

TG Motion

verze 4

Skupina typu SERVO

návod k obsluze

Historie revizí

datum	verze	popis
10. 1. 2017	1.0	
31. 7. 2017	1.1	upravena typografie

Obsah

1. Skupina SERVO	4
1.1 Popis skupiny Servo	4
1.2 Součásti skupiny Servo	4
2. Řízení a diagnostika servopohonu	5
2.1 Princip řízení servopohonů	5
2.2 Důležité registry	6
2.3 Popis principu a užití registrů	6
3. PG – generátor žádané polohy (Profile Generator)	9
3.1 Důležité registry	9
3.2 Popis struktury PG	9
3.3 Generátor průběhů ramp	9
3.4 Příklady použití generátoru žádané hodnoty polohy PG	10
4. Generátor žádané polohy GEAR	13
4.1 Popis struktury GEAR	13
4.2 Důležité registry	13
4.3 Vratná vačka	15
4.4 Přírůstková vačka	15
4.5 Externí tabulky dat	16
4.6 Příklady použití generátoru žádané hodnoty polohy GEAR	17
5. Funkční rozhraní COMMAND	19
5.1 Důležité registry	19
5.2 Popis struktury Command	19
5.3 Příklady použití funkčního rozhraní Command	21
6. Rozhraní pro komunikaci s objekty SDO	23
6.1 Popis struktury SDO	23
6.2 Důležité registry	23
6.3 Popis registrů a práce s nimi	23
6.4 Příklady použití struktury SDO	23
7. Rozhraní CAPTURE	25
7.1 Popis struktury Capture	25
7.2 Důležité registry	25
7.3 Popis funkce struktury Capture	25
7.4 Příklad použití rozhraní Capture	25
8. Apendix	26
Přehled a popis registrů skupiny Servo	26

1. Skupina SERVO

1.1 Popis skupiny Servo

Skupina Servo unifikuje ovládací rozhraní pro různé typy servopohonů. Z hlediska uživatele a programátora PLC kódu se všechny servopohony chovají stejně, mají stejné registry, kterými je lze ovládat, nebo měnit jejich nastavení. Lze tedy operativně měnit servopohony, aniž by se musel přepisovat PLC kód. Stejný PLC kód lze aplikovat na více různých servopohonů.

Skupina Servo tvoří rozhraní pro řízení a diagnostiku jednotlivých servopohonů. Umožňuje uživateli kontrolu obecných registrů a jednoduché ovládání servopohonu prostřednictvím virtuálních generátorů žádané polohy, jako je generátor polohového profilu **PG** pro absolutní a relativní polohové řízení z polohy do polohy a pro rychlostní řízení.

Dále nabízí generátor **GEAR** pro lineární synchronizaci a simulaci vratných a přírůstkových vaček. Pro jednodušší práci s generátory PG a GEAR je uživateli k dispozici funkční rozhraní **COMMAND**. Použití tohoto rozhraní zjednoduší programový kód psaný uživatelem a zrychlí běh virtuálního PLC.

Dále rozhraní Servo obsahuje mechanismus pro komunikaci s jednotlivými servopohony prostřednictvím objektů **SDO**. Tato komunikace umožňuje uživateli jednoduše číst nebo zapisovat parametry daného servopohonu.

Rozhraní obsahuje i mechanismus **CAPTURE** pro zachycení poloh jednotlivých servopohonů a externích inkrementálních snímačů s vyšší přesností, než nabízí **Cycle_Time**. Rozhraní lze využít pro synchronizaci na indexovou značku, vyšší přesnost při vysokých rychlostech a další.

1.2 Součásti skupiny Servo

Obecné registry

- základní nastavení a parametry servopohonu
- zjištění a nastavení stavu a pracovního módu servopohonu
- zjištění a nastavení pozice
- chybová hlášení

PG – generátor polohového profilu

- absolutní polohové řízení z polohy do polohy
- relativní polohové řízení
- rychlostní řízení

GEAR – generátor synchronizace servopohonů

- lineární synchronizace – simulace lineární převodovky
- nelineární synchronizace – simulace vratné nebo přírůstkové vačky

COMMAND

- funkční rozhraní pro práci s generátory PG a GEAR z programu psaného uživatelem

SDO

- skupina registrů umožňující čtení a zápis objektů SDO konkrétních servopohonů

CAPTURE

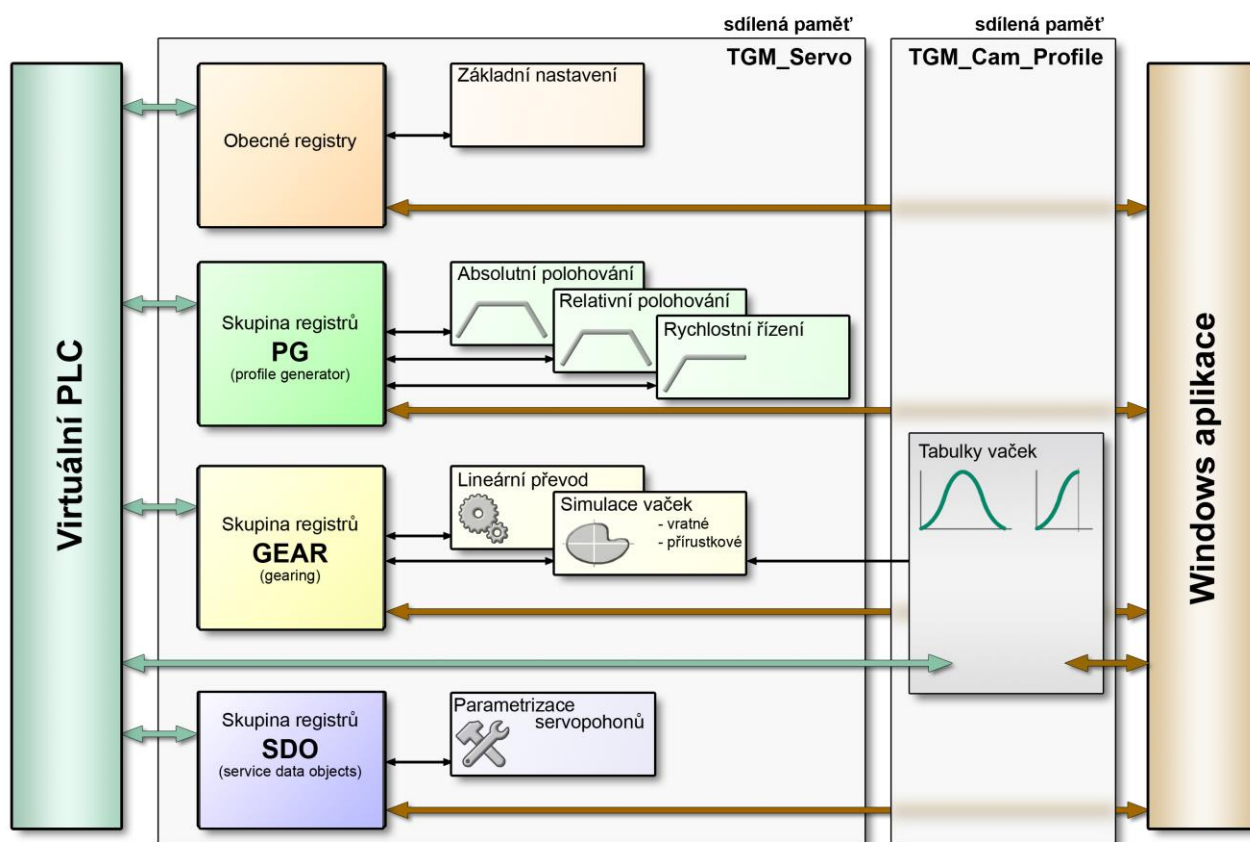
- mechanismus pro zachycení polohy jednotlivých servopohonů

2. Řízení a diagnostika servopohonu

2.1 Princip řízení servopohonů

Výpočet polohy jednotlivých servopohonů pomocí **PG** a **GEAR** se realizuje v rámci jednoho cyklu **Cycle_Time**. Čas cyklu je definován v souboru **TGMotion4xx.ini** (250 μs, 500 μs, 1 ms, 2 ms atd.). V každém **Cycle_Time** zasílá **TG Motion** jednotlivým servopohonům žádané polohy. Mezi dvěma žádanými polohami (tedy uvnitř jednoho intervalu **Cycle_Time**) servozesilovač interpoluje. Průběh a četnost interpolací závisí na konkrétním servozesilovači (viz manuál konkrétního servozesilovače).

Vnitřní regulační struktura servopohonu se snaží regulovat svou aktuální pozici tak, aby platil vztah **Servo.Position = Servo.WritePosition** (skutečná poloha = žádaná poloha). Vzniklý rozdíl pak určuje regulační odchylku, kterou se snaží regulátor servozesilovače zmenšovat.



Obr. Rozhraní PLC – Servopohon – Windows

2.2 Důležité registry

Servo.Status – aktuální stav servopohonu.

Servo.Position – aktuální skutečná poloha servopohonu v inkrementech.

Servo.RefPosition – aktuální skutečná poloha servopohonu posunutá o Offset v inkrementech.

Servo.Offset – posunutí polohy servopohonu v inkrementech.

Servo.Mode – zdroj žádané polohy:

0 – žádaná hodnota není systémem **TG Motion** modifikována; do serva je posílána aktuální hodnota **Servo.WritePosition**.

1 – zdrojem žádané hodnoty je PG.

3 – zdrojem žádané hodnoty je GEAR.

4 – zdrojem žádané hodnoty je PG a GEAR.

6 – zdrojem žádané hodnoty je Interpolátor.

7 – zdrojem žádané hodnoty je PG a Interpolátor.

8 – zdrojem žádané hodnoty je GEAR a Interpolátor.

Servo.Error – chybová hlášení – závisí na konkrétním servopohonu.

Servo.WritePosition – žádaná poloha servopohonu v inkrementech.

Servo.Correction – korekce pozice servopohonu.

Kompletní výčet všech registrů skupiny Servo včetně popisu viz Apendix.

2.3 Popis principu a užití registrů

Každý servopohon je možné ovládat pomocí registru **Servo.Mode**, který umožňuje připojit servopohon k libovolnému generátoru žádané polohy (PG, GEAR, interpolátor), popř. k některé dvojici generátorů žádané polohy, čehož lze využít pro vytváření superponovaných pohybů. Dále je možné přímo nastavovat proudové omezení **Servo.LimitCurrent** příslušného servozesilovače.

Pro diagnostiku pohonu jsou určeny proměnné **Servo.Status** a **Servo.Error**, které určují aktuální stav pohonu s případným kódem chyby. Reset poruch je možné realizovat nastavením příslušného bitu registru **Servo.Control** (hodnoty viz následující výpis). Dále je možné monitorovat polohu, rychlost, popř. proud daného servopohonu.

Registr **Servo.Correction** slouží jako uživatelská korekce polohy vypočítané pomocí PG, GEAR, Interpolátoru, příp. jejich superpozice. Do tohoto registru **TG Motion** nezapisuje žádné hodnoty, do výpočtu žádané pozice příslušného servopohonu jej pouze zahrnuje. Registru lze tedy využít např. k superponování daného pohybu na další uživatelský pohyb. Výpočet registru **Servo.Correction** je třeba provádět v **Program_04**.

Servo.Mode – význam hodnot

Servo.Mode = 0 (nepřipojen)

Registr **Servo.WritePosition** je odpojen od generátorů žádané polohy. Uživatel může v PLC programu **Program_04** generovat žádanou hodnotu polohy.

Servo.Mode = 1 (profile generator)

Do servopohonu je posílána pouze poloha vypočtená generátorem polohy **PG**. V tomto módu je **Servo.WritePosition** počítána podle vztahu:

$$\text{Servo.WritePosition} = \text{Servo.Pg.APos} + \text{Servo.Offset} + \text{Servo.Correction}$$

Servo.Mode = 3 (gear generator)

Do servopohonu je posílána pouze poloha vypočtená generátorem polohy **GEAR**. V tomto módu je **Servo.WritePosition** počítána podle vztahu:

Je-li **Servo.GEAR.Mode = 1** (gear mode):

$$\text{Servo.WritePosition} = \text{Servo.GEAR.Position} + \text{Servo.GEAR.Offset} + \text{Servo.Offset} + \text{Servo.Correction}$$

Je-li **Servo.GEAR.Mode = 2** (cam mode):

```
Servo.WritePosition = Servo.GEAR.CamPosition + Servo.GEAR.Offset + Servo.Offset +  
Servo.Correction
```

Servo.Mode = 4 (profile generator + gear generator)

Do servopohonu je poslán součet poloh vypočtených generátorem polohy **PG** a generátorem polohy **GEAR**. Mód slouží k realizaci superponovaného pohybu. V tomto módu je **Servo.WritePosition** počítána podle vztahu:

Je-li **Servo.GEAR.Mode = 1** (gear mode):

```
Servo.WritePosition = Servo.Pg.APos + Servo.GEAR.Position + Servo.GEAR.Offset +  
Servo.Offset + Servo.Correction
```

Je-li **Servo.GEAR.Mode = 2** (cam mode):

```
Servo.WritePosition = Servo.Pg.APos + Servo.GEAR.CamPosition + Servo.GEAR.Offset +  
Servo.Offset + Servo.Correction
```

Servo.Mode = 6 (interpolator)

Do servopohonu je posílána pouze poloha vypočtená generátorem polohy **Interpolátor**, přičemž registr **Servo.Cnc.Number** určuje číslo interpolátoru, na nějž je servopohon připojen, a registr **Servo.Cnc.NumberAxes** určuje osu interpolátoru, jejíž vypočtená hodnota polohy je zapisována do registru **Servo.WritePosition**. V tomto módu je **Servo.WritePosition** počítána podle vztahu:

```
Intr = Interpolator[Servo.Cnc.Number];  
AxisIdx = Servo.Cnc.NumberAxes;  
  
Servo.WritePosition = Intr.PositionInc[AxisIdx] + Intr.Backlash[AxisIdx] +  
Intr.Offset[AxisIdx] + Servo.Offset + Servo.Correction
```

Servo.Mode = 7 (interpolator + profile generator)

Do servopohonu je poslán součet poloh vypočtených generátorem polohy **Interpolátor** a generátorem polohy **PG**. Mód slouží k realizaci superponovaného pohybu. V tomto módu je **Servo.WritePosition** počítána podle vztahu:

```
Intr = Interpolator[Servo.Cnc.Number];  
AxisIdx = Servo.Cnc.NumberAxes;  
  
Servo.WritePosition = Intr.PositionInc[AxisIdx] + Intr.Backlash[AxisIdx] +  
Intr.Offset[AxisIdx] + Servo.Pg.APos + Servo.Offset + Servo.Correction
```

Servo.Mode = 8 (interpolator + gear generator)

Do servopohonu je poslán součet poloh vypočtených generátorem polohy **Interpolátor** a generátorem polohy **GEAR**. Mód slouží k realizaci superponovaného pohybu. V tomto módu je **Servo.WritePosition** počítána podle vztahu:

Je-li **Servo.GEAR.Mode = 1** (gear mode):

```
Intr = Interpolator[Servo.Cnc.Number];  
AxisIdx = Servo.Cnc.NumberAxes;  
  
Servo.WritePosition = Intr.PositionInc[AxisIdx] + Intr.Backlash[AxisIdx] +  
Intr.Offset[AxisIdx] +  
Servo.GEAR.Position + Servo.GEAR.Offset + Servo.Offset + Servo.Correction
```

Je-li **Servo.GEAR.Mode = 2** (cam mode):

```
Intr = Interpolator[Servo.Cnc.Number];  
AxisIdx = Servo.Cnc.NumberAxes;
```

```
Servo.WritePosition = Intr.PositionInc[AxisIdx] + Intr.Backlash[AxisIdx] +  
Intr.Offset[AxisIdx] +  
Servo.GEAR.CamPosition + Servo.GEAR.Offset + Servo.Offset + Servo.Correction
```

Servo.Control – význam hodnot

K registru se přistupuje bitově; nastavením bitů lze ovládat následující funkce:

bit 0 = reset poruchy.

bit 1 = servopohon není pod momentem.

bit 2 = reset EtherCAT komunikace.

bit 3 = povoluje automatický výpočet registru **Servo.TorqueFeedForward**.



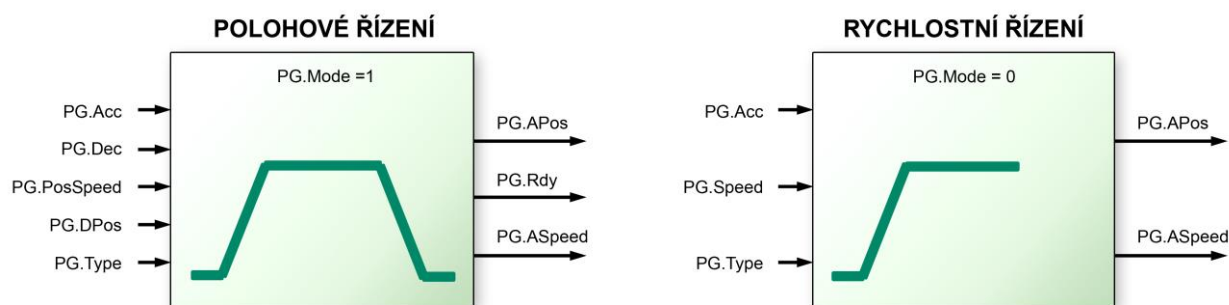
*Při resetu poruchy se musí příslušný bit nastavit minimálně na dobu jednoho **Cycle_Time**. Vynulování bitů se neděje automaticky, musí je zabezpečit programátor.*

3. PG – generátor žádané polohy (Profile Generator)

3.1 Důležité registry

- PG.APos** – aktuální žádaná hodnota polohy.
- PG.ASpeed** – aktuální žádaná hodnota rychlosti.
- PG.Acc**, **PG.Dec** – zrychlení (akcelerace), zpomalení (decelerace) pohonu.
- PG.PosSpeed** – maximální rychlost pohybu.
- PG.DPos** – cílová poloha.
- PG.Speed** – žádaná hodnota rychlosti.
- PG.Mode** – vlastní ovládání generátoru:
 - 0 – rychlostní řízení.
 - 1 – polohové řízení.
- PG.Rdy** – status generátoru PG:
 - 1 – cílové polohy bylo dosaženo.

Kompletní výčet všech registrů skupiny Servo včetně popisu viz Apendix.



Obr. Polohové a rychlostní řízení PG generátorem

3.2 Popis struktury PG

PG (Profile Generator) umožňuje jednoduchým způsobem polohovat z polohy do polohy a rychlostně řídit servopohon. Výstupem generátoru je proměnná **PG.APos**, která určuje žádanou hodnotu polohy a proměnná **PG.ASpeed**, která určuje žádanou hodnotu rychlosti. Parametry generátoru jsou **PG.Acc** a **PG.Dec**, které určují maximální akceleraci, resp. deceleraci pohonu, dále **PG.PosSpeed**, který určuje maximální rychlost pohybu a **PG.DPos**, který určuje cílovou polohu. Parametr **PG.Speed** určuje žádanou hodnotu rychlosti u rychlostního řízení (**PG.Mode = 0**). Pomocí proměnných **PG.Mode** a **PG.Rdy** se realizuje vlastní ovládání generátoru.

Generátor pracuje tak, že v každé periodě servosmyčky na základě registrů **PG.Acc**, **PG.Decel**, **PG.PosSpeed**, **PG.DPos** a stavu proměnných **PG.Mode** a **PG.Speed** vypočítá aktuální žádanou hodnotu polohy **PG.APos** a aktuální žádanou hodnotu rychlosti **PG.ASpeed**, které zapíše do sdílené paměti.

Generátor pracuje jako virtuální, tedy nezávisle na daném servopohonu. Připojení jeho výstupu k servopohonu se provede nastavením **Servo.Mode = 1**.

3.3 Generátor průběhů ramp

Definuje průběh začátku a konce pohybu z hlediska rychlosti (rozjezd a doběh). Používá se k hladkým změnám rychlosti rozběhu a doběhu servopohonu. **TG Motion** nabízí čtyři základní průběhy ramp volitelné hodnotou registru **Pg.Type**.

Pg.Type – význam hodnot

0 – krátký rozběh, dlouhý doběh:

Během rozběhu je rychle dosaženo WriteSpeed, doběh do žádané polohy je pozvolný.

1 – krátký rozběh i doběh – používá se k rychlému zapolohování:

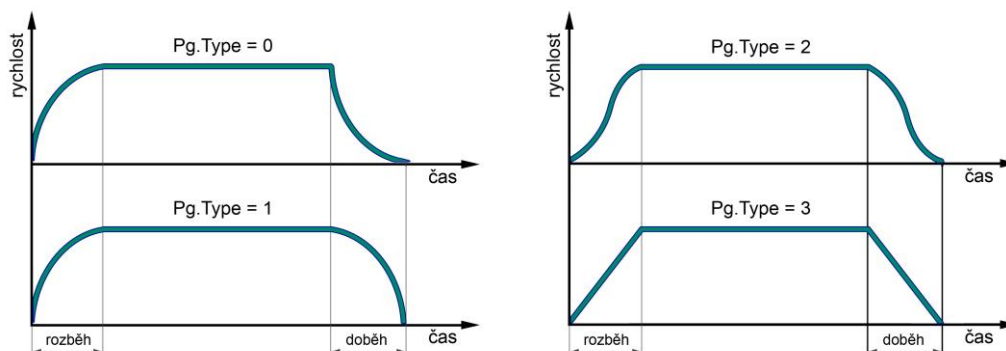
Během rozběhu je rychle dosaženo WriteSpeed, doběh je rychlý včetně dosažení žádané polohy.

2 – harmonický rozběh i doběh – nejpoužívanější průběh:

Obráží přirozené chování mechanických součástí i regulaci. Veškeré průběhy jsou harmonické, respektují setrvačné síly servopohonu a na něj navazujících mechanických částí.

3 – lineární rozběh i doběh – téměř se nepoužívá:

*Veškeré průběhy jsou lineární, lze použít při menších změnách rychlostí. Je vhodný při požadavku nepřekročení hodnot **PG.Acc**, **PG.Dec** (viz následující odstavce).*



Obr. Průběhy generátoru Ramp



TG Motion vždy přesně dodržuje čas rozběhu a doběhu generátoru ramp. Tento čas je nezávislý na tvaru průběhu. Např. lineární i harmonický rozběh trvá stejný čas.

Hodnoty nastavené v registrech **PG.Acc**, **PG.Dec** platí po celou dobu rozběhu a doběhu pouze u lineárního průběhu (**Pg.Type = 3**). U harmonických průběhů obsahují registry střední hodnotu akcelerace a decelerace. Např. u **Pg.Type = 2** je reálné maximální zrychlení/zpomalení 157 % hodnoty registrů **PG.Acc**, **PG.Dec**. Je tedy třeba počítat s tím, že maximální hodnota reálného zrychlení a zpomalení dosahuje u nelineárních průběhů vyšších hodnot, než které obsahují registry **PG.Acc** a **PG.Dec**.

U rychlostního řízení (**PG.Mode = 0**) používá **TG Motion** pro rozběh i doběh pouze registr **PG.Acc**. Hodnota registru **PG.Dec** není do výpočtu žádané polohy zahrnována.

3.4 Příklady použití generátoru žádané hodnoty polohy PG

a) Absolutní polohování

Slouží k zapolohování servopohonu na konkrétní pozici.

Připojení generátoru polohy k pohonu pokud je Mode ≠ 1

Mode = 0

PG.APos = WritePosition – Offset – Correction

Mode = 1

Čekání: 1 perioda Cycle_Time

odpojení generátoru polohy od pohonu

inicializace výstupní polohy

připojení generátoru k pohonu

pauza na připojení (v Program_04 se nemusí provádět)

Nastavení parametru generátoru

PG.Acc = 100000

PG.Decel = 200000

PG.PosSpeed = 50000

PG.DPos = 35000

PG.Type = 2

maximální akcelerace 100000 inc/s²

maximální decelerace 200000 inc/s²

maximální rychlost 50000 inc/s

cílová poloha 35000 inkrementů [inc]

volba typu rampy

Start polohování

 PG.Rdy = 0
 PG.Mode = 1

*reset příznaku zapolohování
 start generátoru*
Test zapolohování

Čekání na PG.Rdy = 1

čekání na dosažení cílové polohy
Přerušení polohování a zastavení

 PG.Mode = 0
 PG.Speed = 0
 Čekání na PG.ASpeed = 0

*zastavení polohování
 zastavení pohybu
 test zastavení*
Změna cílové polohy za běhu

 PG.Mode = 0
 Čekání: 1 perioda Cycle_Time
 PG.DPos = nová poloha
 PG.Mode = 1

*stop polohování
 pauza na odpojení PG
 zápis nové polohy
 start generátoru*
b) Relativní polohování

Slouží k zapolohování servopohonu o určitý počet inkrementů příslušným směrem.

Připojení generátoru polohy k pohonu pokud je Mode ≠ 1

 Mode = 0
 PG.APos = WritePosition – Offset – Correction
 Mode = 1
 Čekání: 1 perioda Cycle_Time

*odpojení generátoru polohy od pohonu
 inicializace výstupní polohy
 připojení generátoru k pohonu
 pauza na připojení (v Program_04 se nemusí provádět)*
Nastavení parametru generátoru

 PG.Acc = 100000
 PG.Decel = 200000
 PG.PosSpeed = 50000
 PG.DPos = PG.APos + 10000
 PG.Type = 1

*maximální akcelerace 100000 inc/s²
 maximální decelerace 200000 inc/s²
 maximální rychlost 50000 inc/s
 cílová poloha = aktuální poloha + 10000
 volba rampy*
Start polohování

 PG.Rdy = 0
 PG.Mode = 1

*reset příznaku zapolohování
 start generátoru*
Test zapolohování

Čekání na PG.Rdy = 1

čekání na dosažení cílové polohy
Přerušení polohování a zastavení

 PG.Mode = 0
 PG.Speed = 0
 Čekání na PG.ASpeed = 0

*stop polohování
 zastavení pohybu
 test zastavení*
c) Rychlostní řízení

Slouží k nastavení rychlosti otáčení nebo posunu bez ohledu na konečnou polohu.

Připojení generátoru polohy k pohonu, pokud je Mode ≠ 1

 Mode = 0
 PG.APos = WritePosition – Offset – Correction
 Mode = 1
 Čekání: 1 perioda Cycle_Time

*odpojení generátoru polohy od pohonu
 inicializace výstupní polohy
 připojení generátoru k pohonu
 pauza na připojení*
Nastavení parametru generátoru

 PG.Acc = 100000
 PG.Speed = 20000

*maximální akcelerace 100000 inc/s²
 nastavení rychlosti na 20000 inc/s*

Změna rychlosti

PG.Speed = -20000

*změna rychlosti na -20000 inc/s***Zastavení**

PG.Speed = 0

Čekání na PG.ASpeed = 0

*zastavení pohybu**test zastavení***d) Nastavení reference**

Slouží k nastavení referenční polohy.

Odpojení generátoru polohy od pohonu

Mode = 0

Čekání: 1 perioda Cycle_Time

*PG odpojen**pauza na odpojení***Nastavení reference aktuální poloha = 100000 inc**

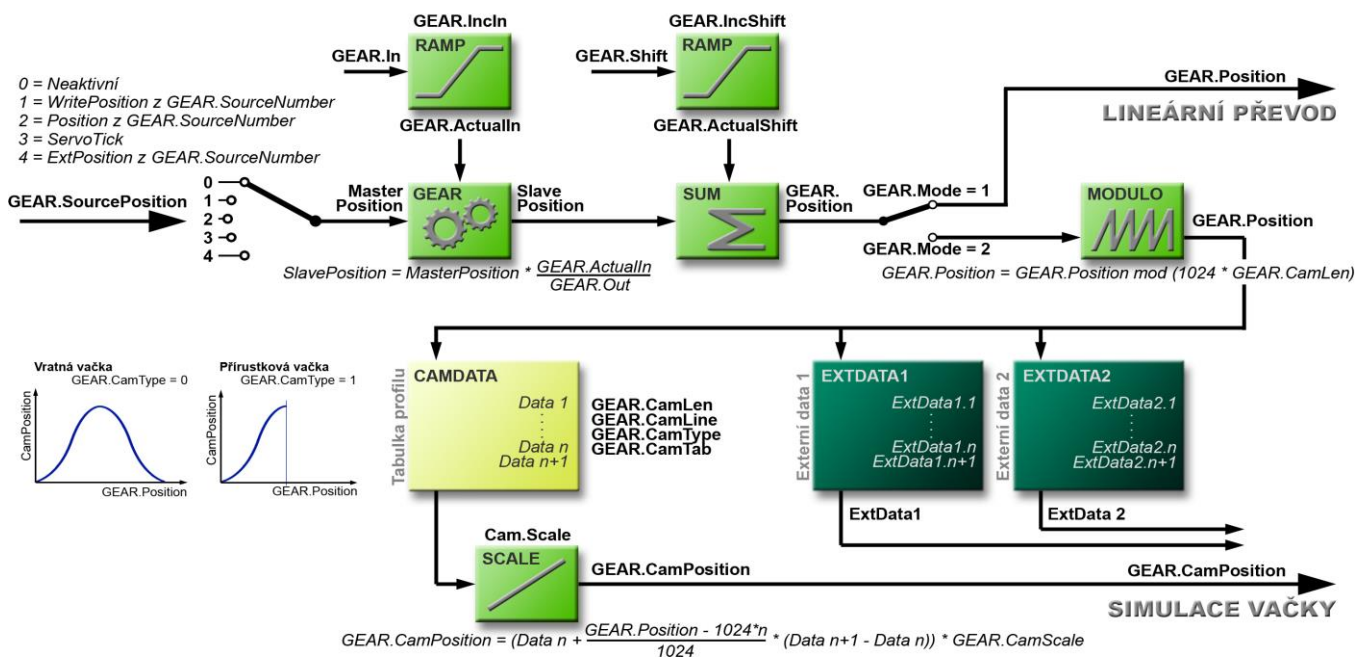
Offset = WritePosition - 100000

Nastavení reference

4. Generátor žádané polohy GEAR

4.1 Popis struktury GEAR

Generátor **GEAR** umožňuje jednoduchým způsobem synchronizovat daný pohon s libovolným jiným pohonem, přičemž zdrojem master polohy může být žádaná poloha, skutečná poloha, tik serva (časová synchronizace), nebo poloha z externího inkrementálního snímače polohy. Vlastní synchronizace pak může být lineární (simulace převodovky) nebo nelineární (simulace vačky). Simulace vačky může být vratná nebo přírůstková. Vačku i výstupní polohu lineárního převodu lze úhlově posouvat s nastavitelnou rychlostí.



Obr. Blokové schéma GEAR generátoru žádané polohy

4.2 Důležité registry

GEAR.Mode – určuje mód generátoru GEAR:

0 – klidový stav (**TG Motion** nepočítá GEAR.Position ani GEAR.CamPosition).

1 – režim lineárního převodu (**TG Motion** počítá GEAR.Position).

2 – režim simulace vačky (**TG Motion** počítá GEAR.Position i GEAR.CamPosition).

GEAR.Position – aktuální žádaná poloha v režimu lineární synchronizace (převodovka) nebo – úhel vačky v režimu nelineární synchronizace (vačky).

GEAR.CamPosition – aktuální žádaná poloha v režimu nelineární synchronizace.

GEAR.SourceNumber – zdroj master polohy.

GEAR.SourcePosition – typ master polohy.

GEAR.In a **GEAR.Out** – převodový poměr.

GEAR.CamTab – určuje sdílenou paměť, kde se nacházejí data profilu vačky:

0 – TGM_Cam_Profile.

1 – TGM_Data.

GEAR.CamLine – počátek dat profilu vačky.

GEAR.CamLen – délka dat profilu vačky.

GEAR.CamType – typ vačky.

GEAR.Shift – úhlový posun vypočtené hodnoty SlavePosition v inkrementech.

GEAR.IncShift – změna tohoto úhlového posunu [inc/servotik].

Kompletní výčet všech registrů skupiny Servo včetně popisu viz Apendix.

Výstupem generátoru je proměnná **GEAR.Position**, která určuje žádanou hodnotu polohy v režimu lineární synchronizace a úhel vačky v režimu nelineární synchronizace. Registry **GEAR.SourceNumber** a **GEAR.SourcePosition** určují zdroj a typ master polohy pro synchronizaci.

GEAR.SourcePosition – hodnoty registru a jejich význam

- 0 – gear je neaktivní.
- 1 – zdrojem polohy je WritePosition master servopohonu.
- 2 – zdrojem polohy je Position master servopohonu.
- 3 – zdrojem polohy je čas (jeden servo tik za Cycle_Time).
- 4 – zdrojem polohy je ExtPosition master servopohonu.

Převodový poměr se nastavuje pomocí dvou registrů **GEAR.In** (čítatel) a **GEAR.Out** (jmenovatel) převodového poměru. Do výpočtu žádané polohy serva může zasahovat ještě registr **GEAR.IncIn** pro simulaci spojky. Slouží k plynulému postupnému navázání na již běžící master servopohon. Registr **GEAR.Shift** určuje fázový posuv výstupní polohy registru **GEAR.Position**, **GEAR.IncShift** určuje změnu fázového posunu až do dosažení posunu daného registrem **GEAR.Shift**.

Pomocí registru **GEAR.Mode** se volí mezi lineárním převodem (převodovka) nebo nelineárním (vačky).

GEAR.Mode – hodnoty registru a jejich význam

- 1 – simulace lineárního převodu – hodnota registru **GEAR.Position** je zdrojem žádané polohy.
- 2 – simulace vačky – hodnota **GEAR.Position** je použita k výpočtu ukazatele do tabulky dat vaček.

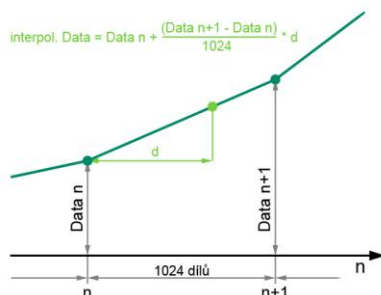
Mód simulace vačky pracuje s daty nelineární synchronizace tvořícími tabulku profilu vačky. Tabulka profilu vačky je uložena buď ve sdílené paměti **TGM_Cam_Profile** (velikosti 1048576 byte), nebo ve sdílené paměti **TGM_Data** (velikosti 524288 byte). Přepínání mezi tabulkami v jednotlivých paměťových oblastech je možné nastavením registru **GEAR.CamTab**. Stejný registr určuje umístění obou tabulek externích dat.

GEAR.CamTab – hodnoty registru a jejich význam

- 0 – data se čtou z **TGM_Cam_Profile**.
- 1 – data se čtou z **TGM_Data**.

O plnění tabulky profilu vaček se stará uživatelská Windows aplikace (např. program vizualizace) nebo PLC. Registry **GEAR.CamLine** a **GEAR.CamLen** určují počátek a počet dat profilu vačky.

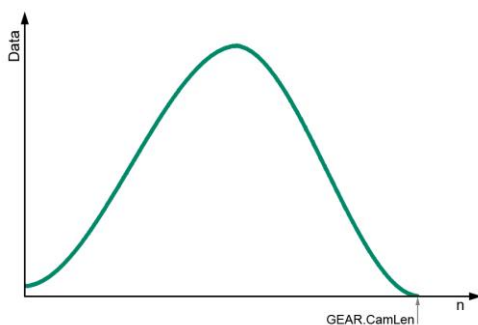
V režimu nelineární synchronizace je výstupem **GEAR** generátoru polohy registr **GEAR.CamPosition**, který určuje aktuální žádanou hodnotu polohy servopohonu. V sekci MODULO je pomocí registru **GEAR.Position** vypočítáno ukazovátko do tabulky vaček, které je 1024krát jemnější než data v tabulce, tedy jeden krok v tabulce je rozdělen na 1024 dílů. Platí, že změna **GEAR.Position** o 1024 inkrementů znamená posun v tabulce profilu vačky o jednu hodnotu. Výsledná hodnota polohy servopohonu (**GEAR.CamPosition**) je počítána lineární interpolací mezi dvěma sousedními hodnotami tabulky.



Obr. Lineární interpolace mezi daty tabulky CAMDATA

4.3 Vratná vačka

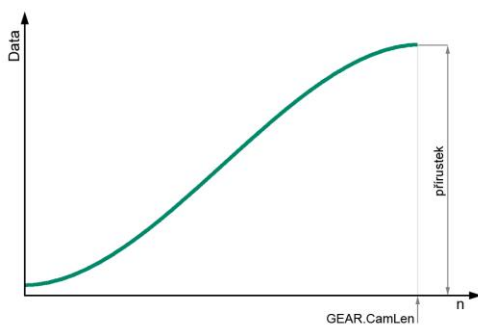
Slouží k periodickému nelineárnímu pohybu po uzavřené křivce – simulaci vačky na hřídeli. **TG Motion** lineárně interpoluje mezi dvěma sousedními hodnotami tabulky rozdělením pomyslného intervalu na 1024 dílů. Při návratu na začátek tabulky lineárně interpoluje mezi posledními a prvními daty tabulky. Tabulku dat je tedy vhodné začínat nenulovou hodnotou a sekvenci dat končit hodnotou nulovou.



Obr. Vratná vačka

4.4 Přírůstková vačka

Slouží k periodickému nelineárnímu pohybu, kde začátek následujícího cyklu plynule navazuje na konec předchozího. **TG Motion** opět lineárně interpoluje mezi dvěma sousedními daty tabulky rozdělením pomyslného intervalu na 1024 dílů. Na začátku tabulky probíhá interpolace mezi explicitní hodnotou **0** a první hodnotou dat tabulky. Při dosažení konce tabulky jsou poslední data brána jako přírůstek (zdvih) přírůstkové vačky, který je při následujícím průchodu tabulkou připočítáván k interpolovaným datům přírůstkové vačky tak, aby byl pohyb servopohonu plynule navázán. Tabulku dat je tedy vhodné začínat nenulovou hodnotou.



Obr. Přírůstková vačka

4.5 Externí tabulky dat

K několika synchronizovaným nelineárním pohybům lze využít dalších dvou nezávislých tabulek dat EXTDATA1 a EXTDATA2. Ty mohou sloužit k ovládní dalších periferních zařízení, která musejí být synchronní s převodem, ale mají mít jiné průběhy než vačka. K výpočtu hodnot jejich výstupů **ExtData1** a **ExtData2** slouží stejné ukazovátka do paměti jako pro paměť CAMDATA. Výstupní hodnoty se pak zapisují do paměti Data na offsety dané registry **CamExpAddress1** a **CamExpAddress2**. Tabulky CAMDATA, EXTDATA1 a EXTDATA2 musejí být stejně velké. Umístění tabulek ve sdílené paměti určuje registr **GEAR.CamTab** (stejný registr, který určuje umístění tabulky dat profilu vačky).

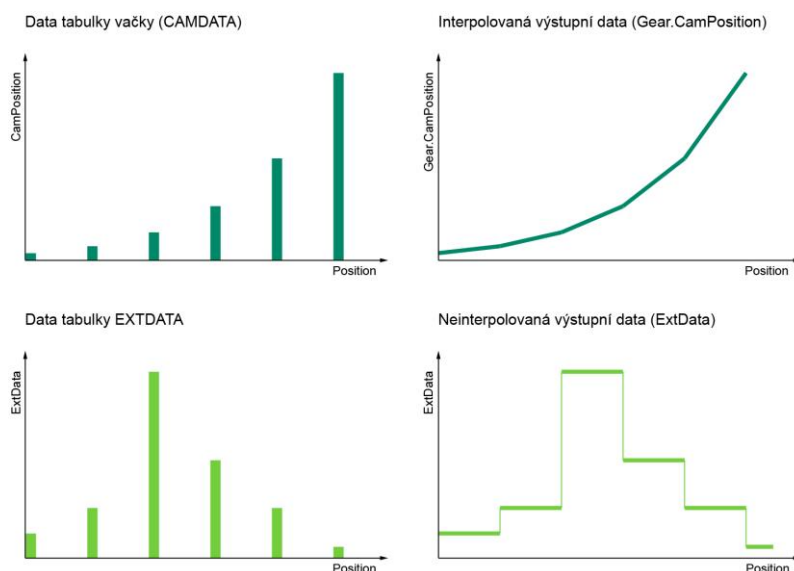
GEAR.CamTab – hodnoty registru a jejich význam

0 – data se čtou z **TGM_Cam_Profile**.

1 – data se čtou z **TGM_Data**.

O plnění tabulky profilu vaček se stará uživatelská Windows aplikace (např. program vizualizace) nebo PLC.

K datům externích tabulek se přistupuje stejně jako k datům tabulky vačky s tím, že při výpočtu výstupních registrů **ExtData1** a **ExtData2** neprobíhá interpolace mezi sousedními daty externích tabulek. Po celou dobu, kdy v tabulce CAMDATA probíhá interpolace, se z tabulek EXTDATA1 a EXTDATA2 používá pro výpočet hodnoty **ExtData1** a **ExtData2** poslední načtená hodnota. Při dosažení dalších dat v tabulce (když je ukazovátka beze zbytku dělitelné 1024) se začnou používat nově načtená data.



Obr. Interpolovaná data tabulky vačky a neinterpolovaná data externích tabulek

4.6 Příklady použití generátoru žádané hodnoty polohy GEAR

a) Lineární převod

Slouží k lineární synchronizaci dvou servopohonů.

Nastavení parametrů převodu

GEAR.SourceNumber = 3
 GEAR.SourcePosition = 1

*synchronizace na servopohon č. 3
 zdrojem polohy je žádaná hodnota polohy servopohonu
 č. 3*

GEAR.ActualIn = 0
 GEAR.IncIn = 10
 GEAR.In = -1000
 GEAR.Out = 2000
 GEAR.Position = 0
 GEAR.Shift = 0
 GEAR.ActualShift = 0

*reset aktuální hodnoty převodového poměru (čitatele)
 nastavení přírůstku převodového poměru smyčky
 určuje žádanou hodnotu převodového poměru (čitatele)
 určuje převodový poměr (jmenovatel)
 reset výstupní polohy generátoru
 reset žádané hodnoty posunutí výstupní polohy
 reset skutečné hodnoty posunutí výstupní polohy*

Spuštění generátoru

GEAR.Mode = 1
 GEAR.Offset = WritePosition - Offset - GEAR.Position
 Mode = 3

*připojení generátoru polohy k pohonu
 inicializace offsetu
 připojení generátoru k pohonu*

Změna převodového poměru za běhu

GEAR.IncIn = 10
 GEAR.In = GEAR.In + 6000
 Čekání na GEAR.ActualIn = GEAR.In

*dynamika změny převodového poměru
 žádaná změna převodového poměru
 test dokončení změny převodového poměru*

Úhlové posunutí

GEAR.IncShift = 10
 GEAR.Shift = 2000
 Čekání na GEAR.ActualShift = GEAR.Shift

*dynamika posunutí
 start posunutí o 2000 inc
 test provedení posunutí*

Desynchronizace

GEAR.IncIn = 40
 GEAR.In = 0
 Čekání na GEAR.ActualIn = GEAR.In
 Mode = 0
 GEAR.Mode = 0

*dynamika změny převodového poměru
 reset převodového poměru
 test provedení desynchronizace
 odpojení generátoru polohy od pohonu
 zastavení generátoru polohy*

b) Nelineární převod

Slouží k nelineární synchronizaci dvou servopohonů (simulace vaček).

Nastavení parametrů převodu

GEAR.SourceNumber = 2
 GEAR.SourcePosition = 4

*synchronizace na servopohon č. 2
 zdrojem polohy je hodnota externího snímače polohy
 (např. IRC servopohonu č. 2)*

GEAR.ActualIn = 0
 GEAR.IncIn = 10

*reset aktuální hodnoty převodového poměru (čitatele)
 nastavení přírůstku převodového poměru – čitatele
 v každé periodě Cycle_Time*

GEAR.In = 0
 GEAR.Out = 2000
 GEAR.Position = 0
 GEAR.Shift = 0
 GEAR.ActualShift = 0
 GEAR.CamType = 0

*reset žádané hodnoty převodového poměru – čitatele
 určuje převodový poměr (jmenovatel)
 reset výstupní polohy generátoru
 reset žádané hodnoty posunutí výstupní polohy
 reset skutečné hodnoty posunutí výstupní polohy
 vratná vačka*

GEAR.CamScale = 1
GEAR.CamTab = 0
GEAR.CamLine = 100
GEAR.CamLen = 1000

*měřítko dat z tabulky vačky (1 odpovídá měřítku 1 : 1)
tabulka dat vačky je uložena v paměti TGM_Cam_Profile
adresa začátku dat vačky v paměti
počet dat vačky*

Spuštění generátoru

GEAR.Mode = 2
Čekání: 1 perioda Cycle_Time

*start simulace vačky
pauza na spuštění generátoru*

Připojení generátoru polohy k pohonu

GEAR.Offset = WritePosition - Offset - GEAR.CamPosition
Mode = 3

*inicializace offsetu
připojení generátoru k pohonu*

Start vačky

GEAR.In = -1000

určuje žádanou hodnotu převodového poměru (čitatele)

Změna převodového poměru za běhu (rychlost průchodu tabulkou vačky)

GEAR.Incln = 10
GEAR.In = GEAR.In + 6000
Čekání na GEAR.Actualln = GEAR.In

*dynamika změny převodového poměru
žádaná změna převodového poměru
test provedení změny převodového poměru*

Úhlové posunutí

GEAR.IncShift = 10
GEAR.Shift = 2000
Čekání na GEAR.ActualShift = GEAR.Shift

*dynamika posunutí
start posunutí o 2000 inkrementů
test provedení posunutí*

Desynchronizace

GEAR.Incln = 40
GEAR.In = 0
Čekání na GEAR.Actualln = GEAR.In
Mode = 0
GEAR.Mode = 0

*dynamika změny převodového poměru
reset převodového poměru
test provedení desynchronizace
odpojení generátoru polohy od pohonu
zastavení generátoru polohy*

5. Funkční rozhraní COMMAND

5.1 Důležité registry

Command.Control – zapsáním čísla funkce do tohoto registru se daná funkce vykoná.

Command_Par [0–14] – parametry jednotlivých funkcí registru **Command.Control**.

Kompletní výčet všech registrů skupiny Servo včetně popisu viz Apendix.

5.2 Popis struktury Command

Funkční rozhraní **COMMAND** uživateli zjednodušuje použití jednotlivých generátorů polohy PG a GEAR. Zjednodušení spočívá v tom, že systém si sám kontroluje časovou synchronizaci zápisu příslušných registrů s cyklem servosmyčky a odpadá tudíž používání časové prodlevy při práci s jednotlivými generátory polohy. Výsledný PLC kód je pak úspornější v počtu použitých instrukcí a jeho vykonávání je rychlejší.

Vlastní rozhraní je realizováno obecně. Uživatel zapsáním hodnoty $\neq 0$ do registru **Command.Control** volá požadovanou funkci. Volané funkci je možné předat až 15 parametrů zapsaných do proměnných **Command.Par [0–14]**. Systém pak signalizuje vykonání funkce vynulováním registru **Command.Control**.

Command.Control – hodnoty a význam předávaných parametrů

Command.Control = 0 – předchozí funkce je vykonána

Po akceptování funkce vynuluje **TG Motion** hodnotu registru **Command.Control** a je připraven vykonat další funkci. Výjimkou je deaktivace GEAR generátoru (**Command.Control = 7** – viz dále).

Command.Control = 1 (run speed) PG – rychlostní řízení

Command.Par [0] = Acc	žádané zrychlení (akceleraace)
Command.Par [2] = Speed	žádaná rychlost
Command.Par [4] = Enable_GEAR	chování se k aktivnímu GEAR pohonu: 0 – pokud je GEAR aktivní, zruší jej a aktivuje PG 1 – na GEAR nasuperponuje PG
Command.Par [5] = Type	typ rampy (0–3)

Command.Control = 2 (pos abs) PG – absolutní polohování

Command.Par [0] = Acc	žádané zrychlení (akceleraace)
Command.Par [1] = Dec	žádané zpomalení (deceleraace)
Command.Par [2] = Speed	žádaná rychlost
Command.Par [3] = Pos	absolutní žádaná poloha
Command.Par [4] = Enable_GEAR	vztah k aktivnímu GEAR pohonu: 0 – pokud je GEAR aktivní, zruší jej a aktivuje PG 1 – na GEAR nasuperponuje PG
Command.Par [5] = Type	typ rampy (0–3)

Command.Control = 3 (pos rel) PG – relativní polohování

Command.Par [0] = Acc	žádané zrychlení (akcelerace)
Command.Par [1] = Dec	žádané zpomalení (decelerace)
Command.Par [2] = Speed	žádaná rychlost
Command.Par [3] = Pos	změna pozice, o kolik inkrementů kterým směrem se má aktuální poloha serva změnit
Command.Par [4] = Enable_GEAR	vztah k aktivnímu GEAR pohonu: 0 – pokud je GEAR aktivní, zruší jej a aktivuje PG 1 – na GEAR nasuperponuje PG
Command.Par [5] = Type	typ rampy (0–3)

Command.Control = 4 (ref pos) PG – nastavení reference

Command.Par [3] = Ref	aktuální poloha je prohlášena za polohu danou tímto parametrem
-----------------------	----------------------------------------------------------------

Command.Control = 5 (pos abort) PG – zrušení jakéhokoli polohování a zastavení servopohonu

Command.Par [1] = Dec	žádané zpomalení (decelerace)
Command.Par [5] = Type	typ rampy (0–3)



Pro aktivaci GEAR generátoru slouží hodnota **Command.Control = 6**. Aktivace **lineárního převodu** nebo **vačky** se liší hodnotou parametru **Command.Par [2]** a nastavením dalších příslušných parametrů.

Command.Control = 6 (sync on) GEAR – aktivace GEAR generátoru – lineární převod

Command.Par [0] = SourceNumber	číslo master servopohonu
Command.Par [1] = SourcePosition	zdroj synchronizace
Command.Par [2] = 1	lineární převod
Command.Par [3] = GIn	převodový poměr (čítatel)
Command.Par [4] = GOut	převodový poměr (jmenovatel)
Command.Par [5] = GInInc	dynamika změny převodového poměru
Command.Par [9] = GEAR.IniPosition	inicializační hodnota GEAR.Position v momentě zapnutí GEAR
Command.Par [12] = ServoMode	nastavení zdroje žádané polohy

Command.Control = 6 (sync on) GEAR – aktivace GEAR generátoru – vačka

Command.Par [0] = SourceNumber	číslo master servopohonu
Command.Par [1] = SourcePosition	zdroj synchronizace
Command.Par [2] = 2 + CamType	vačka
Command.Par [3] = GIn	převodový poměr (čítatel)
Command.Par [4] = GOut	převodový poměr (jmenovatel)
Command.Par [5] = GInInc	dynamika změny převodového poměru
Command.Par [9] = GEAR.IniPosition	inicializační hodnota GEAR.Position v momentě zapnutí GEAR
Command.Par [12] = ServoMode	nastavení zdroje žádané polohy
Command.Par [6] = CamLine	začátek dat vačky
Command.Par [7] = CamLen	délka dat vačky
Command.Par [10] = CamTab	lokalizace paměti dat vačky (TGM_Cam_Profile, TGM_Data)
Command.Par [8] = CamScale	měřítka výstupní hodnoty vačky (koeficient)
Command.Par [13] = CamIncPosition	explicitně zadaný zdvih přírůstkové vačky pro eliminaci chyby vypočteného zdvihu vzniklé převodem reálného čísla (double) na celé číslo (integer) – (pouze u přírůstkové vačky)

Command.Control = 7 (sync off) GEAR – deaktivace GEAR generátoru

Command.Par [5] = GInInc

zadává se kladné číslo jako lineární úbytek čitatele převodového poměru – servopohon začne brzdit

*U **Command.Control = 7** (deaktivace GEAR generátoru) dojde k vynulování registru **Command.Control** až po srampování převodového poměru, tedy až proběhne celá rampa.*

5.3 Příklady použití funkčního rozhraní Command

a) Start absolutního polohování

Slouží k zapolohování příslušného servopohonu na konkrétní polohu/pozici.

Command_Par [0] = 1000000	<i>požadovaná akcelerace [inc/s²]</i>
Command_Par [1] = 2000000	<i>požadovaná decelerace [inc/s²]</i>
Command_Par [2] = 300000	<i>požadovaná rychlost [inc/s]</i>
Command_Par [3] = 200000	<i>požadovaná poloha [inc]</i>
Command.Control = 2	<i>start absolutního polohování</i>

 Čekání na Command.Control = 0
 Čekání na PG.Rdy = 1

*čekání na vykonání funkce
 čekání na zapolohování*

b) Start synchronizace – nelineární převod (vačka)

Slouží k nelineární synchronizaci dvou servopohonů (simulace vaček).

Command_Par [0] = 10	<i>číslo master servopohonu</i>
Command_Par [1] = 2	<i>zdrojem polohy je registr Position master servopohonu</i>
Command_Par [2] = 3	<i>režim simulace přírůstkové vačky</i>
Command_Par [3] = 32000	<i>čítatel převodového poměru synchronizace</i>
Command_Par [4] = 32767	<i>jmenovatel převodového poměru synchronizace</i>
Command_Par [5] = 10	<i>přírůstek (pokles) čitatele převodového poměru synchronizace během rampování převodového poměru za jeden servo cyklus</i>
Command_Par [6] = 0	<i>adresa počátku profilu dat vačky v paměti</i>
Command_Par [7] = 100	<i>počet dat profilu vačky</i>
Command_Par [8] = 1.0	<i>měřítka přepočtu dat tabulky vačky 1 : 1 (koeficient)</i>
Command_Par [9] = 1024	<i>inicializační hodnota ukazatele do tabulky vačky</i>
Command_Par [10] = 1	<i>tabulka vačky ve sdílené paměti TG Motion</i>
Command = 6	<i>start synchronizace</i>

Čekání na Command.Control = 0

čekání na vykonání funkce

c) Start desynchronizace

Slouží k odpojení synchronizovaného servopohonu od zdroje master polohy.

Command_Par [5] = 20	<i>přírůstek (pokles) čitatele převodového poměru synchronizace během rampování převodového poměru za jeden servo cyklus</i>
Command.Control = 7	<i>start desynchronizace</i>

Čekání na Command.Control = 0

čekání na vykonání funkce

*V PLC programu je nutné zajistit, aby funkce **start synchronizace** nebyla volána opakovaně. Pro nové nasyynchronizování se musí nejprve provést desynchronizace a teprve potom může být zavolána funkce synchronizace. Pro funkci **start desynchronizace** platí obdobné pravidlo.*



Pro všechny hodnoty funkce *Command* vynuluje **TG Motion** registr **Command.Control** již po akceptování funkce; poté, co bylo zahájeno její vykonávání. Funkce sama nemusí být dokončena.

Pouze u funkce **start desynchronizace** (**Command.Control = 7**) dojde k vynulování registru **Command.Control** až po srampování převodového poměru; tj. poté, až proběhne celá rampa.

6. Rozhraní pro komunikaci s objekty SDO

6.1 Popis struktury SDO

Struktura pro komunikaci s objekty SDO (Service Data Objects) slouží k parametrizaci konkrétního servozesilovače. Uživatel ze servisního manuálu konkrétního servozesilovače zjistí adresu (index a subindex) potřebného parametru (objektu) a jeho velikost v bytech. Pak může do příslušného parametru zapsat potřebnou hodnotu, nebo aktuální hodnotu parametru přečíst.

6.2 Důležité registry

SDO.Control – řízení zápisu a čtení SDO.

SDO.Status – určuje stav zápisu a čtení SDO, příp. hlášení chyby.

SDO.Index – index objektu SDO.

SDO.SubIndex – subindex objektu SDO.

SDO.Data – data k zápisu nebo přečtená data, v případě chyby (SDO.Status=2) kód chyby.

Kompletní výčet všech registrů skupiny Servo včetně popisu viz Apendix.

6.3 Popis registrů a práce s nimi

Rozhraní slouží pro zápis a čtení libovolných parametrů daného servopohonu prostřednictvím komunikace SDO. Řízení komunikace je umožněno prostřednictvím registrů **SDO.Control** a **SDO.Status**. Adresace jednotlivých parametrů se děje prostřednictvím registrů **SDO.Index** a **SDO.SubIndex**. Hodnota těchto proměnných je specifická podle typu servopohonu; tabulka přiřazení adres jednotlivých parametrů je uvedena v manuálu příslušného servopohonu.

6.4 Příklady použití struktury SDO

a) Zápis parametru

Start zápisu

SDO.NumberByte = 4

SDO.Index = 24672

SDO.SubIndex = 1

SDO.Data = 1265

SDO.Control = 1

počet bytů daného parametru (1–4)

index daného parametru

subindex daného parametru

zapisovaná data

start zápisu

Čekání na SDO.Control = 0

čekání na konec zápisu

Test správnosti provedení zápisu

Když SDO.Status = 0

Když SDO.Status = 2

zápis proveden

chyba zápisu, kód chyby je uložen v registru SDO.Data

b) Čtení parametru**Start čtení**

SDO.NumberByte = 2

SDO.Index = 24672

SDO.SubIndex = 0

SDO.Control = 2

*počet bytů daného parametru (1–4)**index daného parametru**subindex daného parametru**start čtení*

Čekání na SDO.Control = 0

*čekání na konec čtení***Test správnosti provedení čtení***Když SDO.Status = 0**pak Data = SDO.Data**čtení ukončeno, aktuální data jsou v registru v SDO.Data**Když SDO.Status = 2**chyba čtení, kód chyby je uložen v registru SDO.Data*

7. Rozhraní CAPTURE

7.1 Popis struktury Capture

Rozhraní **CAPTURE** slouží k zaznamenání událostí s vyšší přesností, než jakou nabízí **Cycle_Time**. Zachycená událost obsahuje přesnou polohu serva v momentě, kdy k události došlo. Současně jsou zaznamenány i další hodnoty a vepsány do příslušných registrů. Splnění podmínky zachycení hlídá sám servozsilovač na základě nastavených parametrů. **TG Motion** si pak událost přečte v nejbližším cyklu **Cycle_Time**.

7.2 Důležité registry

CAPTURE.Control – ovládání zachycování polohy:

0 – zachycení vypnuto.

1 – povoleno zachycení od externího digitálního vstupu (pouze u servozsilovačů AKD a ServoStar).

2 – povoleno jednorázové zachycení registrů systémem **TG Motion**.

CAPTURE.Type_ExtCapture – číslo digitálního vstupu a typ hrany, na niž se bude zachycování synchronizovat:

bit 0 – povolení zachycení na náběžnou hranu digitálního vstupu 1.

bit 1 – povolení zachycení na sestupnou hranu digitálního vstupu 1.

bit 2 – povolení zachycení na náběžnou hranu digitálního vstupu 2.

bit 3 – povolení zachycení na sestupnou hranu digitálního vstupu 2.

bit 4 – povolení zachycení na nulový puls snímače polohy.

CAPTURE.PPosition – skutečná poloha servopohonu při náběžné hraně digitálního vstupu.

CAPTURE.NPosition – skutečná poloha servopohonu při sestupné hraně digitálního vstupu.

CAPTURE.PExt_Position – hodnota polohy snímače polohy při náběžné hraně digitálního vstupu.

CAPTURE.NExt_Position – hodnota polohy snímače polohy při sestupné hraně digitálního vstupu.

CAPTURE.Write_Position – zachycená požadovaná poloha servopohonu.

CAPTURE.Offset – adresový offset proměnné zachycené do **CAPTURE.Data**.

CAPTURE.DData – zachycená data ze sdílené paměti **TGM_Data** z registru **Offset** jako double (8 bytů).

CAPTURE.Zero_Position – skutečná poloha servopohonu v momentě příchodu nulového pulsu snímače polohy (**CAPTURE.Type_ExtCapture** – bit 4).

Kompletní výčet všech registrů skupiny Servo včetně popisu viz Apendix.

7.3 Popis funkce struktury Capture

Rozhraní **CAPTURE** umožňuje v okamžiku příchodu externího digitálního signálu zachytit aktuální polohu daného servopohonu a polohu ze snímače polohy s časovou tolerancí menší než 100 μs. Externí digitální signál je přiveden přímo na digitální vstup servozsilovače.

Dále rozhraní **CAPTURE** umožňuje zachytit v rámci jednoho **Cycle_Time** žádanou polohu servopohonu a libovolný registr sdílené paměti.

Po nastavení registru **Capture.Control** na nenulovou hodnotu dojde k načtení a uložení hodnot do příslušných registrů. Poté je registr **Capture.Control** vynulován; tak **TG Motion** informuje o tom, že k zachycení došlo. Následně lze vyčíst požadované hodnoty.

7.4 Příklad použití rozhraní Capture

Zachycení na externí digitální signál

Capture.Control = 1

Čekání na Capture.Control = 0

aktivace zachycení

test, zda došlo k zachycení a vyčtení zachycených poloh

8. Apendix

Přehled a popis registrů skupiny Servo

Základní registry

název	přístup	offset	popis
Number	R	0	číslo namapovaného servopohonu, logické číslo interface (Servo 0 – Number = 0, Servo 1 – Number = 1 atd.)
Node	R	4	adresa fyzického servopohonu nastavená na servozesilovači
Axe	R	8	číslo osy na daném servozesilovači (v případě víceosých servozesilovačů)
Type	R	12	typ serva připojeného k danému servozesilovači (uvedeno v souboru TGMotion4xx.ini): 0 = nepřipojen žádný servopohon 1 = připojen servopohon ServoStar 700 2 = připojen servopohon TGA 21 = připojen servopohon TGA firmware 3.00 a vyšší 3 = připojen servopohon TGA300 4 = připojen servopohon TGP 41 = připojen servopohon TGP s externím snímačem polohy 42 = připojen servopohon TGP (32 bit control word) 43 = připojen servopohon TGP s externím snímačem polohy (32 bit control word) 5 = připojen servopohon AKD 6 = připojen servopohon TGA220 7 = připojen servopohon PIERCE retrofit 8 = připojen servopohon DCM 9 = připojen servopohon TGZ
Resolution	R	16	počet inkrementů na 1 otáčku servomotoru [inc]
SerialNumber	R	20	výrobní číslo servozesilovače
Control	RW	24	bit 0 = reset poruchy bit 1 = není pod momentem bit 2 = reset EtherCAT komunikace bit 3 = povoluje automatický výpočet registru TorqueFeedForward
Status	R	28	udává aktuální stav servopohonu: -1 = nekomunikuje (po předchozí komunikaci) 0 = nekomunikuje, komunikaci se vůbec nepodařilo navázat 1 = servopohon v poruše 2 = připraven, bez momentu 3 = připraven, pod momentem
Mode	RW	32	určuje zdroj žádané polohy: 0 = žádaná hodnota odpojena (do serva je posílána jeho aktuální poloha) 1 = zdrojem žádané hodnoty je profil generátor PG 3 = zdrojem žádané hodnoty je GEAR generátor polohy 4 = zdrojem žádané hodnoty je PG a GEAR generátor 6 = zdrojem žádané hodnoty je interpolátor 7 = zdrojem žádané hodnoty je PG a interpolátor 8 = zdrojem žádané hodnoty je GEAR generátor a interpolátor
Error	R	36	určuje kód první zachycené poruchy, pokud se pohon nachází v poruše; význam hodnot závisí na konkrétním typu servopohonu
EtherCATState	R	40	stav komunikace servopohonu z hlediska EtherCAT: 0x01 = Init 0x02 = Pre-Operational 0x04 = Safe-Operational 0x08 = Operational
DigitalIn	R	44	stavy digitálních vstupů; počet vstupů závisí na konkrétním typu servozesilovače
DigitalOut	RW	48	stavy digitálních výstupů; nastavením a nulováním jednotlivých bitů lze ovládat jednotlivé digitální výstupy; počet výstupů závisí na konkrétním typu servozesilovače
TorqueFeedForward	RW	52	hodnota momentu posílaná do servopohonu; funkce je podporována u servozesilovačů AKD a TGZ: AKD: rozsah -3000 až +3000, kde 3000 = špičkový proud servozesilovače TGZ: rozsah -32767 až +32767, kde 32767 = špičkový proud servozesilovače
ControlWord	R	56	kopie CAN control word
StatusWord	R	60	kopie CAN status word
SysTimeDifference	R	64	poslední známý rozdíl mezi lokálním časem a referenčním časem [ns]
WritePosition	RW	464	určuje hodnotu žádané polohy, která je aktuálně posílaná do servopohonu [inc]
Position	R	472	skutečná aktuální poloha servopohonu [inc]
RefPosition	R	480	skutečná poloha motoru posunutá o Offset (Position + Offset) [inc]
ExtPosition	R	488	poloha čtená z externího snímače polohy servozesilovače
Offset	RW	496	umožňuje posunutí polohy motoru v inkrementech [inc]

název	přístup	offset	popis
Correction	RW	504	korekce WritePosition akceptovaná v případě Mode ≠ 0. TG Motion do tohoto registru nezapisuje [inc]
PositionError	R	512	rozdíl zapisované hodnoty pozice a reálné pozice načtené ze servopohonu; WritePosition – Position [inc]
WriteSpeed	R	520	vypočtená žádaná rychlost servopohonu (derivace registru WritePosition) [inc/s]
Speed	R	528	vypočtená aktuální rychlost servopohonu (derivace registru Position) [inc/s]
LimitCurrent	RW	536	omezení proudu servozesilovače; hodnota závisí na konkrétním typu servozesilovače
Current	R	544	proud tekoucí do motoru; hodnota závisí na konkrétním typu servozesilovače
MaxCurrent	R	552	určuje maximální proud servozesilovače (registru není podporován pro všechny typy servozesilovačů)
AnalogIn	R	560	hodnota analogového vstupu daného servozesilovače; jednotky a rozsah závisí na konkrétním typu servozesilovače
WriteAcc	R	568	žádané zrychlení (derivace WriteSpeed) [inc/s ²]
AccMaxTorqueFeedForward	RW	576	měřítka (scale) – při nastaveném bitu 3 registru Control se podle hodnoty WriteAcc a AccMaxTorqueFeedForward počítá TorqueFeedForward

Registry generátoru profilu (PG)

název	přístup	offset	popis
Acc	RW	1296	žádaná hodnota zrychlení [inc/s ²]
Dec	RW	1304	žádaná hodnota zpomalení [inc/s ²]
APos	RW	1312	aktuální hodnota polohy [inc]
DPos	RW	1320	cílová poloha servopohonu [inc]
ASpeed	R	1328	aktuální vypočtená rychlost [inc/s]
PosSpeed	RW	1336	maximální rychlost pohybu v polohovém řízení [inc/s]
Speed	RW	1344	žádaná hodnota rychlosti servopohonu v rychlostním řízení [inc/s]
Mode	RW	1432	mód generátoru polohy: 0 = rychlostní řízení nebo uživatelem vynucené ukončení polohového řízení 1 = polohové řízení 3 = probíhá fáze brždění (doběh generátoru rampy)
Rdy	RW	1436	status PG generátoru: 1 = dosažena cílová poloha
Type	RW	1440	typ rampy, podle níž se mění rychlost (0–3)

Registry pro práci s objekty SDO

název	přístup	offset	popis
Control	RW	1496	řízení zápisu a čtení SDO: 0 = komunikace proběhla 1 = požadavek zápisu 2 = požadavek čtení
Status	R	1500	průběh komunikace, příp. zpráva, zda komunikace proběhla úspěšně: 0 = komunikace proběhla v pořádku 1 = probíhá komunikace 2 = chyba komunikace
NumberByte	W	1504	datová velikost SDO [byty]
Index	W	1508	adresa, odkud se mají data číst, nebo kam se mají zapisovat [byty]
SubIndex	W	1512	podadresa, odkud se mají data číst nebo kam se mají zapisovat [byty]
Data	RW	1516	data k zápisu nebo načtená data, příp. kód chyby

Registry GEAR generátoru

název	přístup	offset	popis
Position	RW	1520	žádaná poloha v režimu lineární synchronizace, nebo úhel vačky v režimu nelineární synchronizace [inc]
CamPosition	RW	1528	žádaná poloha v režimu nelineární synchronizace [inc]
Offset	RW	1536	offset žádané polohy [inc]
Shift	RW	1544	úhlový posun žádané hodnoty [inc]
ActualShift	R	1552	skutečná hodnota posunutí žádané polohy [inc]
IncShift	RW	1560	změna úhlového posunu [inc/servotick]
Incln	RW	1568	rychlost změny převodového poměru GEAR [inc/servotick]
CamIncPosition	RW	1576	úhlový posun přírůstkové vačky [inc]
CamScale	RW	1584	měřítka dat vačky – koeficient, kterým se násobí data vačky
SourceNumber	RW	1656	číslo namapovaného servopohonu sloužícího jako zdroj master polohy
SourcePosition	RW	1660	typ master polohy: 0 = GEAR neaktivní 1 = WritePosition master servopohonu (žádaná poloha) 2 = Position master servopohonu (aktuální skutečná poloha) 3 = ServoTick master servopohonu 4 = ExtPosition master servopohonu (aktuální poloha z externího snímače master servopohonu)
Mode	RW	1664	mód generátoru GEAR: 0 = GEAR neaktivní 1 = režim lineárního převodu 2 = režim simulace vačky
In	RW	1668	čítatel převodového poměru
Out	RW	1672	jmenovatel převodového poměru
ActualIn	R	1676	aktuální čítatel převodového poměru
CamLine	RW	1680	offset počátku dat profilu vačky [byty]
CamLen	RW	1684	počet dat vačky – délka tabulky dat vačky
CamType	RW	1688	typ vačky: 0 = vratná vačka 1 = přírůstková vačka
CamTab	RW	1692	umístění tabulky dat vačky ve sdílené paměti: 0 = TGM_Cam_Profile 1 = TGM_Data
CamExpControl	RW	1696	stav aktivity externích tabulek dat (bitová mapa): bit 0 = použije se tabulka EXTDATA1 (0 = neaktivní, 1 = aktivní) bit 1 = použije se tabulka EXTDATA1 (0 = neaktivní, 1 = aktivní)
CamExpLine1	W	1700	offset počátku dat profilu EXTDATA1 [byty]
CamExpLine2	W	1704	offset počátku dat profilu EXTDATA2 [byty]
CamExpAddress1	W	1708	offset proměnné ve sdílené paměti TGM_Data, kam TG Motion zapisuje aktuální výstupní hodnotu tabulky EXTDATA1 [byty]
CamExpAddress2	W	1712	offset proměnné ve sdílené paměti TGM_Data, kam TG Motion zapisuje aktuální výstupní hodnotu tabulky EXTDATA2 [byty]

Registry MAP_CNC (interpolátor)

název	přístup	offset	popis
Number	W	1776	určuje číslo Interpolátoru, na který je daný servopohon namapován: 0 = je mapován na CNC 0 1 = je mapován na CNC 1 2 = je mapován na CNC 2
NumberAxes	W	1780	určuje číslo osy zvoleného interpolátoru (registr Cnc.Number); číslo osy odpovídá ose v G-kódu podle následující tabulky: 0 = X 1 = Y 2 = Z 3 = C 4 = B 5 = U 6 = V 7 = W 8 = A 9 = O

Registry CAPTURE

název	přístup	offset	popis
Control	RW	1792	ovládání zachycování polohy: 0 = zachycení zakázáno 1 = povoleno zachycení od externího digitálního vstupu (pouze u servozesilovačů AKD a ServoStar) 2 = povoleno jednorázové zachycení registrů systémem TG Motion
Typ_ExtCapture	W	1796	číslo digitálního vstupu a typ hrany, na niž se bude zachycování synchronizovat: bit 0 = povolení zachycení na náběžnou hranu digitálního vstupu 1 bit 1 = povolení zachycení na sestupnou hranu digitálního vstupu 1 bit 2 = povolení zachycení na náběžnou hranu digitálního vstupu 2 bit 3 = povolení zachycení na sestupnou hranu digitálního vstupu 2 bit 4 = povolení zachycení na nulový puls snímače polohy (enkodéru)
Offset	W	1800	adresový offset proměnné zachycené do Capture.Data [byty]
IData	RW	1804	zachycená data ze sdílené paměti TGM_Data z registru Offset; data jsou typu integer (4 byty)
PPosition	R	1824	skutečná poloha servopohonu při náběžné hraně digitálního vstupu [inc]
NPosition	R	1832	skutečná poloha servopohonu při sestupné hraně digitálního vstupu [inc]
PExt_Position	R	1840	hodnota polohy externího snímače polohy při náběžné hraně digitálního vstupu [inc]
NExt_Position	R	1848	hodnota polohy externího snímače polohy při sestupné hraně digitálního vstupu [inc]
Write_Position	R	1856	zachycená požadovaná poloha servopohonu [inc]
DData	R	1864	zachycená data ze sdílené paměti TGM_Data z registru Offset; data jsou typu double (8 bytů)
Zero_Position	R	1872	skutečná poloha servopohonu v momentě příchodu nulového pulsu externího inkrementálního snímače (Capture.Type_ExtCapture – bit 4) [inc]

Registry COMMAND

název	přístup	offset	popis
Control	RW	1920	zapsáním čísla funkce do tohoto registru se daná funkce vykoná (viz kap. 5)
Par [0–14]	W	1928	15 proměnných typu double pro parametry předávané vykonávané funkci; konkrétní význam parametrů závisí na následně volané funkci vložením nenulové hodnoty do registru Command.Control)