verze 1.1f

Program Grbl je volně dostupný interpret G-kódů (podmnožiny standardu RS274) a zároveň ovladač krokových motorů pro CNC stroje, určený pro platformu Arduino.
Je určen především těm, kteří chtějí s minimálními náklady ovládat vlastní CNC stroj poháněný krokovými motory, ale také těm, kteří Grbl chtějí použít jako základ pro jiný projekt a tím pádem ocení modularitu programu.
S programem Grbl je možno komunikovat rozhraním USB nebo sériovým portem Arduina a nevyžaduje od nadřízeného počítače žádné zvláštní vlastnosti ani výkon.
Díky využití všech vlastností mikrokontroléru ATMega328 program umožňuje přesné časování výstupních signálů při krokovací frekvenci do 30 kHz. V programu je zapracován inteligentní způsob akcelerace a decelerace rychlosti pohybu jednotlivých os, takže je možno u strojů dosáhnout vyšší pracovní rychlosti, aniž by docházelo ke ztrátám pozice při náhlých změnách směru pohybu. Program plně podporuje lineární a kruhovou interpolaci.
Většinu parametrů programu je možno nastavovat i za běhu, konfigurace zůstává uložena v paměti EEPROM.

Technická omezení

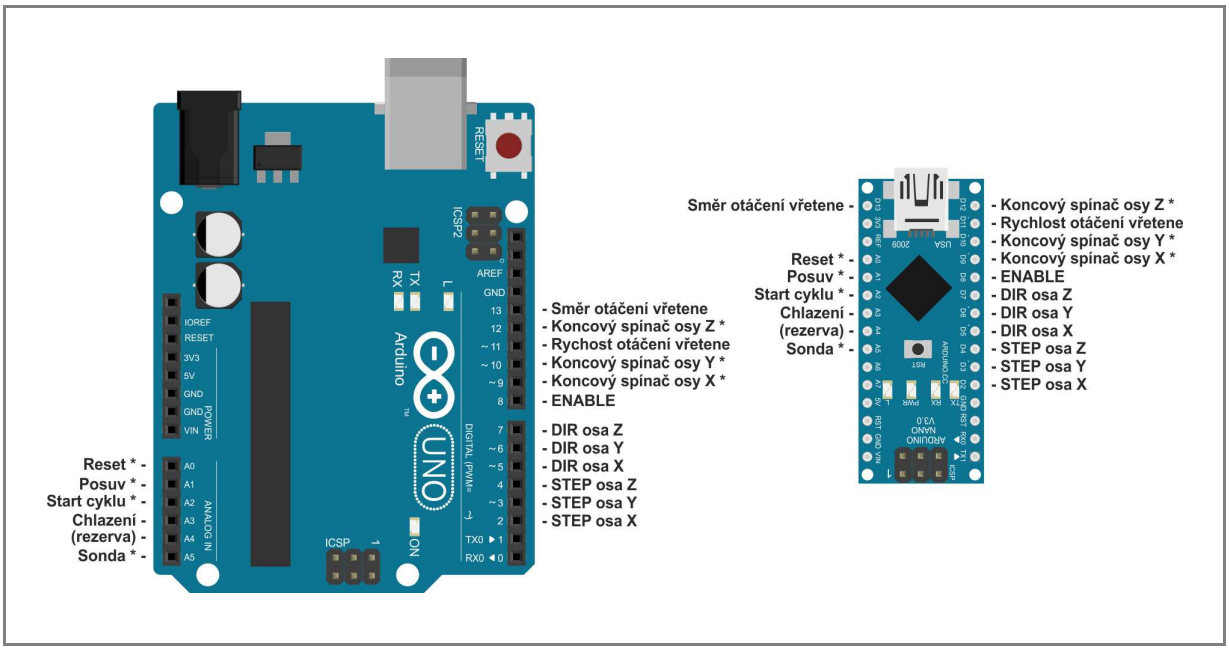
Grbl umožňuje pohyb jen ve třech osách souřadného systému X, Y, a Z. S rotačními osami pracovat prozatím neumí. Množina zpracovávaných G-kódů je záměrně omezena jen na základní příkazy; rozšířené příkazy G-kódu, u kterých se nedá předpokládat, že budou moci být použity pro jednoduché CNC obráběcí stroje, jsou vypuštěny.

Program Grbl ve verzi 1.1f pracuje s těmito G a M kódy:

G00 – Servisní pohyb (Rychlé polohování)
G01 – Lineární interpolace
G02 – Kruhová interpolace ve směru hodinových ručiček
G03 – Kruhová interpolace proti směru hodinových ručiček
G04 – Pauza (až 6 sekund)
G10 L2 – Změna posunutí souřadného systému
G10 L20 – Změna posunutí souřadného systému
G17 – Volba pracovní roviny X-Y
G18 – Volba pracovní roviny X-Z
G19 – Volba pracovní roviny Y-Z
G20 – Nastavení jednotek souřadného systému na palce (inch)
G21 – Nastavení jednotek souřadného systému na milimetry
G28 – Nájezd do referenční polohy
G28.1 – Nastavení souřadnic referenční polohy
G30 – Nájezd do další referenční polohy
G30.1 – Nastavení souřadnic další referenční polohy
G38.2 až G38.5 – sonda
G40 – Vypnutí poloměrové korekce nástroje
G43.1 – Dočasná změna délkového posunutí nástroje
G49 – Vypnutí posunutí nástrojů
G53 – Zrušení souřadného systému obrobku, přepnutí na souřadný systém stroje
G54 až G59 – Volba souřadného systému obrobku
G61 – Režim řízení trajektorie
G80 – Zrušení polohování včetně pevných cyklů
G90 – Absolutní režim odměřování
G91 – Inkrementální (přírůstkový) režim odměřování
G91.1 – Parametry IJK definují polohu středu oblouku relativně k počátečnímu bodu.
G92 – Nastavení lokálního počátku souřadného systému (od aktuální pozice nástroje)
G92.1 – Zrušení lokálního počátku souřadného systému
G93 – Posuv v jednotkách času
G94 – Posuv v délkových jednotkách

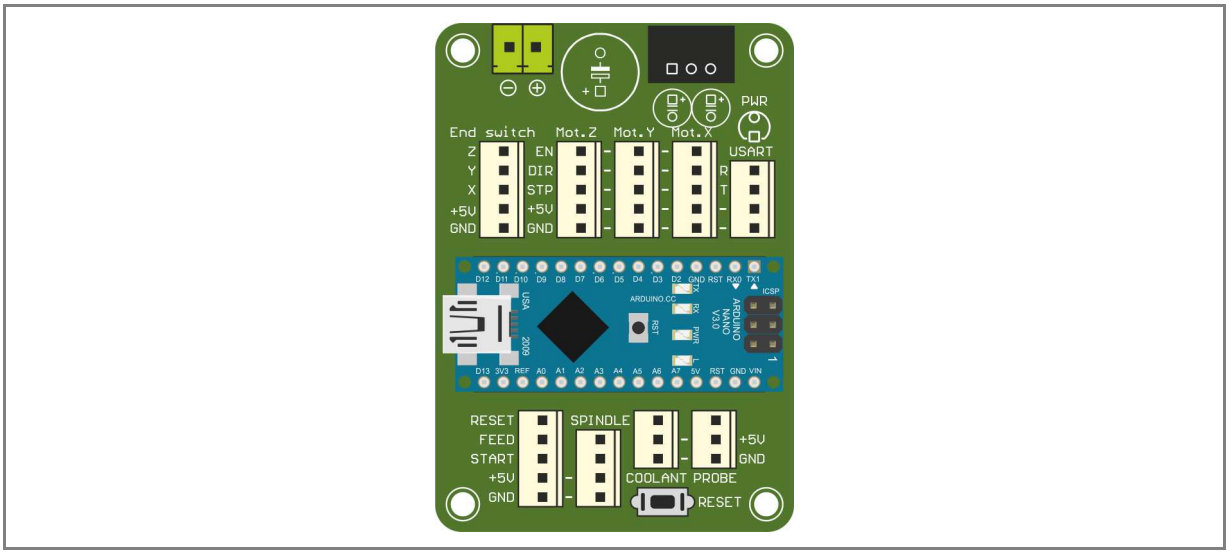
M0 – stop programu
M2 – konec programu
M3 – spuštění vřetene vpravo
M4 – spuštění vřetene vlevo
M5 – stop vřetena
M8 – spuštění chlazení
M9 – zastavení chlazení
M30 – konec programu (reset)

Připojení driverů krokových motorů

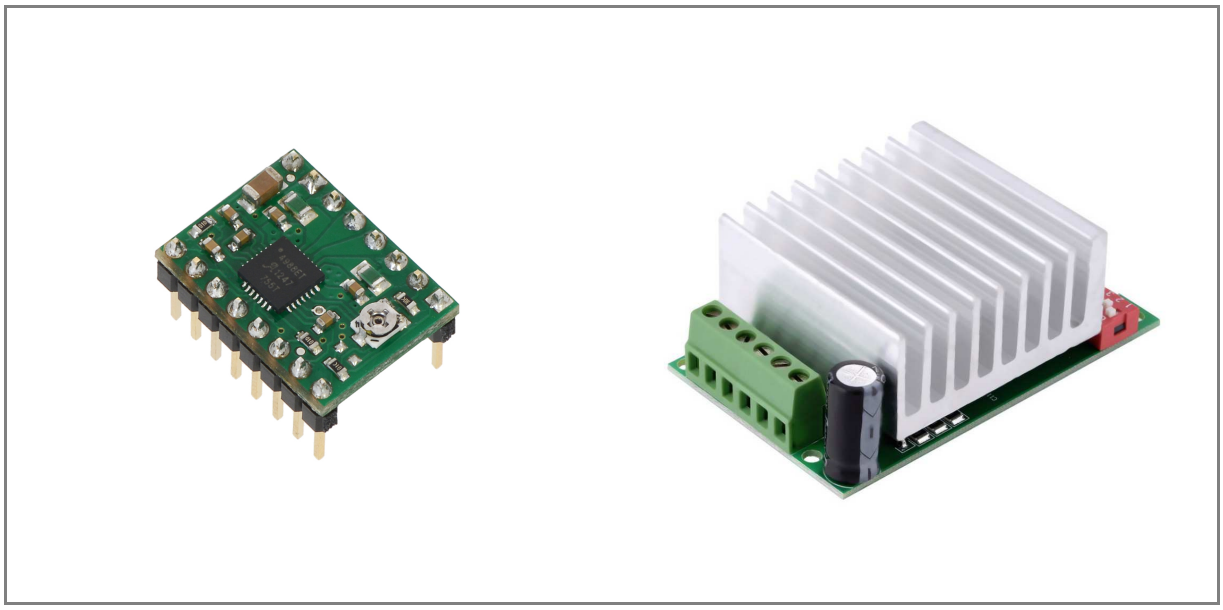


Obr. 1 – rozmístění a funkce pinů mikrokontroléru.
Hvězdičkou označené piny jsou vstupní a je na nich programově zapojen pull-up rezistor. Přesto je doporučeno použít ještě externí pull-up rezistor o hodnotě 4k7 až 10k

Standardní ovladače krokových motorů (drivery) jsou řízeny signály na vstupech STEP a DIR. Impuls, přivedený na vstup STEP pootočí krokovým motorem o jeden krok, logická úroveň na vstupu DIR určuje, kterým směrem se motor bude točit. Některé z driverů jsou vybaveny ještě vstupem ENABLE, jenž umožňuje odpojit buzení motorů.



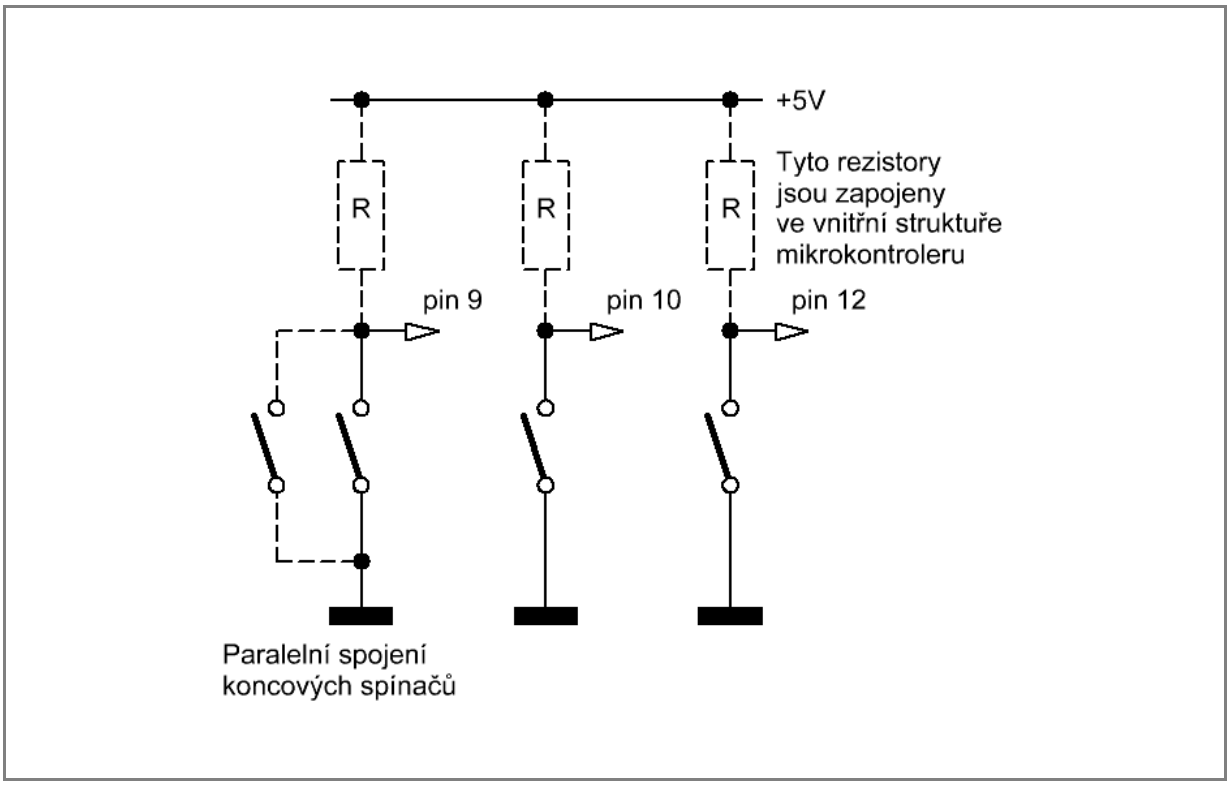
Obr. 2 – Specializovaná základní deska pro Arduino NANO výrazně zjednoduší spojování s drivery a spínači



Obr. 3 – Příklad ovladačů krokového motoru (drivery)

Signály STEP pro osy X, Y, a Z jsou na Arduinu s programem Grbl k dispozici na digitálních výstupech 2, 3 a 4, signály DIR na výstupech 5, 6 a 7. Signál ENABLE je pro všechny drivery společný a je vyveden na výstup 8.

Piny 9, 10 a 12 jsou konfigurovány jako vstupní, jsou na nich programem zapojeny vnitřní zdvihací (pull-up) rezistory a jsou určeny pro připojení koncových spínačů jednotlivých os. Tyto spínače se zapojují proti GND – jejich sepnutím se změní logická úroveň pinu z log. 1 na log. 0. Pokud to vaše konstrukce vyžaduje, můžete zapojit více spínačů paralelně.



Obr. 4 – Zapojení koncových spínačů

Piny 11 a 13 jsou opět výstupní a ovládají rychlost otáčení vřetene (11) a směr jeho otáčení (13). Tyto piny jsou ovládány příkazy, obsaženými v právě prováděném G-kódu, takže jejich činnost je závislá jen na prováděném programu.

Čerpadlo chlazení je ovládáno logickou úrovní na pinu A3. Tento pin je ovládán příkazy, obsaženými v právě prováděném G-kódu, takže jeho činnost je závislá jen na prováděném programu.

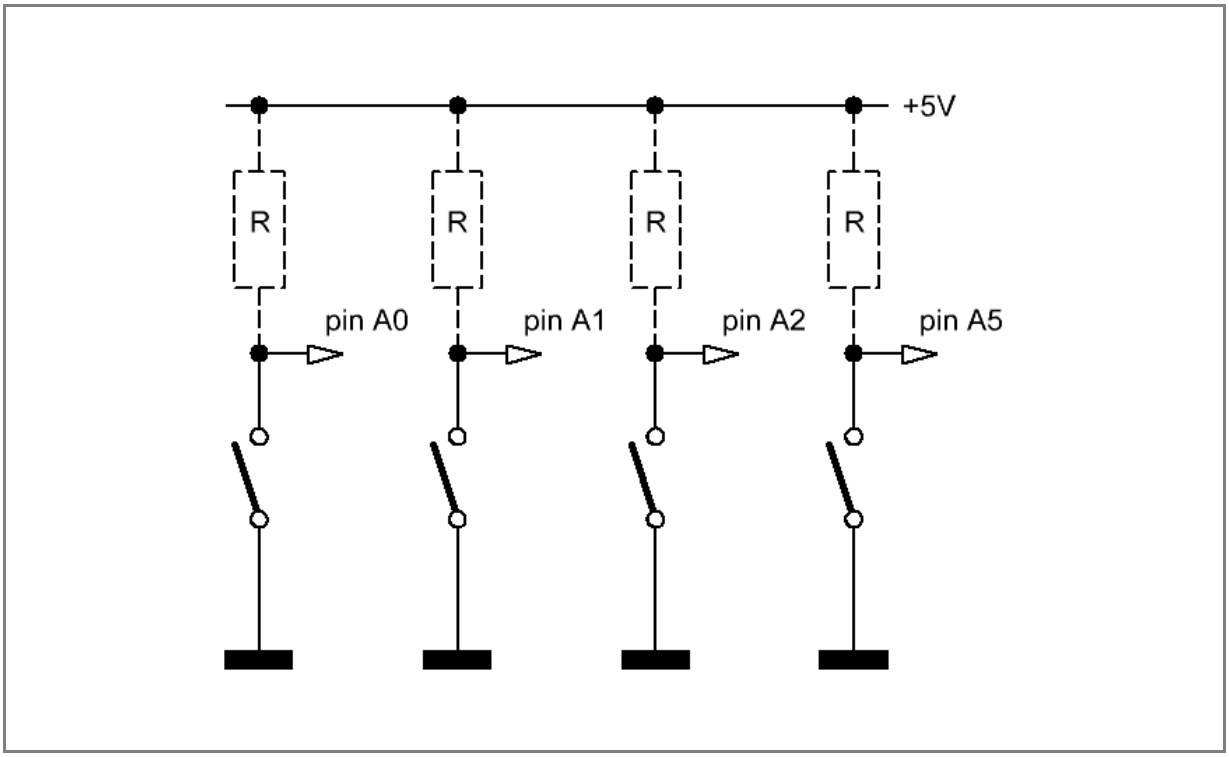
Upozornění:

Na všech výše zmíněných pinech jsou k dispozici jen logické signály, takže pro ovládání vřetene nebo čerpadla chlazení je třeba tyto signály patřičně posílit. Přímé připojení motorů nebo elektromagnetických ventilů by mikrokontroler přetížilo a zničilo.

Piny A0, A1, A2 jsou určeny pro připojení spínačů RESET, POSUV a START CYKLU.

Pin A5 je určen pro připojení kalibrační sondy.

I tyto piny mají programově zapojen pull-up rezistor, postačí tedy spínač zapojit mezi odpovídající pin a GND.



Obr. 5 – Zapojení ovládacích spínačů

Začínáme

Ke Grbl verze 1.1f se můžete připojit pomocí libovolného sériového terminálu komunikační rychlostí 115 200 Bd, (pokud jste ovšem tuto rychlost nezměnili v souboru config.h). Formát dat nastavte na na 8-N-1 (8 bitů, bez parity, 1 stop bit).
Pokud spojení proběhne úspěšně, Grbl vypíše toto úvodní hlášení:

```
Grbl 1.1f ['$' for help]
```

Po stisku klávesy $ a potvrzení klávesou Enter Grbl na obrazovku vypíše tyto zprávy:

```
[HLP:$$ $# $G $I $N $x=val $Nx=line $J=line $SLP $C $X $H ~ ! ? ctrl-x]
ok
$#
[G54:0.000,0.000,0.000]
[G55:0.000,0.000,0.000]
[G56:0.000,0.000,0.000]
[G57:0.000,0.000,0.000]
[G58:0.000,0.000,0.000]
[G59:0.000,0.000,0.000]
[G28:0.000,0.000,0.000]
[G30:0.000,0.000,0.000]
[G92:0.000,0.000,0.000]
[TLO:0.000]
[PRB:0.000,0.000,0.000:0]
ok
$G
[GC:G0 G54 G17 G21 G90 G94 M5 M9 T0 F0 S0]
ok
$I
[VER:1.1f.20170302:]
[OPT:V,15,128]
ok
$N
$N0=
$N1=
ok
?<Idle|MPos:0.000,0.000,0.000|FS:0,0|WCO:0.000,0.000,0.000>
?<Idle|MPos:0.000,0.000,0.000|FS:0,0|Ov:100,100,100>
?<Idle|MPos:0.000,0.000,0.000|FS:0,0>
```

Konfigurační příkazy Grbl

Příkazy, začínající znakem '$' (dolar), mění systémové parametry Grbl.

Poznámka:

Příkazy, které znakem '$' nezačínají, jsou řídící a pracují v reálném čase. Mohou být tedy odeslány kdykoliv, bez ohledu na to, jakou činnost Grbl právě provádí. Tyto příkazy buď okamžitě změní činnost Grbl nebo vynutí odeslání důležitých pracovních údajů, například aktuální polohy.
Tyto příkazy jsou popsány v následující kapitole.

$$

Zobrazení nastavení parametrů Grbl (View Grbl settings)

Napište do okna terminálu **$$** a stiskem **ENTER** příkaz odešlete.
Grbl odpoví vypsáním seznamu aktuálně nastavených parametrů systému, který bude podobný následujícímu příkladu.

```
$0=10
$1=25
$2=0
$3=0
$4=0
$5=0
$6=0
$10=1
$11=0.010
$12=0.002
$13=0
$20=0
$21=0
$22=0
$23=0
$24=25.000
$25=500.000
$26=250
$27=1.000
$30=1000
$31=0
$32=0
$100=250.000
$101=250.000
$102=250.000
$110=500.000
$111=500.000
$112=500.000
$120=10.000
$121=10.000
$122=10.000
$130=200.000
$131=200.000
$132=200.000
ok
```

Všechny nastavené hodnoty parametrů jsou zapsány do paměti EEPROM, takže zůstanou uloženy i po vypnutí napájení.

Poznámka:

Číslování konfiguračních příkazů bylo nutno proti verzi Grbl 0.8 a starším změnit, aby vyhovovalo i budoucím potřebám.

$x

Uložení nastavených parametrů Grbl (Save Grbl setting)

Pokud komunikace proběhne bezchybně, Grbl odpoví OK a uloží všechna nastavení do paměti EEPROM, ve které zůstanou uloženy až do další změny. Povelem **$$** můžete znovu zkontrolovat aktuální systémová nastavení Grbl.

$0

Doba trvání pulzu STEP (Step pulse)

Příklad:

```
$0 = 30
```

Příkaz změní dobu trvání pulsu STEP z 10 (výchozí hodnota) na 30 µs:

Ovladače krokových motorů potřebují pro správnou činnost jistou minimální délku pulsu signálu STEP. Tuto hodnotu naleznete v dokumentaci ovladače, nebo ji zjistíte experimentálně – tím si zároveň prakticky ověříte minimální délku pulsu STEP, kterou je váš ovladač ještě schopen spolehlivě zpracovat.
Pokud ale nastavíte délku pulsů STEP zbytečně velkou, může to působit potíže při vyšších rychlostech posuvu.
Běžné ovladače krokových motorů vyžadují délku pulsu STEP větší než 10 mikrosekund (*µs*).

$1

Délka prodlevy před odpojením motorů (Step idle delay)

Po každém dokončeném pohybu může Grbl změnou logické úrovně na pinu ENABLE odpojit buzení krokových motorů. Příkazem **$1** nastavujete délku prodlevy mezi posledním krokem motoru a odpojením buzení motorů. Většinou je nutné nastavit prodlevu **25 až 50 milisekund**, během níž odezní mechanické kmity stroje, které jsou blokovanými motory účinně utlumeny.

Příklad:

```
$1 = 50
```

Příkaz změní délku prodlevy z 25 (výchozí hodnota) na 50 µs:

Pokud žádnou prodlevu mezi posledním krokem motoru a vypnutím nepotřebujete, nastavte hodnotu **$1 = 0**.

Nastavením **$1 = 255** zůstanou všechny osy trvale aktivní.

$2

Invertování signálu STEP (Step port invert)

Povel k provedení kroku motoru (signál STEP) je představován krátkým pulsem logické úrovně H. Pokud některé ovladače krokových motorů potřebují na vstupu STEP signál s opačnou polaritou, můžete vybraný signál invertovat přímo v Grbl.

Na signál STEP je aplikována logická operace XOR s maskou a výsledek této operace je jako parametr příkazu **$2** odeslán na výstupní port.

Tab. 1: Všechna možná nastavení inverze signálu STEP

Hodnota parametru	Maska	Inverze signálu STEP osy X	Inverze signálu STEP osy Y	Inverze signálu STEP osy Z
0	00000000	ne	ne	ne
1	00000001	ano	ne	ne
2	00000010	ne	ano	ne
3	00000011	ano	ano	ne
4	00000100	ne	ne	ano
5	00000101	ano	ne	ano
6	00000110	ne	ano	ano
7	00000111	ano	ano	ano

$3

Invertování signálu DIR (Direction port invert)

Směr otáčení motoru (signál DIR) je určen logickou úrovní L nebo H na příslušném pinu. Tento parametr umožňuje invertovat signál DIR zvlášť pro každou osu. Ve výchozím nastavení Grbl předpokládá, že osy se pohybují v kladném směru, je-li signál DIR v nízké logické úrovni a v záporném směru, je-li signál DIR ve vysoké logické úrovni. Pokud se některá z os vašeho stroje pohybuje opačným směrem, invertujte její signál DIR.

Na signál DIR je aplikována logická operace XOR s maskou a výsledek této operace je jako parametr příkazu **$3** odeslán na výstupní port.

Tab. 2: Všechna možná nastavení inverze signálu DIR

Hodnota parametru	Maska	Inverze signálu DIR osy X	Inverze signálu DIR osy Y	Inverze signálu DIR osy Z
0	00000000	ne	ne	ne
1	00000001	ano	ne	ne
2	00000010	ne	ano	ne
3	00000011	ano	ano	ne
4	00000100	ne	ne	ano
5	00000101	ano	ne	ano
6	00000110	ne	ano	ano
7	00000111	ano	ano	ano

$4

Invertování signálu ENABLE (Step enable invert)

V základním nastavení Grbl znamená vysoká úroveň na pinech ENABLE zákaz činnosti driveru, nízká úroveň povolení činnosti.

Pokud váš driver používá pro tuto funkci opačné logické úrovně, invertujte funkci výstupu ENABLE příkazem **$4 = 1**.

Původní funkci obnovíte příkazem **$4 = 0**.

Poznámka:

Aby se tato změna projevila, je třeba Arduino resetovat.

$5

Invertování aktivního stavu koncových spínačů (Limit pins invert)

V klidovém stavu jsou piny pro připojení koncových spínačů drženy ve vysoké logické úrovni vnitřními pull-up rezistory Arduina a nízká logická úroveň na nich znamená, že sepnul koncový spínač příslušné osy.

Při použití rozpínacích kontaktů invertujte funkci příslušných pinů Arduina příkazem **$5 = 1**.

Původní funkci obnovíte příkazem **$5 = 0**.

Poznámka:

Aby se tato změna projevila, je třeba Arduino resetovat.

$6

Invertování aktivního stavu sondy (Probe pin invert)

Ve výchozím nastavení je pin pro připojení sondy držen ve vysoké úrovni vnitřním pull-up rezistorem Arduina. Nízká úroveň na tomto pinu znamená, že spínač sondy je sepnut.

Pokud vaše sonda pracuje opačně (rozpínací kontakt), můžete invertovat funkci příslušného pinů příkazem **$6 = 1**.

Původní funkci opět obnovíte příkazem **$6 = 0**.

Poznámka:

Aby se tato změna projevila, je třeba Arduino resetovat.

$10

Výběr odesílaných dat (Status report)

Tímto nastavením určíte, která data bude Grbl odesílat po přijetí příkazu "**?**". Ve výchozím nastavení Grbl odešle zpět stav běží / neběží (tuto volbu nelze vypnout), dále okamžitou polohu stroje a pracovní polohu (poloha stroje plus použité offsety).

K dispozici jsou další dvě užitečné volby pro podávání zpráv: obsazení vyrovnávací paměti při sériovém přenosu a obsazení vyrovnávací paměti bloku plánovače.

Poznámka:

Odesílejte jen ta data, která skutečně potřebujete, protože jejich příprava i odeslání spotřebovávají čas a další zdroje Grbl.

Tab. 3: Všechna možná nastavení souboru odesílaných dat

Hodnota parametru	Maska	Okamžitá poloha	Pracovní poloha	Vyrovnávací paměť plánovače	Vyrovnávací paměť sériového portu
0	00000000	ne	ne	ne	ne
1	00000001	ano	ne	ne	ne
2	00000010	ne	ano	ne	ne
3	00000011	ano	ano	ne	ne
4	00000100	ne	ne	ano	ne
5	00000101	ano	ne	ano	ne
6	00000110	ne	ano	ano	ne
7	00000111	ano	ano	ano	ne
8	00001000	ne	ne	ne	ano
9	00001001	ano	ne	ne	ano
10	00001010	ne	ano	ne	ano
11	00001011	ano	ano	ne	ano
12	00001100	ne	ne	ano	ano
13	00001101	ano	ne	ano	ano
14	00001110	ne	ano	ano	ano
15	00001111	ano	ano	ano	ano

$11

Zpomalení při změně směru pohybu (Junction deviation)

Parametr **$11** určuje, jak rychlý může být pohyb po zakřivené (lomené) dráze. Matematický výpočet tohoto parametru je poněkud komplikovanější, ale obecně platí, že vyšší hodnota parametru umožňuje rychlejší pohyb stroje, ale zároveň se zvyšuje riziko ztráty kroku. Při nižších hodnotách parametru bude pohyb hladší, ale pomalejší.

$12

Přesnost oblouků (Arc tolerance)

Grbl při provádění kódu G2 / G3 skládá kružnice a oblouky z mnoha krátkých přímek (segmentů), napojených na sebe pod určitým úhlem. Výchozí délka segmentů je nastavená na hodnotu 0,002 mm **($12=0.002**), což značně přesahuje skutečnou přesnost většiny CNC strojů.

Zmenšením hodnoty parametru příkazu **$12** se zvyšuje tvarová přesnost oblouku, ale zároveň mohou nastat problémy s výkonem motorů.

$13

Okamžitá poloha stroje (Report inches)

Grbl zasílá v reálném čase zprávy o okamžité poloze jednotlivých os stroje, souřadnicích a o poloze sondy a poskytuje tak uživateli zpětnou vazbu.

Ve výchozím nastavení jsou délkové jednotky těchto zpráv nastaveny na milimetry, ale odesláním příkazu **$13 = 1** můžete jednotky změnit na palce (inch).

Příkazem **$13 = 0** přepnete zobrazování zpět na milimetry.

$20

Programové omezení dráhy stroje (Soft limits)

Příkazem **$20 = 1** programové omezení dráhy aktivujete, příkazem **$20 = 0** ho zakážete.

Programové omezení dráhy stroje zabraňuje vyjetí stroje mimo vymezený pracovní prostor a jeho poškození při chybě v programu. Pokud zadáte Grbl správné rozměry pracovního prostoru, bude po přijetí každého nového G kódu ověřovat, zda se koncová poloha pohybu nenachází mimo tento pracovní prostor. Pokud k tomu dojde, Grbl zastaví pohyb stroje, vypne vřeteno i chlazení a spustí systémový alarm. Protože při tomto způsobu zastavení stroje Grbl pohyb ukončí plynule, nedojde ke ztrátě polohy, jako je tomu v případě aktivace koncových spínačů.

Poznámka:

Pokud používáte programové omezení pohybů, musíte před začátkem práce stroje spustit naváděcí cyklus ($22), aby Grbl znal výchozí polohu, ke které se budou následně vztahovat zadané rozměry pracovního prostoru.

$21

Povolení / zakázání funkce koncových spínačů (Hard limits)

Funkce koncových spínačů se aktivuje příkazem **$21 = 1**, zakazuje příkazem **$21 = 0**.

Koncové spínače obou konců jedné osy se připojují paralelně k odpovídajícímu pinu Grbl (*viz obr. 4*).

Jako koncové spínače se většinou používají běžné mikrospínače se spínacím kontaktem. Ke Grbl se připojují k odpovídajícím pinům a GND. Na těchto pinech je již aktivován pull-up rezistor, který zajišťuje jejich správnou logickou úroveň v klidovém stavu.

Pokud koncové spínače po resetu neustále aktivují režim ALARM, protože stroj stojí v nevhodné pozici a některé z nich jsou dosud sepnuté, zakažte funkci koncových spínačů příkazem **$21=0** a příkazem **$X** alarm vypněte.

Druhou možností je zapojení rozpínacího tlačítka do série se společným vodičem všech koncových spínačů a jeho stiskem tyto spínače dočasně odpojit, takže je možno spustit příkazy, potřebné k mechanickému uvolnění aktivovaných koncových spínačů.

Koncové spínače mají bezpečnostní funkci, která zabraňuje poškození stroje při přejetí koncové polohy. Tyto spínače – mechanické, optické, magnetické nebo indukční – musí být umístěny na koncích všech os. Jakmile je kterýkoli z těchto spínačů sepnut, zastaví se okamžitě veškerý pohyb stroje, uzavře se přívod chladicí kapaliny a vypne se vřeteno (samozřejmě jen pokud je jeho ovládání připojeno na Grbl). Protože se krokové motory okamžitě zastaví, pravděpodobně dojde ke ztrátě polohy. Grbl přejde do nekonečné smyčky režimu ALARM, která vám dá možnost zkontrolovat stroj a přinutí vás Grbl resetovat.

$22

Naváděcí cyklus (Homing cycle)

Naváděcí cyklus se povoluje příkazem **$22 = 1**, zakazuje příkazem **$22 = 0**.

Naváděcí cyklus se používá k přesnému nastavení výchozí pozice stroje (*home position*) pokaždé, když Grbl spouští nový pracovní cyklus.

Po spuštění naváděcího cyklu Grbl nejprve zdvihne osu Z do její nejvyšší polohy, aby se předešlo možné kolizi nástroje s překážkami na pracovní ploše a a teprve potom se začnou pohybovat osy X i Y v kladném směru.

Před dokončením naváděcího cyklu jsou všechny pracovní osy uzamčeny a nemohou se pohybovat. Jak bude vysvětleno dále, uzamčení je možno v případě nutnosti zrušit příkazem **$X**. Uzamčení má bezpečnostní funkci, která zabrání neúmyslným kolizím.

Možnost opakovaného nastavení přesné výchozí pozice stroje je velmi důležitá v případě, že došlo při práci stroje k nějakým problémům, například ke ztrátě kroku nebo k výpadku

napájení a stroj tak ztratil informaci o své aktuální pozici. Tuto pozici můžete obnovit opětovným spuštěním naváděcího cyklu a pak pokračovat v práci.

Pokud používáte koncové spínače zároveň jako spínače referenční, musíte je mít velmi dobře upevněny, protože pokud se spínač při sepnutí pohne nebo posune (třeba jen nepatrným propružením držáku) bude se nastavení výchozí pozice pokaždé lišit.

Referenční spínače, určující výchozí pozici, jsou obvykle umístěny na kladných koncích os X, Y a Z.

Poznámka:

V souboru config.h je možno měnit vlastnosti naváděcího cyklu Grbl. Lze například trvale vypnout blokování pohybu před provedením naváděcího cyklu, nastavit pořadí a směr pohybu jednotlivých os a mnoho dalšího.

$23

Změna polohy referenčních spínačů (Homing dir invert)

Ve výchozím nastavení program Grbl předpokládá, že referenční spínače, označující výchozí polohu, jsou umístěny na kladných koncích jednotlivých os.

Pokud musí mít váš stroj referenční spínače umístěné na opačném konci os, můžete směr pohybu os při naváděcím cyklu invertovat stejně, jako se invertují pulsy STEP a DIR.

Tab. 4: Všechna možná nastavení konfigurace referenčních spínačů

Hodnota parametru	Maska	Změna polohy spínače osy X	Změna polohy spínače osy Y	Změna polohy spínače osy Z
0	00000000	ne	ne	ne
1	00000001	ano	ne	ne
2	00000010	ne	ano	ne
3	00000011	ano	ano	ne
4	00000100	ne	ne	ano
5	00000101	ano	ne	ano
6	00000110	ne	ano	ano
7	00000111	ano	ano	ano

$24

Rychlost zpětného pohybu při referencování (Homing feed)

Při naváděcím cyklu nejprve vyhledává svou referenční pozici osa Z, teprve potom následují osy X a Y současně kladným směrem až do okamžiku sepnutí referenčního spínače. Pak se každá osa začne pohybovat pomalu zpět a přesná poloha referenčního bodu je určena okamžikem rozpojením referenčního spínače.

Příkazem **$24** se nastavuje rychlost tohoto zpětného pohybu. Zvolte ji tak, aby byl výchozí bod každé z os lokalizován s nejvyšší možnou přesností.

$25

Rychlost nájezdu do referenční polohy (Homing seek)

Touto rychlostí se budou pohybovat osy při vyhledávání referenčních spínačů. Rychlost nastavte co nejvyšší, ale takovou, aby při nájezdu na referenční spínače nedošlo k jejich deformaci nebo dokonce poškození.

$26

Potlačení zákmitů referenčních spínačů (Homing debounce)

Po sepnutí spínače s mechanickými kontakty vždy dochází po dobu několika milisekund k odskokům a zákmitům těchto kontaktů a tím k jejich několikanásobnému opakovanému sepnutí a rozpojení. Program Grbl pro potlačení zákmitů kontaktu vkládá před zahájení zpětného pohybu při vyhledávání referenční pozice krátkou prodlevu. Nastavte délku trvání této prodlevy takovou, aby zaručeně odezněly všechny zákmity vámi použitého spínače.

Ve většině případů vyhoví hodnota mezi **5 a 25 milisekundami**.

$27

Ochranná vzdálenost referenčního spínače (Homing pull-off)

Protože jeden spínač může být používán zároveň jako koncový i referenční, nesmí být jeho sepnutí při hledání refrenčních bodů vyhodnoceno zároveň jako sepnutí koncového spínače.

Tento příkaz nastavuje vzdálenost v milimetrech, o kterou se musí osa po nalezení referenčního bodu posunout, než začne být spínač znovu aktivní jako koncový.

$30

Maximální rychlost vřetene (Max spindle speed)

Rychlost otáčení vřetene v otáčkách za minutu, odpovídající 100% plnění pwm (255 = 5 V)

$31

Minimální rychlost vřetene (Min spindle speed)

Rychlost vřetene v otáčkách za minutu odpovídající minimálnímu plnění pwm (1 = 0,02 V)

$32

Mód laseru (Laser mode)

Aktivuje/deaktivuje laser mód, při kterém je potlačeno zastavení pohybu při změně rychlosti otáčení vřetene

$100, $101 a $102

Počet kroků motoru na jeden milimetr dráhy (x, y, z)

Do Grbl musíte zadat počet kroků pro posuv každé osy vašeho stroje o jeden mm. Pro výpočet počtu kroků na mm pro jednotlivé osy vašeho stroje musíte znát:

- Posuv v mm na jednu otáčku pohybového mechanismu
- Počet kroků, které vykoná váš krokový motor na jednu otáčku (typicky 200)
- Nastavení mikrokrokování na vašem ovladači (typicky 1, 2, 4, 8, 16, 32, 64 nebo 128).

Použití zbytečně velkého počtu mikrokroků (např. 128) ale může snížit točivý moment a přesnost polohování krokového motoru.
Počet kroků motoru, potřebný pro jeden jeden milimetr posuvu osy vypočtete ze vzorce:

Spmm = (Spr * Ms) / Fpr

Kde:

Spmm je hledaný počet kroků na milimetr posuvu osy (*step per milimeter*)
Spr je počet kroků na jednu otáčku motoru (*step per revolution*)
Ms je počet mikrokroků na jeden základní krok motoru (*microstep per step*)
Fpr je posuv osy v milimetrech na jednu otáčku motoru (*feed per revolution*)

Vypočtěte tyto hodnoty pro každou osu zvlášť a zadejte je jako parametr příkazů $100, $101 a $102 do Grbl.

$110, $111, $112

Nejvyšší rychlost pohybu os (x, y, z max rate)

Těmito parametry nastavíte nejvyšší povolenou rychlost pohybu pro osy X, Y, a Z. Pokud je v některém G kódu volbou F nastavena rychlost vyšší, než těmito parametry povolená, Grbl místo ní použije hodnotu, nastavenou parametrem příkazů $110, $111 nebo $112.

Poznámka:

Tyto parametry ovlivňují zároveň i maximální rychlost při rychlém pozicování (G0).

$120, $121, $122

Zrychlení v osách X, Y a Z (x, y, z accel)

Tímto parametrem nastavujete zrychlení v jednotkách **mm/sec**^{2}. Postačí, když si zapamatujete, že nižší hodnota parametru znamená menší zrychlení, zatímco vyšší hodnota zrychlení větší. Čím větší je hodnota zrychlení, tím dříve dosáhnete požadované rychlosti posuvu, ale tím je také větší nebezpečí ztráty kroku motoru. Nastavte proto hodnotu zrychlení experimentálně tak, aby u žádné z os ke ztrátám kroků zaručeně nedocházelo.

Nastavte tento parametr s dostatečnou rezervou; když dojde ke ztrátě kroků, Grbl to nemůže nijak zjistit, protože krokové motory nemají zpětnou vazbu a tak stroj bude pokračovat v další práci s chybou.

$130, $131, $132

Maximální délka pohybu os X, Y a Z (x, y, z max travel, mm)

Těmito parametry se nastavuje maximální délka dráhy pohybu jednotlivých os v milimetrech. Nastavit tento parametr má smysl jen v případě, že na začátku práce určíte jednotlivým osám výchozí pozice naváděcím cyklem.

Další příkazy, začínající znakem $

Další příkazy začínající znakem $ umožňují uživateli nastavovat ovládací prvky, jako je například zobrazení pracovních souřadnic a offsetů, nastavení pracovních režimů, analyzátoru modálních stavu nebo spuštění naváděcího cyklu.

$#

Zobrazení parametrů G-kódů (View gcode parameters)

Parametry kódů **G54-G59** a **G28 / G30** jsou po každé změně trvale uloženy do paměti EEPROM. Dočasné parametry **G92, G43.1 a G38.2** se do EEPROM neukládají a jsou vynulovány resetem nebo vypnutím napájení.
Na příkaz **$#** Grbl zobrazí aktuálně uložené souřadnice offsetů, korekci délky nástroje (TLO – Tool Lenght Offset) a souřadnice, zjištěné při posledním měření sondou (PRB).

```
[G54:4.000,0.000,0.000]
[G55:4.000,6.000,7.000]
[G56:0.000,0.000,0.000]
[G57:0.000,0.000,0.000]
[G58:0.000,0.000,0.000]
[G59:0.000,0.000,0.000]
[G28:1.000,2.000,0.000]
[G30:4.000,6.000,0.000]
[G92:0.000,0.000,0.000]
[TLO:0.000,0.000,0.000]
[PRB:0.000,0.000,0.000]
```

$G

Zobrazení aktivních režimů analyzátoru G-kódů (View gcode parser state)

Tento příkaz vypíše všechny G-kódy aktivních režimů v analyzátoru G-kódu.

Při odeslání tohoto příkazu Grbl odpoví například takto:

```
[G0 G54 G17 G21 G90 G94 M0 M5 M9 T0 S0.0 F500.0]
```

Tyto režimy určují, jak bude příští blok G-kódu nebo příkaz interpretován Grbl. Režimy nastaví analyzátor do určitého stavu, takže mu nemusíte neustále určovat, jak má příkazy zpracovat. Tyto režimy jsou uspořádány do skupin, zvaných "modální", které nemohou být aktivní ve stejnou dobu. Například modální skupina jednotek určuje, zda je G-kód programu interpretován v palcích nebo milimetrech.

Tab. 5: Seznam modálních skupin podporovaných Grbl

Modální skupiny	**Member Words**
Motion Mode	G0, G1, G2, G3, G38.2, G38.3, G38.4, G38.5, G80
Coordinate System Select	G54, G55, G56, G57, G58, G59
Plane Select	G17, G18, G19
Distance Mode	G90, G91
Arc IJK Distance Mode	G91.1
Feed Rate Mode	G93, G94
Units Mode	G20, G21
Cutter Radius Compensation	G40
Tool Length Offset	G43.1, G49
Program Mode	M0, M1, M2, M30
Spindle State	M3, M4, M5
Coolant State	M7, M8, M9

Grbl kromě režimů analyzátoru G-kódu též vypisuje číslo aktivního nástroje T, otáčky vřetena S a posuv F, které jsou po resetu nastaveny na 0. Tyto parametry nespadají do modálních skupin, ale jsou stejně důležité pro určení stavu analyzátoru.

$I

Zobrazení verze Grbl (View build info)

Zadáním tohoto příkazu se zobrazí verze Grbl a datum sestavení.
Můžete si také příkazem **$I = xxx** (kde xxx je řetězec o maximální délce 80 znaků) uložit krátkou informační zprávu o stroji, pro jehož řízení je tento Grbl nastaven. Zpráva se bude zobrazovat společně s ostatními údaji.

$N

Zobrazení obsahu bloků prováděných po startu (View startup blocks)

$Nx jsou spouštěcí bloky, tedy řada G kódů, které se v Grbl spustí po každém zapnutí nebo resetu Grbl. V současné době jsou v Grbl připraveny dva bloky G kódů pro výchozí nastavení systému.
Napište do okna terminálu příkaz $N a odešlete ho.

Grbl zareaguje krátkou zprávou, například:

```
$N0=
$N1=
ok
```

Znamená to, že ve spouštěcích blocích $N0 a $N1 není uložen žádný G kód, který se má po startu spustit.

$Nx

Vytvoření spouštěcího bloku (Save startup block)

Příkladem typického použití spouštěcího bloku je přestavení souřadného systému z milimetrů na palce kódem **G20**.
Chcete-li vytvořit spouštěcí blok, napište příkaz: **$N0 =** následované blokem platných G kódů a odešlete. Grbl blok spustí a zkontroluje jeho správnost; je-li v pořádku odpoví ok, pokud ne, odpoví error.
Například, pokud chcete do prvního spouštěcího bloku **$N0** vložit **G-kód G54** (volba souřadného systému), **G21** (nastavení jednotek na milimetry) a **G17** (pracovní rovina X Y), zadejte **$N0 = G21 G54 G17** a odešlete; Grbl odpoví **ok**. Pak můžete příkazem **$N** zkontrolovat, zda byl tento blok opravdu uložen do paměti; pokud ano, odpověď bude **$N0 = G21G54G17**.
Jakmile budete mít spouštěcí blok uložen v EEPROM Grbl, po každém startu nebo resetu uvidíte výpis jeho obsahu ihned za úvodním hlášením.

Pokud jste vložili spouštěcí blok podle předchozího příkladu, po startu uvidíte:

```
Grbl 1.1f ['$' for help]
>G20G54G17:ok
```

Pokud používáte více spouštěcích bloků, zobrazí se v úvodním výpisu pod sebou. Chcete-li některý z bloků vymazat, zapište do něj prázdný řetězec **($N0=**).
Pokud je povolen naváděcí cyklus, budou bloky příkazů provedeny až po jeho vykonání, nikoli ihned po spuštění programu a měnit je lze také až po dokončení naváděcího cyklu.

Důležité:

Buďte velmi opatrní při ukládání jakýchkoli příkazů pro ovládání pohybu (G0/1, G2/3, G28/30) do spouštěcích bloků. Tyto příkazy se provedou vždy po startu nebo resetu Grbl, takže pokud máte nouzový spínač připojen na reset, může spuštění takového bloku způsobit nechtěné kolize.

Do spouštěcího bloku také neukládejte žádné příkazy, které zapisují údaje do vnitřní paměti EEPROM Arduina (např. G10 / G28.1 / G30.1). Tato data by se do paměti ukládala při každém spuštění a po každém resetu a paměť EEPROM by se rychle opotřebila.

$C

Ověření G-kódu (Check gcode mode)

Příkaz **$C** přepíná analyzátor G-kódu do kontrolního módu, kdy se všechny příchozí bloky G-kódu zpracují jako v běžném provozu, ale nepohybuje se žádná osa, ignorují se prodlevy a je vypnuto vřeteno i čerpadlo chladicí kapaliny. To uživateli umožňuje zkontrolovat nové programy analyzátorem G-kódu a odhalit případné chyby. Také lze zkontrolovat, zda program nepřekračuje povolené limity rozměru, pokud je to příslušnou volbou povoleno.

Pokud se tato volba režimu vypne, Grbl se po ukončení kontroly automaticky resetuje (soft-reset Ctrl-X).

Poznámka:

Grbl se resetuje ze dvou důvodů. Jednak se tím zjednodušuje správa kódu, za druhé reset brání spuštění stroje s chybným nastavením Grbl. Reset Grbl umožní začít práci vždy v jednoznačně definovaném nastavení.

$X

Zrušení poplachového režimu (Kill alarm lock)

Poplachový režim Grbl je stav, do něhož program přechází při kritické chybě, například při aktivaci koncového spínače nebo při ztrátě polohy. Ve výchozím nastavení, pokud je povolen naváděcí cyklus, přejde Grbl do poplachového režimu ihned po zapnutí Arduina, protože nezná svou polohu. Režim alarm zablokuje všechny příkazy G-kódu, dokud příkazem **$H** nespustíte naváděcí cyklus. Pokud potřebujete poplachový režim zrušit, například proto, aby bylo možno stroj přesunout z koncových spínačů na jinou pozici, použijte příkaz **$X** který poplachový režim ukončí.

Tento příkaz by ale měl být použit jen v nouzových situacích, protože pravděpodobně je ztracena informace o aktuální poloze stroje. Pro uvolnění z havarijní polohy je doporučeno nastavit inkrementální režim G91, pak spustit naváděcí cyklus nebo Grbl resetovat.

$H

Spuštění naváděcího cyklu (Run homing cycle)

Toto je jediný korektní způsob spuštění naváděcího cyklu. Použití jiných příkazů pro nastavení výchozí pozice stroje je podle norem G-kódu nesprávné.

Tip:

Po dokončení naváděcího cyklu, nastavte stroj na střed pracovního prostoru ručně, nebo můžete příkazy G28 nebo G30 předem definovat pozici a příkazem G28.1 (nebo G30.1) ji uložit. Obecně platí, že po ukončení naváděcího cyklu se osy X a Y přesouvají na střed pracovní plochy a osa Z se nastavuje do nejvyšší polohy, takže je minimalizována možnost kolize mezi nástrojem a obrobkem.

$Jx

Spuštění ručního posuvu (Run jogging motion)

Tento příkaz, který vyvolá speciální ruční posuv (jogging motion) je k dispozici od verze Grbl 1.1.
Mezi ručním posuvem a pohybem vyvolaným standardními G-kódy jsou tři hlavní rozdíly.

- Stejně jako běžné G-kódy, mohou být i příkazy pro ruční posuv řazeny do fronty plánovacího bufferu, ale ruční posuv může být snadno přerušen příkazy Zrušit ruční posuv (Jog Cancel) a Pozastavení programu (Feed Hold). Grbl v takovém případě okamžitě zastaví ruční posuv a z bufferu automaticky vymaže zbývající příkazy.
- Příkazy pro ruční posuv jsou zcela nezávislé na stavu analyzátoru G-kódů. Nezmění žádný režim jako je například inkrementální (přírůstkový) režim odměřování G91, takže už nemusíte pamatovat na nastavení zpět na absolutní režim odměřování G90. To snižuje nebezpečí, že bude spuštěn kód při chybném nastavení režimu.
- Je-li aktivováno programové omezení dráhy (soft-limits), příkazy ručního posuvu, které by měly tyto limity překročit, vrátí pouze chybu a nezpůsobí alarm jako normální g-kódy. To usnadňuje použití GUI nebo joysticku.

Používání ručního posuvu vyžaduje specifickou strukturu příkazu:

- První tři znaky musejí být **$J=**.
- Příkazy pro posuv musejí následovat bezprostředně za rovnítkem; fungují jako běžný příkaz G1.
- Rychlost posuvu je použita v jednotkách G94 za minutu, stav G93 se ignoruje.

Povinné parametry:

- XYZ: jedna nebo více os s cílovou hodnotou.
- F: rychlost posuvu.

Poznámka:

Každý ruční posuv tuto hodnotu vyžaduje a ta není považována za modální.

Volitelné parametry:

- Posuv je vykonán podle aktuálního stavu analyzátoru G-kódu G20/G21 a G90/G91. Je-li poslán některý z následujících příkazů, tento stav je přepsán pouze pro aktuální příkaz.

- G20 nebo G21: režim práce v palcích nebo v milimetrech
- G90 nebo G91: absolutní nebo relativní vzdálenosti
- G53: pohyb v souřadném systému stroje.
- V příkazu ručního posuvu nejsou akceptovány žádné jiné G-kódy, M-kódy ani hodnoty parametrů.
- V příkazu jsou povoleny mezery a komentáře; ty jsou při předzpracování vypuštěny.

Příklad:

Před ručním posuvem byly nastaveny aktivní modální stavy **G21** a **G90**.

Následují příkazy:

- $J=X10.0 Y-1.5 způsobí pohyb na X = 10,0 mm a Y = -1,5 mm v pracovním souřadném systému (work coordinate frame, WPos).
- $J=G91 G20 X0.5 způsobí pohyb o +0,5 palce (12,7 mm) na X = 22,7 mm (WPos). Povšimněte si, že příkazy G91 a G20 se použijí jen pro tento jeden příkaz posuvu.
- $J=G53 Y5.0 způsobí pohyb na Y = 5,0 mm v souřadném systému stroje (machine coordinate frame, MPos). Je-li pracovní offset v ose Y 2,0 mm, pak Y je ve WPos 3,0 mm.

Příkazy pro ruční posuv se chovají téměř stejně jako normální sekvence g-kódů. Každý příkaz posuvu je poté, co úspěšně projde parserem a je připraven k vykonání, potvrzen ‚ok'.
Pokud příkaz není správný nebo překročí programové omezení dráhy (soft-limits), Grbl vrátí ‚error:'.
Příkazů pro ruční posuv může být posláno více v řadě.

Poznámka:

V další dokumentaci ručního posuvu naleznete detaily užitečné pro vytvoření joysticku s rychlou odezvou nebo rozhraní s rotačním enkodérem.

$RST=

Obnovení původního nastavení a dat (Restore Grbl settings and data to defaults)

Příkaz **$RST=$** vymaže veškeré změny konfigurace Grbl, které byly provedeny příkazy **$$** a obnoví výchozí nastavení, definované souborem, použitým při sestavování Grbl, umožní tedy po pokusných nastaveních rychle uvést Grbl do výchozího stavu.
Příkaz **$RST=#** vymaže z paměti EEPROM všechny hodnoty offsetů, definovaných příkazy G54-G59 a pozic, definovaných příkazy G28/G30 a a nastaví hodnoty výchozí, definované souborem použitým při sestavování Grbl. Tímto způsobem je možno všechny hodnoty snadno přepsat naráz, aniž by je bylo nutné měnit jednotlivě nastavením pomocí příkazů G20 L2/20 nebo G28.1/30.1.
Příkaz **$RST=*** vymaže a obnoví všechna konfigurační data EEPROM používaná Grbl, která byla nastavena pomocí příkazů $$, $#, $N a $I. Nemaže se pochopitelně celá EEPROM, ale jen oblasti, do kterých GRBL ukládá data.

Poznámka:

Tyto příkazy nejsou uvedeny v nápovědě, zobrazované příkazem $, ale jsou k dispozici uživatelům pro obnovení části nebo celého obsahu paměti EEPROM. Po vykonání každého z těchto příkazů se Grbl automaticky zresetuje, aby byla zajištěna korektní inicializace.

$SLP

Povolení režimu spánku (Enable Sleep Mode)

Příkaz **$SLP** uvede Grbl do režimu spánku, ve kterém přestane reagovat na příkazy, zastaví vřeteno, vypne chlazení a nastavením pinu Enable do stavu OFF odpojí ovladače krokových motorů. Zpět do provozního stavu může být Grbl uveden pouze měkkým resetem (Ctrl-X) nebo spínačem Start cyklu. Po přechodu zpět do provozního stavu zůstane Grbl v poplachovém stavu, aby obsluha mohla před započetím další práce nejprve zkontrolovat stroj a pak teprve poplachový stav zrušit příkazem **$X**.

Na konec vašeho G-kódu je vhodné umístit příkazy, které přesunou stroj do výchozí pozice a pak teprve Grbl uspat.

Příkazy pracující okamžitě:

Tyto příkazy Grbl pracují v reálném čase, což znamená, že mohou být zaslány kdykoli a Grbl je zpracuje a zareaguje okamžitě (obvykle během několika milisekund), bez ohledu na právě probíhající činnost.

~

Start cyklu (Cycle Start)

Příkaz povoluje nebo zakazuje automatický start zaslaných příkazů.

Profesionální stroje pracují tak, že načtou celý program a spustí jej teprve po stisku tlačítka Start cyklu. Grbl umožňuje totéž, ale nikoli ve výchozím nastavení. Při výuce a ladění programů je výhodnější, když se každý příkaz k pohybu začne provádět okamžitě po přijetí. To usnadňuje pochopení funkce příkazů a ladění stroje řízeného Grbl.
V opačném případě by bylo nutné neustále používat tlačítko „Start cyklu“, kdykoli byste chtěli spustit pohyb.

Poznámka:

Příkaz bude Grbl přijat jen v případě, že ve vyrovnávací paměti je uložen program v G-kódu a že není zaneprázdněn jinou činností, například probíhajícím naváděcím cyklem.

!

Pozastavení programu (Feed Hold)

Příkaz **!** přeruší vykonávanou činnost tak, aby nedošlo ke ztrátě pozice stroje. Grbl pak očekává povel ke startu cyklu pro pokračování v činnosti.

Poznámka:

Příkazem **!** *lze pozastavit pouze probíhající pracovní cyklus, na náváděcí a jiné cykly nemá vliv.*
Pokud nehrozí krizová situace, používejte tento příkaz i k předčasnému ukončení probíhajícího cyklu, vyhnete se tím nutnosti znovunastavení výchozí pozice stroje.

?

Okamžitý stav (Current Status)

Příkaz **?** zobrazí aktivní stav Grbl a aktuální pozice stroje v obráběcích i pracovních souřadnicích. Volitelně může Grbl také zobrazit aktuální obsazení vyrovnávací paměti, a paměti plánovače. souřadnice jsou nastavovány podle volby parametru **$13** buď v milimetrech nebo palcích.

Ihned po odeslání příkazu ? Grbl odpoví zprávou, podobnou této:

```
<Idle,MPos:5.529,0.560,7.000,WPos:1.529,-5.440,-0.000>
```

Aktivní stavy Grbl:

Idle – Grbl je připraven přijmout libovolný příkaz.

Run – probíhá cyklus.

Hold – program je pozastaven.

Door – pokud je sepnut spínač bezpečnostních dveří, je pozastaven veškerý pohyb, vypnuto vřeteno a chlazení.

Home – probíhá naváděcí cyklus. Zobrazování pozice je nastaveno na aktuální hodnotu až po dokončení naváděcího cyklu.

Alarm – došlo k závadě nebo Grbl ztratil pozici.

Check – Grbl je v režimu probíhající kontroly G-kódu, zpracovává a reaguje tedy na všechny příkazy G-kódu, ale nepohybuje přitom strojem.

Ctrl-X

Měkký reset (Reset Grbl)

Takzvaný měkký reset ukončuje kontrolovaným způsobem veškerou činnost, takže stroj neztratí pozici. Následně se Grbl znovu restartuje a je možno pokracovat v práci.

Příkaz Ctrl-X je vhodné použít pokaždé před spuštěním nové úlohy. To zaručí, že v paměti Arduina nezůstaly žádné předchozí příkazy a že není nastaven nevhodný režim činnosti.