

# TG Drives

## Modul CNC

### popis datových struktur

**Historie revizí**

<b>datum</b>	<b>verze</b>	<b>popis</b>
12. 1. 2018	1.0	

Obsah	
1. Popis datových struktur CNC modulu .....	4
Struktura P10DOUBLE .....	4
Struktura P10BOOL .....	4
Struktura P26FLOAT .....	5
Struktura DLL_INIT_STRUCT .....	6
Struktura SIMPLE_GM_PARAMS .....	6
Struktura GM_PARAMS_ENHANCED .....	8
Struktura CALLBACK_INFO .....	9
Struktura GCODE_SIZE .....	9
Struktura GCODE_INFO .....	10
Struktura MACHINE_PARAMETERS .....	10
Struktura GCODE_FEED .....	10
Struktura GCODE_CORRECTION .....	11
Struktura ARC_FEED_TABLE .....	11
Struktura INTERNAL_MFUNC_INSERT_TABLE .....	11
Struktura AXIS_LIMITS .....	11
Struktura GCODE_PARAMETERS .....	12

# 1. Popis datových struktur CNC modulu

## Struktura P10DOUBLE

```
struct P10DOUBLE
  double x Osa X
  double y Osa Y
  double z Osa Z
  double c Další osy z G-kódu
  double b
  double u
  double v
  double w
  double a
  double o
```

## Struktura P10BOOL

```
struct P10BOOL
  INT32 x Osa X
  INT32 y Osa Y
  INT32 z Osa Z
  INT32 c Další osy z G-kódu
  INT32 b
  INT32 u
  INT32 v
  INT32 w
  INT32 a
  INT32 o
```

**Struktura P26FLOAT**

struct P26FLOAT

float A *Hodnota adresy A z G-kódu*  
float B *Další hodnoty podle příslušných písmen*  
float C  
float D  
float E  
float F  
float G  
float H  
float I  
float J  
float K  
float L  
float M  
float N  
float O  
float P  
float Q  
float R  
float S  
float T  
float U  
float V  
float W  
float X  
float Y  
float Z

**Struktura DLL\_INIT\_STRUCT**

```
struct DLL_INIT_STRUCT
LPCTSTR CNC_Dll_Dir
LPCTSTR Macros_Dir
LPCTSTR Comm_Int2_Dir
INT32 Local_Core_Mode
```

*Ukazatel na řetězec, kde je uložen adresář s DLL knihovnamy CNC. Text je typu UNICODE (Windows), nebo ANSI (Linux).*

*Ukazatel na řetězec, kde je uložen adresář s makry*

*Ukazatel na řetězec, kde je uložen adresář s komunikačními DLL knihovnamy*

*Režim CNC jádra*

- 0 - výpočty se provádějí v TG Motion
- 1 - výpočty se provádějí v DLL
- 2 - OFFLINE (demo) režim

**Struktura SIMPLE\_GM\_PARAMS**

```
struct SIMPLE_GM_PARAMS
P10DOUBLE m_s
P10DOUBLE m_e
P10DOUBLE m_calcS
P10DOUBLE m_calcE
P10DOUBLE m_zero
P10DOUBLE m_cc
P26FLOAT m_pars
```

*Počáteční poloha (originální z G-kódu)*

*Koncová poloha (originální)*

*Korigovaná počáteční poloha nebo aktuální poloha*

*Korigovaná koncová poloha*

*Aktuální offset nulového bodu*

*Střed kruhového oblouku a jeho radius (v proměnné m\_cc.c)*

*Všechny adresy z G-kódu (26 písmen anglické abecedy). Pokud nejsou některé adresy v G-kódu uvedeny, příslušná buňka pole obsahuje hodnotu > 1e38.*

Při volání **callback** funkcí jsou nastaveny následující buňky pole:

F - rychlost posuvu

G - číslo G funkce (kopie proměnné m\_func)

I, J, K - střed kruhového oblouku

N - délka aktuálního úseku

R - poloměr oblouku (originální)

S - otáčky vřetene

Při volání funkce **ToolChainCore\_GetPos** jsou nastaveny následující buňky pole:

G - číslo G funkce

M - číslo M funkce, pouze čekací M funkce

N - aktuální délka pracovních úseků (hodnota se mění po úsecích; přesnější a plynule se měnící hodnoty jsou ve struktuře GM\_PARAMS\_ENHANCED)

P - procentuální zaplnění bufferu interpolátoru (0 – 100)

T - aktuální délka rychloposuvů (hodnota se mění po úsecích; přesnější hodnoty jsou ve struktuře GM\_PARAMS\_ENHANCED)

X, Y, Z, C, B, U, V, Q, A, O – aktuální polohy interpolátoru (bez offsetu nastaveném funkcí G92)

I, J, K, L, Q, R, D, E, S, H – příslušné offsety nastavené pomocí G92

INT32	m_func	Číslo G nebo M funkce. Při volání ToolChainCore_GetPos jsou čísla M funkcí zvětšena o 10000. Tj. pro m_func < 10000 je aktuální funkce G, pro m_func > 10000 právě probíhá M funkce.
INT32	m_planeTool	Spodních 16 bitů: Rovina kruhové interpolace (17: XY, 18: XZ, 19: YZ, -18: ZX). Hodnota -18 je použita pouze pro soustruh. Horních 16 bitů: číslo nástroje
UINT32	m_reserved	Rezervováno
UINT32	m_editorLine	Číslo řádku v G-kódu
UINT32	m_fileIdx	Index souboru
INT32	m_gCodeIndex	Index G a M funkcí, vždy vzrůstající jedinečná hodnota, zvyšuje se buď o 1, nebo o 2
INT32	m_bFeedIsMin	Spodních 16 bitů: Posuv je mm/min (1) nebo mm/ot (0), horních 16 bitů: číslo korekce 40, 41, 42
float	m_fp	Hodnota FP (procentuální změna rychlosti posuvu F zapsaná v G-kódu jako FP, obvykle je rovna 1.0)

**Struktura GM\_PARAMS\_ENHANCED**

struct GM_PARAMS_ENHANCED		
double	m_setFeed	Požadovaná rychlost posuvu v mm/min
double	m_calcFeed	Vypočítaná rychlost posuvu v mm/min
double	m_actFeed	Aktuální rychlost posuvu
double	m_relFeed	Aktuální override posuvu (1 = 100%)
double	m_actPartLen	Celková délka právě prováděného úseku (mění se mezi úseky)
double	m_actPartRemain	Zbývající délka právě prováděného úseku (mění se průběžně)
double	m_actWorkPartsLen	Aktuální celková délka všech řezných úseků G1, G2, G3 (mění se průběžně, pokud je řezný úsek)
double	m_actG00PartsLen	Aktuální celková délka všech rychloposuvů G0 (mění se průběžně, pokud je rychloposuv)
double	m_actPartsLen	Aktuální celková délka pohybových úseků (tj. všechna G bez M funkcí)
INT32	m_nWorkCounter	Počítadlo řezných úseků
INT32	m_nG00Counter	Počítadlo rychloposuvů
INT32	m_nG00ToWorkChanges	Počítadlo přechodů rychloposuv -> řezná část
INT32	m_nWorkToG00Changes	Počítadlo přechodů řezná část -> rychloposuv
INT32	m_nMCounter	Počítadlo M funkcí
UINT32	m_MasterEditorLine	Aktuální řádek v hlavním souboru
UINT32	m_MasterFileIdx	Index hlavního souboru
INT32	m_statusCNC	Aktuální status CNC (stejná hodnota jako při volání funkce ToolChainCore_StatusCNC())
double	m_cncRunTime	Vypočítaný přibližný čas běhu G-kódu. Hodnota je platná pouze po simulaci běhu G-kódu



## Struktura CALLBACK\_INFO

struct CALLBACK_INFO		
INT32	stringId	Číslo textu z textového .INI souboru, zároveň číslo chyby, může být -1, pak jsou platné jen položky CALLBACK_INFO.nLines a CALLBACK_INFO.actLine
INT32	nLines	Celkový počet řádků v G-kódu (pokud je znám, jinak 0)
INT32	actLine	Aktuální řádek, kde probíhá kompilace nebo nastala chyba, počítáno od 1, může být i 0 pak se položka nepoužívá
INT32	fileIdx	Číslo souboru, kde je chyba nebo který se kompiluje. Jméno souboru se zjistí pomocí funkce ToolChain_IdxToFile. Může být -1, pak jsou platné jen položky CALLBACK_INFO.nLines a CALLBACK_INFO.actLine
LPCSTR	m_fullError	Pouze pro ToolChain_GetLastError: Ukazatel na kompletní chybový text (ANSI řetězec). Může být prázdný řetězec. Obsah je třeba přkopírovat do lokální proměnné ihned po volání ToolChain_GetLastError.
LPCSTR	m_errorArguments	Pouze pro ToolChain_GetLastError: Ukazatel na řetězec ANSI, který obsahuje případné argumenty chybového řetězce, nebo prázdný řetězec. Jednotlivé argumenty jsou odděleny LF (0x10). Obsah je třeba přkopírovat do lokální proměnné ihned po volání ToolChain_GetLastError.

### Příklad:

Formát chybové zprávy:

"Wrongly entered centre of circular interpolation XY\nStart X = %g Y = %g | End X = %g Y = %g\nDelta (%.7f) > 2\*R (%.7f)"

Kompletní text chyby (m\_fullError):

"Wrongly entered centre of circular interpolation XY\nStart X = 20.123 Y = 30.653 | End X = 43.987 Y = -1.26\n Delta (46.990) > 2\*R (30.000)"

Argumenty chybového řetězce (m\_errorArguments):

"20.123<LF>  
 30.653<LF>  
 43.987<LF>  
 -1.26<LF>  
 46.990<LF>  
 30.000"

## Struktura GCODE\_SIZE

struct GCODE_SIZE		
P10DOUBLE	m_startPos	Počáteční poloha
P10DOUBLE	m_endPos	Koncová poloha
P10DOUBLE	m_minimalPos	Minimum
P10DOUBLE	m_maximalPos	Maximum

### Struktura GCODE\_INFO

```
struct GCODE_INFO
  GCODE_SIZE  m_size[4]    4 struktury popisující rozměry G-kódu
                        0 - originální G-kód neotočený
                        1 - korigovaný G-kód neotočený
                        2 - originální G-kód otočený
                        3 - korigovaný G-kód otočený
```

### Struktura MACHINE\_PARAMETERS

```
struct MACHINE_PARAMETERS
  P10DOUBLE   minCoords    Minimální povolené souřadnice
  P10DOUBLE   maxCoords    Maximální povolené souřadnice
  P10DOUBLE   refCoords    Vztažený bod, ke kterému se vztahují min max souřadnice
  P10BOOL     enableMask   Maska povolení testu limitů pro jednotlivé osy
  P10DOUBLE   backlash     Vůle v osách v mm
  P10DOUBLE   backlashInc  Změna vůle v mm za 100 μs
  INT32       backlashValuesSource  Zdroj hodnot vůlí
                        0 – hodnoty se čtou z .INI souboru
                        1 – hodnoty použijí z výše uvedených proměnných
  INT32       enhancedInterpolatorParameters  bity 0-9: blokáce omezení rychlosti podle tečnosti pro jednotlivé osy
                                                bity 10-19: pokud je bit nastaven, příslušná osa se nebere v úvahu pro výpočet rychlosti posuvu
                                                bit 20: pokud se kterákoliv osa nastavená v bitu 0-9 rozjede z klidu, zastaví se předtím ostatní osy
                                                bit 21: při změně směru pohybu kterékoliv osy nastavené v bitu 0-9 se předtím zastaví všechny osy
```

### Struktura GCODE\_FEED

```
struct GCODE_FEED
  double      feed          Rychlost posuvu
  double      acc           Zrychlení
  double      dec           Zpomalení
  INT32       rampType      Typ rampy: 1 - trapezoid, 2 – sin
  INT32       padding1      Rezervováno
```

**Struktura GCODE\_CORRECTION**

```
struct GCODE_CORRECTION
  double    size      Velikost korekce (poloměr nástroje). Je-li záporná, pak se přehodí význam G41/G42
  INT32     gCode     Číslo G funkce pro korekci (41, 42), zatím se nevyužívá
  INT32     padding1  Rezervováno
```

**Struktura ARC\_FEED\_TABLE**

```
struct ARC_FEED_TABLE
  double    m_r       Rádus
  double    m_relFeed Relativní rychlost posuvu
```

**Struktura INTERNAL\_MFUNC\_INSERT\_TABLE**

```
struct INTERNAL_MFUNC_INSERT_TABLE
  INT32     mFunc     Číslo M funkce nebo 0 (pak se nepoužívá)
  INT32     padding1  Rezervováno
  double    relFeed   Relativní rychlost, při které se M funkce vloží. Musí být v rozsahu <0.01;1.0>. Nemá význam pro vkládání před G00.
  double    distance  Vzdálenost před / za relativní rychlostí, kde se M funkce vloží
```

**Struktura AXIS\_LIMITS**

```
struct AXIS_LIMITS
  double    maxFeed   Maximální povolená rychlost
  double    maxAcc    Maximální povolené zrychlení
  double    maxDec    Maximální povolené zpomalení
```

**Struktura GCODE\_PARAMETERS**

struct GCODE\_PARAMETERS

INT32 absMode

*Význam souřadnic G-kódu: 0 - relativní G-kód, 1 - absolutní G-kód*

INT32 swapG2G3

*Přehození G2/G3*

UINT32 correctionStrategy

*Strategie výpočtu korekcí:*

0 - G41, G42 znamená už korigovanou souřadnici; G40 říká, že až konec následujícího pohybu nebude korigován

1 - G41, G42 koriguje až koncovou souřadnici následujícího pohybu; G40 říká, že konec předchozího pohybu už nebude korigován

0x00000100 – Hledá se nejméně 10 úseků a jejich možné průsečíky

0x00000200 – Hledá se nejméně 50 úseků a jejich možné průsečíky

0x00000300 – Hledá se nejméně 100 úseků a jejich možné průsečíky

0x00000400 – Hledá se nejméně 500 úseků a jejich možné průsečíky

0x00000500 – Hledá se nejméně 1000 úseků a jejich možné průsečíky

0x00000600 – Hledá se nejméně 2000 úseků a jejich možné průsečíky (výchozí hodnota)

0x00000700 – Prohledává se celý vyrovnávací buffer

0x00020000 – Pokud po výpočtu korekcí zůstanou nějaké nespojené úseky (chybná korekce), spojí se pomocí G1

0x00040000 – Pokud po výpočtu korekcí zůstanou nespojené úseky, spojí se přesunutím počátečního bodu úseku do koncového bodu předchozího úseku

0x00080000 – Odstraní křížící se úseky

0x00100000 – Pokud nelze spočítat korekci, je povoleno zpětné procházení bufferu

0x00200000 – oblouky, jejichž ekvidistanta vyjde pro danou korekci záporná, se vypustí z řezného plánu a spojí se předchozí a následující úsek

0x00400000 – Pro režim 0x00800000 (viz níže) se jako rozhodovací kritérium použije dvojnásobek korekce (místo trojnásobku)

0x00800000 – Pokud je nastaven tento bit, pak pro vnější korekci se nejdříve testuje průsečík prodloužených ekvidistant. Pokud průsečík existuje a je vzdálen od původního vrcholu méně než trojnásobek korekce (nebo dvojnásobek, pokud je nastaveno i 0x00400000), je použit tento průsečík a nekládá se oblouk.

0x40000000 – Pokud je nastaven tento bit, je G00 vždy bez korekce.

0x80000000 – Pokud je nastaven tento bit, pak je nastaven režim compatibility se starým systémem. Konkrétně se ignoruje první G41/G42, pokud před ní není žádný pohyb. Dále se ignoruje dvojice G40 - G41/G42, pokud následují těsně za sebou (tj. mezi G40 a G41/G42 není žádný pohyb).

INT32	forceMode	<i>Význam adresy F v G-kódu: 0 - rychlosti F, FF, FP jsou podle G-kódu, 1 - rychlosti F, FF, FP se ignorují, všechny rychlosti jsou dle parametrů, 2 - rychlosti F, FF, jsou dle parametrů, FP určuje relativní rychlost dle parametru</i>
GCODE_FEED	motion[8]	<i>Rychlosti pro následující posuvy (pro forceMode 1 nebo 2):  0 - rychloposuv  1 - dopředu  2 - dozadu  3 - konekt na trajektorii  4 - rychlost pohybu bez M funkcí (režim 31 funkce ToolChainCore_StartCNC)  5 - rychlost posuvu ve funkci ToolChainCore_GoToPosition  6 - zpomalení pro normální stop, pouze pole dec  7 - zpomalení pro emergency stop, pouze pole dec</i>
GCODE_CORRECTION	correction[8]	<i>Tabulka hodnot korekcí pro následující G funkce  0 - G41  1 - G42  2 - G43  3 - G44  4 - G45  5 - G46  6 - G47  7 - G48</i>
double	maxConnectionDistance	<i>Maximální vzdálenost od trajektorie pro ToolChainCore_ConnectCNC, pokud je poloha od úseku trajektorie větší, počítá se vzdálenost jen od koncových bodů a ne průsečík</i>
double	startAngle	<i>Úhel mezi úseky, od kterého se začne zpomalovat [°]</i>
double	endAngle	<i>Úhel mezi úseky, při kterém se zpomalí na rychlost minAngleFeed [°]</i>
double	minAngleFeed	<i>Minimální rychlost, na kterou se zpomalí, pokud vzájemný úhel dvou úseků je větší než endAngle</i>
INT32	disableG0FeedOverride	<i>Je-li 1, pak interpolátor ignoruje override posuvu při rychloposuvu, zároveň vynutí zastavování mezi G0, pokud je jich více za sebou v G-kódu. Rychloposuv používá rychlost z proměnné motion[0].</i>
INT32	interpolatorConfig0	<i>bit 0 - režim override posuvu  1 - relativní rychlost se mění pomocí změny časování, má vliv na rychlost i zrychlení a zpomalení  0 - relativní rychlost se mění pouze během rovnoměrného pohybu, tj. zrychlení a zpomalení se nemění</i>
double	overrideInc	<i>Přírůstek override posuvu, max 0.1 = 10%. Pokud je 0, použije se default 0.001, tj. 0.1%. Přírůstek se připočítává každých 100 µs. Z toho lze určit, za jakou dobu se nastaví požadovaný override.</i>

ARC\_FEED\_TABLE      arcFeedTable[5]      *Tabulka omezení rychlostí v závislosti na poloměru oblouku, celkem 5 dvojic [poloměr; relativní rychlost]. Nulová hodnota poloměru v tabulce znamená, že další hodnoty se už nepoužívají, tzn. tabulka může mít 0 - 5 položek.*

Příklady:

**vše nulové** - tabulka se nepoužívá (výchozí hodnota)

**arcFeedTable[0].m\_r = 0, arcFeedTable[0].m\_relFeed = 0.5** - všechny oblouky se projedou rel. rychlostí 50% (50% z požadované rychlosti oblouku)

**arcFeedTable[0].m\_r = 10, arcFeedTable[0].m\_relFeed = 1.25, arcFeedTable[1].m\_r = 0.9, arcFeedTable[1].m\_relFeed = 0.1** - oblouky s poloměrem  $\geq 10$  budou mít relativní rychlost 125%, oblouky menší než R 10 až do R = 0.9 budou mít postupně menší rychlost až do 10% a všechny oblouky menší než R 0.9 budou mít relativní rychlost 10%

double      centripetalAcc      *Maximální povolené dostředivé zrychlení pro oblouky (zatím neimplementováno)*

INTERNAL\_MFUNC\_INSERT\_TABLE      mFuncInsertTable[3]      *Vkládání průběžných M funkcí pro signalizaci poklesu rychlosti či dosažení řezné rychlosti*  
 0 - M funkce před zpomalením (obloukem) - signalizace budoucího zpomalení.  
*Je-li mFuncInsertTable[0].mFunc = 0, signalizace před zpomalením se nepoužívá.*  
 1 - M funkce za zpomalením (obloukem) - signalizace budoucí plně řezné rychlosti.  
*Je-li mFuncInsertTable[1].mFunc = 0, signalizace za zpomalením se nepoužívá.*  
 2 - M funkce před rychloposuvem G00.  
*Je-li mFuncInsertTable[2].mFunc = 0, signalizace před rychloposuvem se nepoužívá.*

INT32      MfuncInsertType      *Strategie vkládání M funkcí (mFuncInsertTable), zatím jen 0 - vypnuto a 1 - zapnuto*  
 INT32      paddingTo8Bytes\_2      *Rezervováno*  
 double      reserved      *Rezervováno*

INT32      M02\_programEnd1      *Číslo M funkce pro ukončení G-kódu, výchozí hodnota je 2 (M2), 0 - vypnuto*  
 INT32      M30\_programEnd2      *Alternativní číslo M funkce pro ukončení G-kódu, výchozí je M30, 0 - vypnuto*  
 INT32      M03\_startSpindleCW      *Číslo M funkce pro roztočení vřetene CW, výchozí je M3, 0 - vypnuto*  
 INT32      M04\_startSpindleCCW      *Číslo M funkce pro roztočení vřetene CCW, výchozí je M4, 0 - vypnuto*  
 INT32      M05\_stopSpindle      *Číslo M funkce pro zastavení vřetene, výchozí je M5, 0 - vypnuto*  
 INT32      M06\_changeTool      *Číslo M funkce pro výměnu nástroje, výchozí je M6, 0 - vypnuto*  
 INT32      M07\_coolOn      *Číslo M funkce pro zapnutí chlazení, výchozí je M7, 0 - vypnuto*  
 INT32      M08\_oilOn      *Číslo M funkce pro zapnutí mazání, výchozí je M8, 0 - vypnuto*  
 INT32      M09\_coolOilOff      *Číslo M funkce pro vypnutí chlazení i mazání, výchozí je M9, 0 - vypnuto*  
 INT32      M17\_return      *Číslo M funkce pro návrat z podprogramu (RETURN), výchozí je M17, 0 - vypnuto*  
 INT32      M29\_print      *Číslo M funkce pro výstup textového hlášení (PRINT), výchozí je M29, 0 - vypnuto*  
 INT32      M80\_mirrorOff      *Číslo M funkce pro vypnutí zrcadlení, výchozí je M80, 0 - vypnuto*

INT32	M81_mirrorX	Číslo M funkce pro zrcadlení v ose X, výchozí je M81, 0 - vypnuto
INT32	M82_mirrorY	Číslo M funkce pro zrcadlení v ose Y, výchozí je M82, 0 - vypnuto
INT32	M83_mirrorZ	Číslo M funkce pro zrcadlení v ose Z, výchozí je M83, 0 - vypnuto
INT32	M84_mirrorXY	Číslo M funkce pro zrcadlení v ose X a Y, výchozí je M84, 0 - vypnuto
INT32	M85_mirrorXZ	Číslo M funkce pro zrcadlení v ose X a Z, výchozí je M85, 0 - vypnuto
INT32	M86_mirrorYZ	Číslo M funkce pro zrcadlení v ose Y a Z, výchozí je M86, 0 - vypnuto
INT32	M87_mirrorXYZ	Číslo M funkce pro zrcadlení v ose X, Y a Z, výchozí je M87, 0 - vypnuto
INT32	M99_defFeed	Číslo M funkce pro definování výchozí hodnoty posuvu, výchozí je M99, 0 - vypnuto
INT32	insertArcMode	Mód vkládání malých tečných oblouků pro úseky, které na sebe nenavazují tečně: 0 - vypnuto, 1 – zapnuto
INT32	paddingTo8Bytes_3	Rezervováno
double	insertArcToleranceRMax	Maximální povolená vzdálenost mezi průsečíkem dvou netečných úseků a vloženým obloukem, z této vzdálenosti se počítá rádius vloženého oblouku, implicitně 0,03 mm.
double	insertArcToleranceRMin	Minimální povolená vzdálenost mezi průsečíkem dvou netečných úseků a vloženým obloukem, z této vzdálenosti se počítá rádius vloženého oblouku, implicitně 0,03 mm.
double	insertArcMaxAngle	Max. úhel mezi G1 úseky, do kterého se ještě vkládají oblouky
double	insertArcMinAngle	Min. úhel mezi G1 úseky, od kterého se ještě vkládají oblouky (neimplementováno, zatím vždy 0)
INT32	insertArcPlane	Rovina, ve které se oblouky vkládají (17 - XY, 18 - XZ, 19 - YZ nebo 0, pak systém sám zvolí rovinu)
INT32	rezerva1	Rezervováno
INT32	calcTime	0 – výpočet se neprovádí, 1 - během simulace se počítá čas potřebný pro běh G-kódu
INT32	rezerva2	Rezervováno
double	dynamicG41Correction_mm	Přídavná dynamická korekce v závislosti na rychlosti pro G41 v mm
double	dynamicG42Correction_mm	Přídavná dynamická korekce v závislosti na rychlosti pro G42 v mm
AXIS_LIMITS	axisLimits[10]	Limity rychlosti, zrychlení a zpomalení pro jednotlivé osy
double	dynamicG41Correction_percent	Přídavná dynamická korekce v závislosti na rychlosti pro G41 v procentech
double	dynamicG42Correction_percent	Přídavná dynamická korekce v závislosti na rychlosti pro G42 v procentech

double tolerance[10]

*Tolerance:*

- 0 – *povolená odchylka středu oblouku, výchozí hodnota je 0.005 mm*
- 1 – *tolerance výpočtu křížení úseků, výchozí hodnota je 0.001*
- 2 – *úhlová tolerance dvou úseků, pokud je hodnota menší, úseky se spojí, výchozí hodnota je 0.03 [°]*
- 3 – *pokud jsou dva úseky G1 spojeny v jeden, maximální povolená odchylka mezi původními G1 a novou G1 výchozí hodnota je 0.005 [mm]*
- 4 – *minimální povolený poloměr oblouků, menší oblouky budou odstraněny*
- 5 – *násobek korekce pro hledání průsečíku dvou navazujících úseků (ostrý roh). Jinak se vkládá oblouk.*
- 6 – *přesnost navazování úseků*
- 7 – *rezerva*
- 8 – *rezerva*
- 9 – *rezerva*