



CNC Technologie a obráběcí stroje

GVE66 – 4 osý kontrolér



1 Specifikace:

- HW interpolační jednotka s výkonem 75 000 pulzů/s ve 4-osém pohybu. s integrovanými drivery 2f krokových motorů
- Vnitřní buffer pro 420 vektorů, max délka vektoru +- 2147483647 kroků
- Připojení k PC přes RS232 (USB přes opticky oddělený převodník)
- 8 vstupů (2 pro senzor měření nástroje, 4 pro referenční spínače, 2 pro START/STOP box)
- 2 relé výstupy s použitím pro spínání (max 48VDC, 3A) např. odsávání, chlazení atd.
- 4 výstupy pro 2f krokové motory (max 2A, 32V) s tepelnou ochranou proti přehřátí
- LED signalizace stavu výstupů
- Napájení 24 – 32 VDC
- Odběr 4A. (max) při 24V.

2 Aplikace:

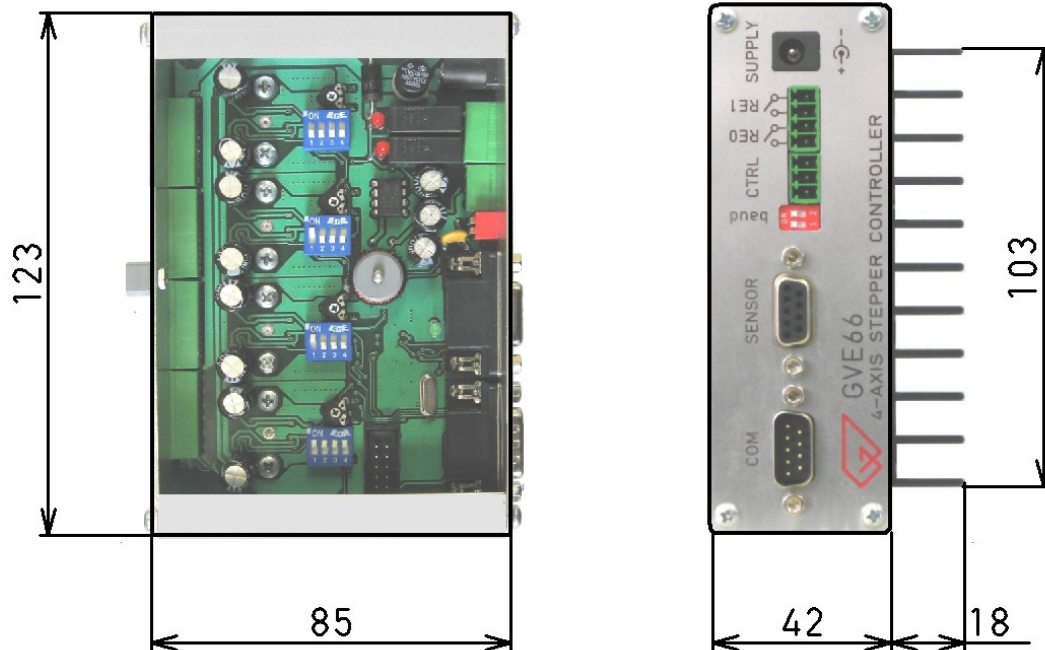
Řízení frézek, gravírek, vrtaček, polohovacích stolů, robotických manipulátorů atd.

3 Součást dodávky:

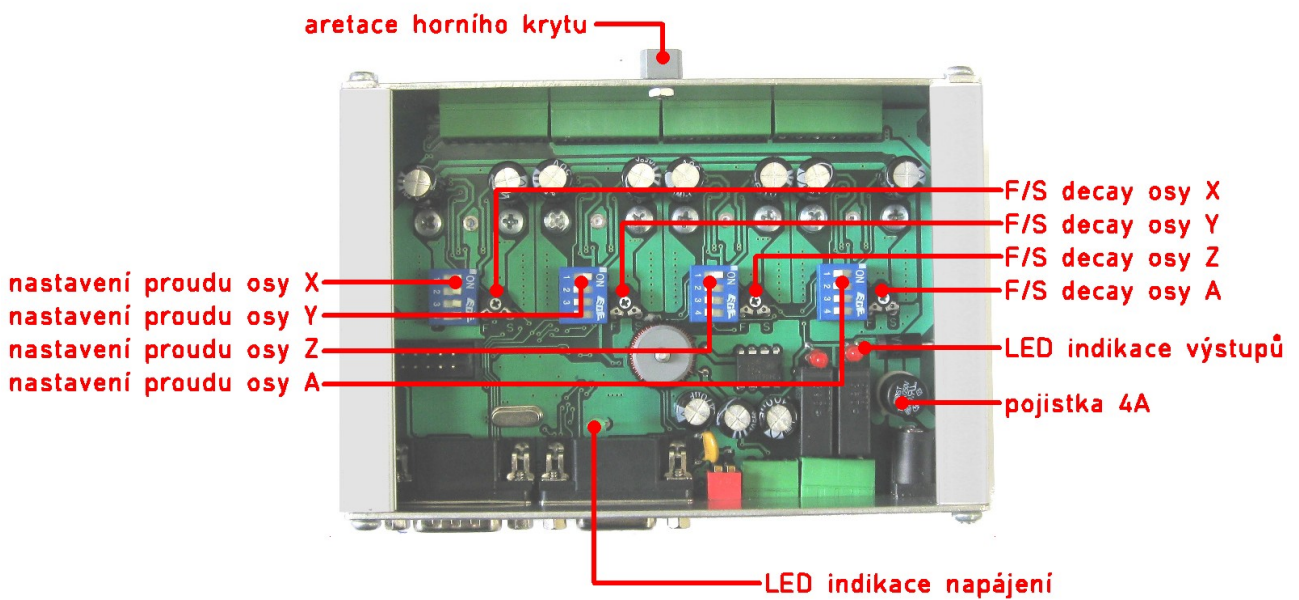
Jednotka GVE66, instalační CD, kabel k připojení k PC, protikusy konektorů, zkušební verze sw ARMOTE pro řízení 4-osé frézky.

4 Rozměry:

123 x 85 x 60mm



5 Přehled



6 Konektory:

7 Popis konektorů:

Axis X	výstup pro motor osy X	SUPPLY	napájení
Axis Y	výstup pro motor osy Y	SENSOR	Senzor nástroje
Axis Z	výstup pro motor osy Z	COM	RS232 připojení k PC
Axis A	výstup pro motor osy A	RE0	Kontakty relé
CTRL	Vstupy	RE1	Kontakty relé

8 Popis vývodů:

konektor	vývod	popis
Axis X	GND	Zem pro referenční spínač
	REF	Připojení ref. spínače
	+A	Fáze A krokového motoru
	-A	Fáze A krokového motoru
	+B	Fáze B krokového motoru
	-B	Fáze B krokového motoru
Axis Y	GND	Zem pro referenční spínač
	REF	Připojení ref. spínače
	+A	Fáze A krokového motoru
	-A	Fáze A krokového motoru
	+B	Fáze B krokového motoru
	-B	Fáze B krokového motoru
Axis Z	GND	Zem pro referenční spínač
	REF	Připojení ref. spínače
	+A	Fáze A krokového motoru
	-A	Fáze A krokového motoru
	+B	Fáze B krokového motoru
	-B	Fáze B krokového motoru
Axis A	GND	Zem pro referenční spínač
	REF	Připojení ref. spínače
	+A	Fáze A krokového motoru

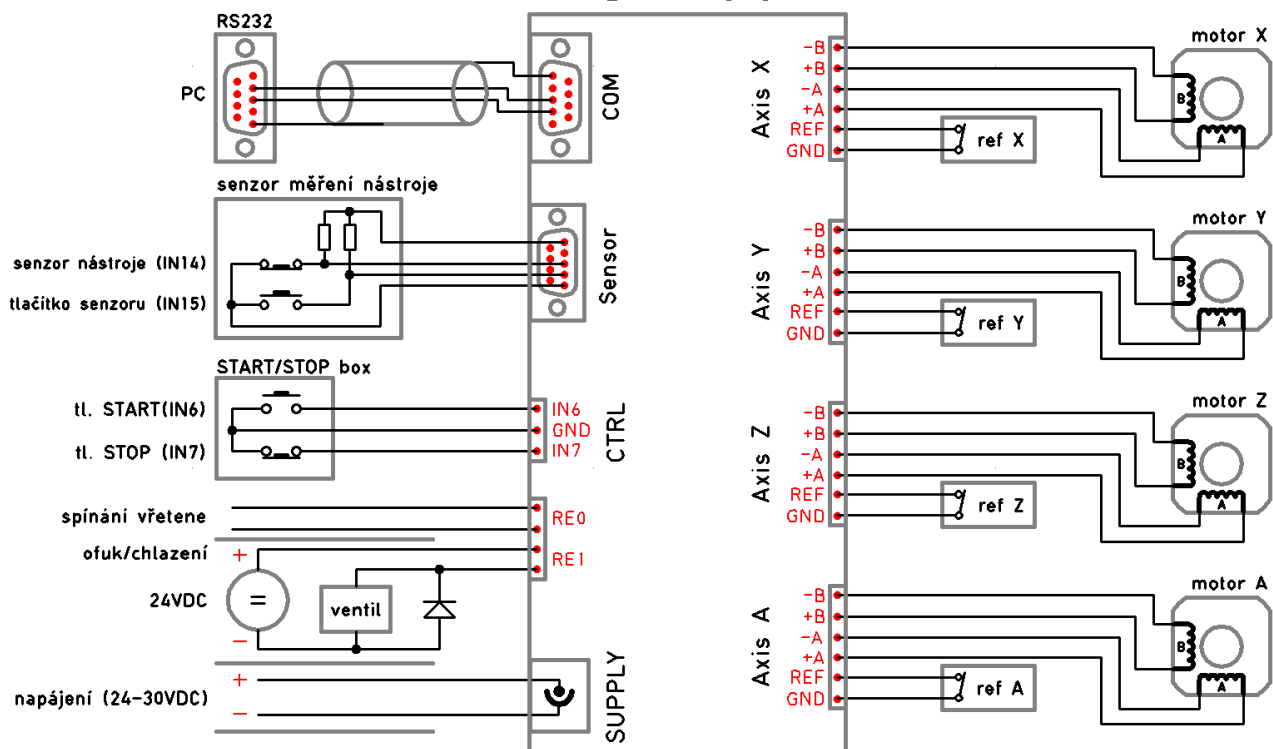
	-A	Fáze A krokového motoru
	+B	Fáze B krokového motoru
	-B	Fáze B krokového motoru
CTRL	ovládání	Vstupy pro připojení START/STOP boxu
RE0	RE0	Kontakty relé (vřeteno start CW) max 48VDC, 3A
RE1	RE1	Kontakty relé (volitelné) max 48VDC, 3A
SUPPLY	napájení	Napájení 32VDC, 4A max viz. Doporučené zapojení
SENSOR	senzor	Konektor senzoru nástroje
COM	RS232	Konektor sériového rozhraní RS232

- Výstup RE0 je vždy používán pro signál roztočení vřetene .
- Výstup RE1 je volitelně konfigurovatelný

logiku spínání výstupu RE1 při použití ovl. sw ARMOTE lze konfigurovat utilitou GVE64_config.exe (volně ke stažení na www.gravos.cz) viz. kapitola 11 - nastavení funkce výstupů.

9 Příklady doporučeného zapojení:

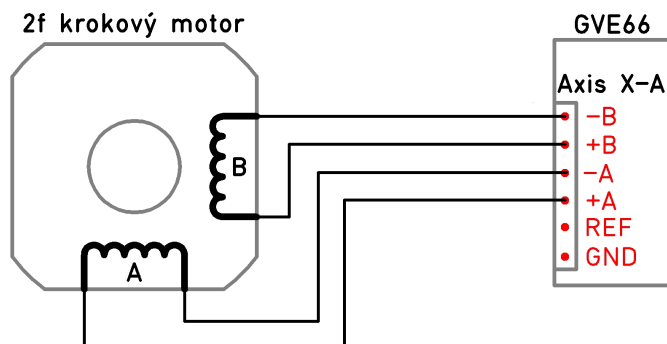
GVE66



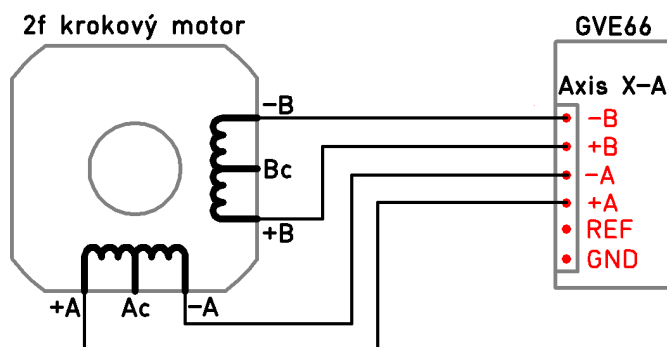
9.1 Připojení krokových motorů, (Axis X – Axis A):

Označení vývodů konkrétního motoru najdete v jeho datasheetu

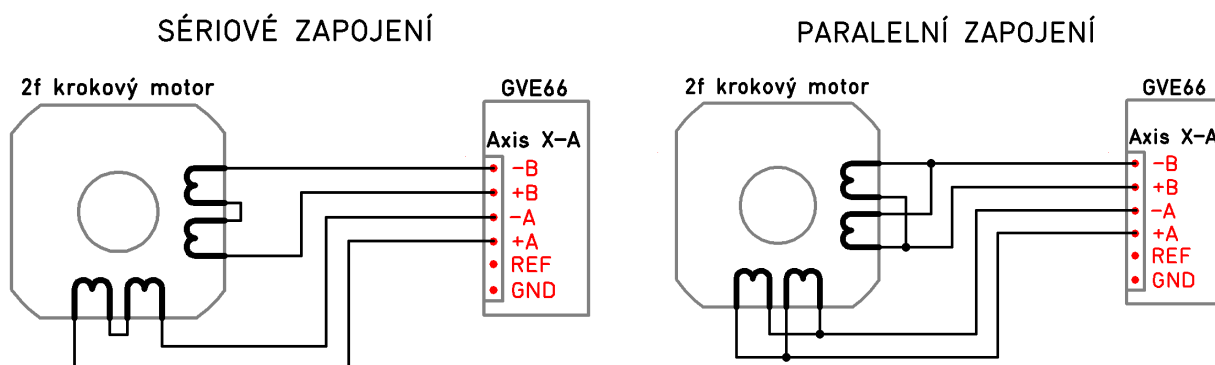
9.1.1 Připojení 2f krokového motoru se 4 vývody



9.1.2 Připojení 2f krokového motoru se 6 vývody



9.1.3 Připojení 2f krokového motoru s 8 vývody

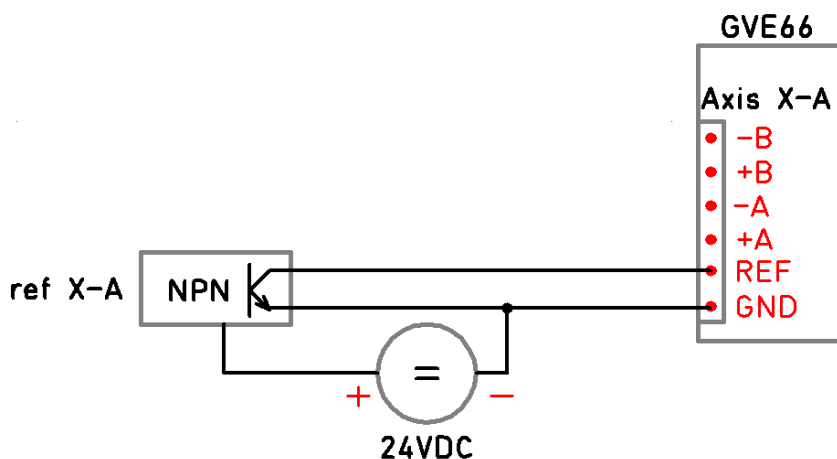


9.2 Zapojení referenčních spínačů (AXIS X – AXIS A, GND/REF):

9.2.1 Připojení indukčních snímačů

Ref. Spínače jsou rozpínací, aby při poškození kabelu došlo k zastavení stroje.

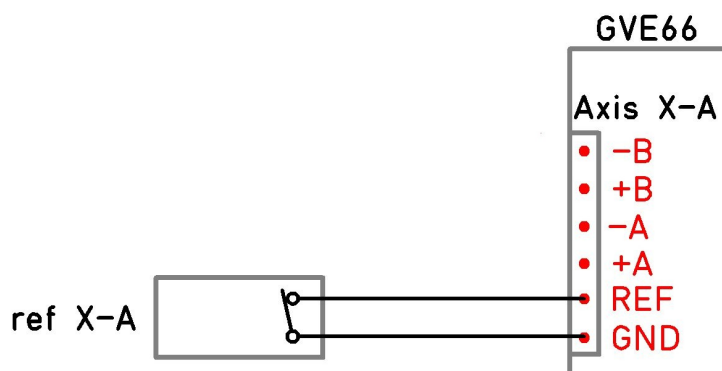
Pro připojení indukčních snímačů je potřeba pro ně použít externí napájení, které může být společné i pro napájení jednotky GVE66



9.2.2 Připojení mechanických spínačů

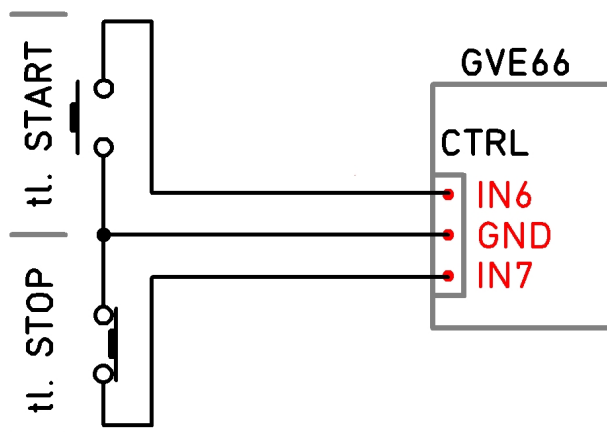
(pro referenční spínače v systémech GRAVOS-ARMOTE)

Ref. Spínače jsou rozpínací, aby při poškození kabelu došlo k zastavení stroje.



9.3 Připojení tlačítek START s STOP (CTRL)

Tl. START je spínací a tl. STOP rozpínací, aby při poškození kabelu došlo zastavení stroje. V systému jsou to vstupy 6 a 7



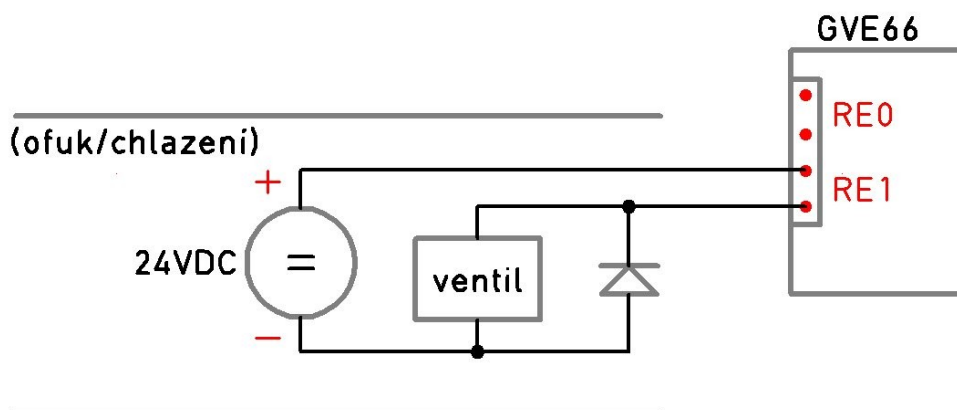
9.4 Zapojení výstupů (RE0 – RE1):

Při použití ovládacího sw Armote je výstup RE0 je použit pro spínání vřetene, funkci výstupu RE1 lze konfigurovat utilitou GVE66 Config (spínání ofuku, chlazení, laseru, plasmového hořáku, otevření krytu stroje nebo brzda osy Z). Kontakty relé max 48VDC, 3A

Konfigurační utilitu najdete na CD dodávané s jednotkou nebo ji lze stáhnout na <http://WWW.GRAVOS.CZ>

9.4.1 Připojení elmag. ventilu

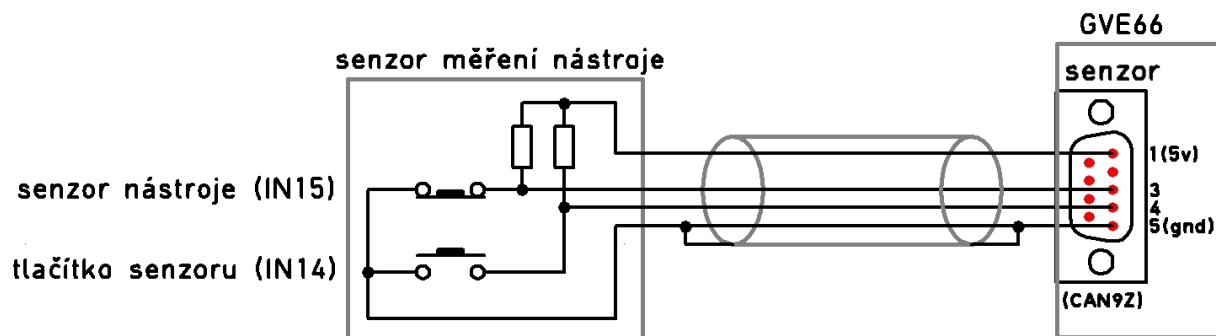
(pro spínání příslušenství v systémech GRAVOS-ARMOTE)



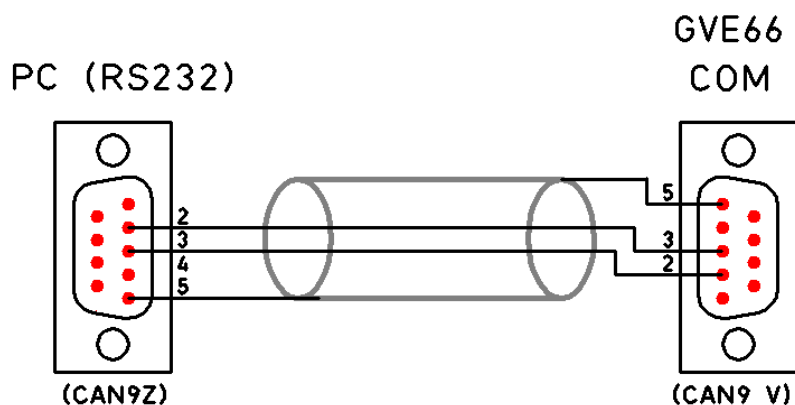
9.5 Připojení senzoru měření nástroje (SENSOR)

Pro senzor je v systému vyhrazen vstup 14 pro tlačítko senzoru a vstup 15 pro měřicí hříbek senzoru.

Pro měřicí hříbek senzoru je třeba použít rozpínací kontakt, aby v případě poškození kabelu nedošlo ke zničení nástroje nebo senzoru.



9.6 Kabel k připojení GVE66 a dalších zařízení k PC (COM)



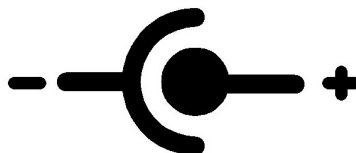
9.7 Napájení, (SUPPLY)

Napájecí napětí

24 – 32 VDC

Doporučený zdroj

ST-C-090-24000375BT (24V/3,75A)



10 Přepínače

10.1 Přepínač Axis Current (nastavení proudu motoru)

Přepínačem se nastavuje proud motoru pro jednotlivé osy, případně je možné osu vypnout úplně (menší příkon a teplota chladiče)

Sw2	Sw3	Sw4	Proud osy
OFF	OFF	OFF	2,06
OFF	OFF	ON	1,85
OFF	ON	OFF	1,65
OFF	ON	ON	1,44
ON	OFF	OFF	1,23
ON	OFF	ON	1,03
ON	ON	OFF	0,81
ON	ON	ON	0,61
Sw1	Stav osy		
ON	Osa zapnuta		
OFF	Osa vypnuta		

10.2 Přepínač COM SPEED

Rychlost komunikace nastavujte před připojením napájení!

Všechna zařízení musí mít nastavenou stejnou komunikační rychlost

sw1	sw2	rychlost
OFF	OFF	19200 Bd
OFF	ON	38400 Bd
ON	OFF	57600 Bd
ON	ON	115200 Bd

10.3 F/S decay (Fast/Slow)

Je určený pro přizpůsobení jednotky k motoru

Trocha teorie:

Výstupní I/O neustále hlídá proud motorem. Pokud je méně, tak sepne můstek tak, aby se proud zvětšoval (nabíjí se).

Pokud dojde k překročení proudu, tak je možné se zachovat 2 způsoby. Buďto se proud nechá cirkulovat přes výstupní tranzistory, pak jeho hodnota klesá relativně pomalu (Slow). Nebo se můstek vypne, a pak se vrací proud zpět přes diody zpátky do zdroje (kondenzátoru). Hodnota proudu klesá rychle (Fast). Trimr má na jedné straně plně Slow, na druhé straně plně Fast.

V mezipoloze je de facto mixed decay, kde se doba vypnutí dělí mezi stavy Slow a Fast podle natočení trimru.

10.3.1 Jak přizpůsobit motoru s osciloskopem:

Nastavit stav Slow. Pustit motor na maximální žádanou rychlost.

Točit trimrem směrem k Fast, až je videt pěkná sinusovka.

Dále netočít, lepší to nebude, jen se zvýší tepelné ztráty.

10.3.2 Jak přizpůsobit motoru bez osciloskopu:

Nastavit stav Slow. Pustit motor na maximální žádanou rychlost.

Poslouchat motor, a pomalu točit směrem k Fast, jakmile se zvuk stane pěkný, čistý, tak dál netočít.

Dá se tak (až překvapivě) přesně nastavit.

11 Nastavení funkce výstupů

Funkce výstupů pro ovládací sw ARMOTE lze konfigurovat pomocí příkazu write zapsáním hodnot na příslušné adrese výstupu, nebo použít utilitu GVE64 config z instalačního CD (utilitu lze stáhnout i na www.gravos.cz v části ke stažení.)

11.1 Adresy EEPROM pro výstupy

adresa	výstup
0x01	RE1
0xFD	Čas po zapnutí pro odbrždění brzdy v 0,1s (max 5s)
0x0C	Sdružení osy A k jiné ose

11.2 Hodnoty nastavení pro RE1

hodnota	funkce
0x00	nepoužito
0x01	ovládání laseru
0x02	chlazení nástroje
0x03	ofuk nástroje
0x04	zámek krytu stroje
0x05	uvolnění nástroje
0x06	otevření krytu nástrojů
0x07	rezerva
0x08	brzda
0x09 - 0xFF	nepoužito

(pokud je výstup nastaven jako brzda, tak je automaticky sepnut po připojení napájení po uplynutí doby nastavené na adrese FD, brzda se používá u strojů s těžším vřeteníkem, kde by po vypnutí stroje došlo ke sjetí osy dolů)

11.3 Hodnoty nastavení pro sdružení osy A s jinou osou

hodnota	připojení
0x01	Osa A je sdružena s osou X
0x02	Osa A je sdružena s osou Y
0x03	Osa A je sdružena s osou Z
0xFF	Osa A je samostatná

Sdružení osy A se hodí zejména pro stroje, které mají na jedné ose 2 motory. Pokud je osa A sdružena s další osou, motor vykonává stejné pohyby jako osa, se kterou je sdružena. Pokud je osa A přidružená k jiné ose, je nefunkční příkaz Switch pro osu A.

12 GVE66 – popis vnitřních instrukcí

(pro tvorbu vlastních uživatelských aplikací)

12.1 CPU

procesor ARM7 32bit

12.2 Program

IP66v7/75 17.9.2012

12.3 Sériový přenos

sériový přenos 8 bitů, 1 stop bit, bez parity
přenosová rychlost BaudRate a Adresa jednotky jsou volitelné přepínači (adresy jsou pevné)
Stav přepínačů je vyhodnocen jen jednou po zapnutí napájení.
Jednotka má pro komunikaci konektor CN1

12.3.1 Komunikace

je čistě simplexní, t.j.: nadřazený počítač pošle povel a čeká na odpověď. Až mu dorazí odpověď, tak si ji analyzuje a pošle další, atd...
Nelze posílat příkazy bez čekání na odpověď. Karta odpoví vždy co nejdříve, s výjimkou příkazu Halt, kdy odpoví až po zabrzdění.

12.3.2 Zabezpečení přenosu pomocí Checksumu:

Přenos po sériové lince je vhodné zabezpečit, aby v případě nějakého rušení jednotka nebo nadřazený počítač poznali, že se případně přenos příkazu nebo odpovědi nepovedl. Například pokud by z příkazu !0L1000,100 vypadla nějaká nula, pojedete se úplně jinam, což by mohlo mít velmi nepříjemné důsledky. Pokud je ale aktivován systém kontrolních součtů, a nějaké číslo by třeba vypadlo, tak kontrolní součet nebude souhlasit a jednotka příkaz neprovede a nahlasí chybu.

Po zapnutí napájení/resetu jsou kontrolní součty vypnuté.

Zapne se příkazem: !0%+ odpověď je už se součtem: 0,5C

Vypne se příkazem: !0%-,CF odpověď je už bez součtu: 0

Součet se počítá tak, že se za příkaz dá místo Enteru čárka a sečtou se všechny Ascii hodnoty všech znaků a modulo 256 přidá za čárku součet v hexadecimální podobě, doplní Enterem a odešle.

např: !0A100, = 0x21 + 0x30 + 0x41 + 0x31 + 0x30 + 0x30 + 0x2C = 0x14F,
doplníme 4F, výsledek bude !0A100,4F

Ascii kódy lze zjistit např. přímo z příslušenství Windows: Charmap.exe
Vypadá to složitě, ale není. Pro člověka takové výpočty moc nejsou, ale pro SW to představuje pár řádků.

12.3.3 Paketizace příkazů:

Pokud používáte pro přenos dat mezi jednotkou a počítačem převodník USB, je vhodné paketizaci použít. USB porty jsou stavěné trochu jinak než COM porty, které již bohužel pomalu z počítačů mizí. USB porty jsou stavěné sice pro rychlý přenos velkého objemu dat, ale dávkově. Jsou zde časová okna, ve kterých se data přenesou (pokud je zrovna co).

Takže od zadání příkazu do jeho skutečného odeslání vznikne časová "díra" - je to označované jako Latence, bývá od 1 do 16ms. A při příjmu odpovědi to samé. Takže je možné, že máme rychlé porty USB 2.0 (až 480Mb/s), rychlý počítač, max. komunikační rychlost a přesto se to loudá.

V případě přenosu malého množství dat třeba pro manipulátory to je většinou nepodstatné, ale pokud budeme chtít např. gravírovat složité křivky, tak jednotka zpracuje příkazy mnohem rychleji, než stačíme dodávat data.

Proto je vhodné sdružit více příkazů do jednoho paketu (stringu) a ten poslat najednou. Jednotka odpoví po přijetí konce paketu.

Např. včetně kontrolních součtů to může vypadat třeba takto:

!0*S,FA	-start paketu
!0C697,30,0,51	-vektor
!0C692,92,0,54	-vektor
!0C682,151,0,7F	-vektor
!0C665,209,0,84	-vektor
!0C645,268,0,87	-vektor
!0C619,322,0,7F	-vektor
!0C589,375,0,8D	-vektor
!0C553,425,0,80	-vektor
!0C515,471,0,7F	-vektor
!0C471,515,0,7F	-vektor
!0*E10,4D	-konec (bylo 10 příkazů)

a odpověď jednotky: 0,10,E9 - bez chyby, bylo 10 příkazů (a kontrolní součet)

Do paketu je možné a účelné dávat jen příkazy typu C (cont.line) a B (brake), ze kterých se vytváří mapa brždění v koncových bodech vektoru.

Minimální délka paketu je alespoň 5 pohybových vektorů typu C.

Pro použití se skutečným sériovým portem paketizaci nedoporučujeme, programová obsluha komunikace je zbytečně složitá (i když to funguje také).

Virtualní sériové porty nedoporučujeme používat vůbec (nespolehlivé, pomalé).

Vůbec hlavně začátečníkům doporučujeme k fríze/manipulátoru počítač se skutečným(i) sériovým portem, třeba i starší. Dnes se lidí houfně starších počítačů zbavují, kvůli výkonu, který pozřou nenažrané programy a operační systém. U stroje mohou ještě dobře posloužit. Jen to chce většinou vyčistit, někdy nový ventilátor a jede se dál...

V případě použití USB převodníku důrazně doporučujeme, aby byl galvanicky oddělený (na strane RS232, USB oddělit nejde).

USB porty jsou totiž často také citlivé na statickou elektřinu. Stane se, že člověk vstane ze židle, dotkne se kovové části stroje, a spojení po USB spadne.

Gravos takový převodník dodává, jsou na něm rychlé optočleny a pod nimi 4mm izolační mezera (případně si můžete podobný zhotovit). Je léty ověřený a spolehlivý.

12.4 Reset

jednotka je po připojení napájení nebo po příkazu J=JUMP na reset cca po 2s ve stavu:

ST0..ST6 = 00	- veškerá přerušení neaktivní
A20	- zrychlení 20000 kr/s ²
V1000	- Rychlost pohybu 1000 kr/s
VL0,0,0,0	- Max. rychlost všech os 0 kr/s
\$512	- short vektory jsou menší než 512 kroků
B75000	- bez omezení brzděné rychlosti mezi cont.vektory.
N	- čítač polohy vynulován
O0,FF	- výstupy vypnuté
MD	- motory na 1/2 proudu

12.5 Příkazy

! Adresa Příkaz [parametry] Enter
mezi jednotlivými parametry je čárka

Adr.0 = interpolační jednotka

12.5.1 Identifikace jednotky

? - VERSION dotaz na verzi programu

Q - QUESTION dotaz na ID procesoru, vrací řetězec 8 čísel
jejich význam: ddmmrrpp
dd = den pálení procesoru
mm = měsíc pálení procesoru
rr = rok pálení procesoru
pp = kolikátý procesor toho dne
např.: 25020304 znamená 25.2.2003 čtvrtý kus toho dne
toto číslo je jedinečné - neexistují 2 procesory se stejným číslem

12.5.2 Zadávání pohybových vektorů

L_{x,y,z,a} - LINE vektor (přímka)

x = počet pulsů v ose X v rozsahu -2147483647 až 2147483647

y = počet pulsů v ose Y v rozsahu -2147483647 až 2147483647

z = počet pulsů v ose Z v rozsahu -2147483647 až 2147483647

a = počet pulsů v ose A v rozsahu -2147483647 až 2147483647

např.: !0L1000,1000,20,500

Zvláštní možnost se nabízí při použití vektoru L0,0,0 , který program považuje za normální vektor, i když nemá žádný pohybový efekt. Tento vektor je výhodné zařadit na konec fronty vektorů, kde může indikovat konec zpracování předchozí fronty.

Dokud není přijat, karta hlasí chybu 1 a tudíž fronta před ním není hotová. Jakmile ho karta přijme, ohlásí 0 (OK) a tudíž je fronta před tímto vektorem hotová. Při tomto způsobu je neustále k dispozici bit INTA.

C_{x,y,z,a} - CONT.LINE pokračující vektor (přímka)

x = počet pulsů v ose X v rozsahu -2147483647 až 2147483647

y = počet pulsů v ose Y v rozsahu -2147483647 až 2147483647

z = počet pulsů v ose Z v rozsahu -2147483647 až 2147483647

a = počet pulsů v ose A v rozsahu -2147483647 až 2147483647

např.: !0C1000,1000,20,500

Určit jestli je vektor pokračující je výpočetně dost složité a tudíž časově náročné a proto to musí určit nadřazený počítač. U pokračujícího vektoru se nesmí příliš změnit úhel, jinak by nebylo fyzikálně možné vektor správně interpretovat.

Jednotka má buffer na 420 CONT. vektorů.
(1 je vykonáván, a další mohou být ve frontě)

Frontou pokračujících vektorů lze velmi zrychlit práci, protože jednotlivé vektory nemusí neustále zrychlovat z nulové rychlosti a následně opět do nulové rychlosti zpomalovat. Také se tím omezí vibrace stroje, a následně selepší kvalita obráběného povrchu.

Vektory (L i C) se zadávají v relativních souřadnicích od posledního bodu. (Absolutní souřadnice by představovaly příliš dlouhé řetězce znaků a proto by klesala skutečná rychlost přenosu informací po seriové lince)

- Tn** - **TIME** prodleva mezi nenavazujícími vektory v milisekundách
 n= 10 az 100 milisekund
 doporučená hodnota je podle hmotnosti stroje asi 10 az 100 ms
 Mezi CONT.vektory tato prodleva není.
 např.: !0T10 - prodleva 10ms
 Příkaz je modální, platí až do zadání jiné hodnoty.
- Sn** - **SHORT** je hraniční hodnota pro rozlišení krátkého a dlouhého vektoru. Chovají se trochu odlišně.
 n = 1..4294967295
 Dlouhý vektor se snaží dostat pomocí zrychlení A až k maximální rychlosti V.
 Krátký vektor se snaží dostat pomocí zrychlení A jen k brzdě rychlosti B na svém konci.
 Tímto se stává fronta krátkých vektorů plynulejší, a průjezd libovolnou spojitou křivkou, která je rozumně rozsekána na úsečky je plynulý také.

12.5.3 Rychlosti

- VL_{x,y,z,a}** – **VELOCITY LIMIT** rychlostní limit
 omezení max. rychlosti, které můžou jednotlivé osy dosáhnout
 X = max. rychlost v tis. pulsů/s osy X v rozsahu 10 - 75000
 Y = max. rychlost v tis. pulsů/s osy Y v rozsahu 10 - 75000
 Z = max. rychlost v tis. pulsů/s osy Z v rozsahu 10 - 75000
 A = max. rychlost v tis. pulsů/s osy A v rozsahu 10 – 75000

např. !0VL25000,25000,30000,10000

- An** - **AKCELERATION** zrychlení a zpomalení následujících vektorů
 n= tisíců pulsů/s²
 např.: !0A50 - akcelerace 50000 pulsů/s²

Příkaz je modální, platí až do zadání jiné hodnoty.

- Vn** - **VELOCITY** rychlost následujících vektorů
 n = 10 az 75000 pulsů/s
 např.: !0V10000 - rychlost 10000 pulsů/s

Příkaz je modální, platí až do zadání jiné hodnoty.

- Bn** - **BRAKE** rychlost, na kterou má vektor dobrzit, pokud za ním ve frontě je další Cont.vektor. Pokud za ním není další, tak stejně dobrzdí do nuly.

n = 10 az 75000 pulsů/s
 např.: !B1000

To má význam hlavně u navazujících vektorů, kdy je nutné před zatačkou přibrzdit, ale ne úplně.
 Příkaz je modální, platí až do zadání jiné hodnoty.

12.5.4 Korekce rychlostí

- VK** - **VELOCITY CORECTION** korekce rychlosti podle směru rychlosti jsou určeny podle $d = \sqrt{dx*dx + dy*dy + dz*dz} / d$
Použití hlavně pro pohyby za předpokladu že všechny osy jsou lineární
- VN** - **NO VELOCITY CORECTION** rychlosti bez korekcí podle směru použití pro pohyby s rotační osou A, kde rychlost pro každý vektor musí určit nadřazené PC s ohledem na směr pohybů jednotlivých os a vzdálenost od osy rotace nebo při velkém rozdílu převodu os.

12.5.5 Změny rychlosti během pohybu

- XMn** - **EXCHANGE MULTIPLIER** násobitel rychlosti parametr n je násobitel rychlosti v procentech.
např. !0XM100 nastaví rychlost na poslední zadanou rychlost příkazem V
např. !0XM200 nastaví rychlost na dvojnásobek (200%) rychlosti nastavené příkazem V
Pokud je nastaven rychlostní limit příkazem VL, rychlosti jednotlivých os můžou dosáhnout max. rychlosti nastavené tímto limitem.

12.5.6 Poloha

- P** - **POSITION** dotaz na polohu X,Y
Odpovědí je okamžitá absolutní poloha x,y , takže během chodu nějakého vektoru se neustále mění.
Po zastavení je hodnota stabilní.
Hodí se pro kreslení okamžité pozice nástroje v rovině XY.
- PF** - **POSITION** dotaz na polohu X,Y,Z
Odpovědí je okamžitá absolutní poloha x,y,z , takže během chodu nějakého vektoru se neustále mění.
Po zastavení je hodnota stabilní.
- P4** - **POSITION** dotaz na polohu X,Y,Z,A
Odpovědí je okamžitá absolutní poloha x,y,z,a , takže během chodu nějakého vektoru se neustále mění.
Po zastavení je hodnota stabilní.
- N** - **NULL** Vynuluje všechny 4 osy čítače pozice nelze použít za chodu vektoru (pri RUN=1)
- Nan** - **NULL AXIS** Vynuluje nebo nastaví čítač pozice vybrané osy parametr a je osa, které se má čítač nastavit (X,Y,Z,A)
parametr n je v rozsahu -2147483647 až 2147483647
např. !0NX0 – vynuluje čítač pozice osy X
např. !0NX100 – nastaví čítač pozice na hodnotu 100

12.5.7 Nalezení spínače osy - referenční pohyb

Wn - **Switch** nalezení spínače osy (referenční pohyb)
Jednotka odpoví až po ukončení reference
Příkaz je ve formátu Wn1,n2,n3,n4,n5

parametry:

n1 = Osa (x,y,z,a)
n2 = Max. délka a směr kterou osa jede ke spínači [pulsy]
n3 = Rychlost ke spínači [pulsů/s]
n4 = Max. Délka kterou osa jede od spínače [pulsy]
n5 = Rychlost od spínače [pulsů/s]

např.reference osy Y: !0WY-20000,1000,2000,500

12.5.8 Obsluha fronty a zpracování vektorů

K - **KEEP** {obdoba PUSH}
Zachytí v operační paměti stav fronty vektorů po přerušení a ST4,ST5 a ST6.
Potom smaže ST4=00,ST5=00 a ST6=00.
Vyhradí v operační paměti místo pro 1 vektor, takže je možno opět zadávat vektory, ale již jen typu L (C ne).
KEEP lze použít bez odpovídajícího RESTORE jen jednou.
Nelze použít za chodu.

R - **RESTORE** {obdoba POP}
inverzní rutina ke KEEP
Obnoví stav operační paměti s frontou vektorů a ST4,ST5,ST6 tak, jak byla uložena příkazem KEEP. Nelze použít za chodu.

Tato dvojice inverzních rutin umožňuje transparentci vektorů po přerušení.

Např.: Obsluha zastaví obrábění tlačítkem STOP nebo příkazem HALT apod. Potom je obrábění zastaveno, ale v jednotce je ještě zbytek vektorů ve frontě. Tento zbytek lze dodělat příkazem GO nebo smazat příkazem DELETE, ale někdy je potřeba zvednout nástroj a nezničit zbytek fronty. Potom je potřeba zachytit stav paměti, vymazat ji, udělat zadané vektory (např.vzhůru a zpět dolů), a potom obnovit paměť a pokračovat v obrábění.

např.:

!0H	zastaví vektor
!0P	zjistí souřadnice zastavení (kde to jsme?)
!0SR4	zjištění stavu systému přerušení (a proc se to stalo?)
!0SR5	zjištění stavu systému přerušení
!0SR6	zjištění stavu systému
!0SW0,1A	nová maska přerušení
!0SW1,3F	nová maska přerušení
!0K	zachytí stav operační paměti
!0L0,0,-1000	zvedne nástroj (pro jeho výměnu)

....tady se čeká na reakci uživatele....

....a když se rozhodne pokračovat třeba změněnou rychlostí....

!0L0,0,1000	spustí nástroj
!0R	obnoví operační paměť
!0V500	nastaví novou rychlost budoucích vektorů
!0XA	nastaví tuto rychlost i pro zbytek vektorů ve frontě
!0SW0,A2	normální maska přerušení
!0SW1,78	normální maska přerušení
!0G	pokračování už jinou rychlostí

- H** - **HALT** zastavení zpracovávaného vektoru, pokud nějaký běží
Nastaví bit INTA=1. Bit RUN signalizuje, zda byl příkaz
HALT použit za chodu (1) nebo ne (0).
Odpoví až po zastavení. To může trvat i dost dlouho - neztracovat zatím komunikaci.
Příkazem HALT se zároveň nastaví bit INTRCOM pro účel identifikace přerušení.
- D** - **DELETE** smaže veškeré vektory ve frontě
Hodí se pro smazání zbytku fronty po přerušení.
Smaže všechny příznaky přerušení
(ST4=00(hex), ST5=00(hex), ST6=00(hex))
Čítač pozice neovlivní.
Nelze použít za chodu.
- G** - **GO** nastartuje dokončení zastaveného vektoru a zbytku fronty
jen pokud INTA=1, jinak bez efektu.
Smaže bit INTA=0,INTRCOM=0,ST4=00,ST5=00.

12.5.9 Obsluha paměti EEPROM

- ERn** - **READ** přečte byte z EEPROM na adrese n a pošle jej po sériové lince
n = 00..FF(hex)
- EWn,x** - **WRITE** zapíše byte x EEPROM na adresu n
n = 00..FF(hex) x = 00..FF(hex)

12.5.10 Ovládání relé

- O0,n** - **OUTPUT** zapíše byte x v hexadecimálním tvaru na výstupní port 0
jsou aktivní v log.0:
bit 0 = CNOUT - RE0 (v kontrolérech Gravos vřeten)
bit 1 = CNOUT - OUT3 (v kontrolérech Gravos brzda)
bit 2 = CNOUT - OUT1 (v kontrolérech Gravos chlazení)
bit 3 = CNOUT - OUT2 (v kontrolérech Gravos ofuk)
výstupy jsou aktivní v log.0, po zapnutí jsou neaktivní log.1

např. spuštění chlazení (OUT1): !000,FB
vypnutí : !000,FF

12.5.11 Čtení stavu vstupů a obsluha přerušení

I1 - **INPUT** přečte vstupní port 1
odpovědí je stav portu v hexadecimálním tvaru
bit 0 = Intr0 (RefX)
bit 1 = Intr1 (RefY)
bit 2 = Intr2 (RefZ)
bit 6 = Intr6 (Ctrl Start)
bit 7 = Intr7 (Ctrl Stop)

I2 - **INPUT** přečte vstupní port 2
odpovědí je stav portu v hexadecimálním tvaru
bit 0 - 5 = nepoužit
bit 6 = CN2 - pin 4 Intr14 (tlačítko sensoru)
bit 7 = CN2 - pin 3 Intr15 (hříbek sensoru)

Intry nedělají nic jiného, než že při své aktivaci přinutí interpolátor zabrzdit (po rampě).

Je zde popsáno jak využívá Intry systém Gravos, to by však nemělo být omezující, lze je použít libovolně jinak. Toto info je jen pro případnou snahu o kompatibilitu.

SRn - **STATUS READ** přečte status n = 0..5
odpovědí je hodnota zadaného status slova

SWn,x - **STATUS WRITE** zapíše do statusu n = 0..5, byte x (v hex.tvaru)
Status slova:
ST0 = povolení uživatel.přerušení INTR0-7 (0 = zakázáno)
ST1 = povolení uživatel.přerušení INTR8-15 (1 = povoleno)
ST2 = polarita uživatel.přerušení INTR0-7 (0 = aktivní v log.0)
ST3 = polarita uživatel.přerušení INTR8-15 (1 = aktivní v log.1)
ST4 = příčina přerušení INTR0-7 (0 = přerušení nebylo)
ST5 = příčina přerušení INTR8-15 (1 = přerušení bylo)

Jednotlivá přerušení korespondují se vstupy. Pomocí přečtení vstupů lze přečíst okamžitý stav. Každé aktivované přerušení zastaví pohyb, a nastaví bit INTA, aby o tom řídicí SW věděl. Libovolný Intr není nutné použít, (lze zamaskovat) a je ho možno použít jako obecný vstupní bit.

SR6 - STATUS READ přečte ST6

význam jednotlivých bitů: (ostatní jsou nepoužité)

STOP = 0 žádost o zastavení

FREE = 1 příznak volného str.času

INTCOM = 4 nastavuje se po přerušení HALTem

RUN = 6 je zpracováván vektor

INTA = 7 akceptováno zastavení

pro uživatele mají význam především bity RUN a INTA

INTA=0 RUN=0 ;nic není spuštěno, klidový stav

INTA=0 RUN=1 ;provozní stav, jsou zpracovávány vektory

INTA=1 RUN=0 ;bylo přerušeno, při brždění vektory doběhly

INTA=1 RUN=1 ;bylo přerušeno, zbytek vektorů je ve frontě

SW6 - STATUS WRITE zapíše do ST6, byte x (v hex.tvaru)

raději nepoužívat, lépe použít instrukce G,D,H apod...

F - FLAG to samé jako SR6, ale je doplněn stav fronty vektorů - bit 5

log.1 = fronta je plná - nelze přijmout vektor

log.0 = do fronty se další vektor vejde

12.5.12 Příkazy pro opravu chyb komunikace

@ - INDEX pošle index posledního příkazu.

Všechny příkazy jsou indexovány modulo 256.

V případě nejistoty zda příkaz do Interpolátoru dorazil, je možné vyžádat tento index a porovnat s vlastním indexováním v programu a tak zjistit, zda ho interpolátor přijal nebo ne. Většina příkazů se dá zopakovat (A,V,PF atd..), ale zadávání polohy ne, to se musí v případě chyby přenosu exaktně dohledat, jinak by se jelo jinam.

> - REPEAT - zopakuje poslední přijatý příkaz a odpověď na něj.

Toto se hodí, pokud dojde k chybě přenosu a nadřiznému počítači přijde místo odpovědi nějaký nesmysl.

12.5.13 Snížení proudu motorů při neaktivitě

Mm - MOTOR MODE (nelze použít za chodu vektoru)

m = U - Power Up, proud motorů podle nastavení DIP přepínači

m = D - Power Down, proud motorů 1/2 z nastavení DIP přepínači

12.5.14 Reset jednotky

J - JUMP na RESET zresetuje včetně vynulování čítače polohy

Obsah

1 SPECIFIKACE:	1
2 APLIKACE:	1
3 SOUČÁST DODÁVKY:	1
4 ROZMĚRY:	2
5 PŘEHLED	2
6 KONEKTORY:	3
7 POPIS KONEKTORŮ:	3
8 POPIS VÝVODŮ:	3
9 PŘÍKLADY DOPORUČENÉHO ZAPOJENÍ:	4
9.1 Připojení krokových motorů, (Axis X – Axis A):	5
9.1.1 Připojení 2f krokového motoru se 4 vývody:	5
9.1.2 Připojení 2f krokového motoru se 6 vývody:	5
9.1.3 Připojení 2f krokového motoru s 8 vývody:	5
9.2 Zapojení referenčních spínačů (AXIS X – AXIS A, GND/REF):	6
9.2.1 Připojení indukčních snímačů:	6
9.2.2 Připojení mechanických spínačů:	6
9.3 Připojení tlačítek START s STOP (CTRL):	7
9.4 Zapojení výstupů (RE0 – RE1):	7
9.4.1 Připojení elmag. ventilu:	7
9.5 Připojení senzoru měření nástroje (SENSOR):	8
9.6 Kabel k připojení GVE66 a dalších zařízení k PC (COM):	8
9.7 Napájení, (SUPPLY):	8
10 PŘEPÍNAČE:	9
10.1 Přepínač Axis Current (nastavení proudu motoru):	9
10.2 Přepínač COM SPEED:	9
10.3 F/S decay (Fast/Slow):	10
10.3.1 Jak přizpůsobit motoru s osciloskopem:	10
10.3.2 Jak přizpůsobit motoru bez osciloskopu:	10
11 NASTAVENÍ FUNKCE VÝSTUPŮ:	10
11.1 Adresy EEPROM pro výstupy:	10
11.2 Hodnoty nastavení pro RE1:	11
11.3 Hodnoty nastavení pro sdružení osy A s jinou osou:	11
12 GVE66 – POPIS VNITŘNÍCH INSTRUKCÍ:	12
12.1 CPU:	12
12.2 Program:	12
12.3 Sériový přenos:	12
12.3.1 Komunikace:	12
12.3.2 Zabezpečení přenosu pomocí Checksumu:	12
12.3.3 Paketizace příkazů:	13
12.4 Reset:	14
12.5 Příkazy:	14
12.5.1 Identifikace jednotky:	14
12.5.2 Zadávání pohybových vektorů:	15
12.5.3 Rychlosti:	16
12.5.4 Korekce rychlostí:	17
12.5.5 Změny rychlosti během pohybu:	17

12.5.6 Poloha.....	17
12.5.7 Nalezení spínače osy - referenční pohyb.....	18
12.5.8 Obsluha fronty a zpracování vektorů	18
12.5.9 Obsluha paměti EEPROM.....	19
12.5.10 Ovládání relé.....	19
12.5.11 Čtení stavu vstupů a obsluha přerušeni	20
12.5.12 Příkazy pro opravu chyb komunikace.....	21
12.5.13 Snížení proudu motorů při neaktivitě.....	21
12.5.14 Reset jednotky.....	21