

Uživatelský manuál

RKM04 S PROTOKOLEM

Příručka uživatele



SofCon,[®] spol. s r.o.
Křenova 11
162 00 Praha 6
tel/fax: +420 235 090 891
E-mail: sofcon@sofcon.cz
www: <http://www.sofcon.cz>

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenes odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.

SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsáno. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.

SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 26.08.2009

Datum posledního uložení dokumentu: 26.08.2009

(Datum vydání a posledního uložení dokumentu musí být stejné)

Upozornění:

V dokumentu použité názvy výrobků, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Obsah :

1.	O dokumentu	6
1.1.	Revize dokumentu	6
1.2.	Účel dokumentu	6
1.3.	Rozsah platnosti	6
1.4.	Související dokumenty	6
2.	Termíny a definice	6
3.	Přehled základních parametrů	7
4.	Popis jednotky RKM04	8
4.1.	Jednotka RKM04	8
4.2.	Komunikační rozhraní	8
4.3.	Paměť	8
4.4.	Vstupy a výstupy	8
4.5.	Led diody	9
4.6.	Příklad chodu krokového motoru	9
5.	Zápis programu	10
5.1.	Navázání komunikace	11
5.1.1.	Formát zpráv protokolu RKM04	11
5.2.	Vybírací okno Motor	12
5.3.	Vybírací okno Typ zprávy	12
5.3.1.	Povel Version	12
5.3.1.1.	Formát povelu	12
5.3.1.2.	Formát odpovědi	12
5.3.2.	Povel Set Command	13
5.3.2.1.	Formát povelu	13
5.3.2.2.	Formát odpovědi	13
5.3.3.	Povel Store Flash	14
5.3.3.1.	Formát povelu	14
5.3.3.2.	Formát odpovědi	14
5.3.4.	Povel Get Position	14
5.3.4.1.	Formát povelu	14
5.3.4.2.	Formát odpovědi	15
5.3.5.	Povel Get Command	15
5.3.5.1.	Formát povelu	15
5.3.5.2.	Formát odpovědi	15
5.3.6.	Povel Get Tabul Run	16
5.3.6.1.	Formát povelu	16
5.3.6.2.	Formát odpovědi	16
5.3.7.	Povel Get Pozic Run	16
5.3.7.1.	Formát povelu	16
5.3.7.2.	Formát odpovědi	17
5.3.8.	Nesprávný povel	17
5.3.8.1.	Formát odpovědi	17
6.	Přehled příkazů	18
7.	Příkazy	20
7.1.	Reset	20
7.2.	Návěští	20
7.3.	Zrychlení	20
7.4.	Pohyb vzad	20

7.5.	Zruš výstup	20
7.6.	Zruš proměnnou	21
7.7.	Smaž čítač polohy	21
7.8.	Změna směru pohybu	21
7.9.	Konec smyčky	21
7.10.	Pohyb vpřed	21
7.11.	Absolutní poloha	22
7.12.	Trvale vpřed	22
7.13.	Trvale vzad	22
7.14.	Výchozí poloha	22
7.15.	Podmíněný skok	23
7.16.	Skok	23
7.17.	Konec pohybu	23
7.18.	Obnovení pohybu	23
7.19.	Stop od BI True	24
7.20.	Stop od BI False	24
7.21.	Reset stop od BI	24
7.22.	Začátek smyčky	25
7.23.	BI True	25
7.24.	Proved' pohyb	25
7.25.	Startovní rychlost	25
7.26.	Nastav výstup	26
7.27.	Nastav proměnnou	26
7.28.	Maximální rychlost	26
7.29.	Změna rychlosti za pohybu	27
7.30.	Čekání	27
7.31.	Total stop od BI True	27
7.32.	BI False	28
7.33.	Limit vpřed	28
7.34.	Limit vzad	28
7.35.	Zapiš absolutní polohu	28
7.36.	Složený pohyb	29
7.37.	Volání podprogramu	30
7.38.	Konec podprogramu	30
7.39.	Program	30
7.40.	Konec programu	30
8.	Funkce parametrizace	31
8.1.	Blokace přepisu zadání programu – parametr 40	31
8.2.	Simulační režim – parametr 41	31
8.3.	Funkce – parametr 42 - 50	31
8.4.	Nastav výstup – parametr 51 - 58	32
9.	Příklady povelové sady	33
9.1.	Příklad 1	33
9.2.	Příklad 2	34
9.3.	Příklad 3	35
9.4.	Příklad 4	36
9.5.	Příklad 5	37
9.6.	Příklad 6	38
9.7.	Příklad 7	39
9.8.	Příklad 8	40

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Autor	Datum vydání	Verze LIB	Verze LIBT	Verze LIBV	Popis změn
1.00	CHW	1.09.2007	05.13			První vydání.
1.01	CHW	6.05.2009	05.13			Přidání několika příkazů
1.02	CHW	28.05.2009	05.14			Změna knihovny, přidání příkazu
1.03	CHW	23.06.2009	05.14			Přidání několika příkazů
1.04	CHW	26.08.2009	05.14			Přidání funkcí T51- T58

1.2. Účel dokumentu

1.3. Rozsah platnosti

Určen pro uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu není potřeba číst žádný další manuál, je však třeba se orientovat v programovém vybavení SofCon.

2. Termíny a definice

Příkaz - jeden definovaný znak s parametrem

Program - sada příkazů

Protokol PRT – komunikační protokol firmy SofCon, spol.s r.o.

3. Přehled základních parametrů

- **PROGRAMOVATELNÁ JEDNOTKA PRO 4 NEZÁVISLÉ MOTORY**
- **RYCHLOST AŽ DO 16 000 KROKŮ/S**
- **DÉLKA DRÁHY AŽ 16 000 000 KROKŮ**
- **VÝKONNÝ SOUBOR PŘÍKAZŮ**
- **PLYNULÉ ZRYCHLENÍ A BRZDĚNÍ**
- **8 UNIVERZÁLNÍCH VSTUPŮ A 8 UNIVERZÁLNÍCH VÝSTUPŮ**
- **SÉRIOVÝ VSTUP, AŽ 16 JEDNOTEK MŮŽE BÝT PŘIPOJENO NA JEDEN PORT NADŘAZENÉHO POČÍTAČE**
- **VNITŘNÍ PAMĚŤ UMOŽŇUJE, ABY PŘEDPROGRAMOVANÉ PŘÍKAZOVÉ SOUBORY BYLY PROVÁDĚNY BEZ PŘIPOJENÍ NADŘAZENÉHO POČÍTAČE**
- **PROGRAMOVATELNÁ MAXIMÁLNÍ RYCHLOST, START/STOP RYCHLOST, ZRYCHLENÍ**
- **MOŽNOST ZADÁNÍ SLOŽENÉHO POHYBU, SLOŽEN AŽ Z DESETI POHYBŮ**
- **MOŽNOST ZMĚNY RYCHLOSTI MOTORU ZA POHYBU**

4. Popis jednotky RKM04

4.1. Jednotka RKM04

Jednotka RKM04 slouží pro nezávislé ovládání čtyř krokových motorů. Je programovatelná pomocí dále definované sady příkazů. Na jednotce jsou binární vstupy a výstupy a komunikační rozhraní RS485. Signalizace vysílání povelů na krokový motor je pomocí ledek na čele jednotky. Napájení je 24VDC.

4.2. Komunikační rozhraní

Jednotka RKM04 se připojuje linkou RS485 s komunikačním protokolem PRT. Přenosová rychlost je 19200Bd, parita NONE, 8 bitů a 1 stop bit. Toto nastavení se nedá měnit.

4.3. Paměť

Program, sada příkazů, se posílá v ASCII tvaru. Pro editaci programu lze použít jakýkoliv textový editor, ovšem nesmí zapisovat do textového souboru při ukládání souboru (NOTEPAD, ...). Kapacita paměti umožní napsat program maximálně do 700 příkazů na jeden motor. Program vyslaný po komunikační lince se uloží do nezálohované paměti, při resetu jednotky RKM04 je zapomenut. Program lze poslat najednou nebo po částech. Příchozí příkazy se zapisují za sebou. Pokud překročíme kapacitu paměti jsou přepisovány nejdříve zaslané příkazy a je nahlášeno přetečení paměti.

Pokud si přejeme zachovat napsaný program po resetu je nutno jej uložit do zálohované paměti příkazem **Store Flash**. Každý motor má svoji vyhrazenou část paměti pro zápis jednoho programu. Po resetu je tento program načten do pracovní části paměti a ihned se začne vykonávat.

4.4. Vstupy a výstupy

Jednotka RKM04 má 8 binárních vstupů a 8 binárních výstupů. Jsou přístupné z programu jednotlivých motorů. Ovládání výstupů a čtení vstupů je záležitostí uživatele a provádí se pomocí příkazů z programu.

Pro synchronizaci motorů na jednotce RKM04 je možno použít 16 vnitřních binárních proměnných adresovaných 80 – 95. Na tyto proměnné lze zapsat hodnotu logická jedna nebo logická nula a lze tuto hodnotu číst.

4.5. Led diody

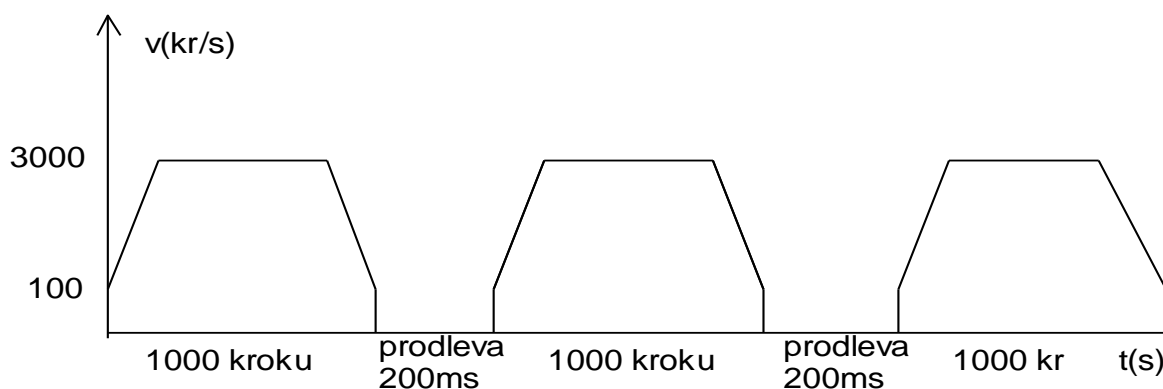
Na čele jednotky RKM04 jsou čtyři signalizační Led diody. Každý motor má jednu led diodu. Hlášené stavy:

- Zhaslá na motor se nevysílají žádné pulsy
- Svítí na motor se vysílají pulsy
- Bliká 0.5sec/0.5sec byla chyba při výpočtu pohybu motoru

4.6. Příklad chodu krokového motoru

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000, maximální rychlost 3000kroků/s. Pojede vpřed, provede čekání 200 ms, provede skok na začátek smyčky(L2), pojede vpřed, provede čekání 200 ms, provede skok na začátek smyčky(L2), pojede vpřed, provede čekání 200 ms, pak nastaví 300 kroků zpět :

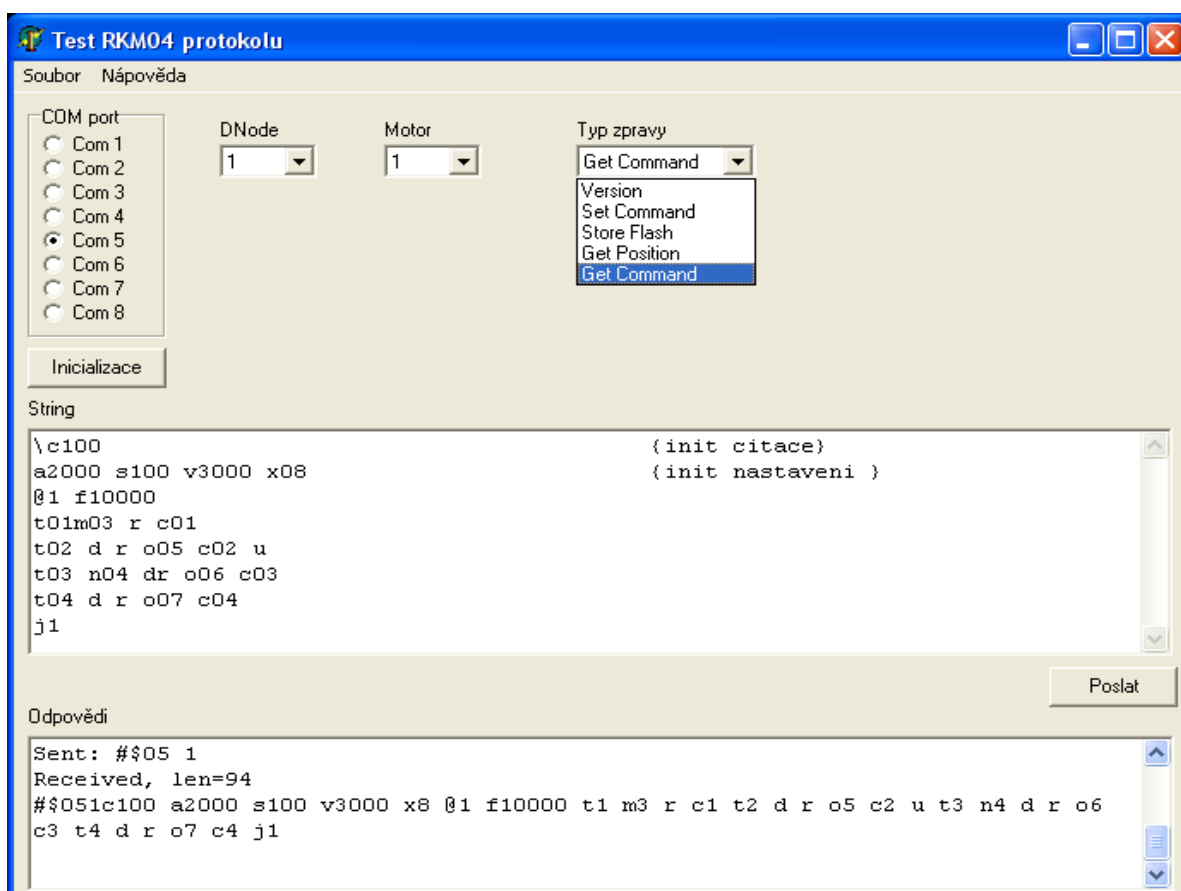
```
S100 A2000 F1000 V3000  
L3 R W200 E  
B300
```



5. Zápis programu

Pro servis jednotky RKM04 byl vyvinut program na PC, který je dodáván společně s jednotkou. Jedná se o jednoduchou operátorskou aplikaci pro zasílání jednotlivých povelů na jednotku RKM04. Okno aplikace obsahuje několik vybiracích oken (COM port, DNode, Motor, Typ zprávy), editovací pole (String) a pole pro zápis dat na komunikaci (Odpovědi). V editačním okně lze napsat program pro řízení motoru nebo lze do tohoto pole nahrát program z textového souboru (lze jej dále upravovat), vytvořený program lze uložit do textového souboru.

Na následujícím obrázku je okno servisního programu.



5.1. Navázání komunikace

Na obrazovce servisního programu na jednotku RKM04 je nutno zvolit komunikační port **Com1** až **Com8** dle zapojení na PC. Vybírací pole **DNode** slouží k nastavení kontroléru pro komunikaci – na jednotce se nastaví na přepínači S1:

DNode	S1-1	S1-2	S1-3	S1-4
1	OFF	OFF	OFF	OFF
2	ON	OFF	OFF	OFF
3	OFF	ON	OFF	OFF
4	ON	ON	OFF	OFF
5	OFF	OFF	ON	OFF
6	ON	OFF	ON	OFF
7	OFF	ON	ON	OFF
8	ON	ON	ON	OFF
9	OFF	OFF	OFF	ON
10	ON	OFF	OFF	ON
11	OFF	ON	OFF	ON
12	ON	ON	OFF	ON
13	OFF	OFF	ON	ON
14	ON	OFF	ON	ON
15	OFF	ON	ON	ON
16	ON	ON	ON	ON

Navolení **DNode na hodnotu 0** je posílání povelů na všechny připojené jednotky RKM04, ovšem jednotky neodpovídají. Zvolením **DNode** v rozsahu 1 až 16 navážeme komunikaci jen s jednou jednotkou, která posílá odpověď na povel. Po navolení předcházejících parametrů se stiskne tlačítko **Inicializace**, které provede propojení a navázání komunikace mezi PC a jednotkou RKM04.

5.1.1. Formát zpráv protokolu RKM04

Fyzická vrstva je komunikační sběrnice RS485 o baudové rychlosti 19200, 8 datových bitů bez parity, 1 stop bit.

- Zprávy jsou popsány jako pole jednotlivých byte číslovaných od nuly.
- Nultý byte zprávy je vždy číslo zprávy.
- Odpověď má vždy stejné číslo zprávy jako požadavek s výjimkou neznámé zprávy, kdy odpovědí je zpráva č. 255.

- Znaky bez apostrofů jsou přímo hodnoty byte ve zprávě. Znak v apostrofech je vždy ASCII znak, jehož kód je dán do patřičného byte zprávy
- ASCII znak čísla motoru v rozsahu '0' – '4' je číslo 48 ('0') – 52('4')

5.2. Vybírací okno **Motor**

je určeno pro nastavení motoru:

- 0 - povel je pro všechny motory,
- 1,2,3,4 – povel na jednotlivý motor.

5.3. Vybírací okno **Typ zprávy**

volí jaký povel se vyšle:

- Version** – dotaz na verzi SW
- Set Command** – odešle program zapsaný v editačním okně **String** do motoru
- Store Flash** – zápis do FLASH paměti pro motor
- Get Position** – přečte absolutní polohu motoru
- Get Command** - přečte aktuální sadu příkazů z motoru
- Get Tabul Run** – přečte aktuální tabulku pohybu pro motor
- Get Pozic Run** - přečte aktuální pozici ve vykonávaném pohybu motoru

5.3.1. Povel Version

Vrátí číslo verze a datum poslední aktualizace SW, který je nahrán na desce v textové podobě. Povel se odešle stiskem tlačítka **POSLAT**.

5.3.1.1. Formát povelu

Nultý byte – hodnota 1 = povel Version

délka	Data
1	1

5.3.1.2. Formát odpovědi

Nultý byte – hodnota 1 = povel Version

První byte – číslo udávající počet znaků (byte) následujícího textového řetězce

druhý až n-tý byte – ASCII znaky textu

Např.

délka	Data
18	1 16 '0' '1' '.' '0' '2' ',' '0' '6' ':' '0' '3' ':' '2' '0' '0' '9'

5.3.2. Povel Set Command

Editační okno **String** slouží k napsání požadovaného pohybu krokového motoru. K tomuto slouží několik předdefinovaných příkazů, jsou popsány dále. Po napsání celého programu se tento odešle stiskem tlačítka **POSLAT** na určený motor.

Editační okno **String** je víceřádkové a lze psát program do více řádků, lze používat komentáře, které se uzavírají do závorek { }. Je možné také načíst program jako soubor z disku pomocí Soubor–Otevři. Vytvořený program je možno uložit do souboru pomocí Soubor–Ulož nebo Soubor–Ulož jako... Prostředí ukládá přednostně soubory s příponou *.rks, ale lze použít jakoukoliv jinou. K editaci programu je možno použít textový editor, který při ukládání nepřidává žádné znaky (např. Notepad).

Po povelu se vrátí zpráva s poruchovým příznakem a případně místem špatného zápisu:

Příznak **00** – vše je správně

Příznak **01** – špatná hodnota za příkazem + místo špatného zápisu

Příznak **02** - neznámý příkaz + místo špatného zápisu

Příznak **03** - překročena datová oblast - varování

Pokud je vše správně začne se povel vykonávat.

5.3.2.1. Formát povelu

Nultý byte – hodnota 2 = povel Set Comand

První byte – ASCII znak čísla motoru v rozsahu '0' – '4'

Druhý až n-tý byte – ASCII příkazy popsané v dokumentu „V5722 – Popis ovládání RKM04 s protokolem“

Např. příkaz „C100“ poslaný na motor č. 3 bude vypadat následovně:

délka	Data
6	2 '3' 'C' '1' '0' '0'

5.3.2.2. Formát odpovědi

Nultý byte – hodnota 2 = povel Set Comand

První byte – ASCII znak čísla motoru v rozsahu '0' – '4'

Druhý byte – příznak (byte) s následujícími hodnotami

0	vše v pořádku
1	špatná hodnota za příkazem
2	neznámý příkaz
3	překročena datová oblast

Třetí byte – vždy uveden, ale význam má pouze pokud příznak obsahuje 01 nebo 02 a určuje ASCII znak/příkaz který nebyl rozpoznán nebo který je syntakticky nesprávný

Např. odpověď vše v pořádku od motoru č. 1 vypadá následovně

délka	Data
4	2 '1' 0 0

5.3.3. Povel Store Flash

Provede zápis všech příkazů po posledním zadání příkazu \-RESET pro určený motor. Tato sada příkazů se pak bude vykonávat po startu desky samostatně. Tuto paměť lze opětovně přepsat zadáním nového programu a vysláním povelu na zápis do Flash paměti. Povel se odešle stiskem tlačítka **POSLAT**.

5.3.3.1. Formát povelu

Nultý byte – hodnota 3 = povel Store Flash

První byte – ASCII znak čísla motoru v rozsahu '0' – '4'

Např. uložení obsahu paměti motoru č. 4 do flash:

délka	Data
2	3 '4'

5.3.3.2. Formát odpovědi

Nultý byte – hodnota 3 = povel Store Flash

První byte – ASCII znak čísla motoru v rozsahu '0' – '4'

Druhý byte – signalizuje chybu při zápisu do FLASH (byte), 0 = bez chyby

Např. úspěšné uložení motoru č. 3:

délka	Data
1	3 '4' 0

5.3.4. Povel Get Position

Vrátí aktuální pozici, tj. počet kroků od posledního vynulování čítače polohy (C100) určeného motoru. Hodnota je zaslána v textovém formátu. Povel se odešle stiskem tlačítka **POSLAT**.

5.3.4.1. Formát povelu

Nultý byte – hodnota 4 = povel Get Position

První byte – ASCII znak čísla motoru v rozsahu '0' – '4'

Např. zjištění pozice motoru č. 1:

délka	Data
2	4 '1'

5.3.4.2. Formát odpovědi

Nultý byte – hodnota 4 = povel Get Position

První byte – ASCII znak čísla motoru

Je-li číslo motoru v rozsahu 1-4, následuje (byte) číslo 10 (délka následujícího textového řetězce) a dále deset znaků s textem obsahujícím pozici

Je-li číslo motoru 0, následuje 4x za sebou pozice, tedy číslo 10 + deset znaků pro motor 1, číslo 10 + deset znaků pro motor 2, číslo 10 + deset znaků pro motor 3 a číslo 10 + deset znaků pro motor 4.

délka	Data
13	4 '1' 10 ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' ' '0'

5.3.5. Povel Get Command

Tento povel vrátí aktuální program, tj. sadu příkazů, která se právě vykonává pro určený motor. Tento povel **nelze poslat** s parametrem motoru 0 – všechny motory. Povel se odešle stiskem tlačítka **POSLAT**.

5.3.5.1. Formát povelu

Nultý byte – hodnota 5 = povel Get Comand

První byte – ASCII znak čísla motoru v rozsahu '1' – '4'

Např. získání příkazů pro motor č. 2:

délka	Data
2	5 '2'

5.3.5.2. Formát odpovědi

Nultý byte – hodnota 5 = povel Get Comand

První byte – ASCII znak čísla motoru

Druhý až n-tý byte – sada povelů pro motor

(pokud si vyžádáme příkazy najedou od motoru č. „0“ – tedy od všech motorů zároveň, přijde nám zpátky text: „ERROR COMMAND! “

Např. obsahuje-li sada příkazů pro motor č. 2 „C100“, bude zpráva vypadat následovně:

délka	Data
6	5 '2' 'C' '1' '0' '0'

5.3.6. Povel Get Tabul Run

Tento povel vrátí tabulku aktuálního pohybu, tj. sadu zlomových bodů na aktuálním pohybu, který se právě vykonává pro určený motor. Tabulka je uzavřena do závorek [a], jednotlivé zlomové body jsou v (a), skladba bodu je (krok; rychlost). Tento povel **nelze poslat** s parametrem motoru 0 – všechny motory. Povel se odešle stiskem tlačítka **POSLAT**.

5.3.6.1. Formát povelu

Nultý byte – hodnota 6 = povel Get Tabul Run

První byte – ASCII znak čísla motoru v rozsahu '1' – '4'

Např. získání tabulky aktuálního pohybu pro motor č. 2:

délka	Data
2	6 '2'

5.3.6.2. Formát odpovědi

Nultý byte – hodnota 6 = povel Get Tabul Run

První byte – ASCII znak čísla motoru

Druhý až n-tý byte – sada aktuální tabulky pohybu pro motor

(pokud si vyžádáme příkazy najednou od motoru č. „0“ – tedy od všech motorů zároveň, přijde nám zpátky text: „ERROR COMMAND!“

Např. odpověď na povel 6, pro motor 2 má 14 zlomových bodů, jednotlivé body jsou [(1;100),(10;500),..., (2222;0)], bude zpráva vypadat následovně:

délka	Data
161	6 '2' ' ' '1' '4' ';' '[' '(' '1' ';' '1' '0' '0' ')' ';' '(' '1' '0'
.....	Data
.....	2 '2' ';' '0' ')' ']'

5.3.7. Povel Get Pozic Run

Tento povel vrátí aktuální počet kroků v právě prováděném jednotlivém pohybu pro určený motor a aktuální pořadí příkazu v právě vykonávaném programu. Tento povel **nelze poslat** s parametrem motoru 0 – všechny motory. Povel se odešle stiskem tlačítka **POSLAT**.

5.3.7.1. Formát povelu

Nultý byte – hodnota 7 = povel Get Pozic Run

První byte – ASCII znak čísla motoru v rozsahu '1' – '4'

Např. získání příkazů pro motor č. 4:

délka	Data
2	7 '4'

5.3.7.2. Formát odpovědi

Nultý byte – hodnota 7 = povel Get Pozic Run

První byte – ASCII znak čísla motoru

Druhý až n-tý byte – aktuální počet kroků; aktuální pozice v programu

(pokud si vyžádáme příkazy najedou od motoru č. „0“ – tedy od všech motorů zároveň, přijde nám zpátky text: „ERROR COMMAND! “

Např. odpověď na povel 7, pro motor 4 je aktuální počet kroků v právě prováděném jednotlivém pohybu 45823 a aktuální pořadí vykonávaného příkazu v programu je 5, bude zpráva vypadat následovně:

délka	Data										
11	7	'4'	' '	'4'	'5'	'8'	'2'	'3'	','	' '	'5'

5.3.8. Nesprávný povel

Pokud je zjištěn nesprávný povel na RKM04 je vrácen kód 255 s hodnotou neznámého povelu.

5.3.8.1. Formát odpovědi

Na libovolnou jinou zprávu dostaneme odpověď, kde:

Nultý byte – hodnota 255 = „neznámý příkaz“

První byte – číslo zprávy které jednotka nezná (byte), tedy pokud zašleme zprávu číslo 7, bude zde číslo 7

Druhý byte – vždy 0 (byte)

Třetí byte – vždy 0 (byte)

Např.:

délka	Data			
4	255	7	0	0

6. Přehled příkazů

\	„RESET“ – uvedení motoru do výchozího stavu
@(NUM)	„Návěští“ – číslo návěští pro skoky v programu
A(NUM)	„Zrychlení“ – zrychlení nebo zpomalení motoru
B(NUM)	„Pohyb vzad“ – počet kroků motoru směrem vzad
C(NUM01-08)	„Zruš výstup“ – nastaví určený logický výstup na logická nula
C(NUM40-58)	„Zruš funkci“ – zruší zadanou přídatnou funkci
C(NUM80-95)	„Zruš proměnnou“ – nastaví určenou logickou vnitřní proměnnou na logická nula
C75	„Obnovení pohybu“ – start programu z místa kde byl ukončen
C100	„Smaž čítač polohy“ – vynulování čítače absolutní polohy
D	„Změna směru pohybu“ - otočí směr pohybu motoru
E	„Konec smyčky“ – označí konec programové smyčky
F(NUM)	„Pohyb vpřed“ – počet kroků motoru směrem vpřed
G(NUM)	„Absolutní poloha“ – zadání absolutní polohy motoru
G+	„Trvale vpřed“ – pohyb vpřed maximální rychlostí motoru
G-	„Trvale vzad“ – pohyb vzad maximální rychlostí motoru
H	„Výchozí poloha“ – návrat motoru na polohu absolutní nula
I(NUMBI)(VALUE)(NUM)	„Podmíněný skok“ – odskok na návěští při splnění podmínice
J(NUM)	„Skok“ – odskok na zadané návěští
K	„Konec pohybu“ – ukončení pohybu motoru po sestupné rampě a stop programu
L(NUM)	„Začátek smyčky“ – počátek programové smyčky
M(NUM)	„Stop od BI True“ – ukončení pohybu motoru po sestupné rampě od fyzického vstupu ve stavu logická jedna, program pokračuje
N(NUM)	„Stop od BI False“ - ukončení pohybu motoru po sestupné rampě od fyzického vstupu ve stavu logická nula, program pokračuje
O(NUM)	„BI True“ – čekání na stav logického vstupu logická jedna
R	„Proved' pohyb“ – vykoná pohyb dle naposled zadaných parametrů
S(NUM)	„Startovní rychlost“ – nastaví start/stop rychlost motoru
T(NUM01-08)	„Nastav výstup“ - nastaví určený logický výstup na logická jedna
T(NUM40-58)	„Nastav funkci“ - nastaví zadanou přídatnou funkci
T(NUM80-95)	„Nastav proměnnou“ - nastaví určenou logickou vnitřní proměnnou na logická jedna
U	„Reset Stop od BI“ – zruší nastavení povelu M(NUM) nebo N(NUM)
V(NUM)	„Maximální rychlost“ – nastaví maximální rychlost motoru
V(NUM)	„Změna rychlosti za pohyb“ – provede změnu rychlosti za pohyb
W(NUM)	„Čekání“ – provede prodlevu zadaný počet milisekund
X(NUM)	„Total stop od BI True“ - ukončení pohybu motoru po sestupné rampě od fyzického vstupu ve stavu logická jedna, program se ukončí
Y	„Složený pohyb“ – zapíše zadání jednoho úseku složeného pohybu do tabulky
Z(NUM)	„BI False“ - čekání na stav logického vstupu logická nula
)(NUM)	„Limit vpřed“ – dojezd směrem vpřed na fyzický BI vstup start/stop rychlostí
((NUM)	„Limit vzad“ - dojezd směrem vzad na fyzický BI vstup start/stop

	rychlostí
=(NUM)	„ Zapiš absolutní polohu “ – čítač absolutní polohy přepíše zadanou hodnotou
‘(NUM)	„ Volání podprogramu “ – odskok na začátek podprogramu
.	„ Konec podprogramu “ – návrat do hlavního programu
[„ Program “ – blokáce okamžitého vykonávání programu
]	„ Konec programu “ – povolení vykonat zadaný program

7. Příkazy

7.1. Reset

\ - uvedení motoru do výchozího stavu. Nahrají se počáteční hodnoty pro start desky.

7.2. Návěští

@(NUM)- návěští (NUM). Číslo NUM v rozsahu 1 – 80

Nastavení návěští číslo 2

@2

7.3. Zrychlení

A(NUM) - s touto hodnotou NUM bude motor zvyšovat nebo snižovat svoji rychlost.

Rozsah 1 – 60 000.

Nastaví zrychlení/zpomalení motoru na 2000 kroků/s²

A2000

7.4. Pohyb vzad

B(NUM) - zadání počtu kroků NUM směrem vzad relativně k současné poloze.

Rozsah 1 – 16 000 000.

Nastaví požadovaný počet kroků pro směr vzad na 1000 kroků

B1000

7.5. Zruš výstup

C(NUM01-08) - nastaví zadaný binární výstup NUM na hodnotu logická 0.

Rozsah fyzické výstupy 01 – 08

Nastaví binární výstup 06 na logická nula

C06

7.6. Zruš proměnnou

C(NUM80-95) - nastaví zadanou logickou vnitřní proměnnou NUM na hodnotu logická 0.

Rozsah vnitřní proměnné 80 - 95

Nastaví vnitřní proměnnou 86 na logická nula

C86

7.7. Smaž čítač polohy

C100 - vynuluje čítač absolutní polohy motoru.

7.8. Změna směru pohybu

D - otáčí směr pohybu motoru.

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000. Pojede vpřed 1000 kroků, provede změnu směru pohybu a pojede vzad 1000 kroků

S100 A2000 F1000

R D R

7.9. Konec smyčky

E - označuje konec právě vykonávané smyčky.

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000. Pojede vpřed, provede čekání 200 ms, provede skok na začátek smyčky(L2), pojede vpřed, provede čekání 200 ms, pak nastaví 300 kroků zpět

S100 A2000 F1000

L2 R W200 E

B300

7.10. Pohyb vpřed

F(NUM) - zadání počtu kroků NUM směrem vpřed relativně k současné poloze.

Rozsah 1 – 16 000 000.

Nastaví požadovaný počet kroků pro směr vpřed na 1000 kroků

F1000

7.11. Absolutní poloha

G(NUM) - zadání počtu kroků NUM na absolutní polohu motoru. Je zadána absolutní poloha.

Rozsah 1 – 2 000 000 000.

Nastaví požadovaný počet kroků jako rozdíl zadaného počtu kroků a skutečné polohy a provede dojezd na novou absolutní polohu

G10000

7.12. Trvale vpřed

G+ - zadání trvalého pohybu směrem vpřed. Dá se ukončit příkazem konec pohybu K nebo příkazem Stop od BI True (False).

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000 kroků. Nastavíme hlídání binárního vstupu 5 a rozjedeme trvale vpřed maximální rychlostí. Při BI5 ve stavu True se běh zastaví po sestupné rampě

S100 A2000 F1000
M05 G+

7.13. Trvale vzad

G- - zadání trvalého pohybu směrem vzad. Dá se ukončit příkazem konec pohybu K nebo příkazem Stop od BI True (False).

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000 kroků. Nastavíme hlídání binárního vstupu 5 a rozjedeme trvale vzad maximální rychlostí. Při BI5 ve stavu True se běh zastaví po sestupné rampě

S100 A2000 F1000
M05 G-

7.14. Výchozí poloha

H - provede pohyb motoru do polohy absolutní nula.

Motor se rozjede směrem k dosažení absolutní polohy nula

H

7.15. Podmíněný skok

I(NUMBI)(VALUE)(NUM) - odskok na zadané návěští při splnění podmínce. Pokud zadaný binární vstup NUMBI splňuje podmínku VALUE(H nebo L) je proveden odskok na návěští NUM.

Rozsah: NUMBI fyzické vstupy 01 – 08
vnitřní proměnné 80 – 95
NUM 1 – 80

Pokud je binární vstup 01 logická 1 je proveden skok na návěští 20 a provedeno 200 kroků směrem vzad, pokud je binární vstup 01 logická 0 je provedeno 200 kroků směrem vpřed a skok na návěští 21,...

@19 I01H20 F200 R J21
@20 B200 R
@21

7.16. Skok

J(NUM) - odskok na zadané návěští NUM.

Rozsah 1 - 80

Provedu pohyb 200 kroků směrem vpřed, počkám 500 ms a pak se skokem vrátím na návěští 20

@20 F200 R W500 J20

7.17. Konec pohybu

K - ukončení pohybu motoru po sestupné rampě. Tento příkaz je v bloku příkazů ignorován!!!

Tento příkaz lze poslat pouze samostatně.

7.18. Obnovení pohybu

C75 - start pohybu motoru po ukončení pohybu. Pohyb pokračuje od místa ve kterém byl přerušen. Platí pouze od přerušeni pomocí příkazu **K – konec pohybu**. Tento příkaz je v bloku příkazů ignorován!!!

Tento příkaz lze poslat pouze samostatně.

7.19. Stop od BI True

M(NUM) - Následující pohyby jsou hlídány zvoleným fyzickým binárním vstupem NUM. Pokud se nastaví na logická 1 je pohyb přerušen po sestupné rampě a program pokračuje následujícím příkazem. Po prvním použití je tento příkaz zrušen, chceme-li ho použít opětovně je nutno ho znovu zadat.

Rozsah 01 – 08

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000 kroků. Nastavíme hlídání binárního vstupu 5 a rozjedeme motor po rampě. Při BI5 ve stavu True se běh zastaví po sestupné rampě a nastaví se počet kroků vzad na 200 kroků, pokud se BI nenastaví na True je proveden celý pohyb a nakonec se nastaví počet kroků vzad na 200 kroků

S100 A2000 F1000
M05 R B200

7.20. Stop od BI False

N(NUM) - Následující pohyby jsou hlídány zvoleným fyzickým binárním vstupem NUM. Pokud se nastaví na logická 0 je pohyb přerušen po sestupné rampě a program pokračuje následujícím příkazem. Po prvním použití je tento příkaz zrušen, chceme-li ho použít opětovně je nutno ho znovu zadat.

Rozsah 01 – 08

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000. Nastavíme hlídání binárního vstupu 5 a rozjedeme motor po rampě. Při BI5 ve stavu False se běh zastaví po sestupné rampě a nastaví se počet kroků vzad na 200 kroků, pokud se BI nenastaví na False je proveden celý pohyb a nakonec se nastaví počet kroků vzad na 200 kroků

S100 A2000 F1000
N05 R B200

7.21. Reset stop od BI

U – provede se zrušení nastavení příkazu **M(NUM)** nebo **N(NUM)**.

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000 kroků. Nastavíme hlídání binárního vstupu 5 a rozjedeme motor po rampě. Při BI5 ve stavu False se běh zastaví po sestupné rampě a zruší nastavení hlídání BI5, pokud se BI nenastaví na False je proveden celý pohyb a nakonec zruší nastavení hlídání BI5

S100 A2000 F1000
N05 R U

7.22. Začátek smyčky

L(NUM) - počátek povelové smyčky a číslo NUM určuje počet opakování. Je možno použít až 8 vnořených smyček.

Rozsah 1 – 255

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000. Pojede vpřed, provede čekání 200 ms, provede skok na začátek smyčky(L2), pojede vpřed, provede čekání 200 ms, pak nastaví 300 kroků zpět

S100 A1000 F1000
L2 R W200 E
B300

7.23. BI True

O(NUM) - čeká na stav binárního vstupu NUM logická 1.

Rozsah fyzické vstupy 01 – 08

vnitřní proměnné 80 - 95

Čekání na stav binárního vstupu 3 logická jedna TRUE

O03

7.24. Proved' pohyb

R - vykoná pohyb dle naposled zadaných parametrů.

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000 kroků. Rozjedeme motor po rampě a vykonáme 1000 kroků. PO dokončení pohybu se vrátíme zpět do nulové absolutní polohy

S100 A2000 F1000
R H

7.25. Startovní rychlost

S(NUM) - nastaví start/stop rychlost motoru NUM. Rozsah je dán parametry motoru.

Rozsah 1 – 2 000

Nastavíme startovní rychlost na 100 kroků/s

S100

7.26. Nastav výstup

T(NUM) - nastaví zadaný binární výstup NUM na hodnotu logická 1.

Rozsah fyzické výstupy 01 – 08

Nastaví binární výstup 06 na logická jedna

C06

7.27. Nastav proměnnou

T(NUM) - nastaví zadanou logickou vnitřní proměnnou NUM na hodnotu logická 1.

Rozsah vnitřní proměnné 80 - 95

Nastaví vnitřní proměnnou 86 na logická jedna

C86

7.28. Maximální rychlost

V(NUM) - nastaví maximální rychlost motoru NUM. Rozsah je dán parametry motoru.

Rozsah 1 – 16 000

Nastaví maximální rychlost motoru na 3000 kroků/s

V3000

7.29. Změna rychlosti za pohyb

V(NUM) - nastaví novou rychlost motoru NUM. Rozsah je dán parametry motoru.

Pokud motor vykonává pohyb, zadanou maximální rychlostí a je nutno provést změnu rychlosti motoru, je možno po komunikaci odeslat příkaz na změnu rychlosti motoru příkazem **V**. Nová rychlost se začne nastavovat do cca 2 sekund od příjmu příkazu. Příkaz na změnu rychlosti **V** se musí poslat **v samostatné zprávě na první pozici**. Provést změnu rychlosti za pohyb lze během jednoho pohybu opakovaně. Všechny změny rychlosti za pohyb jsou dočasné pro aktuální průběh úseku programu. Pokud máme program s opakovaným spouštěním pohybu, nemá příkaz na změnu rychlosti za pohyb na další spuštění tohoto pohybu žádný vliv (provede pohyb s původně zadanými parametry).

Rozsah 1 – 16 000

Tento příkaz lze poslat pouze samostatně.

Nastaví rychlost motoru na 6000 kroků/s

V6000

7.30. Čekání

W(NUM) - provede prodlevu NUM milisekund.

Rozsah 1 – 16 000 000.

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000 kroků. Pojede vpřed, provede čekání 200 ms, provede skok na začátek smyčky(L2), pojede vpřed, provede čekání 200 ms, pak nastaví 300 kroků zpět

S100 A1000 F1000

L2 R W200 E

B300

7.31. Total stop od BI True

X(NUM) - při nastavení fyzického binárního vstupu NUM do logické 1 se běh programu zastaví, pokud byl vykonáván pohyb je zastaven po sestupné rampě.

Rozsah 01 – 08

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000 kroků. Nastavíme hlídání binárního vstupu 5 a rozjedeme motor po rampě. Při BI5 ve stavu TRUE se běh zastaví po sestupné rampě a zruší další chod programu, pokud se BI5 nenastaví na TRUE je proveden celý pohyb a nastaví se počet kroků vzad na 300 kroků

S100 A2000 F1000

X05 R B300

7.32. BI False

Z(NUM) - čeká na stav binárního vstupu NUM logická 0.

Rozsah fyzické vstupy 01 – 08

vnitřní proměnné 80 - 95

Čekání na stav binárního vstupu 3 logická nula False

Z03

7.33. Limit vpřed

)(NUM) - pohyb vpřed motoru start/stop rychlostí na stav fyzického binárního vstupu NUM logická 1.

Rozsah 01 – 08

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000 kroků. Spustíme pohyb vpřed startovní rychlostí, při dosažení BI5 hodnoty TRUE se pohyb zastaví

S100 A2000 F1000

)05

7.34. Limit vzad

((NUM) - pohyb vzad motoru start/stop rychlostí na stav fyzického binárního vstupu NUM logická 1.

Rozsah 01 – 08

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000 kroků. Spustíme pohyb vzad startovní rychlostí, při dosažení BI5 hodnoty TRUE se pohyb zastaví

S100 A2000 F1000

(05

7.35. Zapiš absolutní polohu

=(NUM) - přepíše čítač absolutní polohy na zadanou hodnotu NUM.

Přepíšeme čítač absolutní polohy na hodnotu 5000 kroků

=5000

7.36. Složený pohyb

Y - zapíše zadání jednoho úseku složeného pohybu do tabulky. Každý úsek se zadává počtem kroků, rychlostí a zrychlením. Pohyb se začne vykonávat příkazem **R**. Není povoleno během složeného pohybu měnit směr otáčení motoru. Maximálně lze zadat 10 úseků složeného pohybu. Maximální součet všech kroků složeného pohybu je 160 000 000 kroků.

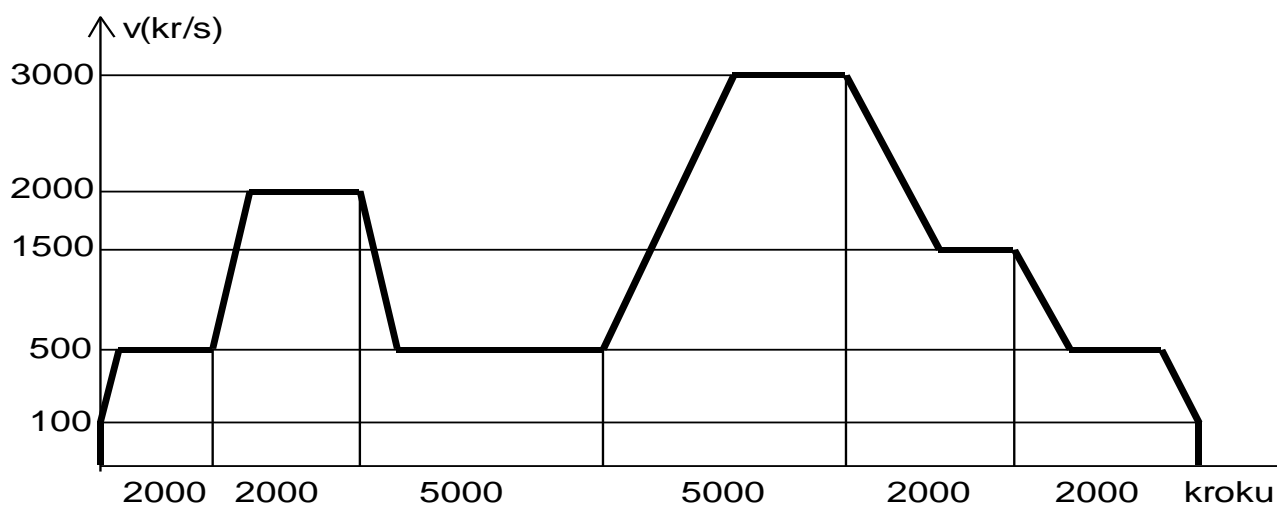
Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², maximální rychlost na 500 kroků/s, počet kroků vpřed na 2000 kroků. Zapišeme první úsek do tabulky. Změníme maximální rychlost na 2000 kroků/s a zapišeme druhý úsek do tabulky. Změníme maximální rychlost na 500 kroků/s, počet kroků vpřed na 5000 kroků a zapišeme třetí úsek do tabulky. Změníme maximální rychlost na 3000 kroků/s, zrychlení na 1000 kroků/s² a zapišeme čtvrtý úsek do tabulky. Změníme maximální rychlost na 1500 kroků/s, počet kroků vpřed na 2000 kroků a zapišeme pátý úsek do tabulky. Změníme maximální rychlost na 500 kroků/s a zapišeme šestý úsek do tabulky. Následně se celý složený pohyb vykoná

```
S100 A2000 V500 F2000 Y
V2000 Y
V500 F5000 Y
V3000 A1000 Y
V1500 F2000 Y
V500 Y R
```

Tabulka zadaného složeného pohybu:

úsek	A	V	F
1	2000	500	2000
2	2000	2000	2000
3	2000	500	5000
4	1000	3000	5000
5	1000	1500	2000
6	1000	500	2000

Schéma složeného pohybu



7.37. Volání podprogramu

‘(NUM) - provádí skok na návěští NUM na začátku podprogramu. Maximálně 6 úrovní vnoření.

Rozsah 1 – 80

V závislosti na binárním vstupu 1 jsou definovány různé činnosti. Podprogram začínající na návěští 3 obsahuje povely společné pro obě činnosti

```
@1 I01L6
F2500 '3 Z3 J1
@6 B2500 '3 Z4 J1
@3 R W2000 D R .
```

7.38. Konec podprogramu

. – konec podprogramu provede návrat do hlavního programu za příkaz volání podprogramu.

V závislosti na binárním vstupu 1 jsou definovány různé činnosti. Podprogram začínající na návěští 3 obsahuje povely společné pro obě činnosti

```
@1 I01L6
F2500 '3 Z3 J1
@6 B2500 '3 Z4 J1
@3 R W2000 D R .
```

7.39. Program

[– první příkaz souboru povelů na motor jenž není okamžitě vykonáván a čeká na Konec programu.

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000 kroků. Další vykonávání programu je blokováno do příchodu povolení běhu programu

```
S100 A2000 F1000
[ X05 R B300
```

7.40. Konec programu

] – poslední příkaz souboru povelů na motor a povely od znaku Program se vykonají.

Nastavíme startovní rychlost na 100 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000 kroků. Další vykonávání programu je blokováno do příchodu povolení běhu programu. Po načtení příznaku konec programu je nastaveno hlídání binárního vstupu 5 a rozjede se motor po rampě. Při BI5 ve stavu TRUE se běh zastaví po sestupné rampě a zruší další chod programu, pokud se BI5 nenastaví na TRUE je proveden celý pohyb a nastaví se počet kroků vzad na 300 kroků

```
S100 A2000 F1000
[ X05 R B300 ]
```

8. Funkce parametrizace

Funkce parametrizace umožňují měnit konfiguraci jednotky RKM04 pro přizpůsobení se požadavkům konkrétní aplikace. Funkce jsou dvoustavové: ON - zapnuto, OFF - vypnuto. Počáteční nastavení funkcí parametrizace je OFF. Po resetu jednotky se funkce nastaví do počátečního nastavení.

Tvar: **T(NUM40-58)** nastaví funkci na ON
 C(NUM40-58) nastaví funkci na OFF

Rozsah 40 – 58

8.1. Blokace přepisu zadání programu – parametr 40

Funkce parametrizace 40 při zaplnění vymezené datové oblasti pro program zablokuje kruhové přepisování zadaných příkazů.

ON: při zaplnění datové oblasti se ohlásí přetečení a dále se nepokračuje v zápisu příkazů do paměti, nové příkazy jsou ignorovány.

OFF: při zaplnění datové oblasti se ohlásí přetečení a dále se pokračuje v zápisu do paměti od pozice 1, nejstarší příkazy jsou přepisovány.

8.2. Simulační režim – parametr 41

Funkce parametrizace 41 nastavuje simulační režim jednotky RKM04.

ON: nastavíme simulační režim výstupních signálů, tj. jednotka RKM04 zpracovává zadaný program normálně, ale jednotlivé výstupy jsou odpojeny a ponechána na nich nulová úroveň.

OFF: zrušíme simulační režim.

8.3. Funkce – parametr 42 - 50

Funkce parametrizace 42 – 50 jsou připraveny pro další použití.

8.4. Nastav výstup – parametr 51 - 58

Funkce parametrizace 51 - 58 nastavuje binární výstup jednotky RKM04 při zahájení vykonávání posledního úseku složeného pohybu. Binární výstup je definován druhou číslicí, tj.: 51 –nastaví výstup 1, 52 –nastaví výstup 2,...,58 –nastaví výstup 8. Funkci je třeba aktivovat před počátkem zadávání složeného pohybu. **Zrušení této funkce neprovede zrušení již nastaveného binárního výstupu.** Zrušit nastavený binární výstup je nutno příkazem Zruš výstup C(NUM01-08).

ON: nastavíme funkci nastav výstup, tj. jednotka RKM04 zpracovává zadaný složený pohyb a při zahájení posledního úseku nastaví definovaný binární výstup do stavu logická jedna.

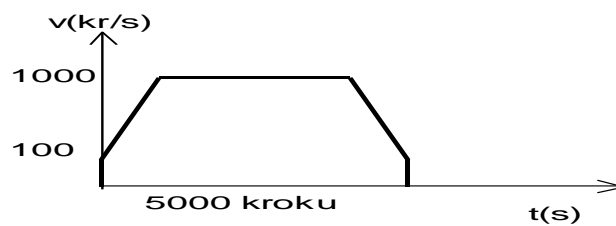
OFF: zrušíme funkci nastav výstup.

9. Příklady povelové sady

9.1. Příklad 1

Provede se vymazání příkazů na krokový motor, vynulování čítače absolutní polohy motoru. Nastavíme Start/Stop rychlost na 100 kroků/s, maximální rychlost na 1000 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 5000 kroků a provedeme tento zadaný pohyb vpřed. Nastavíme počet kroků vzad na 2000 a provedeme pohyb vzad. Nakonec se provede dojezd krokového motoru do polohy absolutní nula.

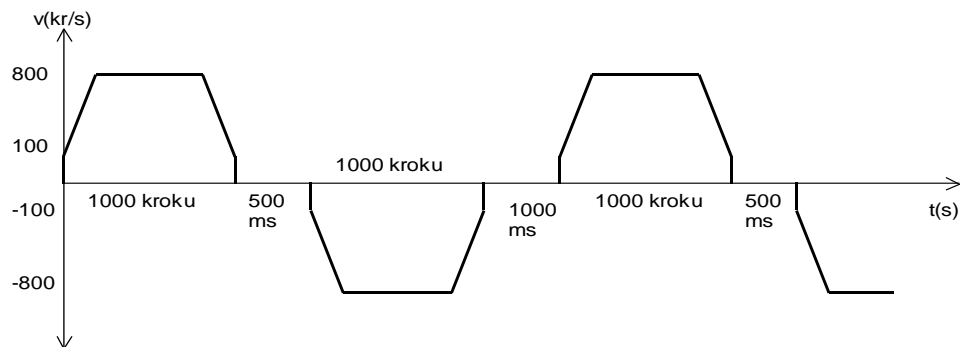
```
\C100  
S100 V1000 A2000 F5000 R  
B2000 R  
H
```



9.2. Příklad 2

Provede se vymazání příkazů na krokový motor. Nastavíme Start/Stop rychlost na 100 kroků/s, maximální rychlost na 800 kroků/s, zrychlení na 2000 kroků/s². Definujeme návěští číslo 1, nastavíme počet kroků vpřed na 1000 kroků a provedeme pohyb vpřed. Provedeme prodlevu 500ms, změním směr pohybu a provedeme pohyb vzad. Provedeme prodlevu 1000ms a provedeme odskok na návěští číslo 1. Od návěští číslo 1 se vše opakuje.

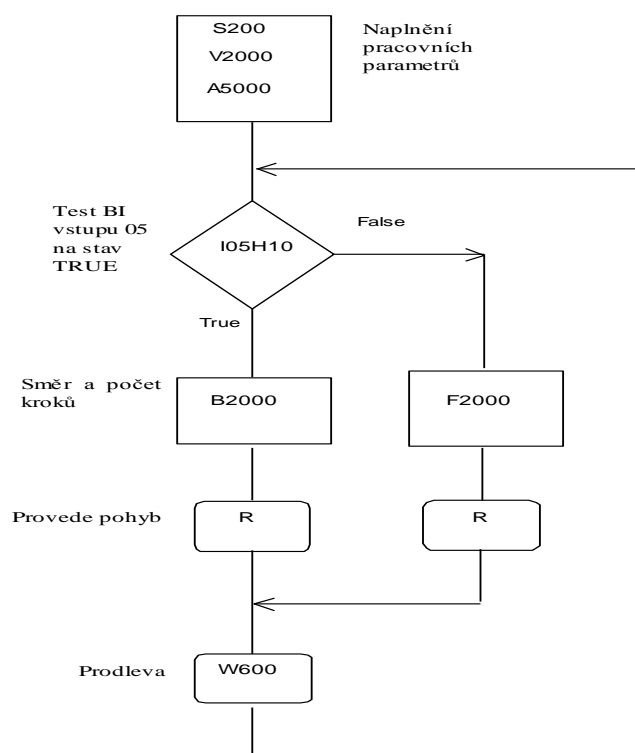
\
S100 V800 A2000
@1 F1000 R
W500 D R
W1000 J1



9.3. Příklad 3

Provedeme smazání všech příkazů na krokový motor. Nastavíme Start/Stop rychlost na 200 kroků/s, maximální rychlost na 2000 kroků/s, zrychlení na 5000 kroků/s². Definujeme návěští číslo 1, pokud je BI vstup 05 na hodnotě True je proveden odskok na definované návěští číslo 10, nastaveno 2000 kroků vzad a je proveden pohyb vzad, následně je proveden odskok na definované návěští číslo 20. Pokud je BI vstup na hodnotě False je nastaveno 2000 kroků vpřed a je proveden pohyb vpřed, následně je proveden odskok na definované návěští číslo 20. Provede se prodleva 600 ms a je uskutečněn odskok na definované návěští číslo 1. A vše se opakuje.

\
S200 V2000 A5000
@1 I05H10 F2000 R J20
@10 B2000 R
@20 W600
J1



9.4. Příklad 4

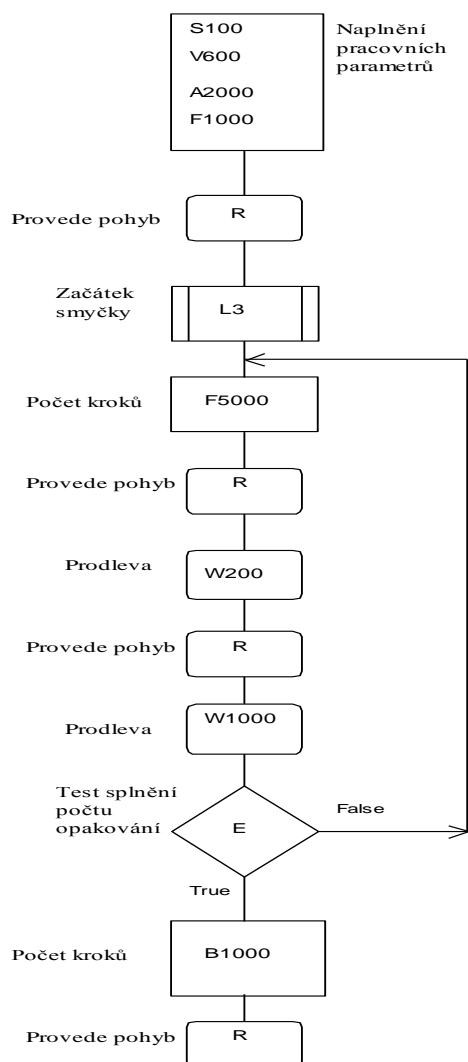
Provede se vymazání příkazů na krokový motor, vynulování čítače absolutní polohy motoru. Nastavíme Start/Stop rychlost na 100 kroků/s, maximální rychlost na 600 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 1000 kroků a provede se pohyb vpřed. Definuje počátek smyčky a počet opakování smyčky na 3, nastaví počet kroků vpřed na 500 kroků, provede se pohyb vpřed, nastane prodleva 200 ms, změní směr pohybu, provede se pohyb vzad, nastane prodleva 1000 ms, otestuje se počet opakování smyčky. Při menším 3 se provede odskok na definici počátku smyčky, pokud byly provedeny již 3 průchody smyčkou ne nastaven počet kroků vzad na 1000 kroků a je proveden pohyb vzad.

\C100

S100 V600 A2000 F1000 R

L3 F500 R W200 D R W1000 E

B1000 R



9.5. Příklad 5

Provede se vymazání příkazů na krokový motor, vynulování čítače absolutní polohy motoru. Nastavíme Start/Stop rychlost na 100 kroků/s, maximální rychlost na 1000 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 5000 kroků a provedeme tento zadaný pohyb vpřed. Počkáme až BI vstup 02 se změní do stavu True, nastavíme počet kroků vzad na 6000 kroků a provedeme pohyb vzad. Počkáme až BI vstup 05 se změní do stavu False, nastavíme počet kroků vpřed na 3000 kroků a provedeme pohyb vpřed. Nakonec provedeme návrat do polohy absolutní nula.

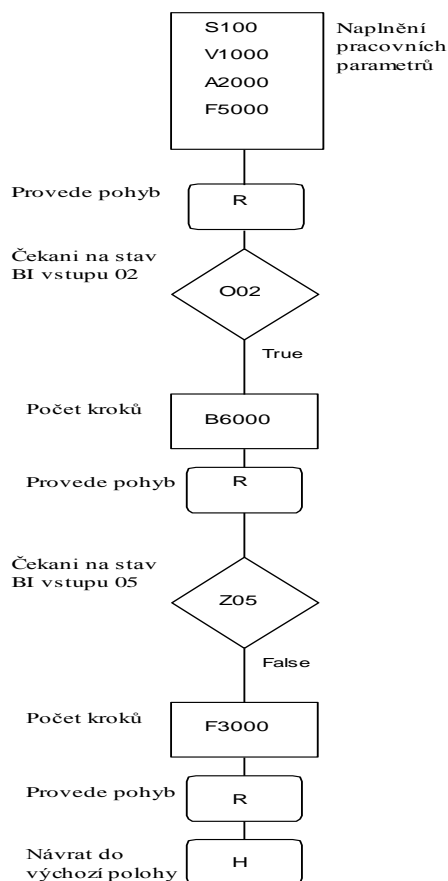
\C100

S100 V1000 A2000 F5000 R

O02 B6000 R

Z05 F3000 R

H

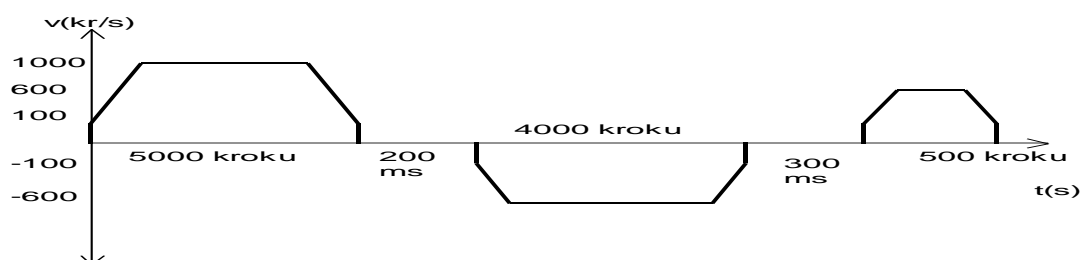


9.6. Příklad 6

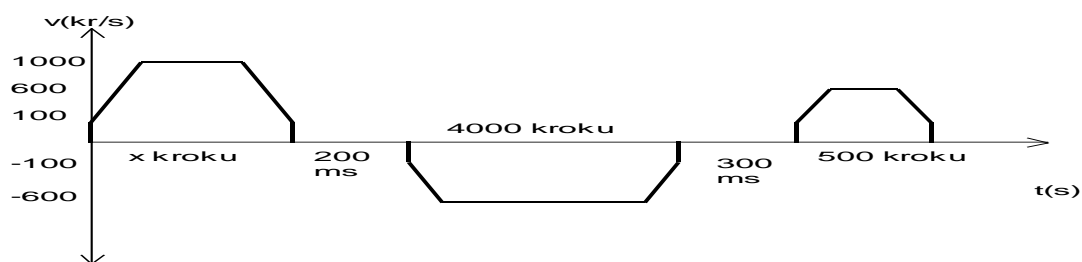
Provede se vymazání příkazů na krokový motor, vynulování čítače absolutní polohy motoru. Nastavíme Start/Stop rychlost na 100 kroků/s, maximální rychlost na 1000 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 5000 kroků. Nastavíme koncový spínač pohybu na BI vstup 03. Spustíme pohyb vpřed, pokud se BI vstup 03 změní na True je pohyb ukončen po sestupné rampě, jinak je pohyb vykonán celý. Zrušíme nastavení koncového spínače. Provedeme prodlevu 200 ms, nastavíme maximální rychlost na 600 kroků/s, nastavíme počet kroků vzad na 4000 kroků. Nastavíme koncový spínač pohybu na BI vstup 06. Spustíme pohyb vzad, pokud se BI vstup 06 změní na False je pohyb ukončen po sestupné rampě, jinak je pohyb vykonán celý. Zrušíme nastavení koncového spínače. Provedeme prodlevu 300 ms, nastavíme počet kroků vpřed na 500 kroků a provedeme pohyb vpřed.

```
\C100
S100 V1000 A2000 F5000
M03 R U
W200 V600 B4000
N06 R U
W300 F500 R
```

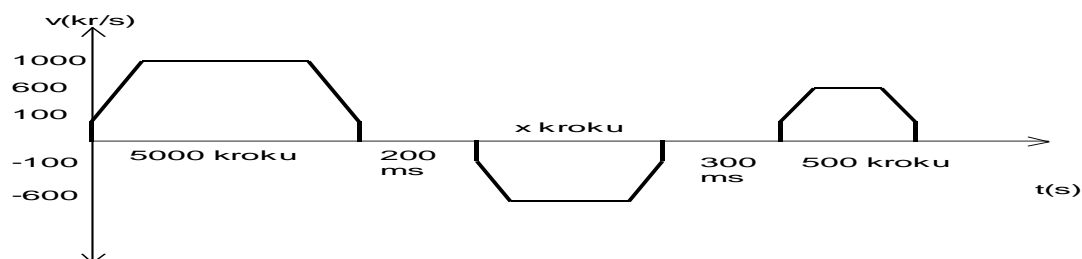
Koncové spínače nereagují



Změna BI vstupu 03 na True



Změna BI vstupu 06 na False



9.7. Příklad 7

Provede se vymazání příkazů na krokový motor, vynulování čítače absolutní polohy motoru. Nastavíme Start/Stop rychlost na 100 kroků/s, maximální rychlost na 1000 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 5000 kroků. Nastavíme total stop na BI vstup 04. Definujeme návěští číslo 1, nastavíme počet kroků vpřed na 5000 a provedeme pohyb vpřed. Provedeme prodlevu 600 ms, nastavíme počet kroků vzad na 5000 kroků a provedeme pohyb vzad. Provedeme prodlevu 1000 ms a odskočíme na návěští číslo 1. A dále se vše opakuje. Pokud během vykonávání příkazů je BI vstup 04 změněn na True je vykonávání příkazů zastaveno (při pohybu je tento ukončen po sestupné rampě).

\C100

S100 V1000 A2000

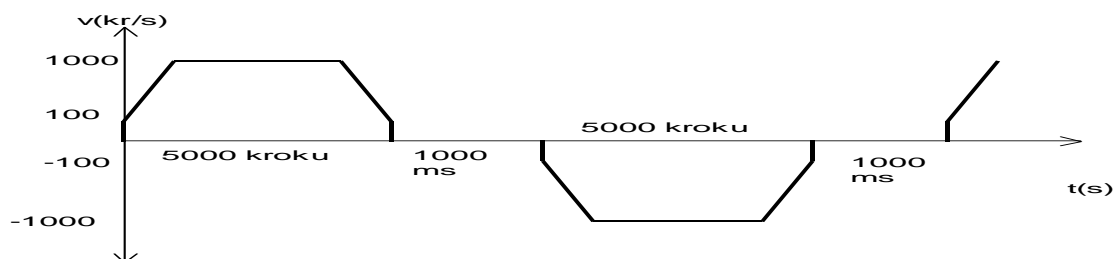
X04

@1 F5000 R

W600 B5000 R

W1000 J1

BI vstup 03 je trvale False



BI vstup 03 se změnil na True



9.8. Příklad 8

Provede se vymazání příkazů na krokový motor, vynulování čítače absolutní polohy motoru. Nastavíme Start/Stop rychlost na 100 kroků/s, maximální rychlost na 1000 kroků/s, zrychlení na 2000 kroků/s². Definujeme návěští číslo 1. Pokud je vnitřní BI proměnná 81 ve stavu False je proveden odskok na návěští číslo 6, nastaven BI výstup 05 na True, nastavíme počet kroků vzad na 2500, je odskočeno na podprogram, který začíná na návěští číslo 3, a hned je vykonán, nastavíme BI výstup 05 na False, nastavena vnitřní BI proměnná 81 na True a odskočíme na návěští číslo 1. Pokud je vnitřní BI proměnná 81 ve stavu True je nastaven BI výstup 02 na True, nastavíme počet kroků vpřed na 2500, je odskočeno na podprogram, který začíná na návěští číslo 3, a hned je vykonán, nastavíme BI výstup 02 na False, nastavena vnitřní BI proměnná 81 na False a odskočíme na návěští číslo 1. Společný podprogram začínající na návěští číslo 3 provede zadaný pohyb, vyčká prodlevu 500 ms, změní směr pohybu, provede pohyb a odskočí zpět do sady příkazů.

\C100

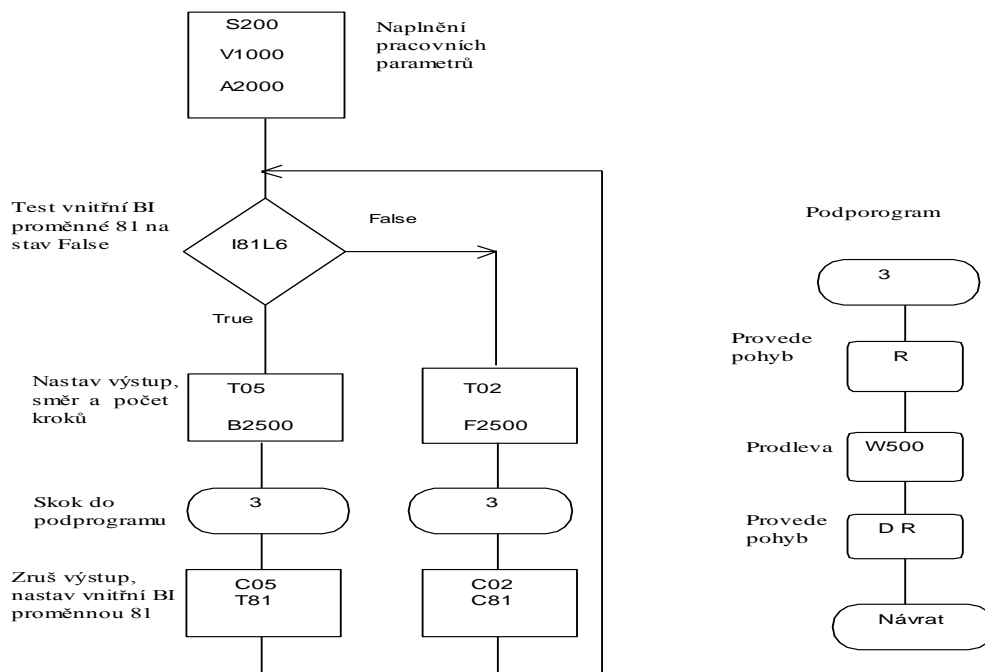
S100 V1000 A2000

@1 I81L6

T02 F2500 '3 C02 C81 J1

@6 T05 B2500 '3 C05 T81 J1

@3 R W500 D R .



9.9. Příklad 9

Provede se vymazání příkazů na krokový motor. Nastavíme Start/Stop rychlost na 100 kroků/s. Zapneme funkci nastavení binárního výstupu 05 při zahájení posledního úseku složeného pohybu. Nastavíme maximální rychlost na 500 kroků/s, zrychlení na 2000 kroků/s², počet kroků vpřed na 2000 kroků a vše uložíme do paměti na pozici 1. Změníme maximální rychlost na 2000 kroků/s a vše uložíme do paměti na pozici 2. Změníme maximální rychlost na 500 kroků/s, počet kroků vpřed na 5000 kroků a vše uložíme do paměti na pozici 3. Změníme maximální rychlost na 3000 kroků/s, zrychlení na 1000 kroků/s² a vše uložíme do paměti na pozici 4. Změníme maximální rychlost na 1500 kroků/s, počet kroků vpřed na 2000 kroků a vše uložíme do paměti na pozici 5. Změníme maximální rychlost na 500 kroků/s a vše uložíme do paměti na pozici 6. Nyní se provede zadaný složený pohyb vpřed (6 úseků pohybu). Při zahájení 6. úseku se nastaví BO05 do stavu logická jedna. Po dokončení pohybu se vyčká 500 milisekund a BO05 se nastaví do stavu logická nula.

```

\
S100 T55
V500 A2000 F2000 Y
V2000 Y
V500 F5000 Y
V3000 A1000 Y
V1500 F2000 Y
V500 Y R
W500 C05

```

