítávat, až na velmi málo výjimek, které je třeba počítat za chodu NC programu. Ukažme si příklad.

REAL "Delka" R10

REAL "Uhel" R11

CALC "R10*COS(R11)" R12

CALC "R10*SIN(R11)" R13

Tento jednoduchý příklad vypočítá koncový bod, zná-li úhel a délku úsečky.

Jak je to s podmínkami? Pokud je zapotřebí vypočítávat jen pokud je platná nějaká podmínka, stačí tuto podmínku uvést do druhých uvozovek. Zde je příklad:

CALC "R10*2" "R11>20" R10.

Pokud bude R11>20, potom se R10 zdvojnásobí.

Zde je tabulka všech operátorů a funkcí, které lze použít při vypočítávání:

Funkce:

SIN – Sinus(x), x je ve stupních

COS – Cosinus(x)

TAN – Tangent(x)

ASIN – ArcSinus(x)

ACOS – ArcCosinus(x)

ATAN2 – ArcTangent2(y,x) – vypočítá úhel úsečky z bodu [0,0] do [x,y]

ATAN – ArcTangent(x)

SQRT - odmocnina

ABS – absolutní hodnota

INT – oříznutí na celé číslo

ROUND – zaokrouhlení na celé číslo

NEG – obrácení znaménka

PI – konstanta PI

Operátory:

+ - * / () = > < <> >= <=

pokud chceme v podmínce použít AND nebo OR používáme k tomu * a +. Zde je příklad:

CALC "R10+R20" "(R1>1)*(R2>10)" R30 – zde je podmínka R1 musí být větší než 1 a zároveň(AND*) R2 musí být větší než 10, potom se výsledek uloží do R30

CALC "R10+R20" "(R1>1)+(R2>10)" R30 – zde je podmínka R1 musí být větší než 1 nebo(OR+) R2 musí být větší než 10, potom se výsledek uloží do R30

Vzhled a řád při psaní skriptů

Zde napíšu několik pravidel pro vytváření skriptů. Při jejich dodržení je zaručena jejich funkčnost.

- 1 přípona scriptu by měla být vždy .SCT
- 2 pokud script edituje tabulku parametrů, mělo by jeho jméno začínat TAB. Výsledné jméno souboru by potom bylo TAB?????.SCT
- 3 v každém scriptu je možné používat // jako komentář
- 4 příkazy DIALOG a REKFILE by měly začínat od levého kraje obrazovky
- 5 ostatní příkazy je dobré od levého kraje obrazovky oddělit tabulátorem

V souboru „vzorový vzhled.txt“ najdete vzorový vzhled scriptu. Většina příkazů v něm budou v maximální verzi, tj. budou mít zadány všechny parametry. Veškerý popis bude v komentářích.

Využívání obrazovek z PLC

Tyto obrazovky můžeme využívat i v PLC. Kam se ale uloží hodnoty z buněk typu REAL, INT a pod? Tabulka parametrů z PLC přeci využít nejde. Pro PLC nebudeme zapisovat REAL R10, ale REAL M1.10. Rozdíl je ten, že v případě scriptu pro NC program potřebujeme uložit data na disk, zatímco pro PLC potřebujeme uložit data do paměti na určenou adresu. Zápis REAL M1.10 znamená uložit do STCH_IN_FIELD(1.) ,ulož na offset 10. Pro STCH_OUT_FIELD by to vypadalo REAL M2.10. Pro jiný offset potom REAL M1.20, kde 20 je právě offset. Tyto obrazovky umějí ukládat jen na STCH_IN_FIELD a STCH_OUT_FIELD. To ale téměř vždy stačí, jde přeci o to vytvořit jednoduchou obrazovku. Ve skutečnosti dialog pro PLC funguje tak, že všude tam, kde bylo napsáno C[cislo] nebo R[cislo] (př. R11, R54, atd.) lze textově nahradit Mfield.offset (př. M1.20, M2.115, atd.). Výsledek je ten, že se bude pracovat s pamětí kterou ovládá PLC a ne s tabulkou parametrů. Zkrátka i pro PLC je vytvoření dialogových obrazovek velmi jednoduché.

Všechny ostatní příkazy fungují naprosto stejně jako je tomu v případě dialogu pro tabulku parametrů. Dokonce lze použít i kombinované obrazovky. Jeden řádek bude mít cíl parametr a druhý paměť

REAL „Do parametru“ R10

REAL „Do PLC“ M1.10

Pro PLC není zapotřebí vkládat do scriptu příkaz REKFILE. V případě PLC jsou všechny buňky aktivní a do všech lze zapisovat. Proto jednoduchý a funkční script pro PC by mohl vypadat takto:

DIALOG "První obrazovka pro PLC" "DLG_MAIN"

REAL M1.0

REAL M1.4

Tento jednoduchý dialog nabídne možnost uživateli zadat dvě desetinná čísla. Po tom, co uživatel stiskne uložit, se do STCH_IN_FIELD na offset 0 uloží první číslo a na offset 4 druhé. Pozor! Data, které dialog uloží do PLC jsou ve formátu 32 bitů se znaménkem (dvojkový doplněk). Proto se musí zadat cíl M1.0 a M1.4 o 4 bajty posunutý.

Stejně jako obrazovky zapisují do paměti PLC, tak i PLC může něco sdělit obrazovkám. Pokud na určitou adresu zapíše PLC nějakou hodnotu, může si tuto hodnotu obrazovka přečíst.

Například chcete sdělit obrazovce, aby vynulovala všechny buňky v dialogu. Na dohodnutou adresu např. M1.20 zapíšeme hodnotu 1, to bude znamenat, že má obrazovka nulovat své buňky. Script potom vypadá takto

DIALOG "Nulování bunek" "DLG_MAIN"

CALC "0" "M1.20=1" M1.0 A

CALC "0" "M1.20=1" M1.4 A

CALC "0" M1.20 A

REAL M1.0

REAL M1.4

Další důležitou věcí je to, že po ukončení dialogu se zapíše do paměti při uložení 1 a při ukončení bez uložení 2 na adresu STCH_IN_HEAD+STCH_IN_REZER a STCH_OUT_HEAD+STCH_OUT_REZER. To může PLC ve smyčce kontrolovat a při změně se dozví, že byla obrazovka ukončena a jakým způsobem k ukončení došlo. Ihned po té je zapotřebí tuto buňku vynulovat, aby dialog při příštím spuštění mohl opět oznamovat ukončení.

Pro PLC lze použít všechny příkazy dialogu proto lze i něco vypočítávat a to i v kombinaci s parametry, např:

CALC "(M1.0 + R10) / M1.5" C10 nebo CALC "(M1.0 + R10) / M1.5" M1.50

Vložení obrazovek do systému

pro vložení obrazovky do řídicího systému je zapotřebí:

- 1 nahrát script do systému, nejčastěji do CMOS
- 2 podle toho jakou velikost scriptu požadujeme (lze vyzkoušet stiskem tlačítek podle toho se mění velikost obrazovky) zadáme cestu ke scriptu do CMOS\CNC836.KNF:

velká obrazovka: \$109 C:\CMOS\TABOBD.SCT

C:\CMOS\TABOBD.SCT je jen příklad, vy zde pochopitelně napíšete cestu ke scriptu z bodu 1)

malá obrazovka: \$110 C:\CMOS\TABOBD.SCT

střední obrazovka: \$111 C:\CMOS\TABOBD.SCT

- každý \$1xx může obsahovat jiný script

3 pro zobrazení obrazovky z tlačítka na panelu nebo klávesnice:
 v SYSFILES\KLAV.KNF najdeme příslušné tlačítko za \$D pro panel a \$P pro klávesnici. Do zvolené klávesy vložíme makro 9200H pro velkou obrazovku, 9300H pro malou a 9400H pro střední.
 Např. pro klávesu F12 by daný řádek v části \$P vypadal 9200H "58...F12. Pokud dáte v souboru KLAV.KNF vyhledávat 58...F12, snadno tento řádek najdete a můžete jej upravit a vyzkoušet to.
 pro PLC je věc o to jednodušší, že podle příslušné velikosti se zavolá SEKV 0F8h,92h a SEKV END a daná velikost obrazovky se spustí z PLC. Velikost obrazovky se mění druhým parametrem v SEKV, kde je nyní uvedeno 92h. malou obrazovku je to 93h a pro střední 94h

To je vše. Nyní nahrajte vzorový script, který najdete v adresáři příklady\Tvary i se všemi doprovodnými soubory na systém do adresáře C:\CMOS (budeme předpokládat, že je CMOS v C:\CMOS).
 V C:\cmos\cnc836.knf nastavte \$109 C:\cmos\tabtvary.sct a \$06 P.
 Do c:\sysfiles\klav.knf v \$P řádek s „58...F12 na 9200H „58...F12.
 Spustěte systém, zreferujte souřadnice a stiskněte F12. Vyberte tvar, nastavte parametry a po té stiskněte uložit. V programech vyberte TVARY.NCP. Klávesu F12 můžete takto stisknout kdykoliv. Zkoušejte nastavit různé parametry a vybírejte různé tvary. Uvidíte, že NC program reaguje na parametry v dialogu. Pozor, po každé změně parametrů v dialogu je nutné znovu zvolit program, aby se změny projevily. Také si všimněme, že pokud stiskneme tlačítko uložit v základním dialogu, NC program nejede. Je to tím, že v základním dialogu (DLG_MAIN) je funkce INIT R30 I500, která nastaví návěští skoku na 500. V tomto bloku je v NC programu M30, která program ukončí. Je to proto, aby byl uživatel dialogu nucen zvolit jeden z tvarů.