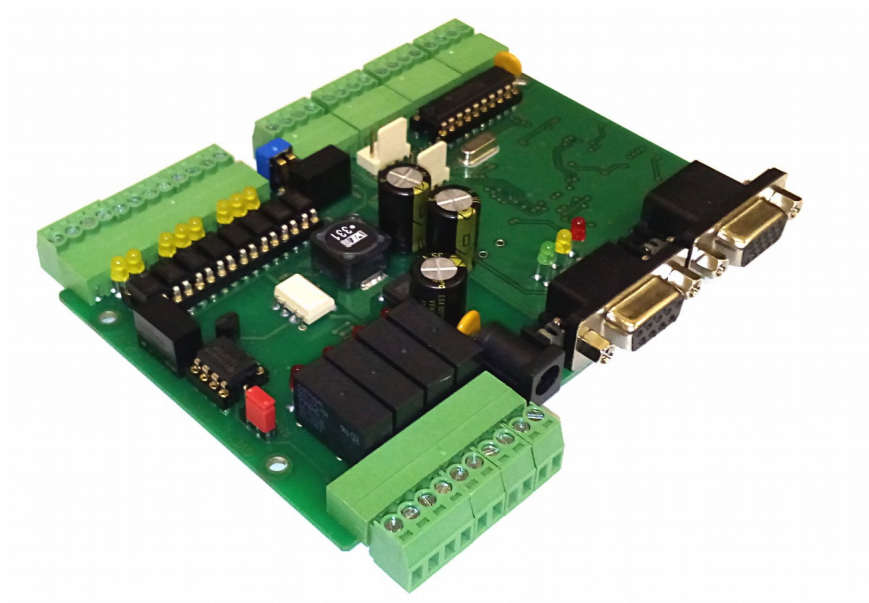


Datasheet

GVE114

HW interpolační jednotka

Pro řízení CNC frézek, gravírek, routerů, vrtaček, polohovacích stolů, robotických manipulátorů atd.



Copyright © Gravos CNC s.r.o. 2018

Obsah	4
1 Aplikace	4
2 Součást dodávky	4
3 Rozměry	5
4 Přehled	5
5 Konektory	6
6 Popis konektorů	6
7 Popis vývodů	7
8 Příklady doporučeného zapojení	9
9.1 Připojení pohonů zařízení, (CNX – CNA)	9
9.1.1 Připojení krokových motorů	9
9.1.2 Časování signálů KROK a SMĚR	9
9.2 Zapojení vstupů (CNIN)	10
9.2.1 Připojení indukčních snímačů	10
9.2.2 Připojení mechanických spínačů	10
9.2.3 Připojení tlačítek START s STOP	11
9.3 Zapojení výstupů (CNOUT)	11
9.3.1 Připojení frekvenčního měniče se signály 0-10V a CW START	11
9.3.2 Připojení frekvenčního měniče pouze se signálem 0-10V	11
9.3.3 Připojení elmag. ventilů	12
9.4 Připojení senzoru měření nástroje (CN2)	12
9.5 Kabel k připojení GVE114 a dalších zařízení k PC (CN1)	13
9.6 Napájení, (CNSUP)	13
10 Přepínače	14
10.1 Přepínač COM SPEED	14
10.2 Přepínač ANALOG/PWM	14
11 Nastavení funkce výstupů	15
11.1 Adresy EEPROM pro výstupy	15
11.2 Hodnoty nastavení pro OUT1 - OUT3	15
11.3 Hodnoty nastavení PWM	15
11.4 Hodnoty nastavení sdružení osy A	16
11.5 Nastavení výstupu signálů STEP/DIR	16
12 Nastavení funkce vstupů	17
12.1 Adresy EEPROM pro vstupy a MPG	17
12.2 Hodnoty nastavení MPG Enable	18
12.3 Nastavení polaroty a povolení přerušování vstupů	18
12.4 Hodnoty nastavení limit	18
12.5 Nastavení počtu kroků na mm	19
12.6 Hodnoty nastavení pohybu pro MPG	19
12.7 Nastavení prodlevy MPG	19
12.8 Nastavení defaultních otáček vřetene	19
12.9 Nastavení velikosti kroku MPG	19
13 GVE114 – popis vnitřních instrukcí	20
13.1 CPU	20
13.2 Program	20
13.3 Sériový přenos	20
13.3.1 Komunikace	20
13.3.2 Zabezpečení přenosu pomocí Checksumu	20
13.4 Paketizace příkazů	21
13.5 Reset	22
13.6 Příkazy	22
13.6.1 Identifikace jednotky	22
13.6.2 Zadávání pohybových vektorů	23

13.6.3	Rychlosti.....	24
13.6.4	Korekce rychlostí.....	25
13.6.5	Změny rychlosti během pohybu.....	25
13.6.6	Poloha.....	25
13.6.7	Nalezení spínače osy - referenční pohyb.....	26
13.6.8	Obsluha ručního ovladače MPG.....	26
13.6.9	Obsluha fronty a zpracování vektorů.....	27
13.6.10	Obsluha paměti EEPROM.....	28
13.6.11	Ovládání relé.....	28
13.6.12	Čtení stavu vstupů a obsluha přerušení.....	29
13.6.13	Příkazy pro opravu chyb komunikace.....	30
13.7	Ovládání vřetene adr.7.....	30
13.8	Odpovědi.....	31

1 Specifikace

- ✓ HW interpolační jednotka s výkonem až 500 000 pulzů/s ve 4-osém pohybu.
 - Další verze 50 000 , 125 000 a 200 000 pulzů/s.
- ✓ Vnitřní buffer pro 2048 vektorů, max délka vektoru +- 2147483647 kroků.
- ✓ Možnost připojení více jednotek na jeden port RS232 (jednotky jsou adresovatelné).
- ✓ Řízení krok/směr pro krokové motory, servo motory nebo lineární motory.
 - plně 4 osá interpolace.
- ✓ Připojení k PC přes RS232. USB přes převodník.
- ✓ Komunikační rychlost standardně přes RS232 115 200 bd.
 - max. 921 000 bd, přes USB 6 000 000 bd (vyžaduje speciální převodník).
- ✓ 8 vstupů (galvanicky oddělené).
- ✓ 2 vstupy pro 1 senzor měření nástroje nebo 1 obrobkovou sondu (neoddělené).
- ✓ 4 relé výstupy s použitím pro spínání (max. 48 VDC, 3 A) např. odsávání, chlazení atd.
- ✓ 1 analogový výstup (0-10V).
 - např. pro řízení frekvenčního měniče, možnost přepnout na PWM. (galvanicky oddělené).
- ✓ LED signalizace stavu vstupů a výstupů.
- ✓ Napájení 9 – 24 VDC.
- ✓ Odběr 300 mA. (max) při 12 V.

2 Aplikace

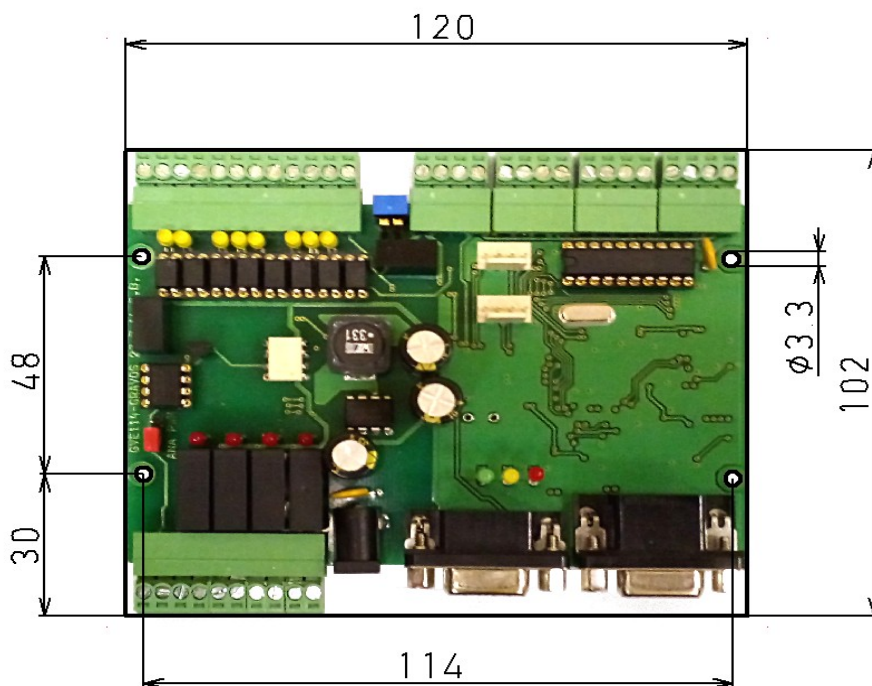
Řízení CNC frézek, gravírek, routerů vrtaček, polohovacích stolů, robotických manipulátorů atd.

3 Součást dodávky

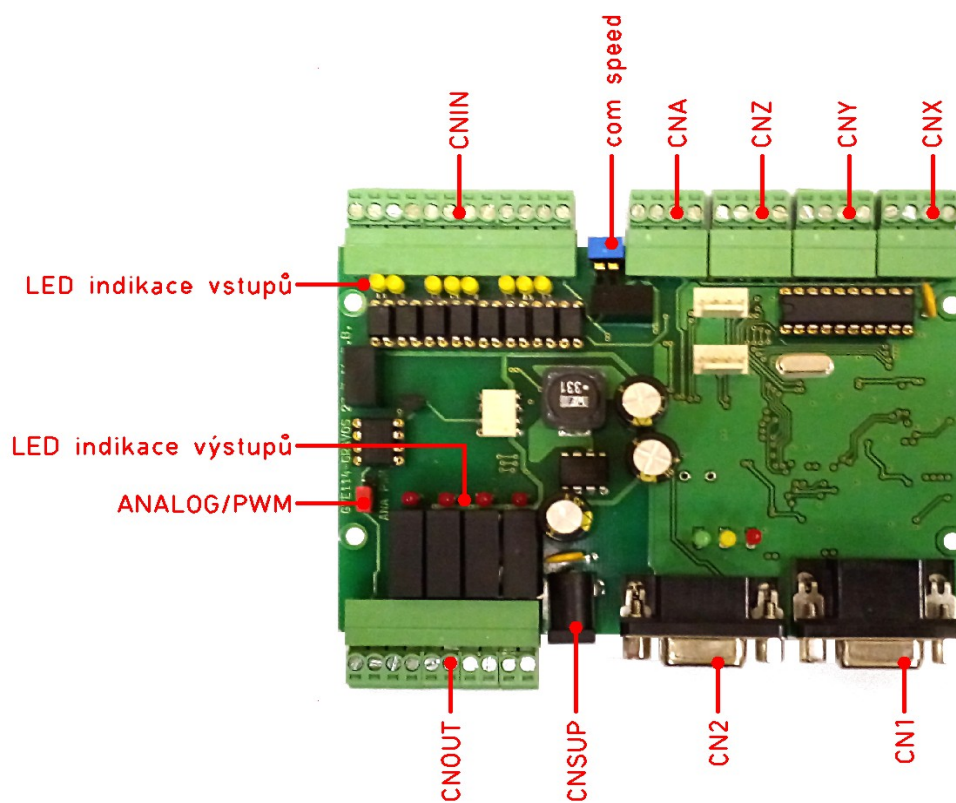
Jednotka GVE114 s protikusy konektorů.

4 Rozměry

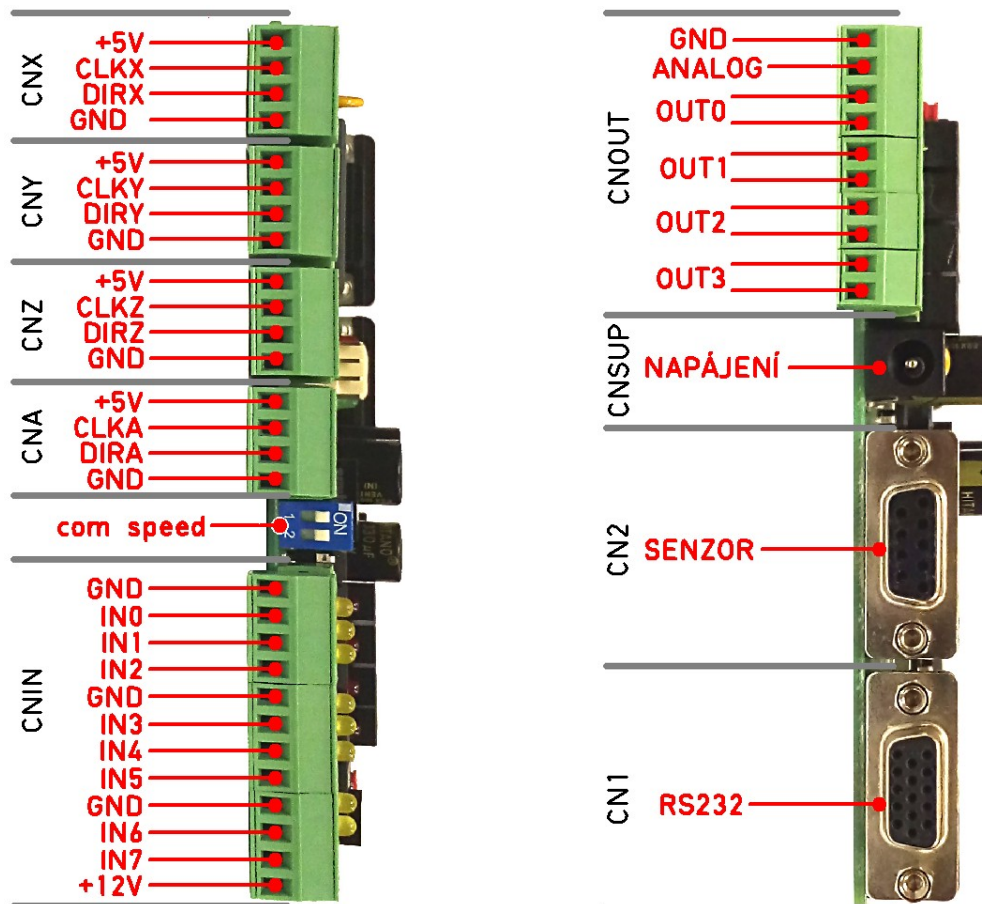
120 x 80 x 22mm



5 Přehled



6 Konektory



7 Popis konektorů

CNX	výstup pro driver osy X	CNOOUT	výstupy
CNY	výstup pro driver osy Y	CNSUP	napájení
CNZ	výstup pro driver osy Z	CN2	Senzor nástroje
CNA	výstup pro driver osy A	CN1	RS232
CNIN	vstupy		

8 Popis vývodů

konektor	vývod	popis
CNX	+5V	Výstup 5 V pro optočlen driveru
	CLKX	Signál KROK (step/clock) pro driver osy X
	DIRX	Signál SMĚR (dir) pro driver osy X
	GND	Napájení - zem
CNY	+5V	Výstup 5 V pro optočlen driveru
	CLKY	Signál KROK (step/clock) pro driver osy Y
	DIRY	Signál SMĚR (dir) pro driver osy Y
	GND	Napájení - zem
CNZ	+5V	Výstup 5 V pro optočlen driveru
	CLKY	Signál KROK (step/clock) pro driver osy Z
	DIRY	Signál SMĚR (dir) pro driver osy Z
	GND	Napájení - zem
CNA	+5V	Výstup 5 V pro optočlen driveru
	CLKA	Signál KROK (step/clock) pro driver osy A
	DIRA	Signál SMĚR (dir) pro driver osy A
	GND	Napájení - zem
CNIN	GND	Zem vstupů
	IN0	Vstup 0, uzemňuje se k GND – CNIN (ref X)
	IN1	Vstup 1, uzemňuje se k GND – CNIN (ref Y)
	IN2	Vstup 2, uzemňuje se k GND – CNIN (ref Z)
	GND	Zem vstupů
	IN3	Vstup 3, uzemňuje se k GND – CNIN (ref A)
	IN4	Vstup 4, uzemňuje se k GND – CNIN
	IN5	Vstup 5, uzemňuje se k GND – CNIN
	GND	Zem vstupů
	IN6	Vstup 6, uzemňuje se k GND – CNIN (tl. START)
	IN7	Vstup 7, uzemňuje se k GND – CNIN (tl. STOP)
	+12V	Zdroj 12 V/50m A (např. pro indukční snímače)
CNOUT	GND	Zem pro analog výstup
	ANALOG	Analog výstup 0 – 10 V nebo PWM výstup
	OUT0	Kontakty relé (vřeten start CW) max 48 VDC, 3 A
	OUT1	Kontakty relé (chlazení nástroje) max 48 VDC, 3 A
	OUT2	Kontakty relé (ofuk nástroje) max 48 VDC, 3 A
OUT3	Kontakty relé (elmag. brzda osy Z) max 48 VDC, 3 A	
CNSUP	napájení	Napájení 9-24 VDC, 300 mA max viz. Doporučené zapojení
CN2	senzor	Konektor senzoru nástroje viz Doporučené zapojení
CN1	RS232	Konektor sériového rozhraní RS232 viz Doporučené zapojení

Pro správnou funkci při použití ovládacího sw ARMOTE je vhodné dodržet následující použití.

- Vstupy IN0, IN1 a IN2 na konektoru CNIN jsou vyhrazeny pro referenční (home) spínače.
- Výstup OUT0 je vždy používán pro signál roztočení včetně pro frekvenční měniče.
- Výstup OUT1 je používán pro ovládání ventilu ofukování nástroje (konfigurovatelné).
- Výstup OUT2 je používán pro ovládání ventilu chlazení nástroje (konfigurovatelné).
- Výstup OUT3 je používán pro odbrzdění elmag brzdy osy Z se zpožděním pro nastartování driverů (konfigurovatelné).

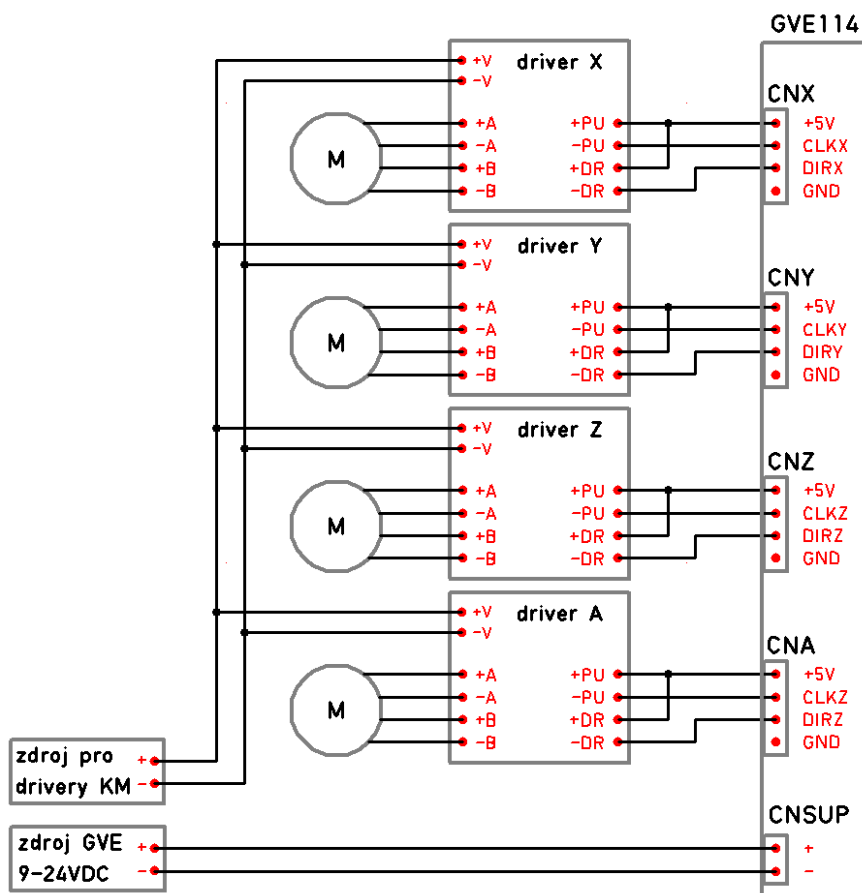
Logiku spínání výstupů při použití sw ARMOTE lze konfigurovat utilitou UniGVEconfig.exe (volně ke stažení na www.gravos.cz).

Pro správnou funkci analogového výstupu (0-10v) na konektoru CNOUT, je nutné nastavit jumper na desce do polohy ANA a na adrese 0x04 v EEPROM nastavit hodnotu 0xFF (standardní signál PWM, jinak nebude analogový výstup fungovat správně).

9 Příklady doporučeného zapojení

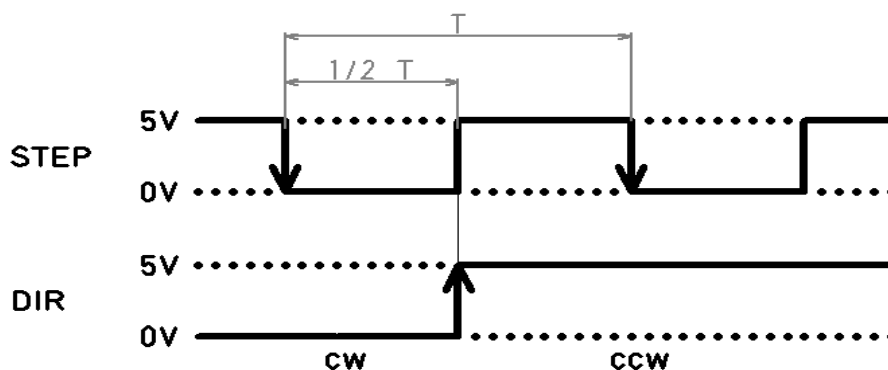
9.1 Připojení pohonů zařízení, (CNX – CNA)

9.1.1 Připojení krokových motorů



9.1.2 Časování signálů KROK a SMĚR

Délka pulzu je vždy $\frac{1}{2}$ periody, při 500 kHz je délka pulzu 1 uS, při 200 kHz je délka pulzu 2.5 uS, při 125 kHz je délka pulzu 4 uS a při 50 kHz je délka pulzu 10 uS.

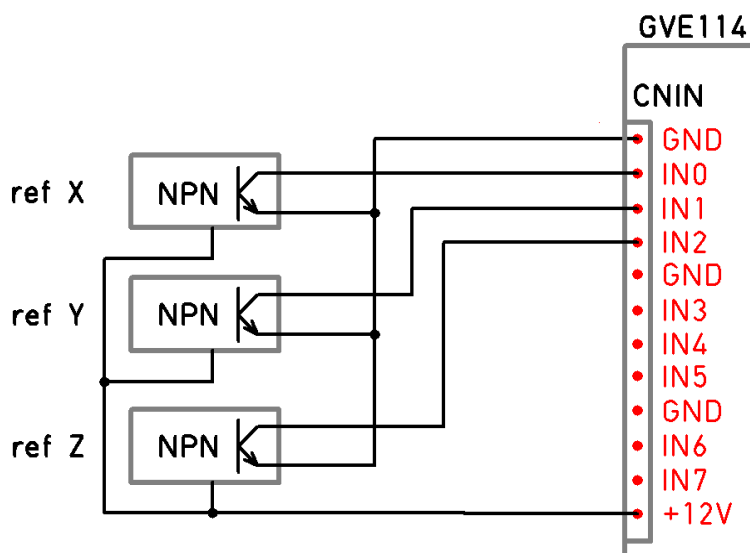


9.2 Zapojení vstupů (CNIN)

9.2.1 Připojení indukčních snímačů

Pro referenční spínače v systémech GRAVOS-ARMOTE.

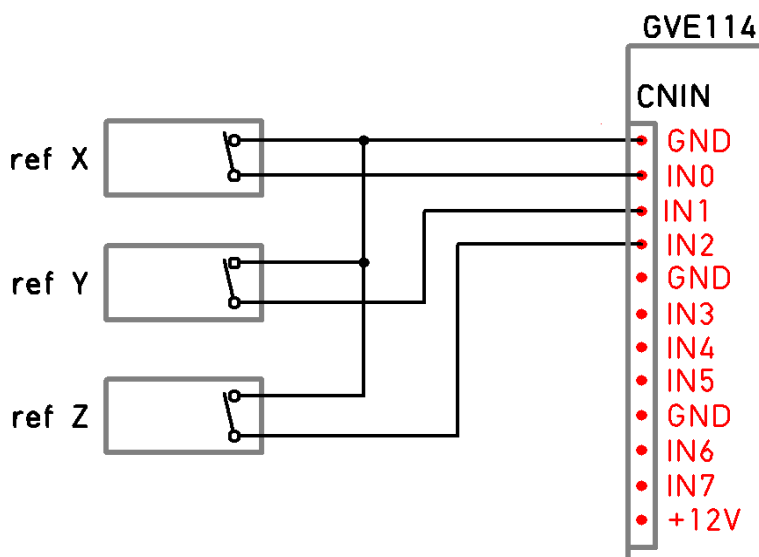
Ref. Spínače jsou rozpínací, aby při poškození kabelu došlo k zastavení stroje.



9.2.2 Připojení mechanických spínačů

Pro referenční spínače v systémech GRAVOS-ARMOTE.

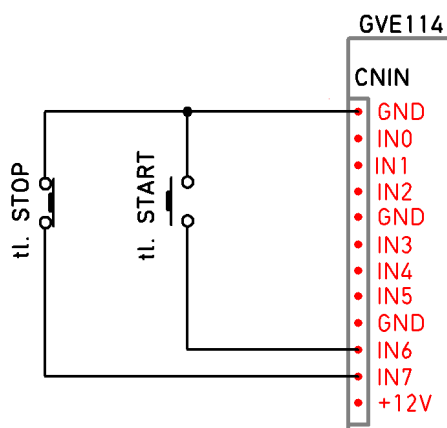
Ref. Spínače jsou rozpínací, aby při poškození kabelu došlo k zastavení stroje.



9.2.3 Připojení tlačítek START s STOP

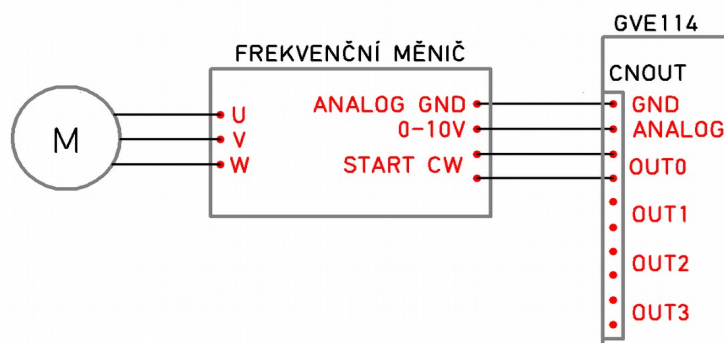
Pro spuštění a zastavení obrábění v systémech GRAVOS-ARMOTE.

Tl. START je spínací a tl. STOP rozpínací, aby při poškození kabelu došlo zastavení stroje.

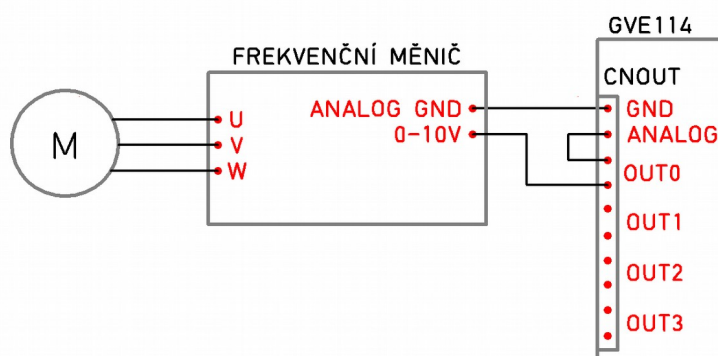


9.3 Zapojení výstupů (CNOOUT)

9.3.1 Připojení frekvenčního měniče se signály 0-10V a CW START

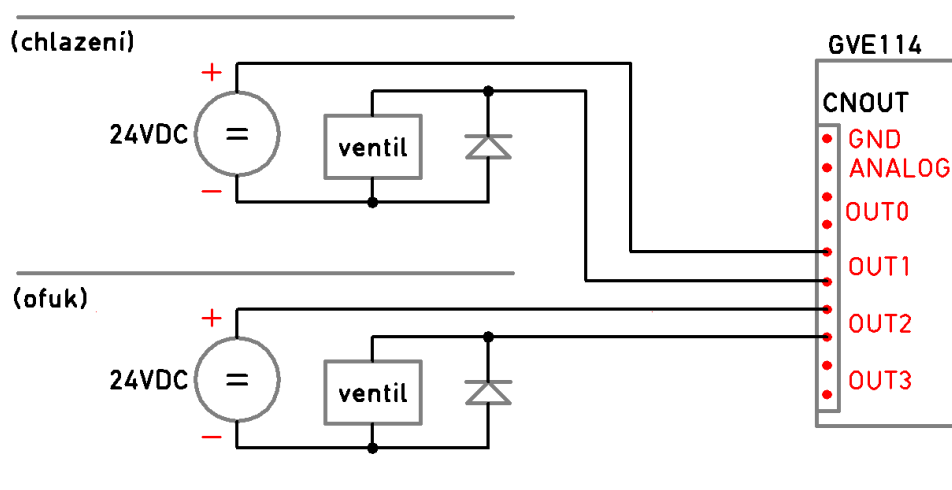


9.3.2 Připojení frekvenčního měniče pouze se signálem 0-10V



9.3.3 Připojení elmag. ventilů

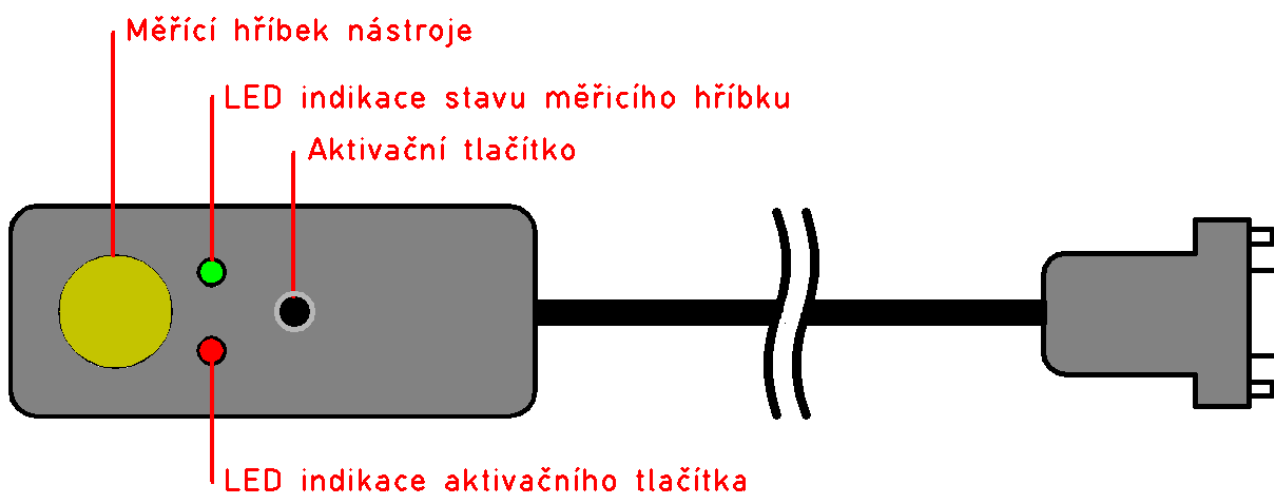
Pro spínání chlazení a ofuku nástroje v systémech GRAVOS-ARMOTE.



9.4 Připojení senzoru měření nástroje (CN2)

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.

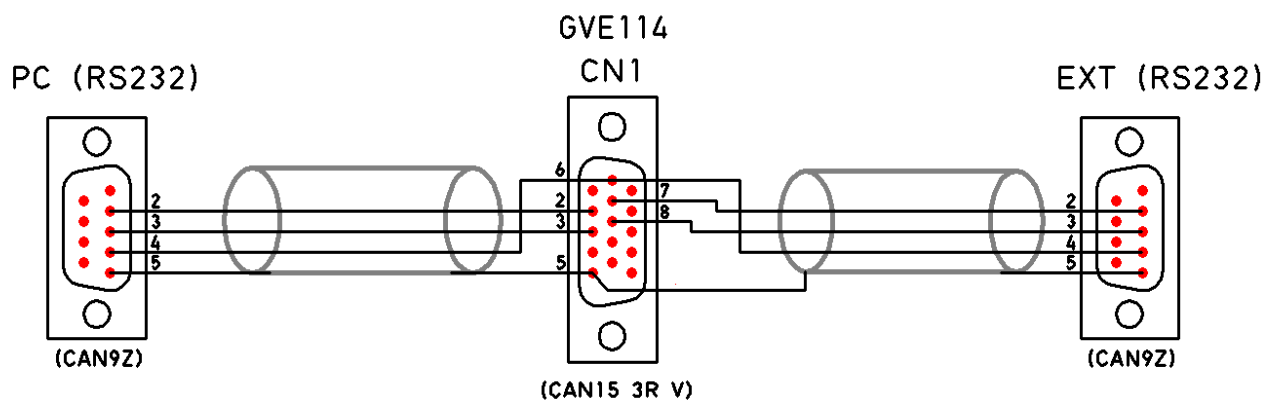


9.5 Kabel k připojení GVE114 a dalších zařízení k PC (CN1)

K připojení GVE114 k PC slouží kabel KAB-GVE-B15 (bez použití GVE65).

Další jednotky GVE je možné na jednu komunikační linku připojit pomocí EXT odbočky (KAB-GVE-BE15).

Všechny jednotky na jedné komunikační lince musí mít nastavenou stejnou komunikační rychlost a každá jednotka musí mít jinou adresu.



9.6 Napájení, (CNSUP)

Připojení souosým napájecím konektorem 5,5/2,1 mm (KAB-SUP-2.1)

Napájecí napětí
9 – 24 VDC



Doporučený zdroj

12 V, 300 mA (adaptér SUP-12V, spínaný zdroj SUP-12V-S)

10 Přepínače

10.1 Přepínač COM SPEED

Rychlost komunikace je potřeba nastavit před připojením napájení.

Všechna zařízení na stejné komunikační lince musí mít nastavenou stejnou komunikační rychlost. Rozsah komunikačních rychlostí lze přepnout na adrese 0x0D, výchozí hodnota je FF.

0x0D=01		
sw1	sw2	rychlost
OFF	OFF	921 000 Bd
OFF	ON	460 800 Bd
ON	OFF	230 400 Bd
ON	ON	115200 Bd

0x0D=FF		
sw1	sw2	rychlost
OFF	OFF	19 200 Bd
OFF	ON	38 400 Bd
ON	OFF	57 600 Bd
ON	ON	115 200 Bd

10.2 Přepínač ANALOG/PWM

Pomocí jumperu lze přepnout funkci výstupu pro ovládání vřetene.

V poloze ANALOG je mezi vývodem ANALOG a GND konektoru CNOUT 0 – 10 V pro řízení frekvenčního měniče. V poloze PWM je na výstupu ANALOG obdélníkový signál PWM v rozsahu 0,0 % až 100,0 % včetně, amplituda 5 V, kmitočet 7,3728 kHz.

11 Nastavení funkce výstupů

Funkce výstupů pro ovládací sw ARMOTE lze konfigurovat pomocí příkazu *Write* zapsáním hodnot na příslušné adrese výstupu nebo použít sw UniGVE config.

11.1 Adresy EEPROM pro výstupy

adresa	výstup
0x01	OUT1
0x02	OUT2
0x03	OUT3
0x04	PWM
0x05	Výstup signálů STEP/DIR
0x0C	Sdružení osy A
0xFD	Čas po zapnutí pro odbrždění brzdy v 0,1s (max 5s)

11.2 Hodnoty nastavení pro OUT1 - OUT3

hodnota	funkce
0x00	nepoužito
0x01	ovládání laseru
0x02	chlazení nástroje
0x03	ofuk nástroje
0x04	zámek krytu stroje
0x05	uvolnění nástroje
0x06	otevření krytu nástrojů
0x08	brzda
0x09	signalizace přerušení
0x10- 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 PWM

Pokud je analogový výstup přepnut jumperem na desce do pozice PWM lze přepnout jeho funkci, která může být buď normální PWM signál nebo signál pro oddělovací desky často používané u sw řídicích systémů s LPT. Nastavení se provádí na adrese 0x04.

hodnota	funkce
0xFF	Standardní signál PWM
0x01	Signál pro oddělovací desky

11.4 Hodnoty nastavení sdružení osy A

hodnota	funkce
0x01	Osa A je sdružená s osou X
0x02	Osa A je sdružena s osou Y
0x03	Osa A je sdružená s osou Z
0x04 - 0xFF	Osa A je samostatná

11.5 Nastavení výstupu signálů STEP/DIR

Nastavení výstupu signálů STEP/DIR na konektorech CNX – CNA se provádí na adrese 0x05, lze zde měnit polaritu signálů STEP pro všechny osy najednou a polaritu signálů DIR pro interpolované osy X,Y,Z,A samostatně. Změnou polarity signálu DIR se mění směr osy. Hodnoty jsou v hexadecimálním tvaru, defaultně nastaveno na FF (všechny bity na 1).

bit 0 = polarita signálu STEP osy X
 bit 1 = polarita signálu STEP osy Y
 bit 2 = polarita signálu STEP osy Z
 bit 3 = polarita signálu STEP osy A

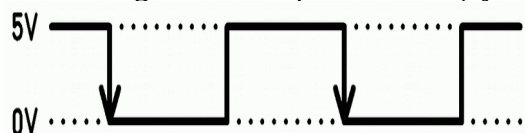
bit 4 = polarita signálu DIR osy X
 bit 5 = polarita signálu DIR osy Y
 bit 6 = polarita signálu DIR osy Z
 bit 7 = polarita signálu DIR osy A

hodnoty se sečtou, a dají výslednou konfiguraci směru všech os

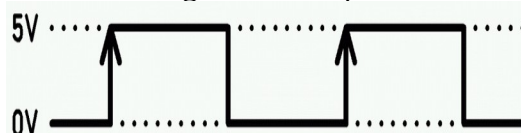
např.: $0 + 0 + 4 + 8 + 0 + 32 + 0 + 128 = 172 = 0xAC$

X,Y sestupná hrana Z,A vzestupná hrana, X otočená, Y normálně, Z otočená, A normálně.

Polarita signálu STEP při bit 4 = 1 (výchozí)



Polarita signálu STEP při bit 4 = 0



Šipka značí aktivní hranu, pro drivery s aktivní sestupnou hranou nastavte bit 4 na 1 a pro driver s aktivní náběžnou hranou nastavte bit 4 na 0, která hrana je pro driver aktivní se dočtete v datasheetu příslušného driveru.

12 Nastavení funkce vstupů

K nastavení lze použít konfigurační utilitu UniGVE Config, při použití sw Armote si program sám hlídá společné parametry, aby byly shodné v jednotce GVE114 i v nastavení programu. Pokud je MPG vypnuto na adrese 0x06, kontrolu parametrů Armote neprovádí.

12.1 Adresy EEPROM pro vstupy a MPG

Při použití sw Armote v1.98 a vyšší, tyto hodnoty nastavuje Armote sám podle nastavení stroje.

adresa	výstup
0x06	MPG Enable, byte
0x07	Ref. Start num., byte
0x08	Intr 0 - 7 Povolení přerušení, byte
0x09	Intr 14 – 15 Povolení přerušení, byte
0x0A	Intr 0 – 7 Polarita vstupu, byte
0x0B	Intr 14 – 15 Polarita vstupu, byte
0x10	Spodni Limit X, float, [mm]
0x14	Spodni Limit Y, float, [mm]
0x18	Spodni Limit Z, float, [mm]
0x20	Horní Limit X, float, [mm]
0x24	Horní Limit Y, float, [mm]
0x28	Horní Limit Z, float, [mm]
0x30	Počet kroků na mm X, float, [kroky]
0x34	Počet kroků na mm Y, float, [kroky]
0x38	Počet kroků na mm Z, float, [kroky]
0x40	Akcelerace MPG, float, [mm/s ²]
0x44	Max Rychlost MPG, float, [mm/s]
0x48	Prodleva MPG, uint32, [ms]
0x4C	Defaultní otáčky vřetene, uint32, [promile]
0x60	Násobitel velikosti kroku, float

12.2 Hodnoty nastavení MPG Enable

Při použití sw Armote v1.98 a vyšší, tyto hodnoty nastavuje Armote sám podle nastavení stroje.

Hodnota	funkce	funkce polohy 4
0x01	MPG povoleno	Vřeteno
0x02	MPG povoleno s omezením limit	Vřeteno
0x81	MPG povoleno	Rotační osa A
0x82	MPG povoleno s omezením limit	Rotační osa A
jiná	MPG nepovoleno	

12.3 Nastavení polarity a povolení přerušení vstupů

Při použití sw Armote v1.98 a vyšší, tyto hodnoty nastavuje Armote sám podle nastavení stroje.

Vstup je aktivní v log. 0

Polarita vstupu 0 – 7 se nastavuje na adrese 0x0A, číslo bitu odpovídá číslu vstupu.

Polarita vstupu 14 – 15 se nastavuje na adrese 0x0B, bit 6 = IN 14, bit 7 = IN 15

Povolení přerušení vstupu 0 – 7 se nastavuje na adrese 0x08, číslo bitu odpovídá číslu vstupu

Povolení přerušení vstupu 14 – 15 se nastavuje na adrese 0x09, bit 6 = IN 14, bit 7 = IN 15

12.4 Hodnoty nastavení limit

Při použití sw Armote v1.98 a vyšší, tyto hodnoty nastavuje Armote sám podle nastavení stroje.

Pokud je hodnota některého parametru mimo rozsah, není možné MPG použít.

Adresa	hodnota
0x10	Spodní limita X, 4 byty float v rozsahu 0 – 10000
0x14	Spodní limita Y, 4 byty float v rozsahu 0 – 10000
0x18	Spodní limita Z, 4 byty float v rozsahu 0 – 10000
0x1C	Spodní limita A, 4 byty float v rozsahu 0 – 10000
0x20	Horní limita X, 4 byty float v rozsahu 0 – 10000
0x24	Horní limita Y, 4 byty float v rozsahu 0 – 10000
0x28	Horní limita Z, 4 byty float v rozsahu 0 – 10000
0x2C	Horní limita A, 4 byty float v rozsahu 0 – 10000

12.5 Nastavení počtu kroků na mm

Při použití sw Armote v1.98 a vyšší, tyto hodnoty nastavuje Armote sám podle nastavení stroje.

Pokud je hodnota některého parametru mimo rozsah, není možné MPG použít.

Adresa	hodnota
0x30	Počet kr/mm X, 4 byty float v rozsahu 1–10000
0x34	Počet kr/mm Y, 4 byty float v rozsahu 1–10000
0x38	Počet kr/mm Z, 4 byty float v rozsahu 1–10000
0x3C	Počet kr/mm A, 4 byty float v rozsahu 1–10000

12.6 Hodnoty nastavení pohybu pro MPG

Při použití sw Armote v1.98 a vyšší, tyto hodnoty nastavuje Armote sám podle nastavení stroje.

Pokud je hodnota některého parametru mimo rozsah, není možné MPG použít.

Adresa	hodnota
0x40	Akcelerace, 4 byty float v rozsahu 1–10000 [mm/s ²]
0x44	Max rychlost, 4 byty float, max ½ inerpolační rychlosti, [mm/s]

12.7 Nastavení prodlevy MPG

Při použití sw Armote v1.98 a vyšší, tyto hodnoty nastavuje Armote sám podle nastavení stroje.

Adresa 0x048. Hodnota je 4 byty uint32 v rozsahu 1-2000, jednotky jsou [ms]

Prodleva MPG je čas, po jehož uplynutí po zastavení otáčení kolečka na MPG začne jednotka brzdít osu po rampě bez ohledu na počet kroků o který bylo kolečko pootočeno, pokud stroj nestihne fyzicky všechny kroky vykonat.

12.8 Nastavení defaultních otáček vřetene

Při použití sw Armote v1.98 a vyšší, tyto hodnoty nastavuje Armote sám podle nastavení stroje.

Adresa 0x4C. Hodnota je 4 byty uint32 v rozsahu 1 – 1000, jednotky jsou promile z rozsahu otáček (z rozsahu 0 – 10V analog. výstupu konektoru CNOOUT, hodnota 500 = 5 V).

12.9 Nastavení velikosti kroku MPG

Velikost kroku se nastavuje na adrese 0x60, hodnota je 4 byty float.

Např.

Hodnota=1	X1=0.01 mm, X10=0.1 mm, X100=1 mm
Hodnota=0.1	X1=0.001 mm, X10=0.01 mm, X100 =0.1 mm

13 GVE114 – popis vnitřních instrukcí

Popis příkazů jednotky pro tvorbu vlastních aplikací.

13.1 CPU

Procesor ARM Cortex M4

13.2 Program

IP114/500 x16 16.9.2018 (c) Gravos (P.Borovsky)

13.3 Sériový přenos

Sériový přenos 8 bitů, 1 stop bit, bez parity, přenosová rychlost BaudRate je volitelná přepínači. Adresa jednotky je pevná a nelze uživatelsky změnit. Při potřebě jiné než výchozí adresy je nutné tento požadavek specifikovat do objednávky.

Stav přepínačů je vyhodnocen jen jednou po zapnutí napájení.

Jednotka má pro komunikaci konektor CN1

13.3.1 Komunikace

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. Jednotka odpoví vždy co nejdříve, s výjimkou příkazu Halt, kdy odpoví až po zabrzdění.

13.3.2 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 stroj 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 nahlásí 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 Výpadá to složitě, ale není. Pro člověka takové výpočty moc nejsou, ale pro SW to představuje pár řádků.

13.4 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 "mezera" - je to označované jako Latence, bývá od 1 do 16ms. A při příjmu odpovědi to samé. Takže i přes vysokou přenosovou rychlost USB portů bývá výsledek horší než u Com portu

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 brzdění v koncových bodech vektoru.

Doporučujeme k fréze/manipulátoru, počítač se skutečným(i) sériovým portem, třeba i starší. U stroje mohou ještě dobře posloužit. V případě použití USB převodníku důrazně doporučujeme, aby byl galvanicky oddělený. USB porty jsou často citlivé na statickou elektřinu.

13.5 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	- max. rychlost 1000 kr/s
\$512	- short vektory jsou menší než 512 kroků
B50000	- bez omezení rychlosti mezi cont. vektory (pro 50 000 hz verzi)
N	- čítač polohy vynulován
O0,FF	- výstup vypnutý

13.6 Příkazy

! Adresa Příkaz [parametry] Enter

mezi jednotlivými parametry je čárka

Adr.0 = interpolační jednotka (Adresa může být v rozsahu 0-7, defaultně je adresa 0, při požadavku jiné adresy (např při ovládání více jednotek najednou) je nutné tento požadavek specifikovat při objednání jednotky.

13.6.1 Identifikace jednotky

? - **VERSION** dotaz na verzi programu

Q - **QUESTION** dotaz na ID procesoru, vrací řetězec 8 čísel

jejich význam: ddmmrrpp

dd = den výroby jednotky

mm = měsíc výroby jednotky

rr = rok výroby jednotky

pp = kolikátá jednotka toho dne

např.: 25022504 znamená 25.2.2025 čtvrtý kus toho dne, toto číslo je jedinečné - neexistují 2 procesory se stejným číslem.

13.6.2 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

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 hlásí 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} - **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

Určit jestli je vektor pokračující je výpočetně složitě a 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 2048 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 se zlepší 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 sériové lince).

Tn - **TIME** prodleva mezi nenavazujícími (L) vektory v milisekundách.

n= 1 az 24 milisekund.

doporučená hodnota je podle hmotnosti stroje asi 5 az 20 ms.

Mezi CONT.vektory tato prodleva není.

např.: !0T5 - prodleva 5 ms

Příkaz je modální, platí až do zadání jiné hodnoty.

\$n - **SHORT** je hraniční hodnota pro rozlišení krátkého a dlouhého vektoru.

n = 1..2147483647

Dlouhý vektor se snaží dostat pomocí zrychlení A až k maximální rychlosti V.

Krátký vektor se snaží dostat pomocí zrychlení A jen k brzdné rychlosti B na svém konci.

Tímto se stává fronta krátkých vektorů plynulejší, a průjezd libovolnou spojitou křivkou, která je rozumně rozložená na úsečky je plynulý také.

13.6.3 Rychlosti

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 100000 pulsů/s (záleží na interpolační rychlosti konkrétní jednotky)

např.: !0V10000 - rychlost 10000 pulsů/s

Příkaz je modální, platí až do zadání jiné hodnoty.

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 Y v rozsahu 10 - 100000

Y = max. rychlost v tis. pulsů/s osy Y v rozsahu 10 - 100000

Z = max. rychlost v tis. pulsů/s osy Y v rozsahu 10 - 100000

např. !0VL25000,25000,30000

Bn - **BRAKE** rychlost, na kterou má vektor dobrzdit, pokud za ním ve frontě je další Cont.vektor.

Pokud za ním není další, tak stejně dobrzdí do nuly.

n = 10 az 100000 pulsů/s

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

13.6.4 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, kde rychlost pro každý vektor musí určit nadřazené PC s ohledem na směr pohybů jednotlivých os.

13.6.5 Změny rychlosti během pohybu

XA - EXCHANGE změni rychlosti u všech vektorů ve frontě na poslední zadanou rychlost V.

XU - EXCHANGE UP změni rychlosti u všech vektorů ve frontě. Hodí se pro změnu parametrů za chodu. Rychlost se zvětší o 1/16 (6,25 %) současného stavu

XD - EXCHANGE DN změni rychlosti u všech vektorů ve frontě. Hodí se pro změnu parametrů za chodu. Rychlost se zmenší o 1/16 (6,25 %) současného stavu

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.

13.6.6 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. Vhodné 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.

N - NULL ALL vynuluje všechny 3 osy čítače pozice. Nelze použít za chodu vektoru (při RUN=1)

Nan - **NULL AXIS** Vynuluje nebo nastaví čítač pozice vybrané osy.

Parametr a je osa, které se má čítač nastavit (X,Y,Z).

parametr n je v rozsahu -2147483647 až 2147483647.

např. !0NX0 – vynuluje čítač pozice osy X.

např. !0NX100 – nastaví čítač pozice na hodnotu 100.

13.6.7 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)

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

13.6.8 Obsluha ručního ovladače MPG

Yn - **YOG (JOG)** řízení ručním ovladačem MPG. Odpovědí je odeslaný příkaz s parametrem.

n = + -MPG zapnuto s posíláním informací o stavu ovladače a polohy

n = & -MPG zapnuto bez posílání informací (po zapnutí nebo resetu)

n = - -MPG vypnuto

Při odeslání Y+ nebo Y- je zakázáno spuštění ref. pojezdů aktivací vstupem.

Při odeslání Y+ nebo Y& je stav stroje v chodu (RUN=1), není možné zadávat vektory.

13.6.9 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 dokončit 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 (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 proč 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. Určeno 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.

13.6.10 Obsluha paměti EEPROM

ERn - **READ BYTE** přečte byte z EEPROM na adrese n a pošle jej po sériové lince

n = 00..FF(hex)

EFRn - **READ FLOAT** přečte float z EEPROM na adrese n a pošle jej po sériové lince

n = 00..FF(hex)

EURn - **READ UINT32** přečte uint32 z EEPROM na adrese n a pošle jej po sériové lince

n = 00..FF(hex)

EWn,x - **WRITE BYTE** zapíše byte x EEPROM na adresu n

n = 00..FF(hex) x = 00..FF(hex)

EFWn,x - **WRITE FLOAT** zapíše float x EEPROM na adresu n

n = 00..FF(hex) x = 00..FF(hex)

EUWn,x - **WRITE UINT32** zapíše uint32 x EEPROM na adresu n

n = 00..FF(hex) x = 00..FF(hex)

13.6.11 Ovládání relé

00,n - **OUTPUT** zapíše byte x v hexadecimálním tvaru na výstupní port 0 jsou aktivní v log.0:

bit 0 = CNOUT - OUT0 (v kontrolérech Gravos vřeteno)

bit 1 = CNOUT - OUT3 (v kontrolérech Gravos brzda)

bit 2 = CNOUT - OUT1 (v kontrolérech Gravos chlazení)

bit 3 = CNOUT - OUT2 (v kontrolérech Gravos ofuk)

bit 4 – 7 = nepoužit

výstupy jsou aktivní v log.0, po zapnutí jsou neaktivní log.1

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

vypnutí všeho: !000,FF

13.6.12 Čtení stavu vstupů a obsluha přerušení

I1 - **INPUT** přečte vstupní port 1. Odpovědí je stav portu v hexadecimálním tvaru.

bit 0 = CNIN - IN0 Intr0 (v kontrolérech Gravos RefX)
 bit 1 = CNIN - IN1 Intr1 (v kontrolérech Gravos RefY)
 bit 2 = CNIN - IN2 Intr2 (v kontrolérech Gravos RefZ)
 bit 3 = CNIN - IN3 Intr3 (v kontrolérech Gravos RefA)
 bit 4 = CNIN - IN4 Intr4
 bit 5 = CNIN - IN5 Intr5
 bit 6 = CNIN - IN6 Intr6 (v kontrolérech Gravos tlačítko Start)
 bit 7 = CNIN - IN7 Intr7 (v kontrolérech Gravos tlačítko Stop)

I2 - **INPUT** přečte vstupní port 2. Odpovědí je stav portu v hexadecimálním tvaru

bit 0 - 5 = nepoužit
 bit 6 = CN2 - pin 4 Intr14 (v kontrolérech Gravos tlačítko senzoru)
 bit 7 = CN2 - pin 3 Intr15 (v kontrolérech Gravos hříbek senzoru)

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 INTR0-2 (0 = zakázáno)
 ST2 = polarita uživatel. přerušení INTR0-2 (0 = aktivní v log.0)
 ST4 = příčina přerušení INTR0-2 (0 = přerušení nebylo)

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 brzdění 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

13.6.13 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.

J - **JUMP** na RESET zresetuje včetně vynulování čítače polohy.

13.7 Ovládaní vřetene adr.7

Vn - **VELOCITY** Ovládaní analogového/PWM výstupu 0,0...10 V nebo střída 0,0 %...100,0 %.

např: !7V300 = výstup na 3V/ nebo PWM na 30% (podle jumperu)

při zadání !7V0 se také vypne OUT0

při zadání !7V1...1000 se také zapne OUT0

Tedy pozor: !7V500 je něco jiného než !0V500

na adrese 7 jednotka reaguje také na (stejně jako na adrese 0)

? - Version

Q - Question

> - Repeat

@ - Index_out

J - Jump

% - ChSum

E - Eeprom

13.8 Odpovědi

1 hexadecimální znak 0..F [další vyžadované parametry] Enter (0Dh) mezi jednotlivými parametry je čárka. Odpověď je odeslána ihned po zadání příkazu.

Jedině pro příkaz HALT je odpověď odeslána až po vykonání instrukce.

INTA, 3 bitový kód chyby

bit 3, 2 .. 0

kód chyby:

0 = OK (žádná chyba)

1 = fronta je plná, nelze zařadit další vektor je nutné počkat, zopakovat

2 = příkaz nepřišel celý včas, přetržení komunikace (zafunguje WATCH DOG)

3 = neznámý příkaz

4 = chyba syntaxe

5 = parametr mimo meze

6 = pro Go, není co spustit

7 = za chodu vektoru nelze

Např.:

Příkaz	Odpověď	Pozn.
!0L1000,0,0	0	OK
!0PF	0,-1000,2000,50	OK
!0C100,20,0	1	vektor nebyl přijat (je nutné ho opakovat)
!0V1000	8	OK, ale je přerušeno (INTA=1)
!0SW8,F1	5	parametr mimo meze
!0SR2	0,2B	OK

© Gravos CNC s.r.o.
Poslední změna 5.11.2018
www.gravos.cz