CNC Technologie a obráběcí stroje

GVE64 – HW interpolátor

1 Specifikace:

- HW interpolační jednotka s výkonem 50 000 pulzů/s ve 4-osém pohybu. Možnost upgradu až na 125 000 pulzů/s a až na 200 000 pulzů/s ve verzi Phantom
- Vnitřní buffer pro 420 vektorů, max délka vektoru +- 2147483647 kroků
- Řízení krok/směr (plně 4 osá interpolace)
- Připojení k PC přes RS232 (USB přes opticky oddělený převodník)
- 8 vstupů (galvanicky oddělené), 2 vstupy pro senzor měření nástroje (neoddělené)
- 4 relé výstupy s použitím pro spínání (max 48VDC, 3A) např. odsávání, chlazení atd.
- 1 analogový výstup (0-10V) např. pro řízení frekvenčního měniče, možnost přepnout na PWM výstup (galvanicky oddělené).
- LED signalizace stavu vstupů a výstupů
- Napájení 9 – 24VDC
- Odběr 300mA. (max) při 12V.

2 Aplikace:

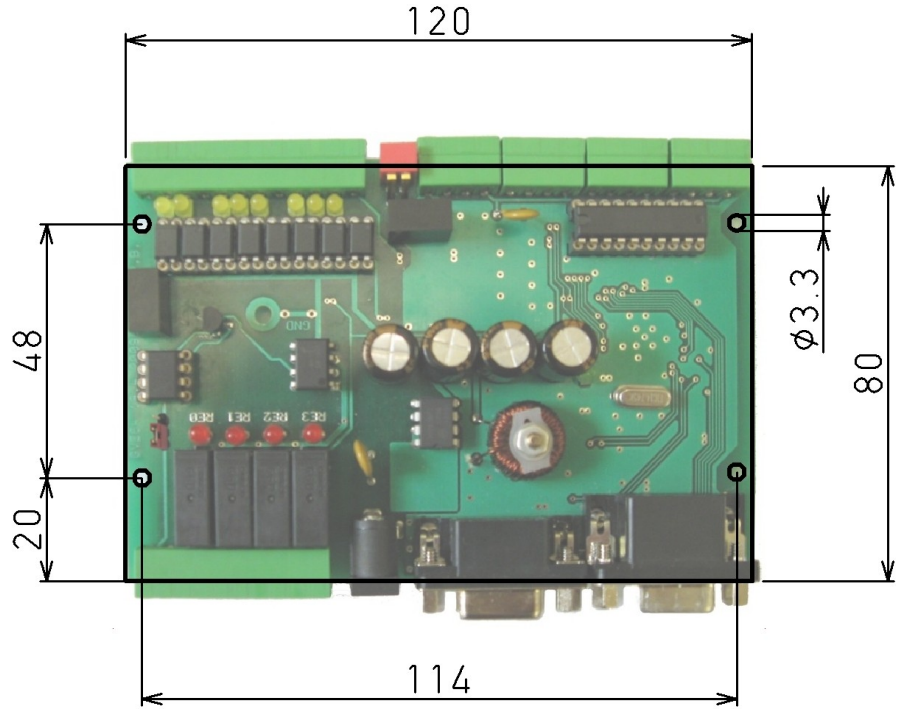
Řízení frézek,gravírek, vrtaček, polohovacích stolů, robotických manipulátorů atd.

3 Součást dodávky:

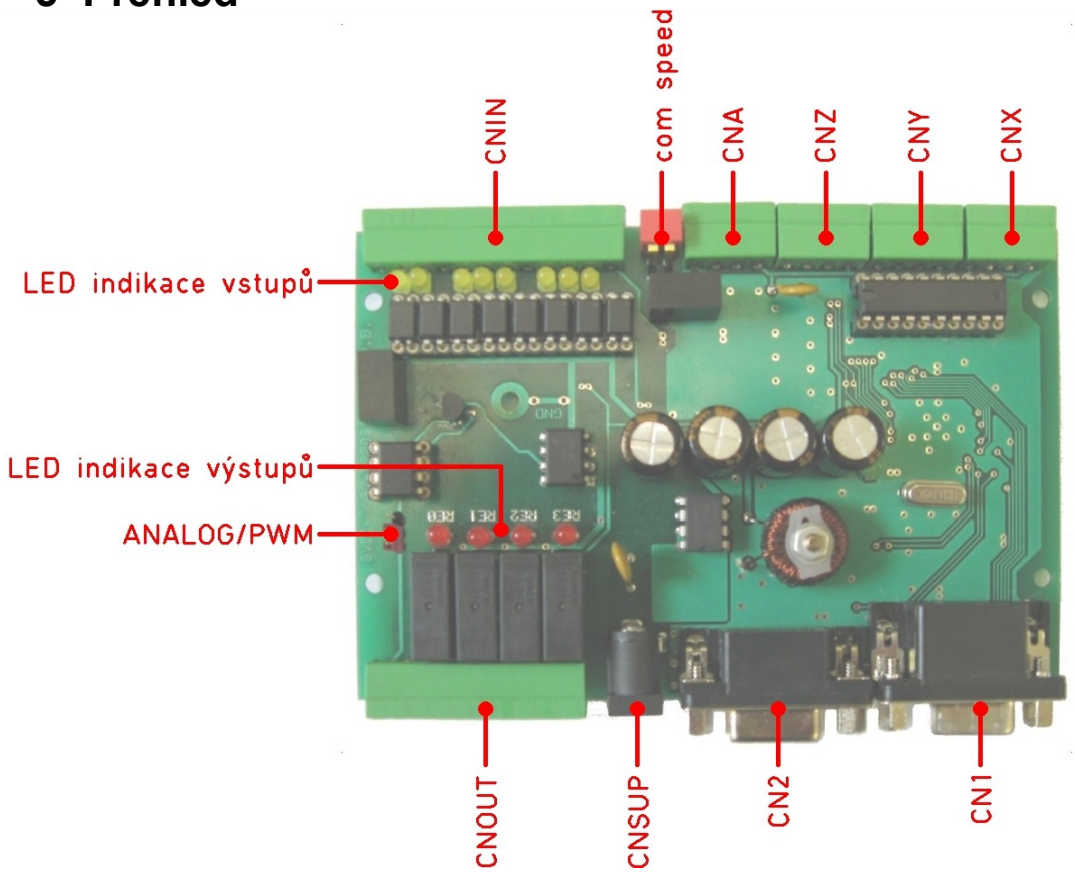
Jednotka GVE64,instalační CD, protikusy konektorů, zkušební verze sw ARMOTE pro řízení 4-osé frézky.

4 Rozměry:

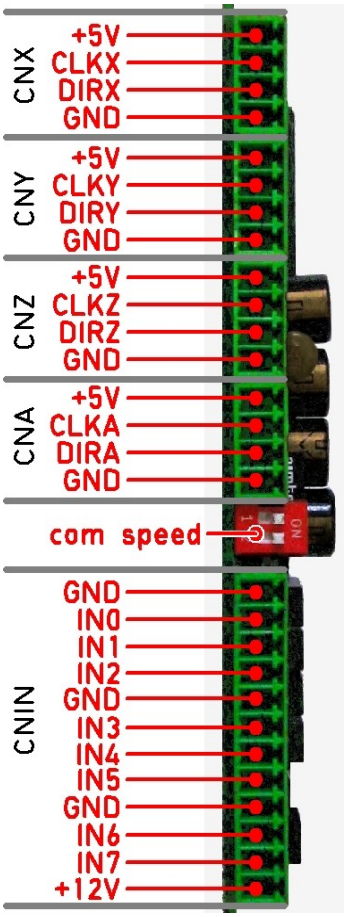
120 x 80 x 22mm

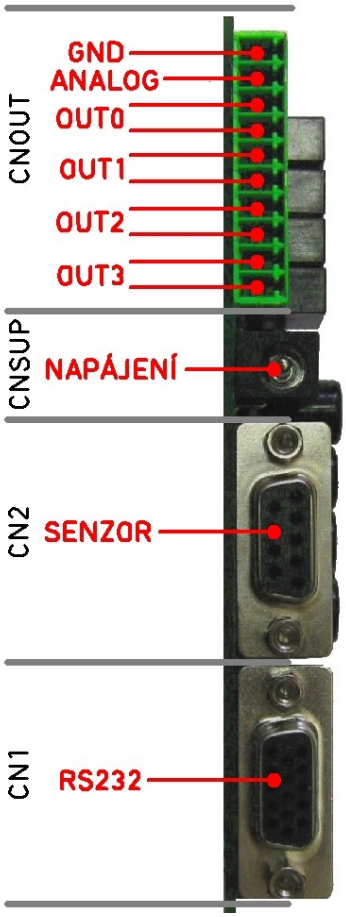


5 Přehled



6 Konektory:





7 Popis konektorů:

CNX	výstup pro driver osy X	**CNOUT**	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 5V pro optočlen driveru
	CLKX	Signál KROK (step) pro driver osy X
	DIRX	Signál SMĚR (dir) pro driver osy X
	GND	Napájení - zem
CNY	+5V	Výstup 5V pro optočlen driveru
	CLKY	Signál KROK (step) pro driver osy Y
	DIRY	Signál SMĚR (dir) pro driver osy Y
	GND	Napájení - zem
CNZ	+5V	Výstup 5V pro optočlen driveru
	CLKY	Signál KROK (step) pro driver osy Z
	DIRY	Signál SMĚR (dir) pro driver osy Z
	GND	Napájení - zem
CNA	+5V	Výstup 5V pro optočlen driveru
	CLKA	Signál KROK (step) 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 (tl. START)
	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
	IN7	Vstup 7, uzemňuje se k GND – CNIN (tl. STOP)
	+12V	Zdroj 12V/50mA (např. pro indukční snímače)
CNOUT	GND	Zem pro analog výstup
	ANALOG	Analog výstup 0 – 10V nebo PWM výstup
	OUT0	Kontakty relé (vřeteno start CW) max 48VDC, 3A

	OUT1	Kontakty relé (chlazení nástroje) max 48VDC, 3A
	OUT2	Kontakty relé (ofuk nástroje) max 48VDC, 3A
	OUT3	Kontakty relé (elmag. Brzda osy Z) max 48VDC, 3A
CNSUP	napájení	Napájení 9-24VDC, 300mA 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 nutné 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 vřetene 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 odbrždění elmag brzdy osy Z se zpožděním pro nastartování driverů (konfigurovatelné).

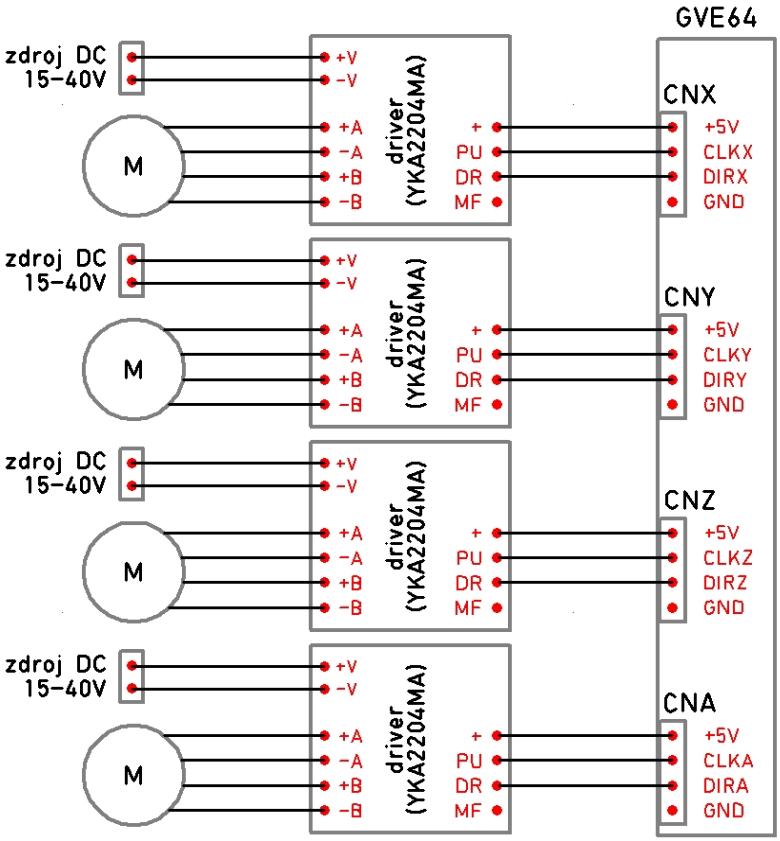
logiku spínání výstupů při použití ovl. sw ARMOTE lze konfigurovat utilitou GVE64_config.exe (volně ke stažení na www.gravos.cz) viz. kapitola 11 - nastavení funkce výstupů.

!!! 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:

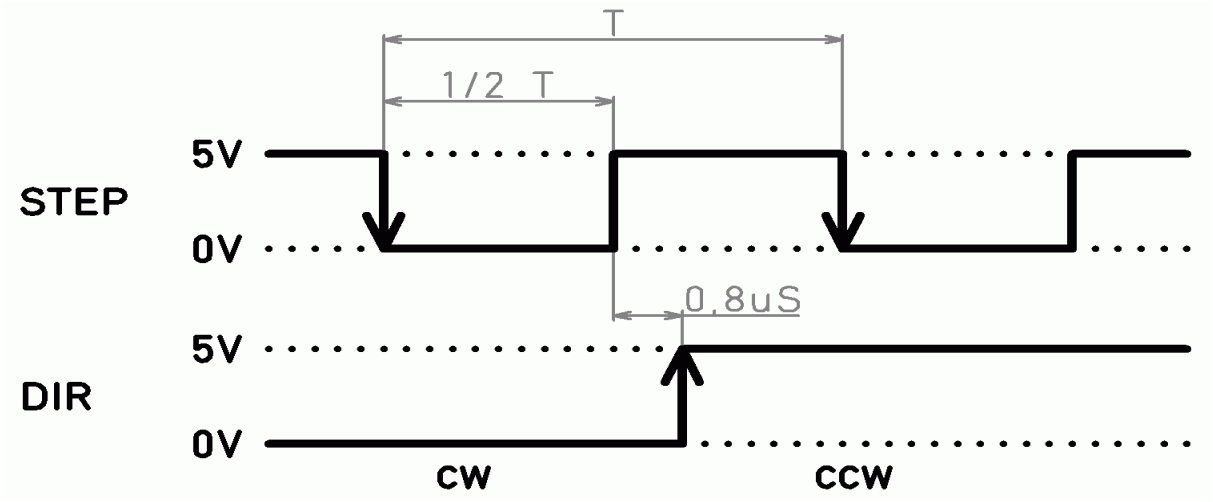
9.1 Připojení pohonů zařízení, (CNX – CNA):

9.1.1 Připojení krokových motorů



9.1.2 Časování signálů KROK a SMĚR

Délka pulzu je vždy ½ periody, při 100khz je délka pulzu 5uS, při 35khz je délka pulzu 14uS

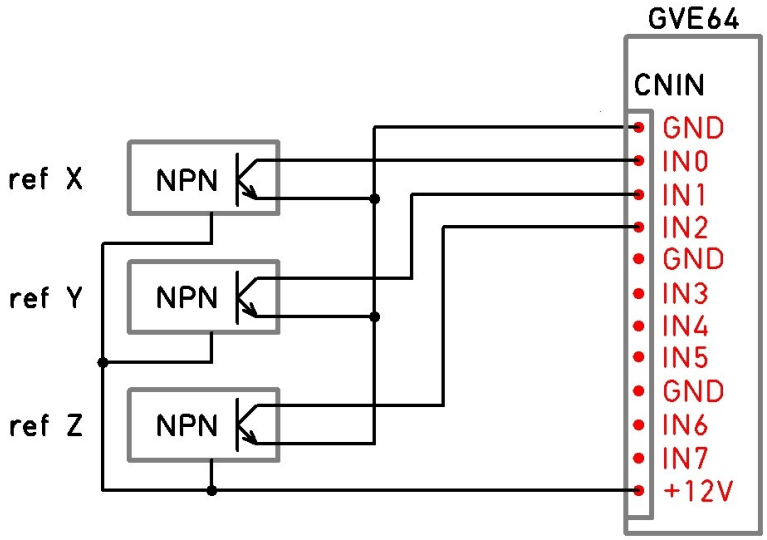


9.2 Zapojení vstupů (CNIN):

9.2.1 Připojení indukčních snímaču

(pro referenční spínače v systémech GRAVOS-ARMOTE)

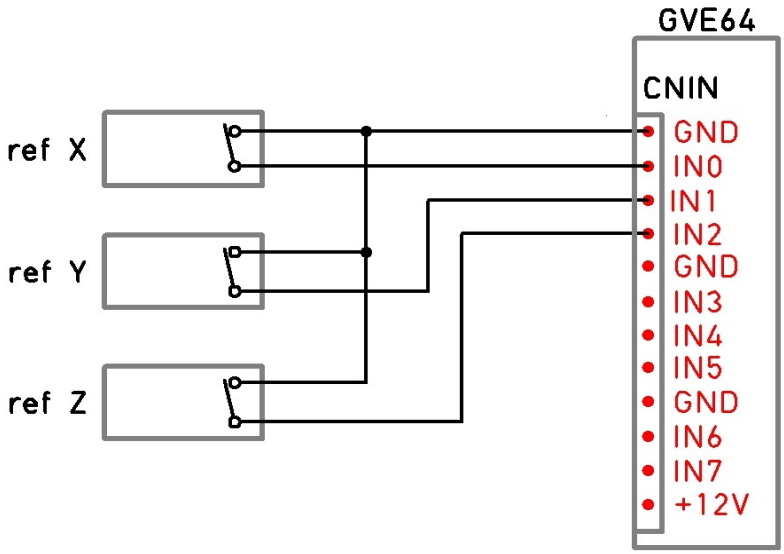
Ref. Spínače jsou rozpínací, aby při poškození kabelu došlo k zastavení stroje.



9.2.2 Připojení mechanických spínačů

(pro referenční spínače v systémech GRAVOS-ARMOTE)

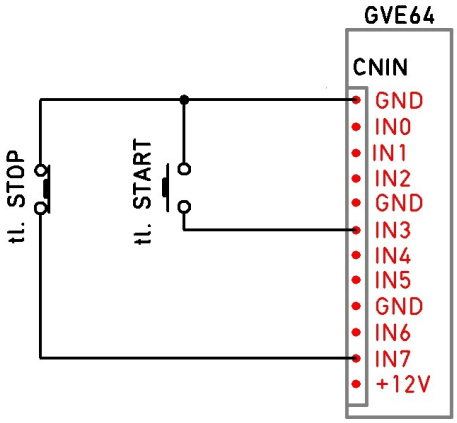
Ref. Spínače jsou rozpínací, aby při poškození kabelu došlo k zastavení stroje.



9.2.3 Připojení tlačítek START s STOP

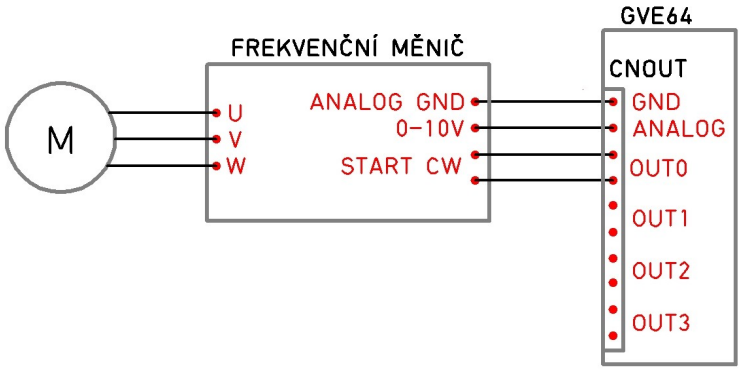
(pro spuštění a zastavení obrábění v systémech GRAVOS-ARMOTE)

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

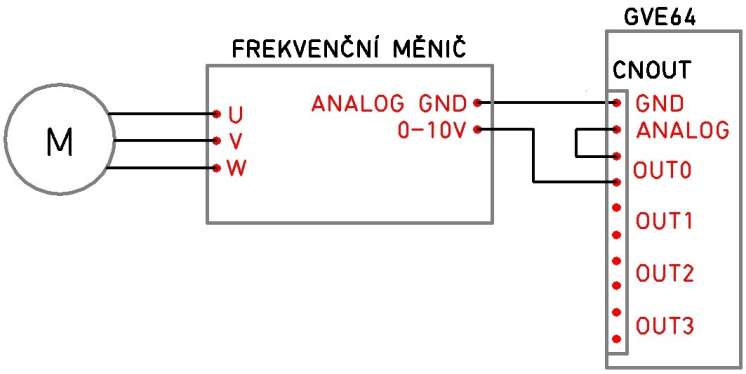


9.3 Zapojení výstupů (CNOUT):

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

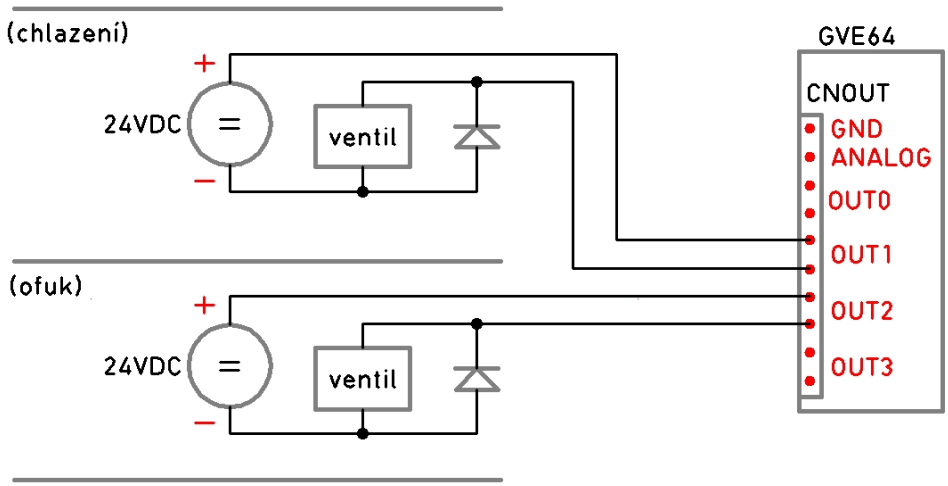


9.3.2 Připojení frekvenčního měniče pouze se signálem 0-10V



9.3.3 Připojení elmag. ventilů

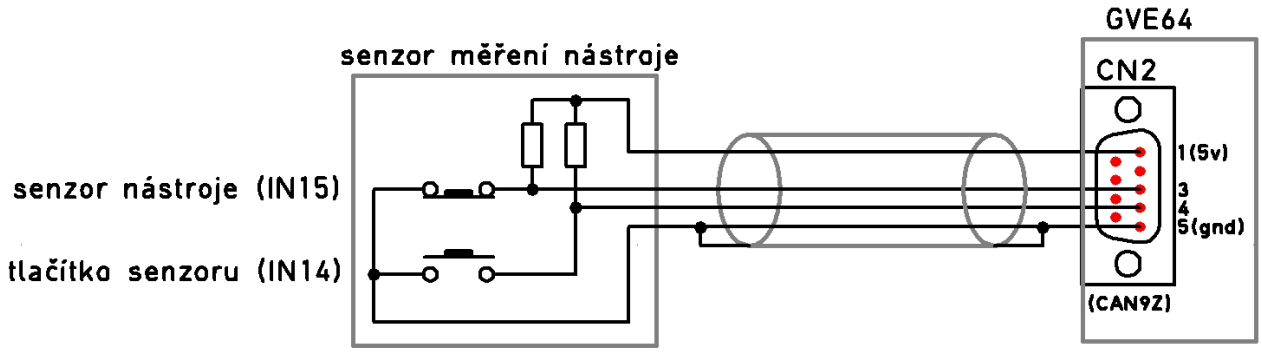
(pro spínání chlazení a ofuku nástroje v systémech GRAVOS-ARMOTE)



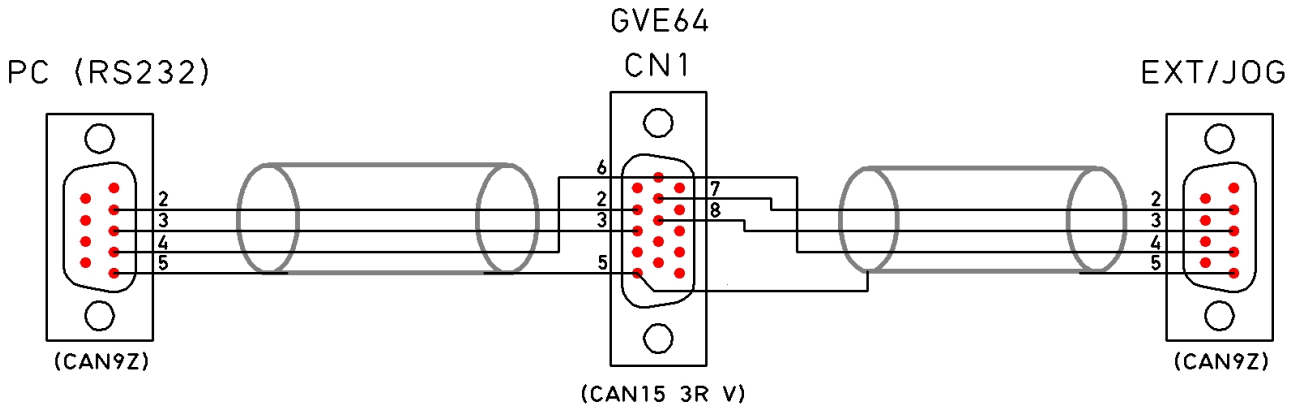
9.4 Připojení senzoru měření nástroje (CN2)

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

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



9.5 Kabel k připojení GVE64 a dalších zařízení k PC (CN1)



9.6 Napájení, (CNSUP)

Napájecí napětí
9 – 24VDC

Doporučený zdroj
MB120D030 (součástí dodávky)
12V, 300mA

10 Přepínače

10.1 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

10.2 Přepínač ANALOG/PWM

Pomocí jumperu lze přepnout funkci výstupu pro ovládání vřetene

V poloze ANALOG je mezi vývodem ANALOG a GND konektoru CNOUT 0 – 10V pro řízení frekvenčního měniče. V poloze PWM je na výstupu ANALOG obdélníkový signál PWM v rozsahu 0,0% až 100,0% včetně, amplituda 5V, kmitočet 7,3728 kHz. Použití pro řízení otáček vřetene při náhradě SW řídících systémů s LPT portem.

Výstup PWM lze dále přepínat na standardní PWM signál nebo na signál PWM pro oddělovací desky (breakout boards) na adrese 0x04

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	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	uvolění 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 řídících 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

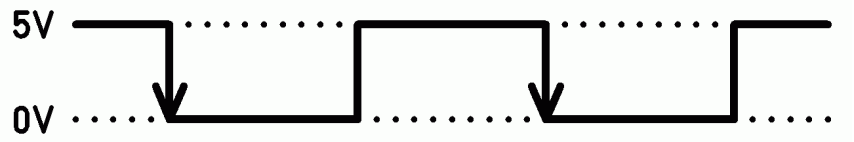
hodnota	funkce
0x01	Osa A je sdružená s osou X
0x02	Osa A je sdružena s osou Y
0x03	Osa A je sdružená s osou Z
0x04 - 0xFF	Osa A je samostatná

11.5 Nastavení výstupu signálů STEP/DIR

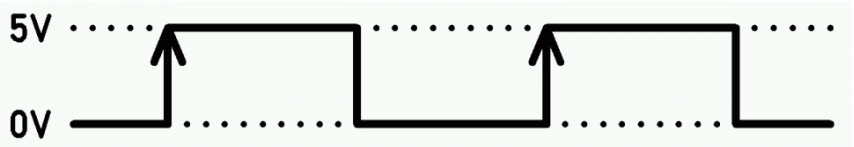
Nastavení výstupu signálů STEP/DIR na konektorech CNX – CNA se provádí na adrese 0x05, lze zde měnit polaritu signálů STEP pro všechny osy najednou a polaritu signálů DIR pro interpolované osy X,Y,Z,A samostatně. Změnou polarity signálu DIR se mění směr osy. Hodnoty jsou v hexadecimálním tvaru, defaultně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

bit 0 = polarita signálu DIR osy X
bit 1 = polarita signálu DIR osy Y
bit 2 = polarita signálu DIR osy Z
bit 3 = polarita signálu DIR osy A
bit 4 = polarita signálu STEP (pro všechny osy najednou)
bit 5 – 7 = nepoužito

Polarita signálu STEP při bit 4 = 1 (defaultně)



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 použí konfigurační utilitu GVE64 Config v1.2, při použití sw Armote si program sám hlídá společné parametry, aby byly shodné v jednotce GVE64 i v nastavení programu. Pokud je MPG vypnuto na adrese 0x06, kontrolu parametrů Armote neprovádí.

12.1 Adresy EEPROM pro vstupy a MPG

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]
0x50	Rychlost ke spínači X pro ref. pohyb, float, [mm/s]
0x54	Rychlost ke spínači Y pro ref. pohyb, float, [mm/s]
0x58	Rychlost ke spínači Z pro ref. pohyb, float, [mm/s]

12.2 Hodnoty nastavení MPG Enable

Hodnota	funkce
0x01	MPG povoleno
0x02	MPG povoleno s omezením limit
jiná	MPG nepovoleno

12.3 Hodnoty nastavení Ref. Start Num

Vstup, kterým se aktivuje referenční pohyb (hledání spínače)

Vstup musí být aktivován déle než 1s a pro správnou funkci musí být správně nastavena polarita a povolení přerušení. Pokud není povoleno MPG nebo byl použit příkaz Y-, jsou reference zakázány. Osa, která má referovat, se nastavuje na přepínači os ručního ovladače MPG.

Pro nalezení spínače se použije maximální dráha osy (horní limit – spodní limit) a Rychlost ke spínači konkrétní osy. Od spínače je rychlost 1mm/s a max vzdálenost 10mm

hodnota	funkce
3.....7	**Číslo vstupu pro start Ref pojezdu**
Jiné hodnoty	**Reference nepovoleny**

12.4 Nastavení polarity a povolení přerušení vstupů

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

12.5 Hodnoty nastavení limit

Při použití sw Armote je třeba aby všechny spodní limity byly 0.
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**
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**

12.6 Nastavení počtu kroků na mm

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**

12.7 Hodnoty nastavení pohybu pro MPG

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]

12.8 Nastavení prodlevy MPG

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 brzdit osu po rampě bez ohledu na počet kroků o který bylo kolečko pootočeno, pokud stroj nestihnul fyzicky všechny kroky vykonat.

12.9 Nastavení defaultních otáček vřetene

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

12.10 Nastavení rychlosti ke spínači ref. Pojezdu

Pokud jsou hodnoty nastaveny na 0, tak ref pojezd je nepovolen

Adresa	hodnota
0x50	Ke spínači X, 4 byty float, max ½ interpolační rychlosti, [mm/s]
0x54	Ke spínači Y, 4 byty float, max ½ interpolační rychlosti, [mm/s]
0x58	Ke spínači Z, 4 byty float, max ½ interpolačního rychlosti, [mm/s]

13 GVE64 – popis vnitřních instrukcí

(pro tvorbu vlastních uživatelských aplikací)

13.1 CPU

procesor ARM7 32bit

13.2 Program

IP64MPG v19 10.9.2010 (c) Gravos (P.Borovsky)

13.3 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

13.3.1 Komunikace

je čistě simplexní, t.j.: nadřízený 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 vyjímkou příkazu Halt, kdy odpoví až po zabržděn.

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, pojede se úlně 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
Výpadá to složitě, ale není. Pro člověka takové výpočty moc nejsou, ale pro SW to představuje pár řádků.

13.3.3 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 odeslaní vznikne časová "díra" - je to označované jako Latence, býva od 1 do 16ms. A při příjmu odpovedi 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 vetšinou 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 učelné dávat jen příkazy typu C (cont.line) a B (brake), ze kterých se vytváří mapa brždění v koncových bodech vektoru.

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).

Hlavně začátečníkům doporučujeme k fréze/manipulátoru, počítač se skutečným(i) sériovým portem, třeba i starší. Dnes se lidi 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.
To se nám stávalo, když byl venku mráz, a tudíž velmi nízká vlhkost vzduchu, byť bylo vše řádně uzemněné.
Gravos takový převodník dodává, jsou na něm rychlé optočleny a pod nimi 4mm izolační mezera (případně si můžete podobný zhotovit). Je léta ověřený, spolehlivý.

13.4 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/s2
V1000	- max.rychlost 1000 kr/s
$512	- short vektory jsou menší než 512 kroků
B35000	- bez omezení rychlosti mezi cont.vektory (pro 35000 hz verzi)
N	- čítač polohy vynulován
O0,FF	- výstupy vypnuté
Y&	- MPG zapnuto bez posílání informací o stavu MPG

13.5 Příkazy

! Adresa Příkaz [parametry] Enter
mezi jednotlivými parametry je čárka
Jednotka se chová jako 2 jednotky najednou, má 2 pevné adresy, na kterých reaguje na povely.

Adr.0 = interpolační jednotka
Adr.7 = jednotka ovládání vřetene

13.5.1 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

13.5.2 Zadávání pohybových vektorů

Lx,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.

Cx,y,z,a - CONT.LINE pokračující vektor (přímka)
x = počet pulsů v ose X v rozsahu -2147483647 az 2147483647
y = počet pulsů v ose Y v rozsahu -2147483647 az 2147483647
z = počet pulsů v ose Z v rozsahu -2147483647 az 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 nadřízený 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 nasledně 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 seriové lince)

Tn - TIME prodleva mezi nenavazujícími 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 5ms
Příkaz je modální, platí až do zadání jiné hodnoty.

$n - 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 brzdé 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.

13.5.3 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 125000 (200000) pulsů/s
např.: !0V10000 - rychlost 10000 pulsů/s
Příkaz je modální, platí až do zadání jiné hodnoty.

VLx,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 - 125000 (200000)
Y = max. rychlost v tis. pulsů/s osy Y v rozsahu 10 - 125000 (200000)
Z = max. rychlost v tis. pulsů/s osy Z v rozsahu 10 - 125000 (200000)
A = max. rychlost v tis. pulsů/s osy A v rozsahu 10 - 125000 (200000)
např. !0VL25000,25000,30000,10000

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 125000 (200000) 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.

13.5.4 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 nadřízené PC s ohledem na směr pohybů jednotlivých os a vzdálenost od osy rotace
nebo při velkém rozdílu převodu os.

13.5.5 Změny rychlosti během pohybu

XA - EXCHANGE změní rychlosti a zrychlení u všech vektorů ve frontě
na poslední zadanou rychlost V.

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

XD - EXCHANGE DN změní rychlosti a zrychlení 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.

13.5.6 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 ALL 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

13.5.7 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

13.5.8 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 možne zadávat vektory.

13.5.9 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 transparenci vektorů po přerušení.
Např.: Obsluha zastaví obrabění 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 - nezatracovat 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.

13.5.10 Obsluha paměti EEPROM

ERn - READ BYTE přecte byte z EEPROM na adrese n a pošle jej po sériové lince
n = 00..FF(hex)

EFRn - READ FLOAT přecte float z EEPROM na adrese n a pošle jej po sériové lince
n = 00..FF(hex)

EURn - READ UINT32 přecte 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)

13.5.11 Ovládání relé

O0,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): !0O0,FB
vypnutí všeho: !0O0,FF

13.5.12 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 tlačítko Start)
bit 4 = CNIN - IN4 Intr4 (v kontrolérech Gravos EndZ)
bit 5 = CNIN - IN5 Intr5 (v kontrolérech Gravos EndY)
bit 6 = CNIN - IN6 Intr6 (v kontrolérech Gravos EndX)
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 sensoru)
bit 7 = CN2 - pin 3 Intr15 (v kontrolérech Gravos 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 řídící SW věděl.
Libovolný Intr není nutné použít, (lze zamaskovat)
a je ho možno použít jako obecný vstupní bit.

SR6 - STATUS READ prečte 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 zpracovávany vektory
INTA=1 RUN=0 ;bylo přerušeno, při brždění vektory doběhly
INTA=1 RUN=1 ;bylo preřušeno, zbytek vekorů 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

13.5.13 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 nadřízenému
počítači přijde místo odpovědi nějaký nesmysl.

J - JUMP na RESET zresetuje včetně vynulování čítače polohy

13.6 Ovládaní vřetene adr.7:

Vn - VELOCITY:
Ovládání analogového/PWM výstupu 0,0...10V 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 .

13.7 Odpovědi

1 hexadecimální znak 0..F [další vyžadované parametry] Enter(0Dh)
mezi jednotlivými parametry je čárka.
Odpoveď je odeslána ihned po zadání příkazu.
Jedině pro příkaz HALT je odpoveď 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 = neznamý 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

14 Rozdíly ve verzích jednotky

14.1 Rozdíly verzí firmware

Kapitola 1 - Specifikace, velikost bufferu 420 cont. vektorů a max. výkon interpolace 125 000 pulzů/s platí pro jednotky verze IP64 v9 a vyšší.

Kapitola 11.2 - Funkce výstupu Signalizace přerušení je v jednotkách verze IP64MPG v10 a vyšší.

Kapitola 11.3 - Přepínání funkce PWM je v jednotkách verze IP64 v3 a vyšší.

Kapitola 11.4 - Funkce sdružení osy A je v jednotkách verze IP64MPG v10 a vyšší.

Kapitola 12 - Nastavení vstupů je platná jen pro jednotky verze IP64MPG v10 a vyšší.

Kapitola 13 - Příkazy YOG, SWITCH, READ FLOAT, READ UINT32, WRITE FLOAT a WRITE UINT32 jsou v jednotkách verze IP64MPG v10 a vyšší.

- Příkazy Velocity Limit, Velocity corection, Exchange multiplier, Null Axis jsou v jednotkách verze IP64MPG v19 a vyšší.

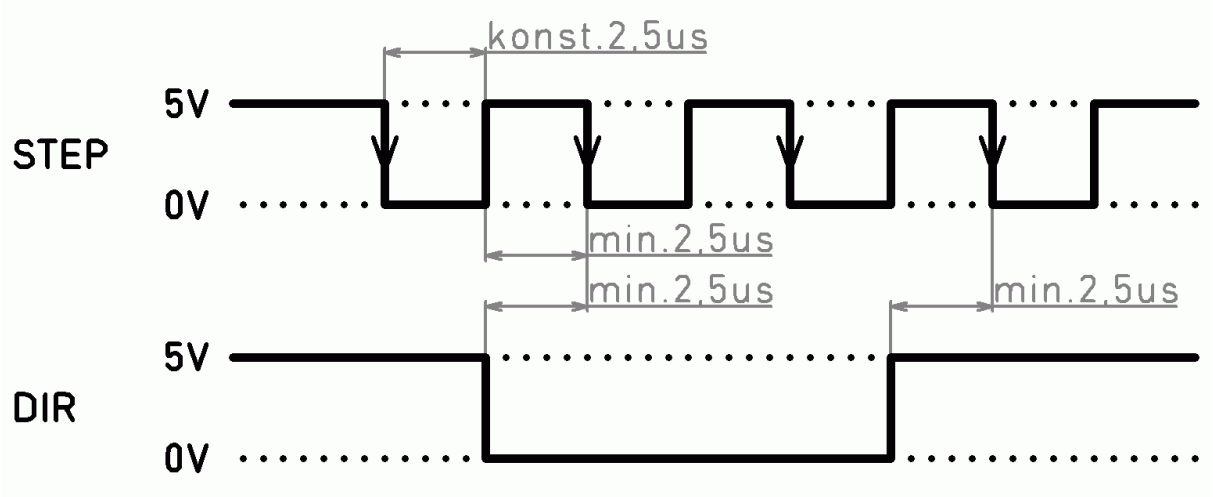
14.2 Odlišnosti verze GVE64 – PHANTOM

Kapitola 1 - Max. Výkon interpolace 200 000 pulzů/s

Kapitola 11.5 - Nastavení výstupu signálů STEP/DIR, není funkční nastavení polarity signálu STEP. Aktivní je sestupná hrana signálu.

Kapitola 9.1.2 - Časování signálů KROK a SMĚR není platné, platný je následující diagram

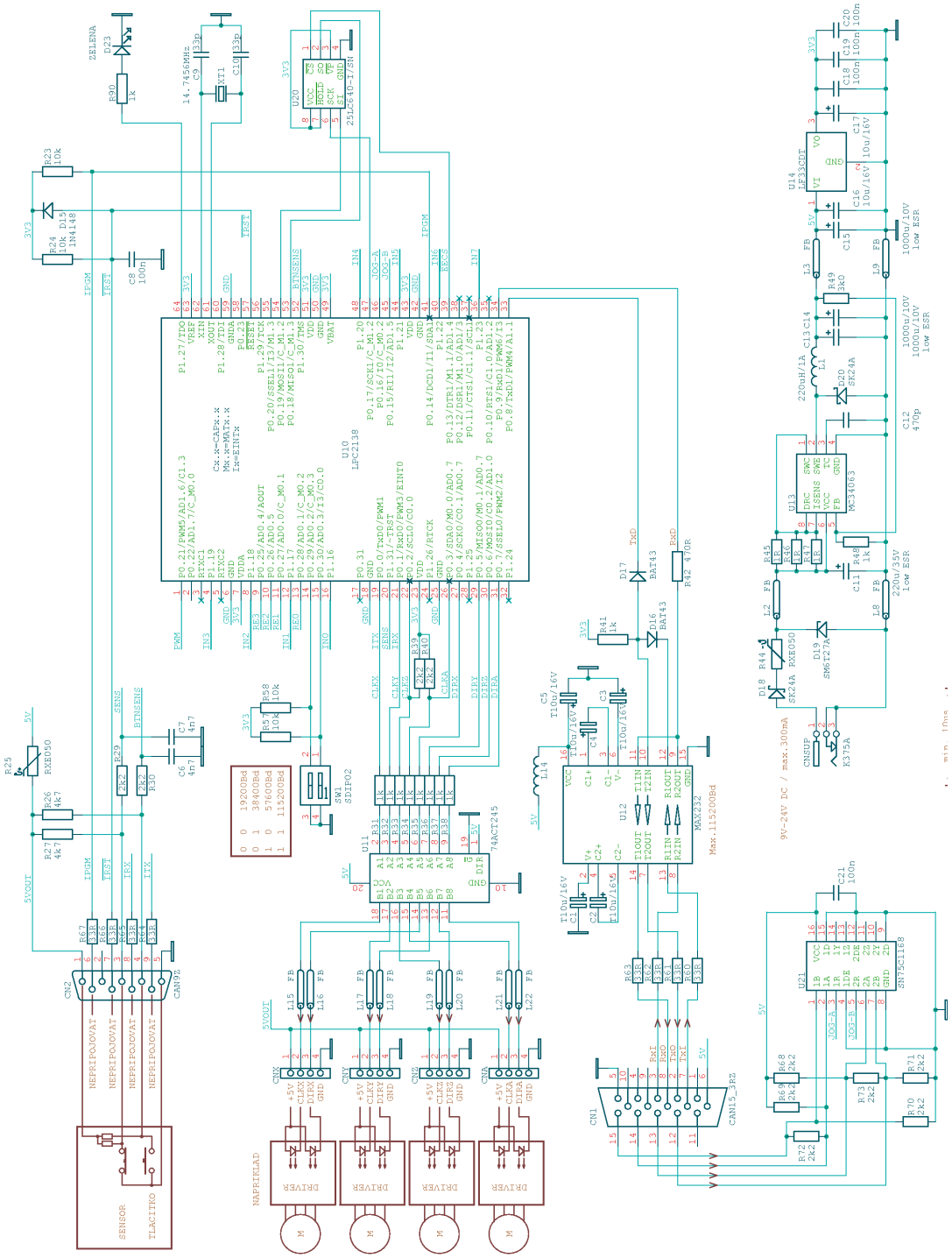
Časování signálů STEP a DIR verze PHANTOM



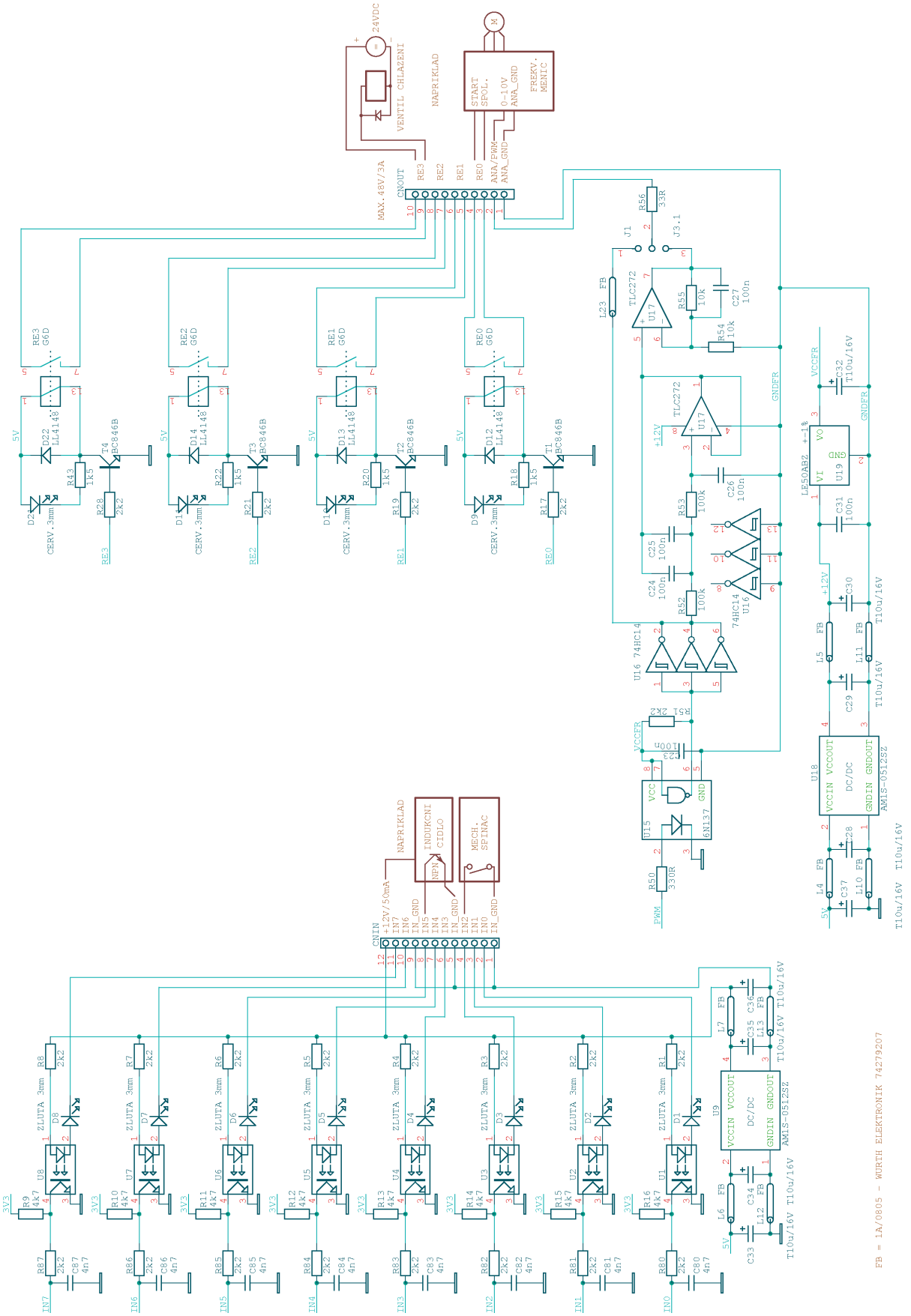
(šipka značí aktivní hranu, verze PHANTOM neumožňuje změnu polarity signálu STEP)

15 Schéma jednotky

15.1 CPU



15.2 I/O


24VDC
VENTIL CHLAZENI
NAPRIKLAD
START SPOL.
0-10V
ANA_GND
FREKV. MENIC
ANA/PWM
ANA_GND
MAX.48V/3A
CNOUT
RE3
RE2
RE1
RE0
J1
J3.1
R56 33R
TLC272
U17
L23 FB
R55 10k
C27 100n
R54 10k
GNDFR
VCCFR
L6502AB2
U19
C32 T10u/16V
C31 100n
C30 T10u/16V
L5 FB
L11 FB
C29 T10u/16V
U18
DC/DC
AMIS-0512S2
VCCIN VCCOUT
GNDIN GNDOUT
C28 T10u/16V
L4 FB
C37 T10u/16V
L10 FB
+12V
5V
3V3
74HC14
U16
R53 100k
R52 100k
C25 100n
C24 100n
C26 100n
R51 2k2
C23 100n
U15
6N137
VCC
GND
R50 330R
PWM
NAPRIKLAD
INDUKCNI CIDLO
NPN
MECH. SPINAC
CNIN
+12V/50mA
IN7
IN6
IN5
IN4
IN3
IN2
IN1
IN0
IN_GND
RE3 G6D
RE2 G6D
RE1 G6D
RE0 G6D
D22 LL4148
D14 LL4148
D13 LL4148
D12 LL4148
T4 BC846B
T3 BC846B
T2 BC846B
T1 BC846B
R43 1k5
R22 1k5
R20 1k5
R18 1k5
R28 2k2
R21 2k2
R19 2k2
R17 2k2
D23 CERV.3mm
D15 CERV.3mm
D16 CERV.3mm
D9 CERV.3mm
ZLUTA 3mm
D8
D7
D6
D5
D4
D3
D2
D1
U8
U7
U6
U5
U4
U3
U2
U1
R9 4k7
R10 4k7
R11 4k7
R12 4k7
R13 4k7
R14 4k7
R15 4k7
R16 4k7
R3 2k2
R7 2k2
R6 2k2
R5 2k2
R4 2k2
R3 2k2
R2 2k2
R1 2k2
R87 2k2
R86 2k2
R85 2k2
R84 2k2
R83 2k2
R82 2k2
R81 2k2
R80 2k2
C87 4n7
C86 4n7
C85 4n7
C84 4n7
C83 4n7
C82 4n7
C81 4n7
C80 4n7
U9
L7 FB
L13 FB
C36 T10u/16V
C35 T10u/16V
L12 FB
C34 T10u/16V
C33 T10u/16V
L6 FB
FB = 1A/0805 - WURTH ELEKTRONIK 74279207

Obsah

Poslední změna 16.09.2010

WWW.GRAVOS.CZ