



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

## OVLADAČE K MĚNIČI PRO KROKOVÉ MOTORY

DRIVERS FOR STEPPER MOTOR DRIVER

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Marek Maralík

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Čala

BRNO 2017



# Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Marek Maralík

**ID:** 164331

**Ročník:** 3

**Akademický rok:** 2016/17

**NÁZEV TÉMATU:**

## Ovladače k měniči pro krokové motory

### POKYNY PRO VYPRACOVÁNÍ:

Úkolem práce je vytvoření ovladačů k měniči pro krokové motory Standa série 8SMC1 v prostředí LabVIEW nebo C/C++. Zadáni lze shrnout do těchto bodů:

1. Nastudujte a popište měniče Standa dostupné na UAMT, zaměřte se zejména na možnosti rozhraní použitého k jejich ovládání.
2. Uvedte strukturu a možnosti již vytvořených ovladačů od výrobce.
3. Navrhněte vylepšení ovladačů nebo jejich náhradu tak, aby bylo možné najednou ovládat více os a byla zachována ostatní funkcionality.
4. Realizujte navržené ovladače v prostředí LabVIEW nebo C/C++ a vytvořte také obslužnou aplikaci. Kód patřičně zdokumentujte.
5. Otestujte funkčnost kódu z bodu 4 a diskutujte silná a slabá místa.

### DOPORUČENÁ LITERATURA:

[1] 8SMC1-USBhF User Manual. Standa Ltd [online]. 2011. Dostupné z:  
<http://www.standa.lt/files/usb/8SMC1-USBhF%20User%20Manual.pdf>

**Termín zadání:** 6. 2. 2017

**Termín odevzdání:** 29.5.2017

**Vedoucí práce:** Ing. Martin Čala

**Konzultant:**

doc. Ing. Václav Jirsík, CSc.  
předseda oborové rady



### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

## **Abstrakt**

Bakalářská práce se věnuje problematice ovladačů k měniči pro krokové motory. V úvodu práce jsou popsány základní principy krokových motorů a způsoby jejich řízení. Druhá část se věnuje popisu měniče Standa 8SMC1-USBhF-B2. Třetí část se zabývá možnostmi ovládání měniče navrženými od výrobce. Součástí práce jsou vylepšené ovladače výrobce a nově navržené funkce, které zachovávají funkčnost a zjednodušují práci s ovladači. K dispozici je základní popis modifikovaných a nových funkcí.

## **Klíčová slova**

Krokový motor, měnič Standa 8SMC1 – USBhF-B2, ovladače krokového motoru, LabView

## **Abstracts**

The bachelor thesis deals with the problems of the drivers for the stepper motor controller. At the beginning of the thesis are described basic principles of stepping motors and ways of their control. The second part deals with the description of Standa 8SMC1-USBhF-B2. The third part discusses options of controlling proposed by the manufacturer. Part of the work is enhanced manufacturer's drivers and newly designed features that keep functionality and simplify driver operation. A basic description of modified and new features is available.

## **Keywords**

Stepper Motor, Standa 8SMC1 - USBhF-B2, Stepper Motor Controllers, LabView

**Bibliografická citace mé práce:**

MARALÍK, M. *Ovladače k měniči pro krokové motory*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 66 s. Vedoucí bakalářské práce Ing. Martin Čala.

## **Prohlášení**

„Prohlašuji, že svou závěrečnou práci na téma Ovladače k měniči pro krokové motory jsem vypracoval samostatně pod vedením vedoucího *bakalářské* práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne **29. května 2017**

.....  
podpis autora

## **Poděkování**

Děkuji vedoucímu *bakalářské* práce Ing. Martinu Čalovi. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne **29. května 2017**

.....  
podpis autora

# Obsah

Úvod .....	9
1 Krokové motory .....	10
1.1.1 Typy krokových motorů.....	10
1.2 Ovládání krokových motorů.....	13
1.2.1 Princip řízení bipolárního dvoufázového motoru.....	14
1.2.2 Princip řízení unipolárního krokového motoru.....	15
1.2.3 Metody řízení krokových motorů.....	16
2 Měnič STANDA 8SMC1.....	20
2.1 Popis měniče Standa 8SMC1- USBhF .....	20
2.1.1 Popis desky měniče Standa 8SMC1-USBhF.....	21
2.2 Popis měniče Standa 8SMC1 – USBhF-B2 .....	22
3 Ovládání měniče Standa 8SMC1-USBhF-B2 .....	24
3.1 Aplikace SMCView .....	24
3.1.1 Popis aplikace SMCView .....	25
3.2 MicroSMC.....	28
3.2.1 Dynamická knihovna USMCDLL.dll.....	29
3.3 LabView ovladače od výrobce .....	30
3.3.1 Struktura LabView ovladačů .....	30
3.3.2 Popis skupiny funkcí.....	32
3.4 Nedostatky ovladačů výrobce u ovladačů v LabView.....	34
4 Vylepšení ovladačů od výrobce.....	35
4.1 Odstranění globálních proměnných.....	35
4.1.1 Hlavní struktura MAX.....	36
4.2 Modifikované funkce .....	36
4.2.1 Hlavní funkce (Main Functions).....	36
4.2.2 Užitečné funkce (Helpful Functions).....	39
4.2.3 Kalibrační funkce (Calibration Functions).....	41
4.2.4 Funkce pro nahrání (Load) .....	42
4.2.5 Funkce sloužící k uložení (Save) .....	43
4.2.6 Ostatní funkce (Other Function).....	43

4.3	Nově přidané funkce.....	46
4.3.1	Jednoduché funkce BASIC.....	47
4.3.2	Jednoduché EXTENDED.....	48
4.3.3	Polymorfní funkce .....	49
4.3.4	Polymorfní funkce Extended .....	54
4.4	Ukázka vytvoření jednoduchého ovládání jednoho krokového motoru....	59
4.5	Modifikace příkladu umožňující ovládání více krokových motorů.....	61
5	Porovnání Vylepšených ovladačů s ovladači od výrobce.....	63
6	Závěr .....	64
7	Literatura.....	65
8	Seznam příloh .....	66



# ÚVOD

S rozvojem automatizace a robotiky vzniká čím dál větší potřeba po pohonech, které se pohybují po pevně daných úsecích, umí nastavit přesnou polohu a následně ji držet. Těmto požadavkům skvěle vyhovují krokové motory. Ovládání krokových pohonů, ale není úplně triviální záležitost. Proto je potřeba pro jejich ovládání měničů, které potřebují kvalitní ovladače.

Tato práce se zabývá modifikací ovladačů výrobce při zachování stávající funkcionality a přidáním nových funkcí, které usnadňují práci s ovladači.

Úvodní kapitola je věnována základním principům krokových motorů a způsobům jakým, je lze řídit. Další kapitola se věnuje popisu měniči Standa 8SMC1-USBhF-B2. V kapitole číslo tři jsou popsány způsoby ovládání měniče a podrobnější pohled na ovladače od výrobce v LabView. Následující kapitola se věnuje návržení a vylepšení již stávajících ovladačů. Navržené ovladače umožňují ovládání více krokových motorů a jejich součástí jsou také vzorové aplikace.

V předposlední kapitole je porovnání ovladačů a v závěti shrnují výsledky práce.

# 1 KROKOVÉ MOTORY

*„Krokový motor je speciální synchronní motor, jehož funkční pohyb (otáčení hřídele) není plynulý, ale je složen z dílčích stejně velkých pohybů (natočení), tzv. kroků. [4]*

Jak už je z názvu motorů patrné, krokové motory jsou charakterizovány tím, že se pohybují po daných krocích. Vždy se pootočí o přesný úhel a následně se zastaví, tento úhel nazýváme jako úhel kroku  $\alpha$ . Běžné velikosti kroků jsou v rozmezí 30 až 1.8 stupně (30°, 15°, 7.5°, 5°, 2.5°, 1.8°). Aby se krokový motor mohl pohnout, je potřeba, aby dostal elektrický impuls od ovládacího prvku nebo kombinaci impulsů, pokud se jedná o motor s několika fázemi. Stejně jako u ostatních druhů elektrických pohonů, tak se i krokové motory skládají z rotoru a statoru.

Zatímco u klasických elektrických pohonů je klíčový moment a úhlová rychlost, u krokových motorů hraje primární roli úhel natočení a statický vazební moment (moment který drží krokový motor na místě). Z toho vyplývá, že krokové motory jsou určeny k polohovému ovládní. Mezi výhody patří jednoduchost, přesnost a také skutečnost, že ovládací prvek nemusí znát přesnou polohu krokového motoru pro určení jeho orientace. Díky těmto vlastnostem se používají v zařízeních, jako jsou třeba výrobní roboti, CNC obráběcí stroje, 2D a 3D tiskárny, analogové hodiny apod. [1, 2, 3, 4]

## 1.1.1 Typy krokových motorů

V současných aplikacích se můžeme setkat zejména se třemi druhy krokových motorů [4]

- **Krokové motory s proměnlivou reluktancí** – výhodné pro možnost velmi malých kroků ovšem s nízkým točivým momentem.
- **Krokové motory s permanentními magnety** – poskytují vysoký točivý moment, ale krok je ve srovnání s reluktančním motorem větší.
- **Hybridní krokové motory** – konstrukčně se jedná o kombinaci předchozích dvou typů s cílem dosažení výhod obou zmíněných typů. Malých kroků a zároveň vysokého točivého momentu.

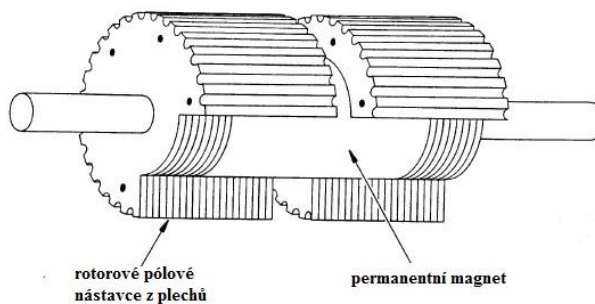
## Hybridní krokové motory

Hybridní krokové motory svou konstrukcí kombinují reluktanční motory s motory s permanentními magnety a dosahují tak výhod obou typů. Poskytují malý úhel kroku a přitom vysoký točivý moment. Příklad hybridního motoru je na Obr. 1.1. Hybridní motory jsou nejčastěji používanými krokovými motory, ale ve srovnání s motory s permanentními magnety mají vyšší cenu. Další nevýhodou je i větší velikost vynucená konstrukcí. [2, 3, 5]



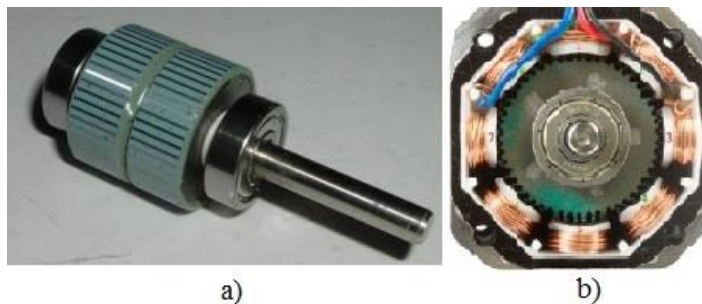
Obr. 1.1 Hybridní krokový motor 42BYGHM810 [5]

Částečný řez rotorem hybridního motoru je znázorněn na Obr. 1.2. Rotor se skládá z permanentního válcového magnetu a je osazen pólovými nástavci se zuby po obvodu.



Obr. 1.2 Částečný řez rotorem hybridního krokového motoru [1]

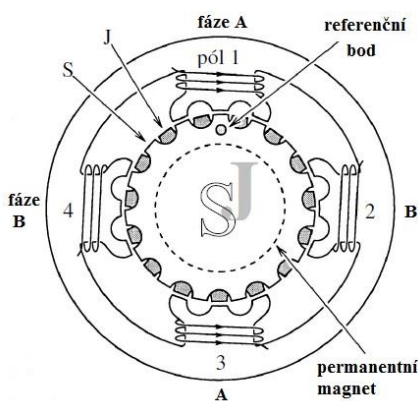
Na Obr. 1.3a je fotografie rotoru krokového motoru. Celkový pohled na hybridní krokový motor je na Obr. 1.3b.



Obr. 1.3 Hybridní krokový motor [6, 5]

Princip fungování lze dobře vysvětlit na Obr. 1.4. Na tomto obrázku je znázorněn dvoufázový hybridní motor. Součástí statoru jsou čtyři vyjádřené póly, každý se třemi zuby. Vinutí pólu 1 a 3 jsou spojeny do série a tvoří fázi A. V nákresu právě prochází proud fáze A a stator vytváří magnetické pole, které udržuje rotor v poloze, kdy zuby severního magnetu rotoru jsou v poloze nejbližší pólu 1 a stejně tak zuby jižního pólu jsou co nejbližší pólu 3. V následujícím kroku je fáze A odpojena od zdroje a napájena je fáze B. Směr proudu fáze B přitom rozhodne o směru pootočení rotoru. Rotor se pootočí o čtvrtinu zubového rozestupu. Postupným zapojováním a odpojováním fází v dané polaritě dochází k otáčení rotoru požadovaným směrem. V praxi je častěji než čtyřpólový motor používán motor osmipólový, který při dvoufázovém zapojení má 200 kroků na jednu otáčku. To odpovídá úhlu kroku  $1,8^\circ$ .

[1]



Obr. 1.4 Dvoufázový hybridní krokový motor [1]

### Krokový motor použitý v lineárním posunu Standa

Pro testování ovladačů měniče byl k dispozici lineární posun Standa 8MT167-100C28. Součástí posunu je hybridní krokový motor s úhlem kroku  $1,8^\circ$ . Hodnoty motoru jsou uvedeny v Tab. 1.1, krokový motor na Obr. 1.5.

Statický vazební moment je maximální moment, kterým můžeme působit na hřídel nabuzeného stojícího motoru, aniž by došlo porušení magnetické vazby. [1].

Tab. 1.1 Charakteristiky krokového motoru posunu Standa 8MT167-100C28

typ	Napětí	Proud fáze,	Statický vazební moment	Statický vazební moment nenabuzeného motoru,	Odpor fáze	Induktance	Kroků na otáčku (Úhel)
	[V]	[A]	[Nm]	[mNm]	[ $\Omega$ ]	[mH]	[ $^\circ$ ]
bipolární	4,56	0,67	0,106	3,75	$6,8 \pm 15\%$	$4,8 \pm 20\%$	200 ( $1,8^\circ$ )



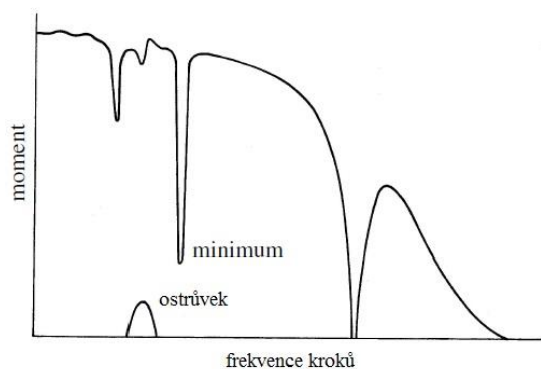
Obr. 1.5 Lineární posun Standu 8MT167-100C28

## 1.2 Ovládání krokových motorů

Krokové motory se ovládají pomocí stejnosměrných elektrických impulsů, které generuje ovládací prvek krokového motoru. Základní vlastností ovládací jednotky je schopnost napájení jednotlivých fází vinutí stejnosměrným proudem s možností velmi rychlého spínání a změny polarity napětí.

Kapitola se zaměřuje na nejčastěji používané dvoufázové krokové motory, tedy na motory hybridní a s permanentními magnety. Pro samotnou volbu ovládání je klíčová skutečnost, zda se jedná o bipolární nebo unipolární stroj. Rozdíl je v konstrukci vinutí pólů stroje a vyvedení vodičů budícího vinutí. Jednoduše lze poznat bipolární dvoufázový motor podle vyvedení dvou vodičů. Unipolární motory mají šest vyvedených vodičů viz Obr. 1.7b. Schématické zapojení pro bipolární a unipolární krokové motory je na Obr. 1.7.

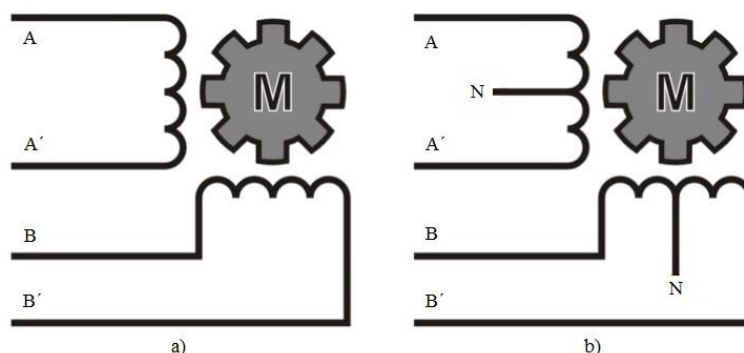
Při samotném ovládání krokových motorů a návrhu jejich použití v praxi je třeba respektovat statické a dynamické vlastnosti stroje vzhledem k jeho limitům a omezením. Na Obr. 1.6 je zobrazen příklad skutečné momentové charakteristiky krokového motoru. Z průběhu závislosti momentu na frekvenci je vidět, že při určitých frekvencích je velikost momentu ovlivněna rezonancí a nestabilitou. [1]



Obr. 1.6 Příklad skutečné provozní charakteristiky krokového motoru [1]

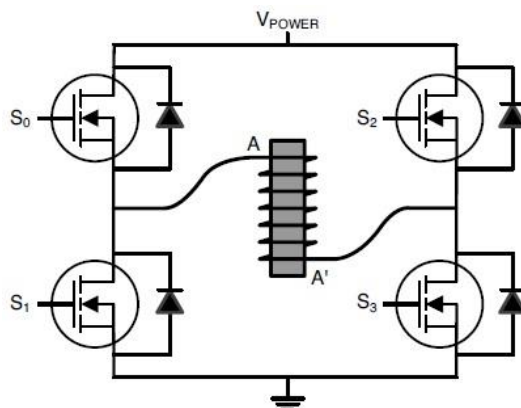
### 1.2.1 Princip řízení bipolárního dvoufázového motoru

Bipolární krokový motor má protilehlé cívky v motoru zapojeny tak, že při napájení fáze jsou tyto cívky buzeny a vytvářejí magnetické pole obrácené polarity. Motor poskytuje vyšší krouticí moment, ale řídicí jednotka je ve srovnání s řízením unipolárního stroje složitější a vzhledem k zapojení dvojnásobného počtu cívek má motor větší spotřebu. Schematické zapojení bipolárního motoru je na Obr. 1.7a.



Obr. 1.7 Schéma vinutí motoru a) bipolárního, b) unipolárního [7]

Pro řízení je třeba zajistit spínání napětí na svorkách A-A' (respektive B-B') v polaritě dle potřeby. Pro zajištění této funkce je pro dvoufázový motor možné použít dva H můstky složené z tranzistorů. Schéma H můstku je na Obr. 1.8. Směr proudu je ovládán přivedením napětí na svorky S<sub>0</sub>, S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>. Jsou-li napájeny svorky S<sub>0</sub> a S<sub>3</sub> proudí proud vinutím ve směru A-A' a vzhledem k orientaci vinutí na obrázku vytvoří sever na pólu A a jih na pólu A'. Při vypnutí svorek S<sub>0</sub> a S<sub>3</sub> a napájení svorek S<sub>1</sub> a S<sub>2</sub> prochází proud opačným směrem – pól A je jižní a pól A' severní. [3, 6]



Obr. 1.8 Schéma H můstku pro řízení fáze bipolárního krokového motoru [3]

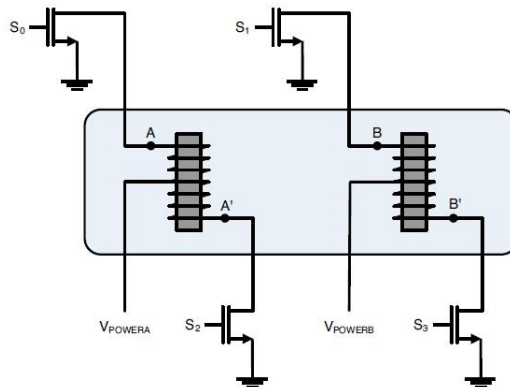
H můstek je možné vytvořit z jednotlivých součástek. Vzhledem k množství prvků a propojení je ovšem výhodnější použít integrované obvody, které jsou běžně dostupné.

## 1.2.2 Princip řízení unipolárního krokového motoru

Výhodou unipolárního řízení krokového motoru jsou nižší nároky na řídicí jednotku. Ve srovnání s bipolárním řízením uvedeným v předchozí kapitole, kde k řízení dvou fází bylo zapotřebí dvou H-můstků s celkem osmi tranzistory, si při unipolárním řízení vystačíme se čtyřmi tranzistory. Schéma vinutí je na obrázku Obr. 1.7b. Motor vzhledem k napájení pouze jedné cívky má nižší spotřebu ovšem tomu odpovídá i menší krouticí moment.

Napájení je přivedeno na místo spojení vinutí A, A' a B, B' (na Obr. 1.7b znázorněno písmenem N). Konec každého vinutí je přiveden na tranzistor, který při sepnutí propojí vodivou cestu se zemí. Postupným spínáním tranzistorů prochází proud směrem od N přes danou cívku.

Na Obr. 1.9 je schéma se zapojením vinutí, napájením a řídicími tranzistory. [3, 6]



Obr. 1.9 Schéma zapojení vinutí unipolárního motoru [4]

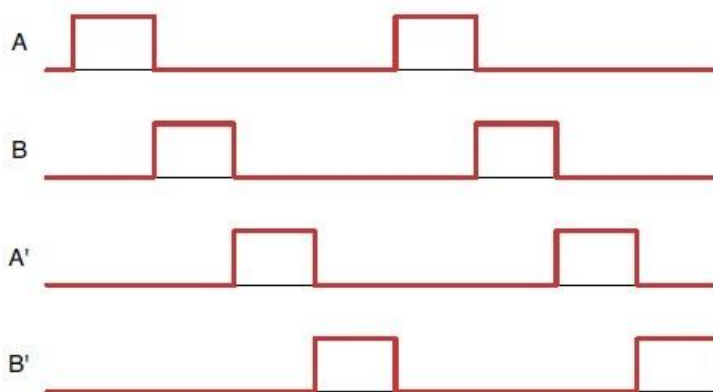
### 1.2.3 Metody řízení krokových motorů

Způsoby řízení krokových lze rozdělit na:

- řízení s plným krokem se zapojením jedné fáze,
- řízení s plným krokem se zapojením dvou fází,
- řízení s polovičním krokem
- řízení mikrokrokové

#### Řízení s plným krokem se zapojením jedné fáze

Nejjednodušším způsobem řízení je řízení za pomoci spínání vždy jedné fáze krokového motoru v určené polaritě na Obr. 1.10.



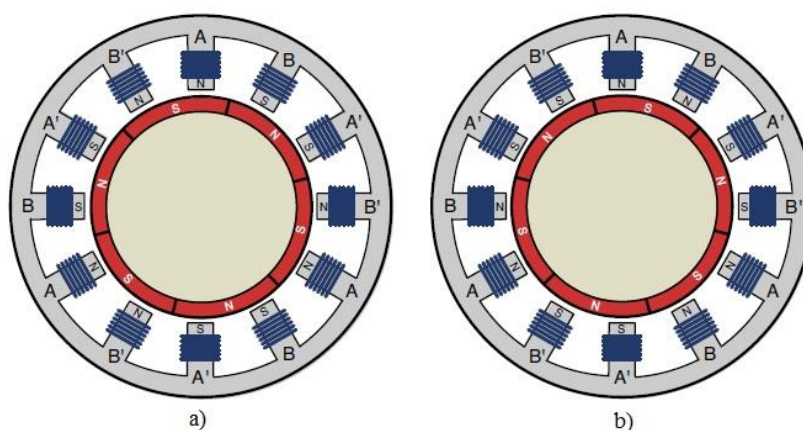
Obr. 1.10 Průběh spínání při řízení plným krokem jedné fáze [3]

Krok při tomto způsobu řízení je právě jmenovitá hodnota kroku motoru. U krokového motoru, použitém v posuvu Standa 8MT167-100C28, který je tématem druhé části této práce je to právě  $1,8^\circ$ . [3]



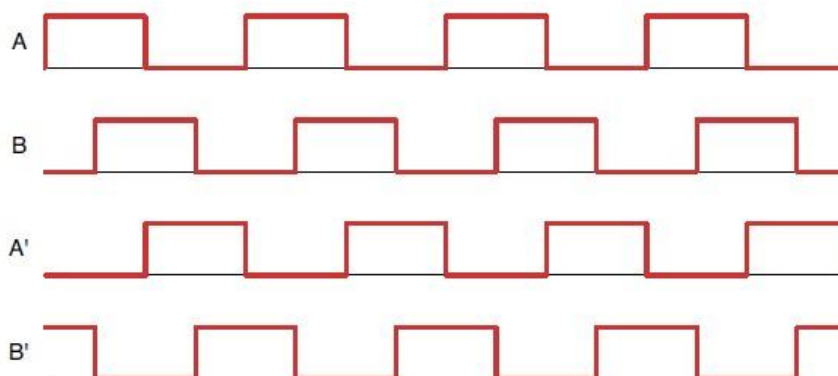
## Řízení s plným krokem se zapojením dvou fází

Při metodě ovládání krokového motoru se zapojením dvou fází jsou napájeny dvě sousedící fáze ve stejný čas. Vytvořené elektromagnetické pole má větší intenzitu a díky této skutečnosti dosahuje točivý moment vyšší velikosti. Poloha středu magnetů rotoru v rovnovážné poloze je uprostřed mezi dvěma shodně nabuzenými póly (Obr. 1.11), a přestože motor odebírá dvojnásobný proud, vzroste točivý moment jen o 30 až 40 %. [3, 6]



Obr. 1.11 Metoda řízení s plným krokem se zapojením dvou fází [3]

Sekvence spínání buzení vinutí je na Obr. 1.12. Po nabuzení fáze A dojde v půlce intervalu k sepnutí fáze B. Stejně jako v předchozím případě odpovídá velikost kroku motoru Standa 8MT167-100C28 jmenovité hodnotě  $1,8^\circ$ .



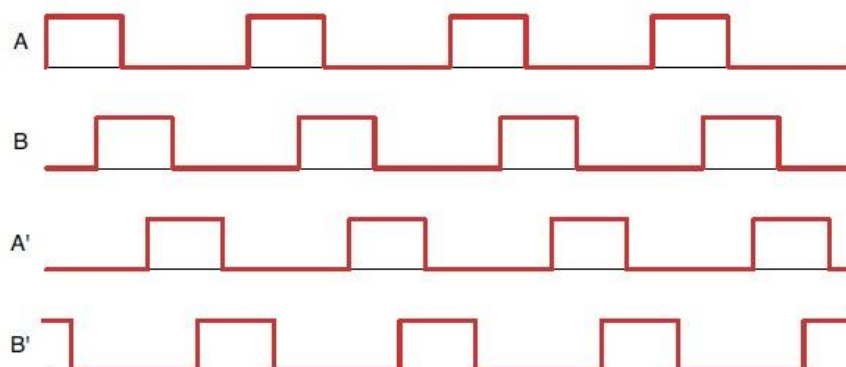
Obr. 1.12 Průběh spínání při řízení plným krokem dvou fází [3]

## Řízení s polovičním krokem

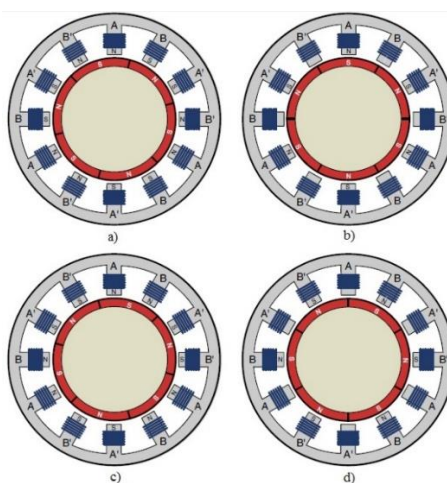
Kombinací předchozích dvou metod řízení, střídavým spínáním dvou a jedné fáze dosáhneme kroku, který odpovídá polovině plného kroku motoru. Situace pro krokový motor s permanentními magnety je znázorněna na Obr. 1.14. V pozici dle

Obr. 1.14a jsou napájeny obě fáze dvoufázového motoru a severní pól permanentních magnetů směřuje do středu mezi dva sousedící shodně polarizované póly. Po vypnutí fáze B dojde k pootočení o půlkrok a střed permanentního magnetu je naproti středu nabuzené fáze A (Obr. 1.14b). V dalším kroku dojde k sepnutí fáze B, ale s opačnou polaritou proudu a rotor se opět otočí o půlkrok viz situace Obr. 1.14c. Následuje vypnutí fáze A a otočení rotoru dle Obr. 1.14d.

Průběh spínání jednotlivých fází je na Obr. 1.13. [3, 7]



Obr. 1.13 Průběh spínání při řízení s polovičním krokem [3]

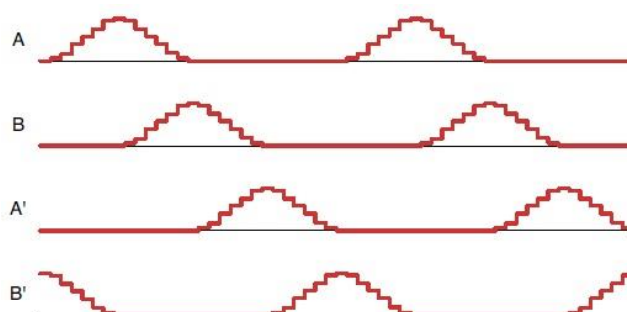


Obr. 1.14 Metoda řízení krokového motoru polovičním krokem [3]

Využitím popisovaného způsobu řízení zmenšíme krok motoru Standa 8MT167-100C28 z  $1,8^\circ$  na  $0,9^\circ$ . Nevýhodou tohoto způsobu řízení je pokles točivého momentu při buzení jedné fáze o 20 %. Kolísání momentu lze eliminovat současným zvýšením budícího proudu fáze, která je buzena při vypnutí druhé fáze. [3]

## Řízení mikrokrokové

S cílem co nejhladšího chodu krokového motoru je možné použít mikrokrokové řízení. Hladký chod je v tomto případě vykoupěn řadou nevýhod. Při provozu dochází k poklesu točivého momentu téměř o 30 % a velké nároky jsou kladeny na řídicí jednotku, která musí na rozdíl od předchozích metod napájet fáze podstatně větším množstvím pulzů o různých velikostech. Vzhledem k náročnosti dochází i k poklesu možné rychlosti otáčení motoru a snižuje se spolehlivost neboli přesnost mikro pozice. Principem je rozdělení proudových impulsů buzení na podstatně kratší impulsy menší velikosti. Průběh je znázorněn na Obr. 1.15. Používané hodnoty rozdělení jsou 8, 64 a 256. Pokud je energetický signál rozdělen na 256 částí může krokový motor s krokem  $1,8^\circ$  vykonávat posun o  $1,8^\circ/256 = 0,007^\circ$ . Přesnost pozice se s vyšším rozdělením snižuje. [3]



Obr. 1.15 Metoda mikrokrokového řízení krokového motoru [3]

## 2 MĚNIČ STANDA 8SMC1

Informace o měniči jsou čerpány z materiálů a manuálů k měniči, které nabízel výrobce a velká část údajů je z nich převzata. [8, 9, 10]

Jedná se o kompaktní, funkční, nízkonákladové, all-in-one zařízení, navržené pro řízení jednoho bipolárního krokového motoru. Měnič 8SMC1-USBhF je na Obr. 2.1.



Obr. 2.1 Měnič 8SMC1-USBhF [10]

### 2.1 Popis měniče Standa 8SMC1- USBhF

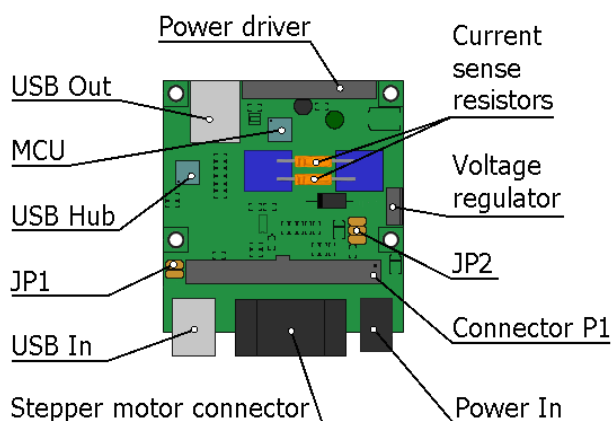
Měnič lze ovládat za pomoci lokálního nebo vzdáleného ovládání. Lokální ovládání bývá realizováno dvěma knoflíky a třemi LED diodami. Vzdálené ovládání bývá řešeno skrze USB rozhraní. Samotný měnič zahrnuje USB controller, tří portový USB hub umístěný na desce, plně integrovaný chip MCU (micro controller unit) a sinusové mikro krokové řízení bipolárního motoru pomocí pulzně šířkové modulace (PWM) typu chopper. Měnič má vbudované funkce ochrany proti přehřátí, přepětí, zkratu a řeší krátkodobé výpadky napájení. Mezi další funkce patří naváděcí algoritmus, inkrementální (kvadrurní) enkodér nebo podpora rozlišení senzoru, přídatné vstupy a výstupy pro synchronizaci, programovatelné limitní spínače.

Vbudovaný chip MCU umožňuje ovládat všechny parametry jako pozice, rychlost, směr, rozlišení, velikost řídicího proudu, .... Tyto parametry lze také uložit do flash paměti MCU nebo do počítače a část z nich lze měnit za chodu programu. Proud pro jednotlivé krokové motory se nastavuje pomocí dvou proudově citlivých rezistorů. V zadržovacím (HOLD) módu je potenciál omezení proudu 40 %. Napájení logiky měniče 8SMC1-USBhF může být řešeno až třemi různými způsoby. Logika může být napájena přímo z USB, z externího zdroje napětí 7-12V nebo externího

zdroje stejnosměrného napětí +5V. Měnič také obsahuje nouzový vypínač, rozlišovací senzor a vstupy pro dva limitní spínače.

Pro komunikaci s deskou 8SMC1-USBhF lze použít 40pinový víceúčelový konektor nebo USB vstup typu B. Stejnosměrný vstup slouží k napájení krokového motoru a 15pinový D-Sub k jeho připojení. Dále jsou také k dispozici dva konektory USB typu A jako výstup. Dva výstupní a jeden vstupní USB porty umožňují zapojit několik měničů za sebou. Na jednoho USB hostitele lze za použití stromové struktury připojit až 30 zařízení bez použití přídatných USB hubů.

### 2.1.1 Popis desky měniče Standa 8SMC1-USBhF



Obr. 2.2 Rozložení na desce měniče [9]

- **Power driver** Zajišťuje ovládání krokového motoru. Sinusové mikro krokové řízení bipolárního motoru.
- **USB Out** Dva výstupní konektory USB typu B, které dále umožňují kaskádní řazení.
- **MCU** Chip MCU si udržuje v paměti aktuální data, zajišťuje fungující rozhraní mezi počítačem a krokovým motorem. Data do počítače posílá na vyžádání počítače. Konkrétně se jedná o Full Speed USB, 16K ISP FLASH Microcontroller.
- **USB hub** Zajišťuje tři downstream porty se specifikací USB 1.1 a jeden upstream port. Jeden downstream port je zapojený do MCU, upstream port je připojen do USB In a Connector P1. Porty se na základě připojeného zařízení automaticky nastaví na full-speed nebo low-speed.

- **JP1** Zapíná a vypíná nouzový vypínač. V poloze směrem nahoru je zapnutý, v poloze směrem dolů je vypnutý
- **USB In** Jeden vstupní USB konektor typu B.
- **Stepper motor connector** Konektor pro připojení krokového motoru. Jedná se o 15pinový D-Sub (female). V tomto konektoru jsou také všechny přídatné kabely jako je nouzový limitní spínač, limitní spínače, rozlišovací senzor a odpovídající napájení.
- **Power In** Nízkonapěťová zástrčka pro vstupní napájení. Nabízí dvě možnosti zapojení. První je napěťové napájení 7-12 V, které bude napájet krokový motor a logiku. Druhou možností je napájení 7-40 V, v tomto případě bude napájen pouze krokový motor. Za pomoci JP2 a Connector P1 je potřeba zvolit správné napájení.
- **Connector P1** Víceúčelový 40pinový konektor, navržený pro embedded aplikace. Jsou zde duplikovány konektory USB In, USB Out, Stepper motor connector and Power In.
- **JP2** Přepínač, který slouží k nastavení režimu napájení pro logiku. Jsou zde 3 režimy pro napájení. Napájení přímo z USB, z externího zdroje napětí 7-12V nebo externího zdroje stejnosměrného napětí +5V.
- **Voltage regulator** Používá se v případě, že je logika napájena externě 7-12 V pro logiku
- **Current sense resistors** Dva rezistory k měření proudu, které lze měnit podle typu motoru (nastavuje se vhodný úbytek napětí).

## 2.2 Popis měniče Standa 8SMC1 – USBhF-B2

Pro vypracování práce a testování byl k dispozici měnič 8SMC1-USBhF-B2 (Obr. 2.3). Tento měnič se skládá ze tří kompaktních měničů 8SMC1-USBhF. Využívá totiž možnosti měniče 8SMC1-USBhF, který díky tříportovému USB hubu (1 vstup, 2 výstupy) umožňuje kaskádní zapojení. Měnič 8SMC1-USBhF-B2 umožňuje ovládání až tří bipolárních krokových motorů (3 osy). K dispozici bylo pouze ovládání dálkové, skrze USB rozhraní. Všechny měniče byly zavřené uvnitř krabice. Krabici

jsem nerozebíral. Výrobce nabízel pouze manuály a schémata zapojení k měniči 8SMC1-USBhF.

Pro komunikaci se všemi měniči slouží vstup USB typu B. Dále jsou k dispozici tři 15pinové konektory D-Sub (female), umožňující ovládání jednotlivých krokových motorů. K připojení napájení slouží konektor KPPX-4P(female). Červená LED dioda indikuje připojení napájení. Logika je napájena zkrze USB port.



Obr. 2.3 Měnič Standa 8SMC1-USBhF-B2

## **3 OVLÁDÁNÍ MĚNIČE STANDU 8SMC1-USBHF-B2**

Ve třetí kapitole jsou využity informace nabyté při seznamování s obslužným programem od výrobce a část údajů je převzata z uživatelského manuálu. [9]

Ovládání je u měniče 8SMC1-USBhF-B2 možné pouze vzdáleně, skrze USB 1.1 rozhraní. Výrobce nabízí k ovládání měniče aplikaci s grafickým rozhráním SMCView. Tato aplikace je založena na VI knihovnách vytvořených v LabView a k jeho spuštění je potřeba mít nainstalované patřičné NI VISA ovladače. Další možností je ovládat měnič přímo vývojářskými nástroji v LabView nebo v jazycích C, C++, Basic Delphi, Matlab. Ovládat měnič by mělo jít i za pomoci mobilního zařízení pracujícího pod Windows Mobile 5.0 a vyšší.

V případě práce s programovacími jazyky C, C++, Basic, Delphi, Matlab nebo s Windows Mobile je nutná instalace MicroSMC ovladačů, které nabízí výrobce. Přestože je možné mít v počítači nainstalováno více USB ovladačů, měnič 8SMC1-USBhF může fungovat pouze s jedním v danou chvíli. Při změně prostředí je tedy nezbytné USB ovladače přeinstalovat (NI VISA – LabView, MicroSMC - C, C++, apod.).

Všechny softwarové nástroje by měly fungovat pod operačními systémy Windows 2000/XP/Vista/7. Windows 8 a 8.1 a fungují pouze pokud mají měniče nainstalován nejnovější firmware v.2505. Na Windows 10 je problém s digitálním podpisem („driver signature enforcement“), který zabraňuje nainstalování ovladačů. I v případě jeho zrušení, nainstalované ovladače nabízejí pouze omezenou funkčnost.

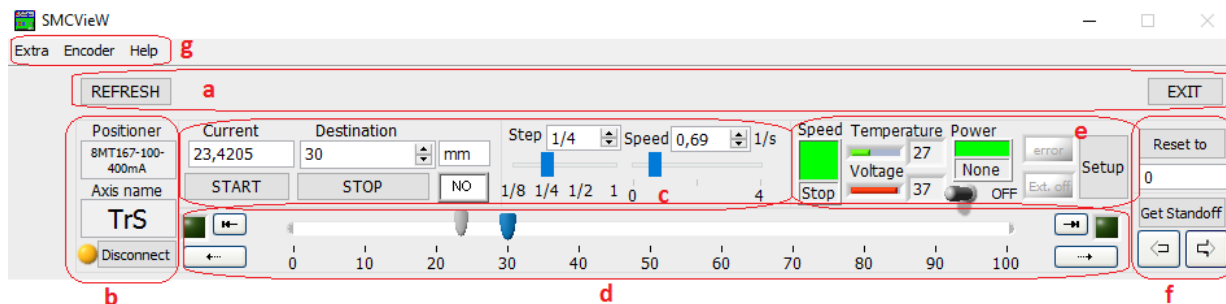
### **3.1 Aplikace SMCView**

Aplikace SMCView nabízí uživatelsky přívětivé grafické rozhraní, s jehož pomocí lze ovládat více os (krokových motorů) a měnit řadu jejich parametrů, které lze následně uložit do binárního souboru. Vzhled rozhraní aplikace SMCView (viz. Obr. 3.1). ke své činnosti vyžaduje nainstalované USB-VISA ovladače od National Instrument a nejnovější ovladače pro daný měnič. Aplikace by měla bezproblémově



fungovat pod operačními systémy Windows 2000/XP/Vista/7/8/8.1. U operačních systémů Windows 8/8.1 záleží na verzi firmwaru měniče.

### 3.1.1 Popis aplikace SMCView

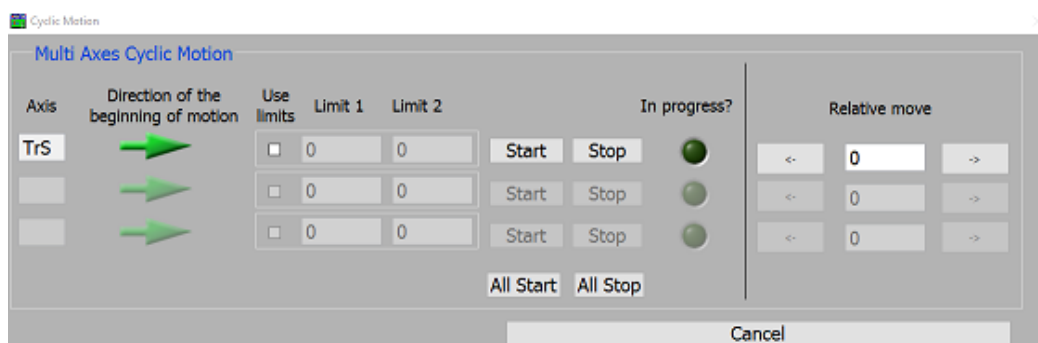


Obr. 3.1 Ovládací rozhraní aplikace SMCView

- Po stisknutí tlačítka **REFRESH** se vypíše všechny dostupné ovladatelné osy. Tlačítkem **EXIT** lze ukončit aplikaci, před jejím ukončením je potřeba, pomocí tlačítka **Disconnect** v části **b** odpojit všechny připojené zařízení.
- Ukazatel **Positioner** ukazuje, o jaký model krkového motoru se jedná a **Axis name** udává název osy. Oba tyto parametry lze měnit v okně po stisku tlačítka **Setup** v oblasti **e**. Tlačítko **Disconnect** slouží k odpojení dané osy.
- Ukazatel **Current** udává aktuální pozici, **Destination** udává žádanou pozici. Jednotku **mm** můžeme po stisku tlačítka **Setup** na kroky **step**. Tlačítkem **START** se spouští krokový motor a tlačítkem **STOP** se zastavuje. Aby se krokový motor začal pohybovat, je potřeba mít zapnuté napájení v části **e**. U ukazatele **Step** lze nastavit velikost kroku a u ukazatele **Speed** rychlost.
- Grafická reprezentace aktuální a žádané polohy, modrým ukazatelem je vyznačena poloha aktuální, šedým žádaná. Ledky na okrajích se rozsvítí při dosažení limitní hodnoty krokového motoru. Po stisku tlačítka **←** nebo **→** se začne pohybovat motor daným směrem, dokud nenarazí na konec nastaveného rozsahu. Tlačítka **|← a →|** spustí krokový motor a zastaví se, dokud se nedostanou na limitní hodnotu krokového motoru.
- Ukazatel **Speed** zobrazuje, zdali krokový motor stojí(zelená), akceleruje/zpomaluje(žlutá) nebo se pohybuje zadanou rychlostí(červená). Ukazatel **Temperature** udává teplotu měniče a **Voltage** velikost napětí měniče. **POWER** ukazuje, zdali je krokový motor bez napájení(zelená), v hold

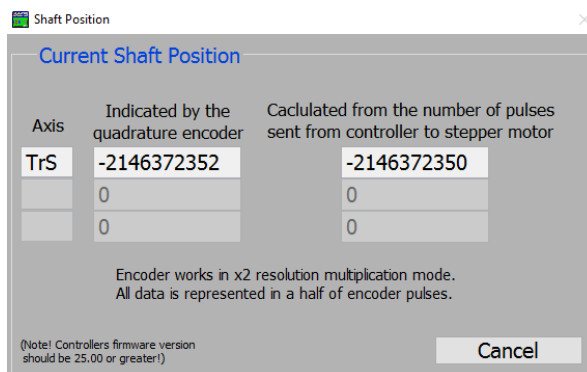
módu(žlutá) nebo plně napájen(Červená). V případě, že senzor otáček nebo inkrementální (kvadrurní) enkodér detekuje chybu, rozsvítí se kontrolka **error**. V případě sepnutí nouzového snímače se rozsvítí kontrolka **Ext. off**. **Setup** okno otevře okno s nastavením pro krokový motor, menu Setup se podrobněji věnuji v kapitole 3.2.1.

- f) Slouží k nastavení referenční pozice. Nejdříve si nastavíte pozici, kterou chcete mít jako referenční. K přesnějšímu nastavení pozice lze použít šipky **<=** a **=>**. Do kolonky pod tlačítkem **Reset to** zadáte hodnotu, kterou by měl mít referenční bod. Poté stisknete tlačítko **Get Standoff**, motor se začne pohybovat směrem k limitní hodnotě krokového motoru, po dosažení limitní hodnoty krokového motoru se vrátí zpátky na původní pozici a uloží si do paměti vzdálenost daného bodu od limitní hodnoty. Poté stisknete tlačítko **Reset to**, motor se znovu začne pohybovat směrem k limitnímu hodnotě krového motoru. Jakmile dosáhne limitní hodnoty vrátí se zpět o hodnotu, kterou zaznamenal pomocí funkce **Get Standoff**. Pozici na, které zastaví, přiřadí zadanou hodnotu pod tlačítkem **Reset to**. Pro nejvyšší přesnost se doporučuje, aby krokový motor měl nastavený co nejmenší krok, což v případě krokových motorů Standa bývá 1/8 kroku. Velikost **Get Standoff**, **Reset to** a směr, kterým se bude pohybovat krokový motor pohybovat, při použití těchto funkcí lze nastavit v nastavení **Setup**.
- g) V záložce **Extra** záložky zahrnující, aplikaci **Cyclic Motion** (Obr. 3.2), umožňující cyklický pohyb až 3 os (krokových motorů) najednou. Dále přídatné aplikace závislé na licenci, jako je například 2D Motion. A seznam licencí.



Obr. 3.2 Cyclic motion umožňující pohyb až tří os najednou

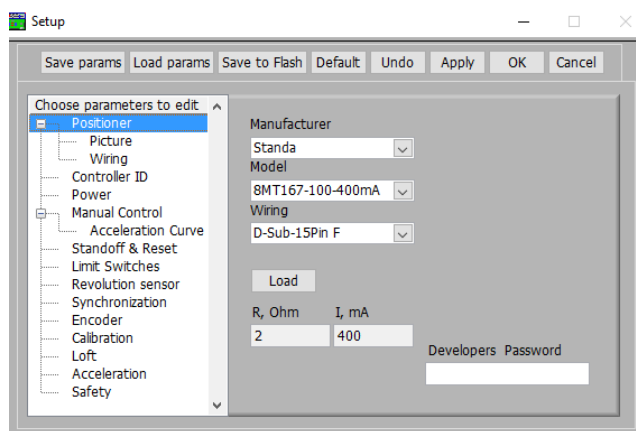
Záložka **Encoder** obsahuje položku **Encoder state**, která po rozklepnutí zobrazí okno **Shaft Position** (Obr. 3.3). Toto okno zobrazuje informace o aktuální poloze hřídele krokového motoru v polovině enkodérových impulsů.



Obr. 3.3 Current shaft position, umožňující znát aktuální polohu hřídele

### Nastavení v okně Setup

Okno **Setup** (na Obr. 3.4) nabízí nastavení spousty parametrů, které lze měnit. Pro usnadnění lze jejich specifické nastavení nahrát/uložit na pevný disk PC (**Save/Load params**) nebo uložit přímo do flash paměti měniče (**Save to Flash**).



Obr. 3.4 Nastavení v okně Setup

V záložce **Positioner** viz Obr. 3.4, lze měnit parametry jako je výrobce (**Manufacturer**), model (**Model**) krokového motoru a jeho zapojení (**Wiring**). Zobrazený odpor **R** rezistorů k měření proudu (**Current sense resistors**) a proud krokového motoru **I** mají pouze informační charakter. Klíčová je ještě záložka **Wiring**, kde lze zkontrolovat schéma zapojení. Záložka **Picture**, slouží pouze pro ilustraci vzhledu krokového motoru daného typu. Po nastavení údajů je potřeba nahrát nastavení pomocí tlačítka **Load**, po jeho stisknutí se změní všechny potřebné parametry tak aby odpovídali vybranému výrobcí, modelu a zapojení.

V záložce **Controller ID** můžeme najít unikátní číselný identifikátor ovladače a verzi firmwaru měniče, kterou můžeme updatovat. Dále zde můžeme změnit název osy **Axis Name**, která je zobrazena v hlavním panelu obrázek 3.1 oblast b.

**POWER** se zabývá power managementem měniče. Umožňuje vypnout/zapnout 40% redukci proudu v zadržovacím módu a nastavit jeho zpoždění v milisekundách, zapnutí/vypnutí zpoždění vypnutí napájení při vypnutí napájení a nastavit jeho zpoždění v milisekundách.

Dále **Setup** nabízí nastavení Manuálního ovládání(**Manual Control**) a parametrů jeho akcelerační křivky(**Acceleration Curve**), nastavení specifických parametrů pro funkce **Reset to** a **Stanoff (Stanoff & Reset)**, limitních spínačů (**Limit Switches**), senzoru otáček (**Revolution Sensor**), synchronizace(**Synchronization**), enkóderu(**Encoder**), parametrů kalibrační křivky (**Calibration**), kompenzátoru vůle (**Loft**), parametrů akcelerační křivky (**Acceleration**), maximální teploty(**Safety**). Jednotlivé parametry jsou podrobněji rozebrány v materiálech od výrobce.

## 3.2 MicroSMC

MicroSMC se využívá v případě použití programovací jazyků kompatibilních s jazykem C (C, C++, Basic Delphi, Matlab, ...). Ovladače MicroSMC se liší od NI ovladačů, jsou totiž založeny na ezusbs.sys ovladačích narozdíl od NI VISA. Proto je také potřeba v případě jejich použití přeinstalovat ovladače pro USB výstup ve správci zařízení počítače. Měnič 8SMC1-USBHF totiž může pracovat jenom s jedním USB ovladačem v daný okamžik. Dále je potřeba mít Microsoft.NET Framework 2.0 nebo vyšší.

Ovladače MicroSMC umožňují kontrolu až 32 měničů 8SMC1-USBhF zároveň z jednoho počítače, což je o 2 více než NI VISA (až 30 měničů). A jsou nezbytné pro používání dynamické knihovny USMCDLL.dll, která obsahuje základní příkazy pro krokový motor.

### 3.2.1 Dynamická knihovna USMCDLL.dll

Dynamická knihovna USMCDLL.dll nabízí všechny funkce pro kontrolu krokového motoru. Všechny funkce a deklarace jsou k dispozici v hlavičkovém souboru USMCDLL.h.

#### Dostupné funkce:

- **USMC\_Init** – funkce vyhledá připojená zařízení a inicializuje
- **USMC\_GetState** – funkce zjistí aktuální stav krokového motoru
- **USMC\_SaveParametersToFlash** – funkce uloží parametry do flash paměti měniče
- **USMC\_GetMode** – funkce vrátí strukturu stavů (USMC\_Mode) nebo výchozí strukturu
- **USMC\_SetMode** – funkce zašle do měniče parametry zařízení uvnitř struktury USMC\_Mode
- **USMC\_GetParameters** - funkce vrátí strukturu parametry (USMC\_Parameters) nebo výchozí strukturu
- **USMC\_SetParameters** – funkce nastaví parametry zařízení podle hodnot ve struktuře USMC\_Parameters
- **USMC\_GetStartParameters** – funkce vrátí strukturu parametrů potřebných ke startu krokového motoru a parametrů informativního rázu (USMC\_StartParameters) nebo výchozí strukturu
- **USMC\_Start** – funkce nastaví startovní parametry (USMC\_StartParameters) a zahájí chod krokového motoru
- **USMC\_Stop** – funkce zastaví chod krokového motoru
- **USMC\_SetCurrentPosition** – funkce nastaví hodnoty pro aktuální pozici
- **USMC\_GetEncoderState** – funkce vrátí strukturu s informacemi o aktuální pozici enkodéru
- **USMC\_GetLastErr** – funkce vrátí string s popisem jednoho poslední chyby

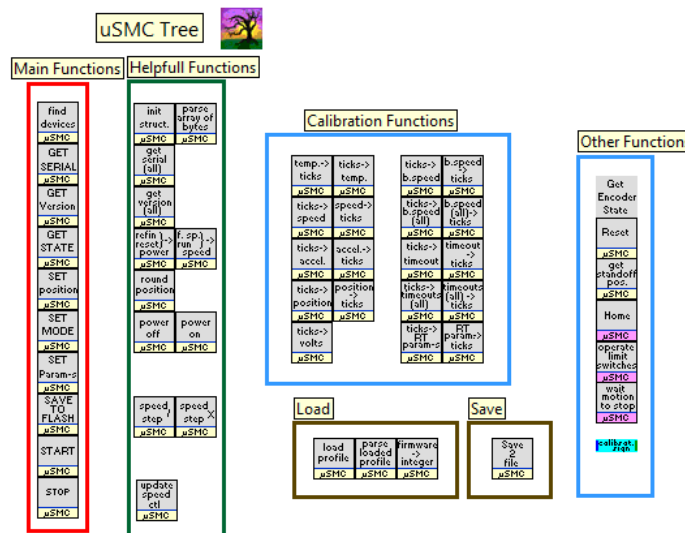
Struktury USMC jsou podrobněji popsány v manuálu výrobce v kapitole 7.6 a jsou podobné využitým strukturám v LabView.

### 3.3 LabView ovladače od výrobce

Struktura ovladačů LabView je koncipována tak, že výrobce pracuje se sadou základních virtuál instrument (VI) viz Obr. 3.5. Tyto VI jsou volně k dispozici ve vývojovém balíčku, nabízí tak možnost úpravy a vylepšení nebo použití ve vlastním kódu. Na základě těchto funkcí je postavena aplikace od výrobce SMCView. Výrobce nedává k dispozici úplně všechny funkce, které v rámci aplikace použil a nedává tak možnost na její modifikaci. Pro snadnější pochopení jsou součástí vývojového balíčku také příklady.

VI jsou rozděleny z důvodu snazší orientace do několika kategorií:

- Hlavní funkce (**Main Functions**)
- Užitečné funkce (**Helpful Functions**)
- Kalibrační funkce (**Calibration Functions**)
- Funkce pro nahrání (**Load**)
- Funkce sloužící k uložení (**Save**)
- Ostatní funkce (**Other Function**)



Obr. 3.5 Přehled funkcí výrobce

#### 3.3.1 Struktura LabView ovladačů

Ovladače výrobce jsou založeny na práci s pěti základními globálními poli struktur, které se objevují ve většině funkcí. Globální pole struktur se inicializují v rámci funkce Initialize All Structures a počet jejich prvků je dán počtem nalezených

měníčů pomocí funkce `find devices`. Základní funkce obsahují vstup a výstup pro VISA array, aby mohly pomocí VISA funkcí nahrávat a získávat informace z měniče ovládajícího krokový motor.

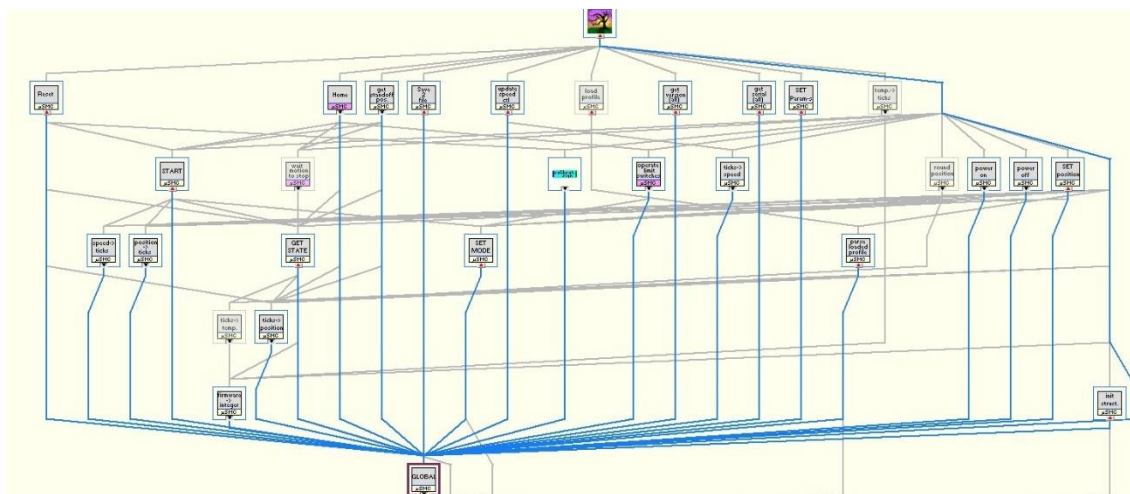
V rámci funkcí se používá 5 hlavních globálních polí struktur a ty jsou **Mode**, **Parameters**, **State**, **StartPlus**, **Position** a string **Profile Version** a pole **stringů Firmware Version**. Tyto jsou podrobněji uvedené jsou v manuálu výrobce kapitola 6.2 Structures.

### **Globální proměnné a jejich prvky:**

- **Mode** – Struktura skládající se z hodnot typu bool, rozhoduje o nastavení (vypnutí/zapnutí) důležitých prvků. Lze s ní nastavit stav limitních snímačů (logický stav, prohození logických snímačů, povolení), napájení krokového motoru, nouzové vypínání napájení, omezení proudu, rotační převodovku, enkodér, počet kroků do synchronizace, atd. K nahrání do měniče slouží funkce `Set Mode`.
- **Parameters** – Struktura základních parametrů převážně v typu u16, polovina parametrů se vztahuje k ovládání krokového motoru za použití tlačítek (u měniče Standa 8SMC1 – USBhF-B2 není lokální ovládání k dispozici). Nastavuje čas zrychlení a zpomalení, maximální teplotu, řadu parametrů týkající tlačítek (např. časový limit, rychlost), doba synchronizace, počet Ticks na jednu obrátku (musí být dělitelné 256), atd. Parametry se nahrávají do měniče za pomoci funkce `Set Parameters`.
- **State** – Struktura shrnující aktuální stav krokového motoru. Obsahuje informace o aktuální pozici, směru rotace, teplotě, třístavová indikace rychlosti, stavu napájení, stavu limitních snímačů, atd. Pro naplnění aktuálními hodnotami slouží funkce `Get State`.
- **StartPlus** - Struktura informací a parametrů, potřebných ke startu krokového motoru. Obsahuje informace o sériovém čísle krokového motoru, název jednotek, maximální rychlost, kalibrační body, typ krokového motoru, verze firmwaru měniče, atd. Mezi důležité parametry patří např. rychlost, krok, pomalý start/stop, cílová pozice. K nahrání některých parametrů do měniče slouží funkce `Start`.

- **Position** – Struktura parametrů vztahujících se k pozici v jednotkách(step/mm) a pozici v Ticks (jednotka se kterou pracuje měnič). Obsahuje parametry aktuální pozice, nejnižší/nejvyšší limit, maximální/minimální hodnotu ve vybraných jednotkách (mm/step).
- **Profile Version** -Aktuální verze programu ve které se ukládá. Tato globální proměnná je doplňková, užívá se pouze v jedné funkci.
- **Firmware Version** – Verze firmware, doplňková a vzhledem existenci stejného parametru ve struktuře StartPlus je nadbytečná.

Globální proměnné jsou umístěny ve VI Globals, hierarchie toho, kterými funkcemi prostupují je patrná na Obr. 3.6



Obr. 3.6 Hierarchie využití globálních proměnných v ostatních funkcích

### 3.3.2 Popis skupiny funkcí

Výrobce v rámci zpřehlednění umístil do složky s funkcemi VI Tree (Obr. 3.5), ve kterém rozdělil funkce vizuálně do několika kategorií podle důležitosti. Kategorie **Other Functions** byla vytvořena dodatečně z funkcí, které výrobce přidal v pozdějších fázích vývoje. Jednotlivým kategoriím je v této části práce věnován pouze nezbytně nutný prostor, protože podrobněji jsou funkce některé funkce popsány v kapitole 4.1.



## **Hlavní funkce (Main Functions)**

Hlavní funkce mají pro komunikaci s krokovým motorem na vstupu a výstupu VISA array (až na funkci `find devices`). VISA array se vytváří pomocí funkce pro nalezení všech připojených zařízení (`find devices`). V této kategorii jsou zahrnuty funkce na získání parametrů, stavů, pozice z měniče funkce GET (`Get Serial`, `Get State`, `Get version`). Získané změny se projeví v globálních strukturách `STATE` a `Position`. Dále jsou zde funkce, které nahrávají změny do měniče SET (`Set Mode`, `SET Position`, `Set Parameters`). Funkce pro spuštění a zastavení (`Start`, `Stop`) pohybu krokového motoru a funkce pro ukládání do flash paměti měniče `Save to flash`.

## **Užitečné funkce (Helpful Functions)**

Nabízí funkce pro zapnutí/vypnutí napájení krokového motoru, inicializaci 5 důležitých globálních polístruktur proměnných výchozími hodnotami, funkce pro přepočítání velikosti rychlosti na základě velikosti kroku, zaokrouhlování zadané hodnoty pozice. Aktualizace hodnot rychlosti a kroku při použití sliderů, či variace na hlavní funkce v podobě získání verze firmwaru a sériového čísla pro všechny motory.

## **Kalibrační funkce (Calibration Functions)**

Jedná se o nízkoúrovňové funkce pro přepočítání parametrů v jednotkách (teploty, napětí, pozice) na hodnotu v jednotkách měniče (`Ticks`) a naopak.

## **Funkce pro nahrání (Load)**

Funkce pro nahrání profilu z binárního souboru. Tento binární soubor lze vytvořit v rozhraní `SMCView` nebo pomocí funkce `Save to File`. Nahraný profil v sobě obsahuje všechny informace pro naplnění všech globálních struktur.

## **Funkce sloužící k uložení (Save)**

Zahrnuje pouze jednu funkci pro uložení nastavených parametrů (`Mode`, `Position`, `StarPlus`, `Parameters`) do binárního souboru. Soubor je potřeba ukládat s příponou `.sav`.

### **Ostatní funkce (Other Function)**

Zahrnuje funkce, které byly přidány při aktualizaci ovladačů výrobce nebo se jedná o nízkoúrovňové funkce. Řada z těchto funkcí nefunguje korektně a nepracuje dle popisu výrobce.

## **3.4 Nedostatky ovladačů výrobce u ovladačů v LabView**

Globální proměnné zhoršují přehlednost programu a bez bližšího prozkoumání jednotlivých funkcí není jasné, kde všude se právě provedené změny v globálních proměnných projeví.

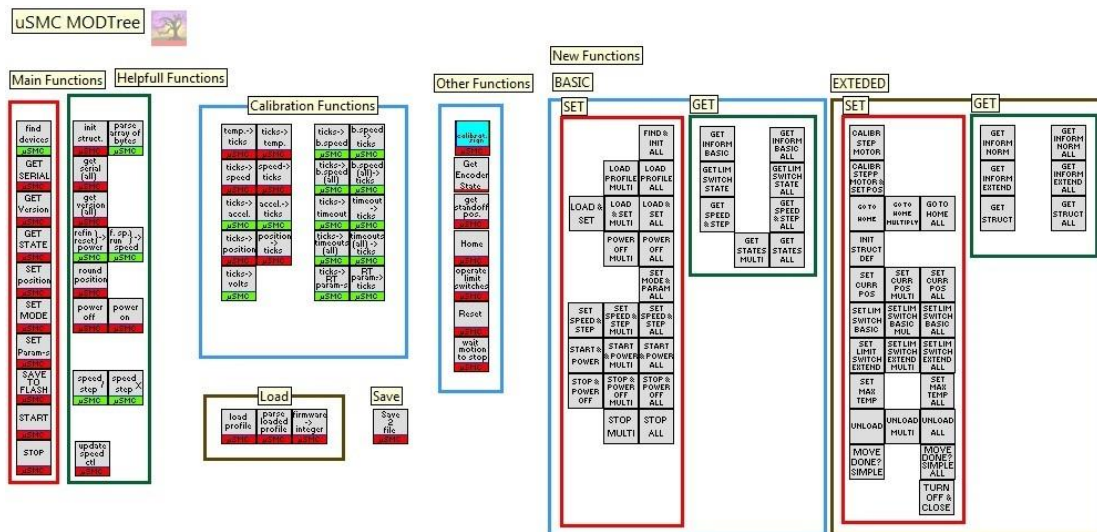
Dalším nedostatkem ovladačů a exemplů výrobce je, že v případě ovládání více krokových motorů najednou může docházet k problému s přepisem hodnot v globálních proměnných.

# 4 VYLEPŠENÍ OVLADAČŮ OD VÝROBCE

Vylepšení ovladačů výrobce spočívá v odstranění globálních proměnných. Globální proměnné jsou nahrazeny jednou strukturou pro každý motor sdružených v poli, které v sobě sdružuje všechny důležité proměnné. Tento krok zahrnuje modifikaci většiny funkcí výrobce. Funkce jsou upraveny tak, aby na vstup i výstup přicházelo hlavní pole struktur MAX, které je ve funkcích programu pro jednoduchost připojováno na terminály MAX IN a MAX OUT. Celá změna zjednodušuje a zpřehledňuje kód, při zachování funkčnosti. Funkce navenek vypadají velice podobně a také se s nimi tak pracuje.

Jsou přidány nové funkce, které zjednodušují práci s krokovými motory. Nově doplněné polymorfní funkce umožňují jednoduché a intuitivní přepínání mezi ovládáním jednoho, více vybraných motorů nebo všech krokových motorů.

Přehled všech upravených funkcí a nových funkcí viz. Obr. 4.1. Červenou barvou jsou vyznačeny upravené funkce, nově přidané funkce nemají žádné barevné označení. Nové funkce jsou pro názornost na obrázku (Obr. 4.1) rozděleny na funkce, které nastavují (Set) a získávají informace (Get).



Obr. 4.1 Modifikované a nové funkce

## 4.1 Odstranění globálních proměnných

Odstraněním globálních proměnných bylo dosaženo vytvořením jednoho velkého pole struktur MAX obsahující všechny parametry a stavy připojených motorů.

## 4.1.1 Hlavní struktura MAX

Počet struktur MAX v hlavním poli taženém celým programem je dán počtem připojených měničů. Inicializaci pole struktur MAX je potřeba provést za pomoci modifikované funkce `Initiaze All Structures`, která naplní prvky výchozími hodnotami. Struktura MAX je tvořena upravenými strukturami výrobce (původně globálními). Tzn., že byl zachován stejný obsah těchto podstruktur. Podrobnější přehled jednotlivých prvků je k dispozici ve třetí kapitole v části globální struktury viz kapitola 3.3.1.

Tab. 4.1 Obsah struktury MAX

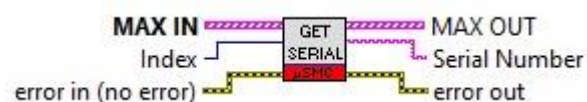
1	VISA resource name
2	StartPlus
3	Position
4	Parameters
5	Mode
6	State

## 4.2 Modifikované funkce

V této podkapitole jsou uvedeny všechny modifikované funkce a jejich popis.

### 4.2.1 Hlavní funkce (Main Functions)

#### Get Serial



Obr. 4.2 Get Serial (MOD)

Zjistí sériové číslo vybraného krokového motoru a uloží tuto hodnotu do vybrané MAX struktury (do parametru `MAX.StartPlus.Steper_motor_ID`). Na vstupu je MAX IN, *Index* udává vybraný krokový motor. Na výstupu je pozměněné pole struktur MAX OUT a *Serial Number* udávající ve formátu string sériové číslo.

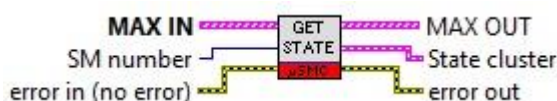
#### Get Version



Obr. 4.3 Get Version (MOD)

Načte verzi firmwaru vybraného krokového motoru, která je také k dispozici na výstupu, zapíše do MAX OUT do parametru MAX.StartPLus.FirmwareVer. Na vstupu je pole struktur MAX a index udává vybraný krokový motor. Na výstupu je MAX OUT a Firmware Version, který udává ve formátu string verzi firmwaru. Do odpovídající struktury MAX se na příslušnou pozici pro vybraný krokový motor uloží verze firmwaru.

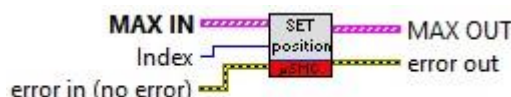
### Get State



Obr. 4.4 Get State (MOD)

Získá informace o aktuálním stavu vybraného zařízení a zapíše změny do příslušné struktury MAX (MAX.State a MAX.Position). Vstup SM number udává zvolený krokový motor. Na výstupu je k dispozici struktura s aktuálním stavem State cluster a obsahuje informace o aktuální pozici krokového motoru, směr rotace, rychlost rotace (rozděleno na tři stavy: nepohybuje se, zrychluje/zpomaluje, pohybuje se plnou rychlostí), teplotě, stavu napájení (bez napájení, poloviční zatížení, plné zatížení), stavu limitního přepínače 1, stav limitního přepínače 2, velikost vůle, indikace stavu a chyby rotační převodovky, zdali je po restartu, velikost kroku (0 - 1/8 kroku, 1 - 1/4, 2 - 1/2, 3 - plný krok), stav synchronizačního vstupu a výstupu na desce, stav nouzového vypínače, zdali je logika napájena z USB, velikost napětí, zdali měnič pracuje, pozice motoru v jednotkách enkodéru a aktuální pozice enkodéru. Na vstupu i výstupu je pole struktur MAX.

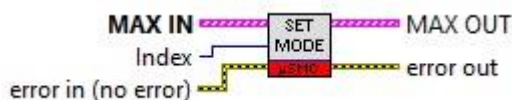
### Set Current Position



Obr. 4.5 Set Current Position (MOD)

Nastaví aktuální pozici krokového motoru. *Index* udává vybraný měnič. Pozici nastavuje z parametru MAX.Position.Set Cur Pos Units. Na vstupu i výstupu je pole struktur MAX.

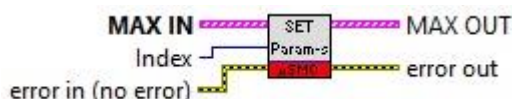
### Set Mode



Obr. 4.6 Set Mode (MOD)

Nastaví měnič podle parametrů ze struktury Max.Mode. Nastavují se parametry jako je třeba přepínání limitních spínačů, jejich zapnutí nebo vypnutí, resetování enkodéru, atd. *Index* udává vybraný měnič. Na vstupu i výstupu je pole struktur MAX.

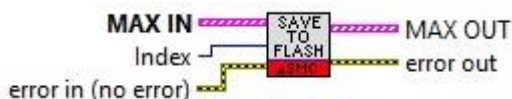
### Set Parameters



Obr. 4.7 Set Parameters (MOD)

Nastaví měnič podle parametrů ze struktury Max.Parameters. *Index* udává vybraný měnič. Na vstupu i výstupu je pole struktur MAX.

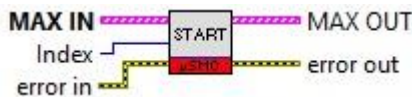
### Save Parameters to Flash



Obr. 4.8 Save Parameters to Flash (MOD)

Uloží parametry do flash paměti vybraného měniče. *Index* určuje zvolený krokový motor. Na vstupu i výstupu je pole struktur MAX.

### Start

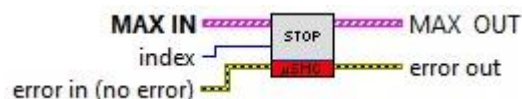


Obr. 4.9 Start (MOD)

Tato funkce spustí vybraný krokový motor v případě, že je zapnuté napájení. *Index* udává krokový motor. Poté co nahraje do měniče parametry struktury

Max.StartPlus, tak se z funkce vystoupí. O tom zdali se dostal krokový motor na cílovou pozici se stará samotný měnič. Na vstupu i výstupu je pole struktur MAX.

## Stop

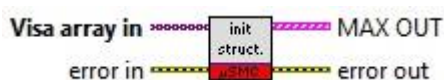


Obr. 4.10 Stop (MOD)

Zastaví vybraný krokový motor. Index udává krokový motor. Na vstupu i výstupu je pole struktur MAX.

## 4.2.2 Užitečné funkce (Helpful Functions)

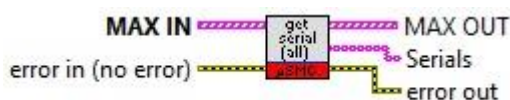
### Initialize All Structures



Obr. 4.11 Initialize All Structures (MOD)

Funkce inicializuje všechny důležité struktury výchozími hodnotami proměnných, které následně uzavírá do jednoho velkého pole struktur MAX. Na vstupu je *VISA array* vytvořené za pomoci funkce Find Devices. Na výstupu je pole struktur MAX sdružující všechny důležité proměnné. Počet prvků MAX je dán počtem nalezených zařízení. Funkce obsahuje také chybový drát *error* na vstupu i výstupu. Podstruktury obsažené v MAX odpovídají podobě globálních proměnných.

### Get Serials of All Controllers



Obr. 4.12 Get Serials of All Controllers (MOD)

Načte verzi firmwaru pro všechny připojené krokové motory, které jsou k dispozici na výstupu (*Serials*). Na vstupu i výstupu je pole struktur MAX, kde výstupní je modifikované (konkrétně *MAX.Versions* následně obsahuje jednotlivé verze krokových motorů).

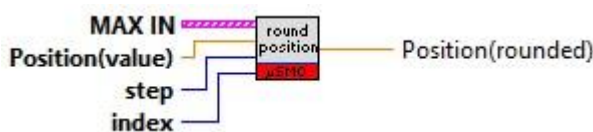
## Get Versions of All Controllers



Obr. 4.13 Get Versions of All Controllers (MOD)

Načte verzi firmware pro všechny připojené krokové motory, které jsou k dispozici na výstupu. Na vstupu i výstupu je pole struktur MAX a pole Versions s jednotlivými verzemi krokových motorů.

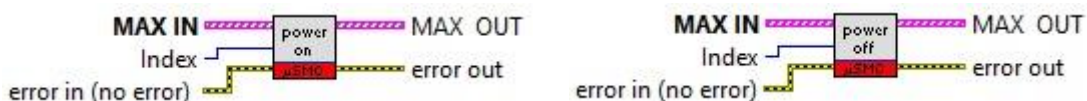
## Round Position



Obr. 4.14 Round Position (MOD)

Zaokrouhlí pozici na základě zvoleného kroku. Na vstupu je MAX IN, Position(value) pro vstupní hodnotu, step indikuje velikost kroku (0-1/8 krok, 1-1/4 krok, 2-1/2 krok, 3-1 krok), Index udává zvolený krokový motor. Position(rounded) je zaokrouhlená pozice na základě velikosti kroku.

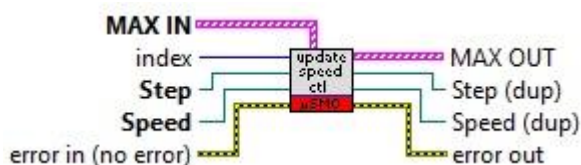
## Power on / Power off



Obr. 4.15 Power On (MOD) /Power Off (MOD)

Funkce pro zapnutí/vypnutí napájení vybraného krokového motoru. Index slouží k vybrání zařízení. Na vstupu i výstupu je pole struktur MAX.

## Update Speed Ctl



Obr. 4.16 Update Speed Ctl (MOD)



Funkce pomáhá aktualizovat a nastavit limity rychlostního a krokového slideru. Index udává vybraný krokový motor, Step a Speed slouží pro vstup reference sliderů a jsou na výstupu funkcí vráceny. Na vstupu i výstupu je pole struktur MAX.

### 4.2.3 Kalibrační funkce (Calibration Functions)

#### Calibrate Temperature 2 Ticks / Calibrate Ticks 2 Temperature



Obr. 4.17 Calibrate Temper. 2 Ticks (MOD)/ Calibrate Ticks 2 Temper. (MOD)

Přepočet teploty v °C na teplotu v Ticks a naopak. SMindex slouží k vybrání požadovaného krokového motoru. Na vstup je potřeba přivést pole struktur MAX.

#### Calibrate Ticks 2 Speed / Calibrate Speed 2 Ticks



Obr. 4.18 Calibrate Ticks 2 Speed (MOD) / Calibrate Ticks 2 Temperature (MOD)

Převod rychlosti v jednotkách (mm/step) na jednotky Ticks. Index slouží k vybrání požadovaného krokového motoru. Na vstup je potřeba přivést pole struktur MAX.

#### Callibrate Position 2 Ticks

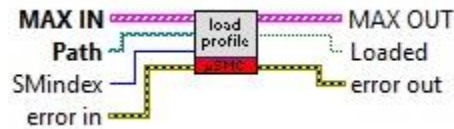


Obr. 4.19 Calibrate Position 2 Ticks (MOD) / Calibrate Ticks 2 Position (MOD)

Přepočet pozice v jednotkách (mm/step) na základě nastavených kalibračních bodů ze struktury MAX(MAX.StartPlus. point... ) na hodnotu v Ticks a naopak. Na vstupu je pole struktur MAX, array index udává vybraný krokový motor.

## 4.2.4 Funkce pro nahrání (Load)

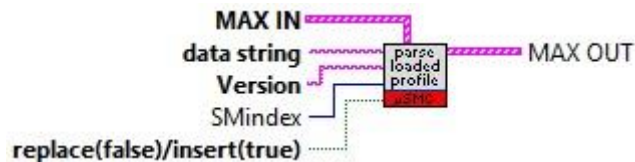
### Load Profile



Obr. 4.20 Load Profile (MOD)

Slouží k nahrání profilu (ve formě binárním souboru) pro daný typ krokového motoru. Tento profil nastavuje řadu parametrů a limitních hodnot. Profil je dostupný od výrobce pro danou verzi krokového motoru, či jej lze vytvořit pomocí programu SMCView v okně Setup nebo využít funkce Save Parametres 2 File. Na vstupu je cesta k souboru Path s uloženým nastavením (soubor s koncovkou .pre nebo .sav). SMindex udává zvolený, krokový motor. Na výstupu předává Loaded informaci, zdali se nahrávání podařilo nebo ne. Na vstupu i výstupu je pole struktur MAX a chybový drát.

### Parse Loaded Profile



Obr. 4.21 Parse Loaded Profile (MOD)

Nízkoúrovňová funkce zajišťující nahrání profilu (souboru základních parametrů) do proměnných. Na vstup data string je potřeba přivést obsah binárního souboru s uloženými nastaveními pro daný krokový motor ve formátu string, Version musí obsahovat verzi SMCView rozhraní ve kterém byl profil uložen, SMindex určuje zvolený krokový motor a pomocí vstupu bool měníme zda se mají nastavení nahradit (false) nebo vložit(true). Funkce přepisuje obsah příslušné struktury MAX. Na vstupu i výstupu je pole struktur MAX.

### FirmwareVer 2 Case

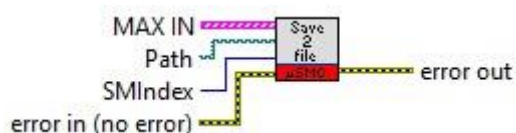


Obr. 4.22 FirmwareVer 2 Case (MOD)

Funkce vrací verzi vybraného krokového motoru ve formátu pro low level funkce. Na vstupu je pole struktur MAX a SMindex pro určení krokového motoru. Na výstupu je *Case* určující verzi krokového motoru (0 pro verzi 23, 1 pro 24 a 2 pro 25).

## 4.2.5 Funkce sloužící k uložení (Save)

### Save Parameters to file



Obr. 4.23 Save Parameters 2 File (MOD)

Slouží k uložení profilu (nastavených parametrů) do binárního souboru. Ukládá struktury MAX.Mode, MAX.Parameters, MAX.Position, MAX.State. Na vstupu je pole struktur MAX a SMindex pro určení krokového motoru. Na vstupu i výstupu je chybový drát.

## 4.2.6 Ostatní funkce (Oher Function)

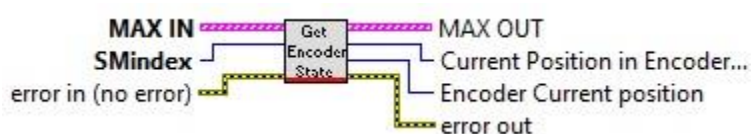
### Get Calibration Sign



Obr. 4.24 Get Calibration Sign (MOD)

Nízkoúrovňová funkce která z kalibračních bodů (viz MAX.StartPlus.Point 1 Units, MAX.StartPlus.Point 1 Ticks, MAX.StartPlus.Point 2 Units a MAX.StartPlus.Point 2 Ticks) vyhodnotí kalibrační znak. Na vstupu je pole struktur MAX IN a SM Index určující zvolený krokový motor. Na výstupu je kalibrační znak Calibration Sign.

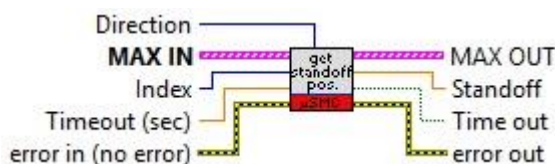
### Get Encoder State



Obr. 4.25 Get Encoder State (MOD)

Vrací aktuální pozici enkodéru. SMindex odpovídá zvolenému krokovému motoru. Na výstupu je aktuální hodnota enkodéru ve dvou druzích enkodérových jednotek a v Ticks. Funkce přepisuje obsah pole struktur MAX. Na vstupu i výstupu je pole struktur MAX.

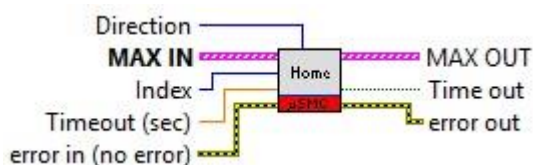
### Get Standoff position



Obr. 4.26 Get Standoff Position (MOD)

Funkce slouží k získání hodnoty vzdálenosti aktuální pozice od limitního snímače. Po spuštění se začne krokový motor pohybovat zvoleným směrem (k maximální nebo minimální hodnotě), dokud nenarazí na limitní snímač. Poté se vrátí na výchozí pozici a zobrazí hodnotu zaznamenané vzdálenosti, kterou zapisuje do struktury MAX.StartPlus.Limit Switch standoff. Direction určuje směr pohybu. Index udává zvolený krokový motor. Vstup Timeout slouží k nastavení časového limitu v sekundách, po jehož vypršení se krokový motor vrací na výchozí pozici. Standoff udává hodnotu naměřené vzdálenosti limitního snímače od krokového motoru. Time out informuje zdali vypršel časový limit. Na vstupu i výstupu je pole struktur MAX a chybový drát. Při spuštění funkce Standoff nelze program přerušit, zastavit či jinak ovládat. Pro vykonání funkce je nutné mít zapnuté napájení.

### Home stage

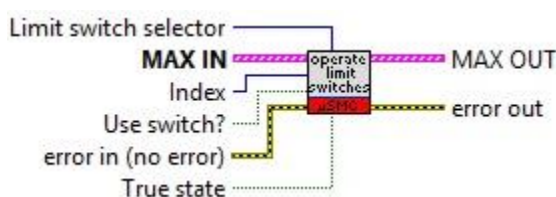


Obr. 4.27 Home stage

Funkce nejprve způsobí, že se krokový motor začne pohybovat od aktuální pozice směrem k limitnímu snímači. Po jeho dosažení, se vrátí na aktuální pozici kterou nastaví na nulu. Po nastavení nuly se má vždy po stisku vrátit na přednastavenou hodnotu. Direction určuje směr pohybu. Index udává zvolený krokový motor. Vstup

Timeout slouží k nastavení časového limitu v sekundách, po jehož vypršení se krokový motor vrací na výchozí pozici. Time out informuje zdali vypršel časový limit. Na vstupu i výstupu je pole struktur MAX a chybový drát. Funkce i po úpravách vykazuje prvky chybového chování, pro její náhradu lze využít obdobně se chovající nově vytvořené funkce Go to HOME, Calibrate stepper motor. Při spuštění funkce Home nelze program přerušit, zastavit či jinak ovládat. Pro vykonání funkce je nutné mít zapnuté napájení.

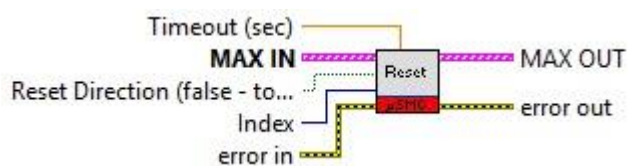
### Operate limit switches



Obr. 4.28 Operate Limit Switches (MOD)

Nízkoúrovňová funkce sloužící k nastavení limitních snímačů. Na vstupu i výstupu je pole struktur MAX a chybový drát. Index určuje zvolený krokový motor. Limit switch selector slouží k vybrání limitního snímače. Vstup True state přepne logický stav snímače. Vstup Use switch? určuje zdali při dojezdu krkového motoru na limitní snímač má dojít k zastavení krokového motoru.

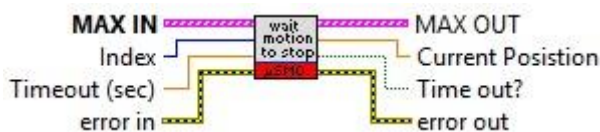
### Reset



Obr. 4.29 Reset (MOD)

Aktuální pozici nastaví jako nulovou, na základě nastaveného směru nastaví znaménka a začne se pohybovat směrem k limitnímu snímači. Poté co na něj narazí nebo vyprší časový limit se zastaví. Reset Direction v případě, že je false, začne se pohybovat k aktuálním maximálním hodnotám. Index udává zvolený krokový motor. Vstup Timeout(sec) slouží k nastavení časového limitu v sekundách. Na vstupu i výstupu je pole struktur MAX a chybový drát. Pro vykonání funkce je nutné mít zapnuté napájení.

## Wait till motion stops



Obr. 4.30 Wait till motion stops (MOD)

Nízkoúrovňová funkce, která se užívá v rámci funkce Get Standoff Position. Informuje nás o tom, zdali krokový motor ukončil svůj pohyb nebo vypršel časový limit. Index udává zvolený krokový motor. Vstup Timeout slouží k nastavení časového limitu v sekundách. Current position vrací aktuální pozici po dojetí na limitní snímač nebo vypršení časového limitu. Time out informuje zdali vypršel časový limit. Na vstupu i výstupu je pole struktur MAX.

## 4.3 Nově přidané funkce

V rámci nově vytvořených funkcí jsou mimo obyčejných pomocných funkcí přidány funkce polymorfni, které zjednodušují vytváření nových aplikací.

V případech kdy nastavují (SET), mají tři režimy:

- Režim pro ovládání jednoho krokového motoru, kdy je potřeba na vstup dovést index vybraného krokového motoru.
- Režim multiplikativní, kdy pomocí vstupního stavového pole Select Array rozhodujeme o tom, které krokové motory se mají ovládat. Na vstupy těchto funkcí by měly přicházet po každé pole s tím, že vždy jednomu krokovému motoru přísluší jeden prvek. V případě, že není zavedeno selektivní pole, tak se funkce provede pro krokové motory podle pole s nejnižším počtem vstupních prvků (Velikost pole struktur MAX také rozhoduje o počtu provedení). Například do funkce Set Speed & Step multiply zavedeme pole struktur MAX (3 inicializované motory - 0,1,2 -> 3 prvky), nezapojené pole Selector Array, Step (2 prvky) a Speed (4 prvky). Funkce se provede pouze pro první dva krokové motory (0,1), protože vstupující pole Step je nejmenší. Tyto funkce mívají na konci označení multiple a na výstupu mají pole proměnných.

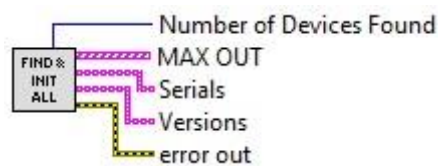
- Režim provádějící funkce pro všechny krokové motory, provede funkci pro všechny inicializované krokové motory (na základě počtu prvků pole MAX).

Tyto funkce mají na konci označení all a na výstupu mají pole.

Pro funkce, které získávají informace (GET) jsou většinou pouze pro jeden krokový motor nebo pro všechny.

### 4.3.1 Jednoduché funkce BASIC

#### Find & Init. All



Obr. 4.31 Find & Init all (NEW)/

Najde a inicializuje všechna připojená zařízení, zjistí jejich sériová čísla, verzi firmwaru. Výstup Number of Devices Found udává počet připojených zařízení. Dále jsou k dispozici výstupní pole Version (Verze firmware), Serials (Sériová čísla). Na výstupu je inicializované pole struktur MAX. Tato funkce je určena k použití na začátku programu, protože pomocí funkce výrobce v ní obsažené Find Devices, nalezne všechna zařízení a následně je inicializuje funkcí Initiaze All Struct.

#### Set Mode & Params All



Obr. 4.32 Set Mode & Params all (NEW)

Nastaví všechny inicializované měniče podle parametrů obsažených ve strukturách Max.Mode a Max. Parameters Array. Jedná se o variaci na funkci výrobce. Na vstupu i výstupu je struktura pole struktur MAX.

#### Turn off & close

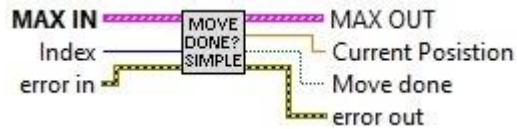


Obr. 4.33 Turn off & close all (NEW)



Funkce zastaví, vypne a uzavře VISY všechn krokových motorů. Tato funkce je vhodná na konec programu.

### Move Done simple

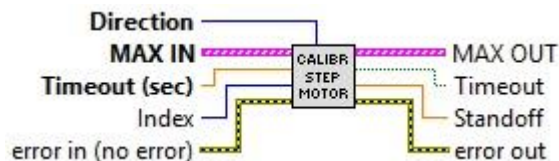


Obr. 4.34 Move Done simple (NEW)

Jednoduchá funkce, která pouze zjistí, zdali krokový motor stojí. Byla vytvořena pro příklad 2 (viz kapitola 4.4). Na výstupu je k dispozici aktuální pozice Current Position a indikátor toho zda už krokový motor stojí Move Done.

## 4.3.2 Jednoduché EXTENDED

### Calibrate stepper motor



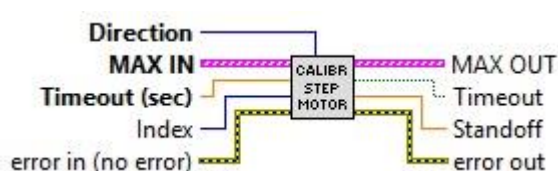
Obr. 4.35 Calibrate stepper motor (NEW)

Kalibrační funkce, slouží ke zjištění vzdálenosti mezi limitními snímači a nastaví krokový motor do středu. Vstupní parametr Index udává krokový motor pro, který se má kalibrace provést. Po spuštění funkce se spustí pohyb (pouze v případě, že je zapnuté napájení krokového motoru) k limitnímu snímači na základě parametru Direction (Ke kladné hodnotě nebo záporné limitní hodnotě  $+\infty/-\infty$ ). Po dojetí na limitní snímač se nastaví pozice na 0 a spustí se pohyb směrem k druhému limitnímu snímači. Po dojetí na druhý snímač se krokový motor začne pohybovat do středu naměřené vzdálenosti mezi snímači. Vstupní parametr Timeout udává časový limit pro dojetí k limitnímu snímači, tento parametr je ve skutečnosti dvojnásobný. Protože udává Timeout pro každý pohyb k oběma limitním snímačům. Například po zadání Timeoutu 400 s, bude pohyb trvat 800 s, protože se jedná o 400 s pro každý pohyb k limitnímu snímači. Na výstupu je k dispozici indikace vypršení časového limitu Timeout. Standoff udává zjištěnou vzdálenost mezi limitními snímači.



Krokový motor se bude pohybovat předešlou nastavenou rychlostí na základě údajů ve struktuře MAX.StartPlus (Rychlost lze nastavit pomocí funkce Set Speed & Step). Pro co největší přesnost je potřeba nastavit malý krok (1/8, 1/4). Nastavením velkého kroku ve spojení s velkou rychlostí může dojít k chybě. Při maximálních rychlostech může krokový motor přejet přes limitní snímače tak rychle, že měnič, který obstarává zastavení krokového motoru na základě sepnutí limitního spínače nestihne zareagovat. Aby se funkce začala vykonávat, je potřeba mít zapnuté napájení (Zapnutí napájení může obstarat funkce POWER ON). Při spuštění funkce nelze program přerušit, či zastavit či jinak ovládat.

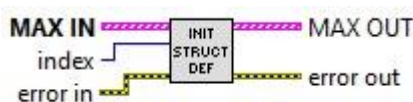
### Calibrate stepper motor & Set position(NEW)



Obr. 4.36 Calibrate stepper motor & Set position(NEW)

Upravená funkce Calibrate stepper motor (NEW). Provede kalibraci a ukončí svůj pohyb uprostřed. Tuto pozici nastaví jako výchozí (Nastaví pozici na 0). Pro vykonání funkce je nutné mít zapnuté napájení.

### Initiaze default values (NEW)



Obr. 4.37 Initiaze default values (NEW)

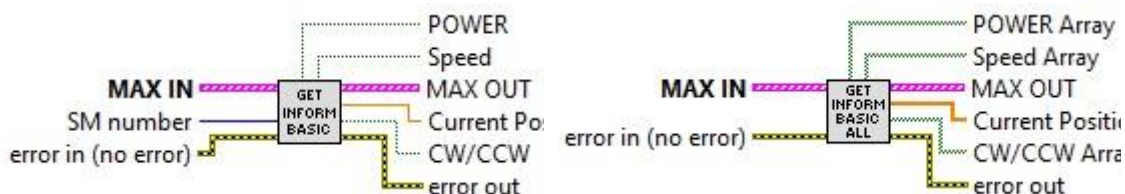
Nastaví výchozí hodnoty prvkům struktury MAX (StartPlus, Position, Parameters, Mode, State) pro vybraný krokový motor. Funkce není náhradou za funkci Initialize All Structures, protože neinicilizuje prvek Max.Visa a na vstup je potřeba přivést již vytvořené pole struktur MAX.

## 4.3.3 Polymorfní funkce

Funkce jsou k dispozici v podobě pro jeden krokový motor, více vybraných krokových motorů (Multiple) a pro všechny(all). Parametr na SM number/Index udává vybraný krokový motor. V případě podoby multiplikativní

(Multiple) a pro všechny(all) mají na výstupu pole. U multiplikativních funkcí vstupním polem rozhodujeme o tom, které krokové motory se mají zpustit.

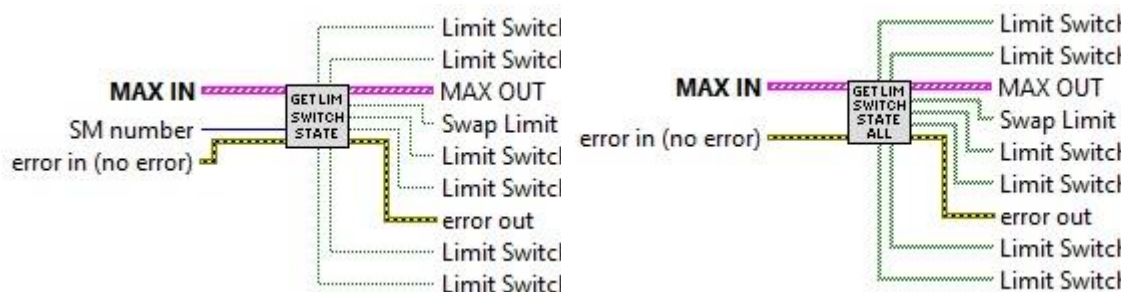
### Get Information BASIC



Obr. 4.38 Get Information BASIC (NEW)/ Get Information BASIC all (NEW)

Funkce zjistí základní stav krokového motoru a změny zapíše do struktury MAX.State. Funkce je k dispozici v podobě pro jeden krokový motor, kdy je potřeba přivést na pozici index vybraného krokového motoru SM number. POWER udává stav napájení krokového motoru(vypnuto/zapnuto), Speed udává stav rychlosti (stojí/v pohybu), Current Position aktuální pozici a CW/CCW směr rotace motoru (po směru/proti směru hodinových ručiček).

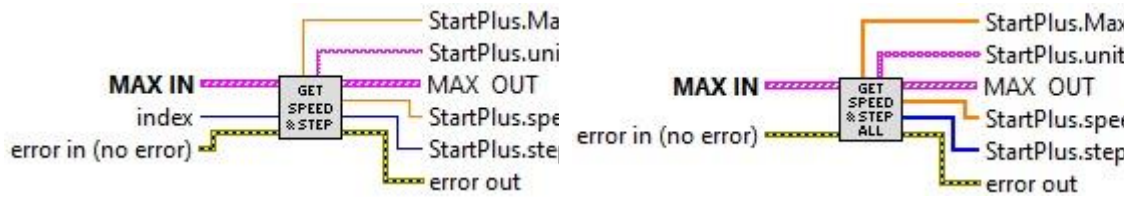
### Get Limit switch state



Obr. 4.39 Get Limit switch state (NEW)/ Get Limit switch state all (NEW)

Funkce zjistí stav limitních snímačů a jejich nastavení. Limit Switch X udává aktuální stav limitního snímače. Limit Switch X Operation Enabled rozhoduje o tom, zdali se má krkový motor nastavit, když dojde na limitní snímač. Swap Limit Switches informuje o prohození snímačů. Limit Switch X True State nastaví limitní snímač do stavu, kdy se jeví jako zapnutý.

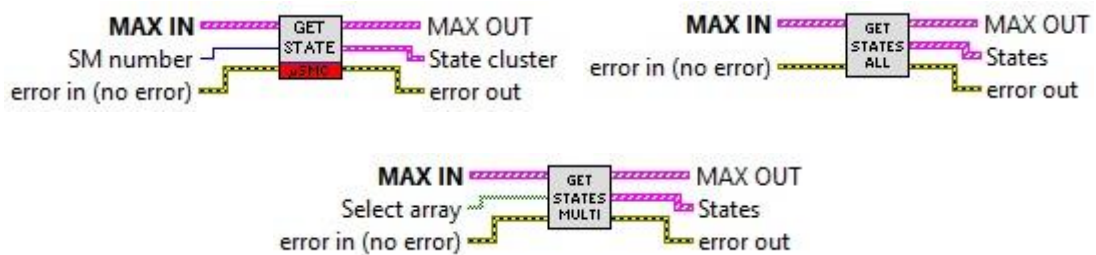
## Get Speed & Step



Obr. 4.40 Get Speed & Step (NEW)/ Get Speed & Step all (NEW)

Zjistí ze struktury MAX.Startplus maximální možnou rychlost (StartPlus.MaximumSpeed), používané jednotky (StartPlus.units) a aktuálně nastavenou rychlost (StartPlus.speed) a krok. (StartPlus.step). Nejedná se o aktuální hodnoty nastavené v měniči, ale o hodnoty ve struktuře MAX.Startplus se do krokového motoru načtou pomocí funkce Start, Start & Power.

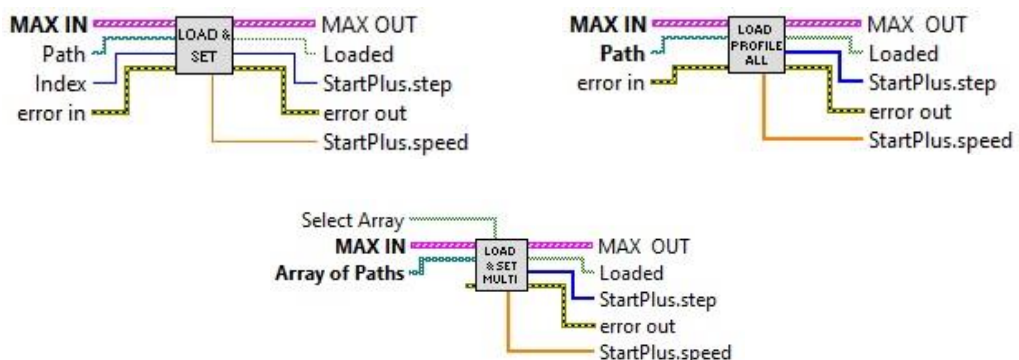
## Get State



Obr. 4.41 Get States (MOD)/Get States multi (NEW)/Get State all (NEW)

Variace na funkci výrobce Get State, zjistí aktuální stav krokového motoru a uloží změny do struktury MAX. Stav je k dispozici i jako pole na výstupu (States).

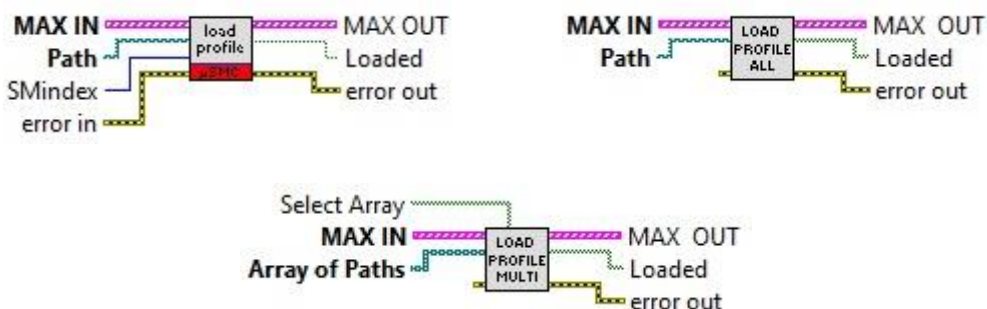
## Load & Set



Obr. 4.42 Load & Set/Load & Set multiple/ Load & Set all (NEW)

Funkce nahraje profil a nastaví měnič. V případě použití multiplikativní funkce nahraje pouze vybrané krkové motory a u zbylých nahraje výchozí hodnoty. Na výstupu je k dispozici indikátor úspěšnosti Loaded nahrání funkce. Nahrané hodnoty nastavené rychlosti a kroku. Rychlost a krok se v rámci funkce měniči nenastavuje, rychlost a krok nastavuje se nastavuje přímo před spuštěním krokového motoru ve funkci Start.

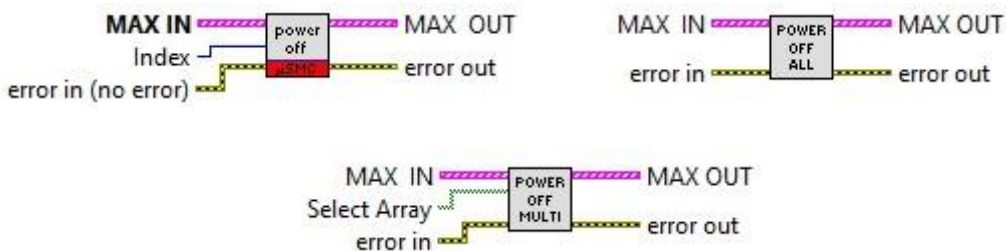
## Load profile



Obr. 4.43 Load profile(MOD)/Load profile multiple(NEW)/ Load profile (NEW)

Funkce nahraje profil, jedná se o variaci na funkci výrobce. Na výstupu je k dispozici indikátor úspěšnosti nahrání funkce Loaded.

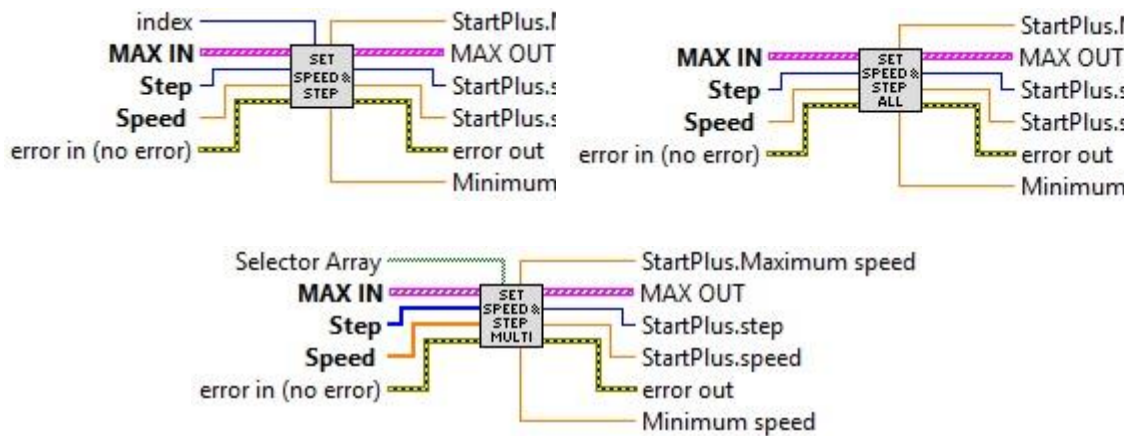
## POWER OFF



Obr. 4.44 POWER OFF (MOD)/ POWER OFF multiple(NEW)/ POWER OFF all (NEW)

Funkce vypne napájení krokovému motoru. K dispozici je také polymorfní funkce pro zapínání krokových motorů POWER ON polymorph.

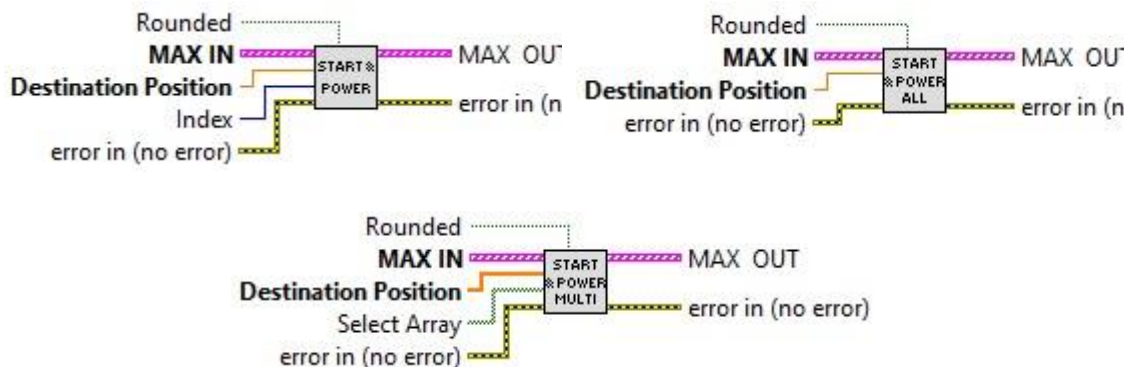
## Set Speed & Step



Obr. 4.45 Set Speed & Step polymorph(NEW)/ Set Speed & Step multiple(NEW)/ Set Speed & Step all(NEW)

Nastaví do pole struktur MAX velikost kroku (1, 1/2, 1/4, 1/8) a rychlost v jednotkách (mm/step). Rychlost se nahraje do měniče až při použití funkce start. V případě použití moc velké rychlosti hodnot zaokrouhlí. Na výstupu jsou nastavené hodnoty maximální, minimální, aktuální rychlosti a kroku. Tyto hodnoty jsou určeny pro načtení do uživatelem vytvořených indikátorů. StartPlus.Step indikuje krok tak jak je zapsaný ve struktuře (0 - 1/8 krok, 1 - 1/4 krok, 2 - 1/2 krok, 3 - 1 krok).

## Start & POWER



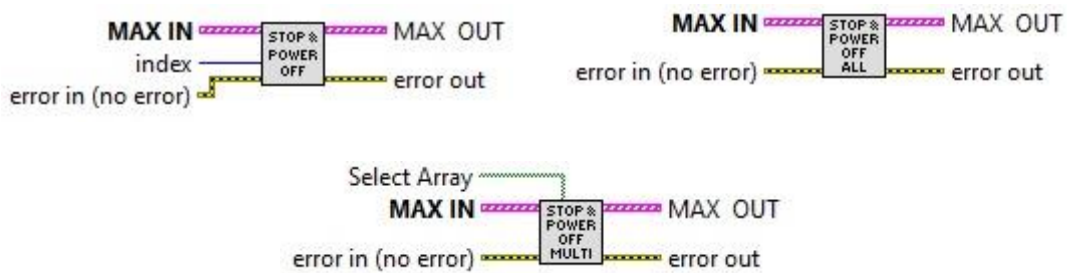
Obr. 4.46 Start & POWER (NEW)/ Start & POWER multiple (NEW)/ Start & POWER all (NEW)

Funkce zapne napájení v případě, že není zapnuté a spustí krokový motor. Stavem Rounded rozhodujeme o zaokrouhlení cílové pozice Destination Position, pokud není zapojený tak se poloha nezaokrouhluje. Funkce start nastaví hodnotu kroku, rychlost ze struktury a ostatních parametrů ze struktury MAX.StartPlus. Po zaslání parametrů a spuštění motoru se z funkce vystoupí a měnič sám kontroluje,



zda-li dosáhl krokový motor cílové polohy nebo limitního snímače. Po dosažení cílové polohy se měnič zastaví, ale napájení se nevypne. Zůstane v zadržovacím módu (Hold Mode).

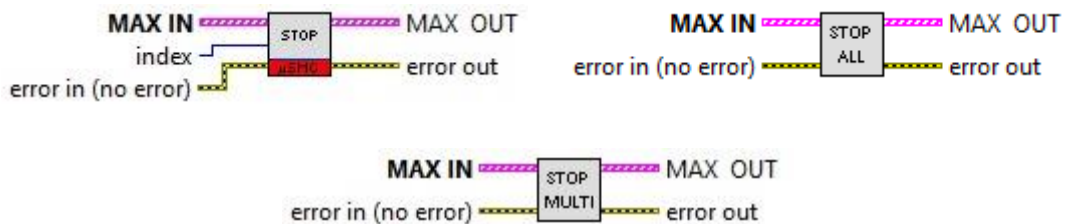
### Stop & Power OFF



Obr. 4.47 Stop & Power OFF/ Stop & Power OFF multiple/ Stop & Power OFF all

Funkce zastaví krokový motor a vypne jeho napájení.

### Stop



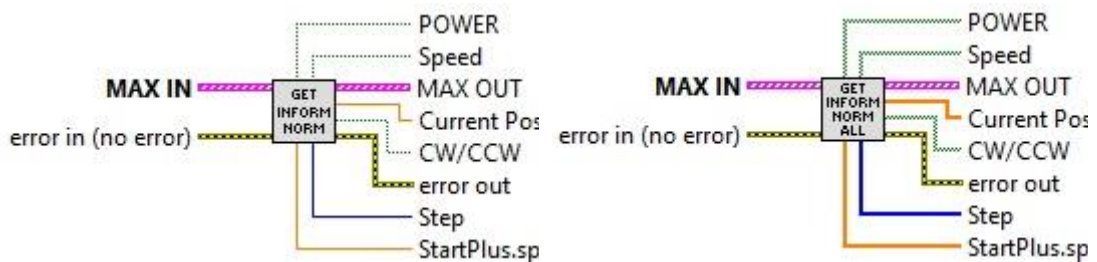
Obr. 4.48 Stop (NEW)/ Stop multiple(NEW)/ Stop all(NEW)

Funkce zastaví pohyb krokového motoru. Stav napájení zůstává nezměněn.

## 4.3.4 Polymorfní funkce Extended

Doplňkové funkce nabízející nový stupeň funkčnosti.

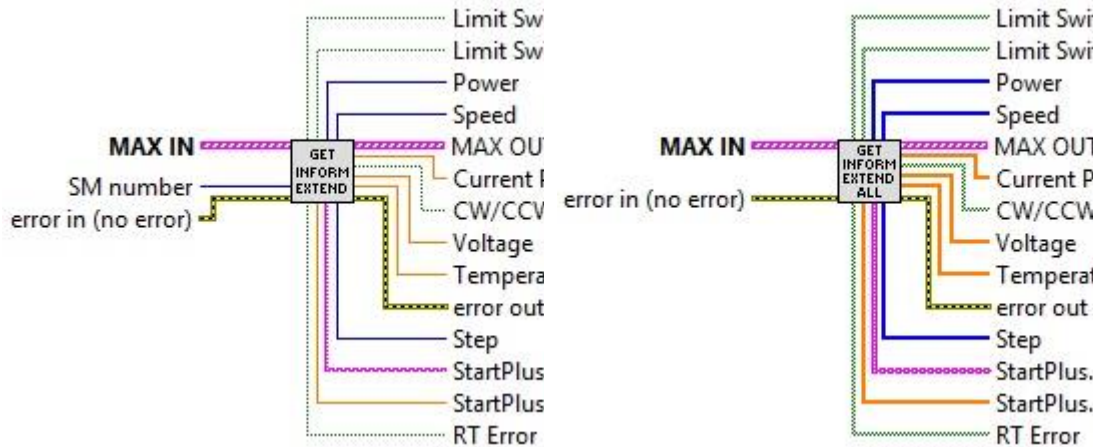
### Get Information NORMAL (NEW)



Obr. 4.49 Get Information NORMAL (NEW)

Modifikovaná funkce Get Information BASIC. Navíc nabízí informace o velikosti rychlosti a kroku ze struktury MAX.Startplus.

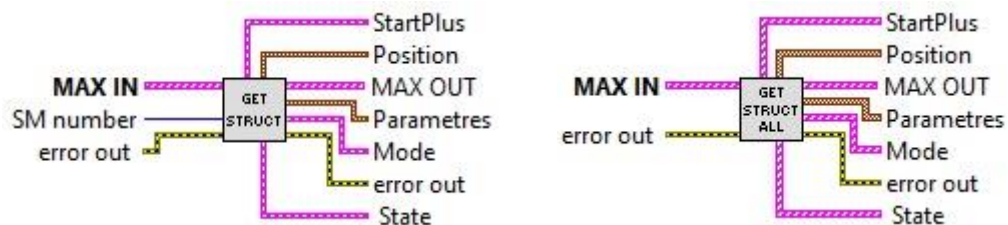
### Get Information EXTENDED polymorph (NEW)



Obr. 4.50 Get Information EXTENDED polymorph (NEW)

Struktura nabízí aktuální stav všech důležitých parametrů. Limit Switch X udává stav limitních snímačů. POWER udává stav napájení (0 – bez napájení, 1 – plovíční zatížení, 2 – plné zatížení), Speed indikuje rychlost (0 – krkový motor se pohybuje, 1 – zrychluje/zpomaluje, 2 – pohybuje se plnou rychlostí). Current Position udává aktuální pozici, směr rotace CW/CCV (po směru/proti), Voltage(velikost napětí ve voltech V). Temperature velikost teploty ve stupních °C. Step udává mód kroku (0 – 1/8 krok, 1 – 1/4 krok, 2 – 1/2 krok, 3 – 1 krok), Startplus.units nastavené jednotky(mm/step), Startplus.Speed (rychlost). RT Error indikuje stav rotační převodovky.

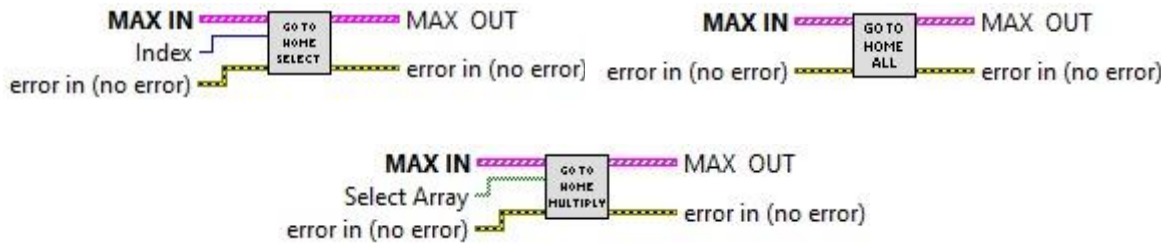
### Get Structs polymorph



Obr. 4.51 Get Structs (NEW)/ Get Structs all(NEW)

Rozbalí pole struktur MAX na jednotlivé podstruktury.

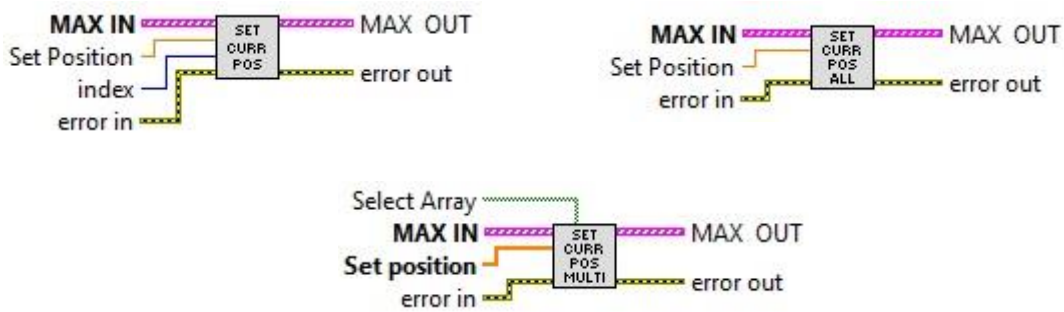
## Go to HOME polymorph



Obr. 4.52 Go to HOME (NEW)/ Go to HOME multiple(NEW)/ Go to HOME (NEW)all

Spustí pohyb krokového motoru na pozici 0. Na to zda se krokový motor dostane do výchozí pozice se nečeká. Pošle se pouze měnič cílová pozice a měnič se o vykonání příkazů postará sám.

## Set Current Position poly

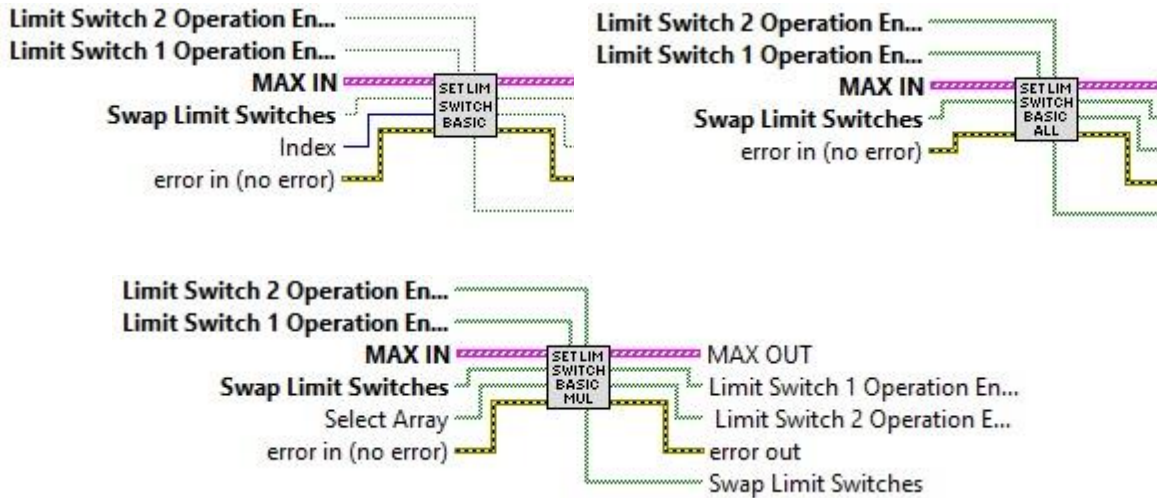


Obr. 4.53 Set Current Position (NEW)/ Current Position multiple (NEW)/ Current Position all(NEW)

Nastaví aktuální pozici krokového motoru na pozici zadanou parametrem Set position.



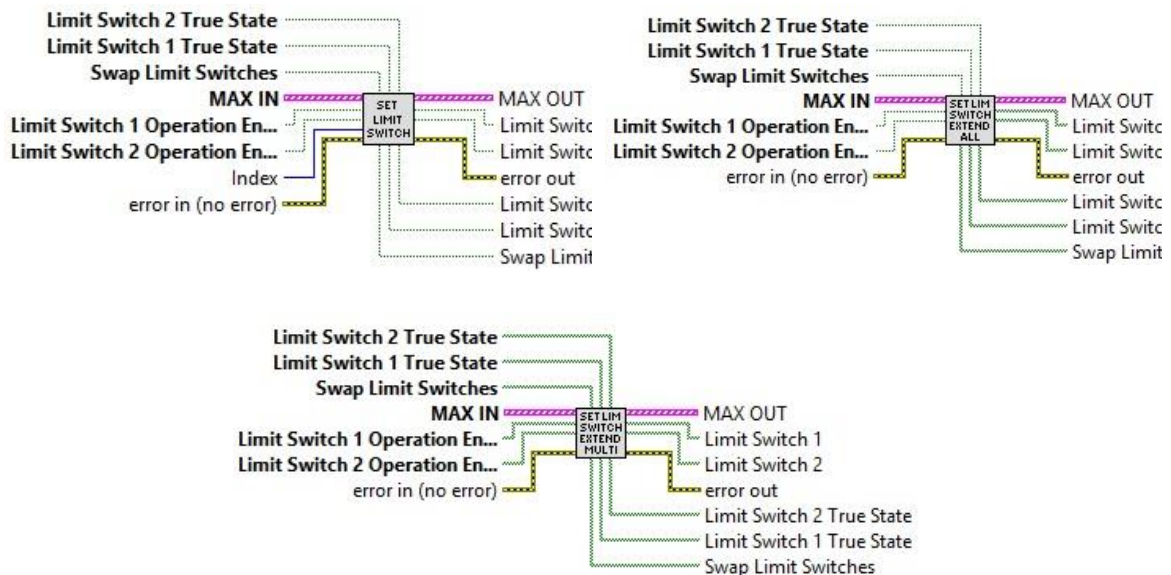
## Set Limit Switches BASIC polymorph (NEW)



Obr. 4.54 Set Limit Switches BASIC polymorph (NEW)

Funkce pro nastavení stavu limitních snímačů. Pomocí Limit Switch X Operation Enabled nastavíme, zda-li se má krokový motor zastavit, když dojde na limitní snímač. Na výstupu je k dispozici aktuální stav Limit Switch X Operation Enabled. Swap Limit Switches prohodí limitní snímače. Na výstupu jsou k dispozici nastavené parametry pro zavedení od indikátorů.

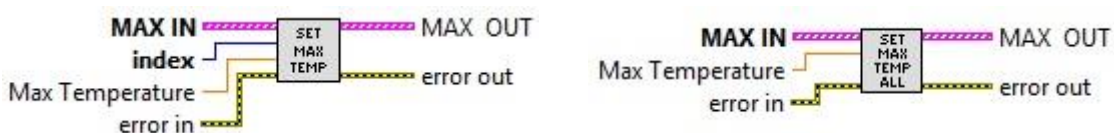
## Set Limit switches EXTENDED polymorph (NEW)



Obr. 4.55 Set Limit switches EXTENDED polymorph (NEW)

Rozšířená funkce Set Limit Switches, umožňující navíc měnit logický stav limitních snímačů Limit Switch X True State. Na výstupu je navíc k dispozici aktuální stav limitních snímačů Limit Switch X.

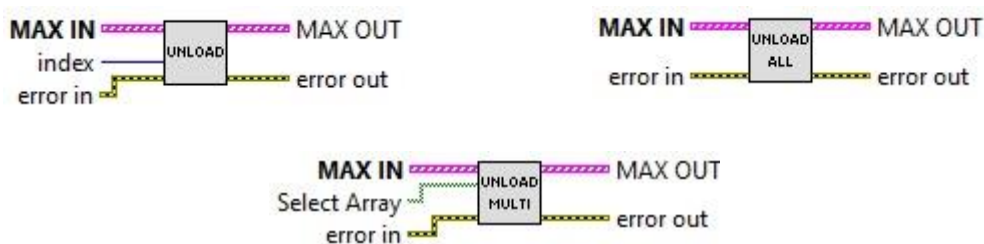
### Set Max Temperature polymorf



Obr. 4.56 Set Max Temperature polymorf (NEW)

Nastaví maximální teplotu krokovému motoru podle vstupního parametru Max Temperature. Při dosažení nebo překročení se krokový motor vypne. Maximální teplota se krokovému motoru nahrává při načítání profilu. V případě nezavedení parametru se funkce nastaví maximální teplotu na 70°C.

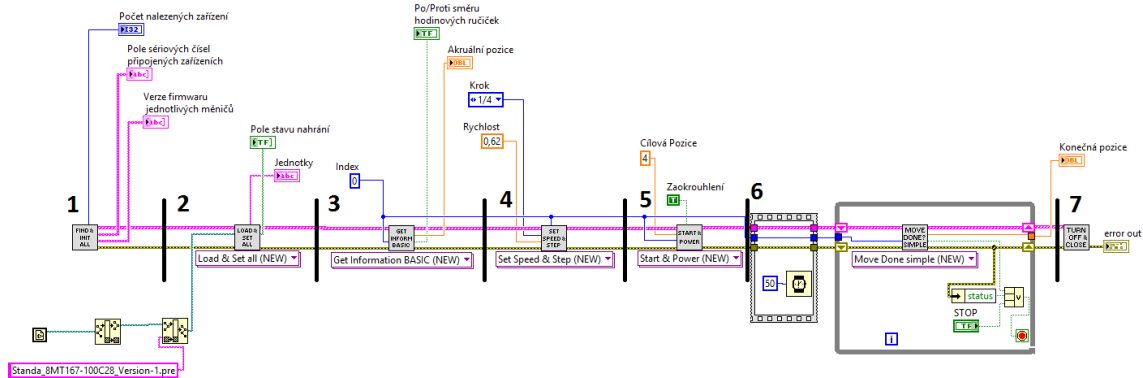
### Unload polymorf (NEW)



Obr. 4.57 Unload polymorf (NEW)

Přehraje nastavení měniče na výchozí hodnoty.

## 4.4 Ukázka vytvoření jednoduchého ovládání jednoho krokového motoru



Obr. 4.58 Příklad jednoduchého ovládání jednoho krokového motoru

Jednoduchý příklad ovládání vybraného krokového motoru, u kterého lze konstantou vybrat, který krokový motor bude ovládán, nastavit cílovou pozici, velikost kroku a rychlost (viz Obr. 4.58). Po provedení zadaných instrukcí se krokový motor zastaví a vypne. Uvedené řešení je k dispozici ve složce vzorové aplikace – příklad 2.

V první části se pomocí funkce najdou všechny dostupné motory, vytvoří pole struktur MAX, která se naplní výchozími hodnotami a zjistí se sériová čísla a verze firmwaru měničů. Verze firmwarů i sériových čísel se zapíšou do příslušných struktur MAX. Tato funkce nebo funkce v ní obsažené (Find Devices, Initialize All Structures) jsou nezbytné k používání dalších funkcí. Na výstupu je k dispozici počet nalezených zařízení a pole sériových čísel a firmwaru.

V druhé části dochází k nahrání profilu (všech kalibračních parametrů) z binárního souboru a nastavení parametrů pro všechny krokové motory. Nahrané informace určují, v jakých jednotkách se budou údaje zobrazovat (mm nebo step) a nastavení limitních hodnot. V příkladu zaváděný soubor obsahuje nastavení přímo pro krokový motor, který jsem měl k dispozici (Lineární posun Standa 8MT167-100C28). Profil lze vytvořit v programu SMCView nebo za pomoci funkce Save to Flash. Nahrávaný binární soubor musí mít koncovku sav, pre nebo usm. O tom, zdali se podařilo profil nahrát informuje pole stavů na výstupu. V případě, že se profil nepodaří nahrát, nastaví se do měniče výchozí hodnoty. Na výstupu je též k dispozici výstup oznamující v jakých jednotkách krokový motor pracuje.

V třetí části je funkce, která získá základní stav vybraného krokového motoru. Funkce `Get Inform Basic` na výstupu kromě indikace aktuální pozice a směru rotace nabízí stav, zdali je krokový motor v pohybu nebo zdali je zatížen. V tomto jednoduchém příkladu, ale tyto informace nejsou zobrazeny. Více parametrů lze zjistit za pomoci funkcí rozšířených funkcí `Get Inform Normal`, `Get Inform Extended` nebo rozbalením struktury `MAX` za pomoci funkce `Get Data`. Důležitým parametrem je konstanta `Index`, udávající zvolený krokový motor.

Ve čtvrté části se nastavuje rychlost a velikost kroku. Přenastavená rychlost a krok se zapíše do struktury `MAX`, údaje se nahrávají do měniče až ve funkci `Start`. `Set Speed & Step` provede zápis do struktury a zaokrouhlí rychlost tak, aby nepřesahovala maximální povolenou rychlost získanou na základě nahraného profilu.

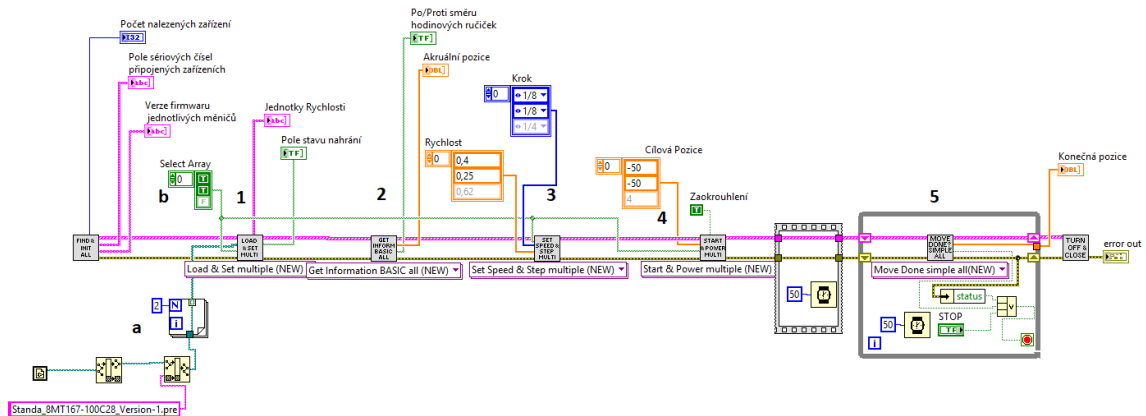
V páté fázi funkce zapne napájení krokovému motoru, v případě, že již není zapnuto, a spustí pohyb krokovému motoru. Funkce nahraje do měniče cílovou pozici, parametry udávající rychlost a krok. Zaokrouhlení určuje, zdali se má zadaná cílová pozice zaokrouhlovat. Zpoždění 50 ms je z důvodu zajištění funkčnosti další fáze.

V šesté fázi, kontrolujeme zdali krokový motor dosáhl cílové pozice. Zpoždění 50 ms je nutné z důvodu zajištění funkčnosti další fáze. Protože krokový motor se vypne sám, po dosažení zadané pozice nebo sepnutím limitního snímače. O jeho zastavení, ale rozhoduje měnič. Z důvodu demonstrace funkčnosti je v tomto místě kódu nutno kontrolovat, zdali už dokončil svůj pohyb. Jinak by se totiž hned po zaslání informací do měniče pomocí funkce `Start & Power`, vykonala poslední funkce `TURN OFF & CLOSE`. Funkce `Move Done? Simple` vrací také poslední dosaženou hodnotu.

V sedmé fázi dochází k zastavení, vypnutí všech krokových motorů. Funkce `Turn Off & Close` se má používat až na konci, protože provádí také zavření parametru `VISA`. V případě potřeby pouze zastavení jsou zde k dispozici funkce `Stop` nebo `Stop & Power OFF`.

## 4.5 Modifikace příkladu umožňující ovládání více krokových motorů

Příklad z obrázku Obr. 4.58. můžeme přepnutím polymorfních funkcí změnit tak, aby fungoval pro více vybraných krokových motorů. Upravený příklad pro několik vybraných krokových motorů je patrný na Obr. 4.59. Uvedené řešení je k dispozici ve složce vzorové aplikace – příklad 3.



Obr. 4.59 Příklad jednoduchého ovládání více vybraných krokových motorů

Na obrázku Obr. 4.59, jsou čísla označeny funkce, které bylo potřeba pozměnit a jednoduchými písmeny jsou označeny nové důležité prvky.

- a) Pole SELECT ARRAY označuje vybrané krokové motory. Selektivní pole je důležitým vstupem multiple funkcí. Momentálně vybrané nastavení značí, že se použijí oba dva vybrané motory. V případě, nezapojení SELECT ARRAY se funkce vykoná podle nejnižšího počtu prvků pole vstupu (I podle pole struktur MAX).
- b) Multiple funkce vyžadují pole vstupů, proto je potřeba vytvořit pole cest ze, kterých se bude nahrávat.

Změněné funkce oproti předchozímu (viz Obr. 4.59), funkce 1, 3, 4 jsou funkce multiple a na jejich vstup je potřeba přivést pole hodnot (viz Rychlost, Krok, Cílová pozice, apod.). Druhým tipem užití funkce je funkce all, tyto funkce ovlivňují všechny krokové motory. Funkce multiple i all, mají na výstupu pole parametrů (viz Pole stavu nahrání, konečná pozice, apod.).

Změněné funkce oproti předchozímu (viz Obr. 4.59), funkce 1, 3, 4 jsou funkce multiple a na jejich vstup je, potřeba přivést pole hodnot (viz Rychlost, Krok, Cílová pozice, apod.). Druhým tipem užití funkce je funkce all, tyto funkce ovlivňují

všechny krokové motory. Funkce `multiple` i `all`, mají na výstupu pole parametrů (Pole stavu nahrání, konečná pozice, atd.).

Na přiloženém disku jsou k dispozici ve složce Vzorové aplikace ještě tři příklady (příklad 1, příklady 4, příklady 5)

Příklad 1 je upravenou verzí příkladu 2 s tím, že nabízí ovládání za pomoci `controlů`.

Příklad 3 nabízí ovládání více krokových motorů za s využitím výběrového pole `Select Array`.

Příklad 5 umožňuje ovládání krokových motorů, ve kterém je potřeba přepínat mezi motory. Nabízí využití pokročilých funkcí.

## 5 POROVNÁNÍ VYLEPŠENÝCH OVLADAČŮ S OVLADAČI OD VÝROBCE

Ovladače výrobce v LabView byly z důvodu použití globálních struktur nepřehledné. Umožňovaly sice ovládat více krokových motorů, ale pouze po vytvoření složitější nastavby.

Nově modifikované ovladače mají zachovanou stejnou funkčnost. Navenek se liší jen tím, že mají místo Visa array zavedené pole struktur MAX, případně se pole struktur MAX objevuje i u funkcí, které předtím vstup VISA array neměly, protože četly data z globálních proměnných. V rámci funkcí se parametry prvků struktur MAX přepisují. Prvek struktury MAX obsahuje další struktury, které mají stejnou podobu jako výrobcem použité globální struktury (StartPlus, Position, Parameters, Mode, State). Obsah těchto podstruktur zůstal stejný. Vedení všech dat a nastaveních jedním velkým polem struktur, která je na vstupu i výstupu funkcí, práci s ovladači zjednodušilo a zpřehlednilo.

Nově přidané funkce zjednodušují práci a zahrnují v sobě více menších funkcí. Multiplikativní funkce umožňují ovládání více vybraných krokových motorů najednou. Zapouzdřené funkce a také nové užitečné funkce umožňují navrhnout jednoduché aplikace pro ovládání více krokových motorů bez toho aby bylo nutné rozbalovat pole struktur MAX.

## 6 ZÁVĚR

Úvodní kapitola je věnována krokovým motorům a způsobům jejich ovládání.

Měnič Standa 8SMC1-USBhF slouží k ovládání jednoho bipolárního krokového motoru. Jeden vstupní a dva výstupní USB porty umožňují kaskádové zapojení měničů za sebou a tím nabízí možnost ovládání více krokových motorů najednou.

Použitý měnič 8SMC1-USBhF obsahuje tři kompaktní měniče 8SMC1-USBhF a umožňuje tak ovládání tří krokových motorů. Za pomoci měniče byly ovládány dva hybridní krokové motory Standa 8MT167-100C28 bipolárního typu. Krokové motory lze ovládat pomocí aplikace dodávané výrobcem SMCView, za použití vývojových nástrojů v LabView, nebo v jazycích kompatibilních s C.

Práce se zaměřuje na ovladače v LabView, protože v MicroSMC jsou k dispozici pouze deklarace funkcí v hlavičkovém souboru. Ovladače výrobce tvořeny v LabView jsou založeny hlavně na pěti základních polích globálních struktur. Tato koncepce je velmi nepřehledná a stěžuje vytváření nových aplikací.

V práci je vytvořeno pole struktur MAX, které nahrazuje globální proměnné výrobce. Struktura MAX v sobě zahrnuje vzory globálních struktur a parametr VISA. Funkce, které využívali globálních proměnných byly přepracovány podle nové logiky. Odstranění globálních proměnných sebou přineslo zpřehlednění a zjednodušení.

Dále byly vytvořeny nové funkce, které usnadňují práci a umožňují ovládat více krokových motorů najednou. Nové funkce jsou zahrnuty do polymorfních funkcí, které umožňují jednoduché přepínání mezi funkcí pro jeden/více vybraných/všechny krokové motory. Funkce pro více vybraných motorů (multiplikativní) umožňují pomocí vstupního pole stavů vybrat, pro které krokové motory se má funkce provést.

Novou přidanou hodnotou ovladačů je kromě modifikovaných a nových funkcí taky přidání několika názorných příkladů aplikací, které jsou na příloženém disku.



## 7 LITERATURA

- [1] KROKOVÉ MOTORY. *Kves* [online]. Žilina: Žilinská univerzita [cit. 2016-12-17]. Dostupné z: <<http://www.kves.uniza.sk/kvesnew/dokumenty/%C5%A1es/krokov%C3%A9%20motory/Krokove%20motory.pdf>>
- [2] Krokový motor. *Pohonnatechnika.cz* [online]. [cit. 2016-12-24]. Dostupné z: <<http://www.pohonnatechnika.cz/skola/motory/krokovy-motor>>
- [3] SCARPINO, Matthew. *Motors for makers: a guide to steppers, servos, and other electrical machines*. ISBN 01-340-3283-7.
- [4] DVOŘÁKOVÁ, Hana. *Silové krokové motory*. Brno: Výzkumný a vývojový ústav elektrických strojů točivých, 1969. Studijní zpráva.
- [5] Stepper Motor and Controller Primer. *Phidgets* [online]. Phidgets, 2015 [cit. 2016-12-25]. Dostupné z: <[http://www.phidgets.com/docs/Stepper\\_Motor\\_and\\_Controller\\_Primer#Bipolar](http://www.phidgets.com/docs/Stepper_Motor_and_Controller_Primer#Bipolar)>
- [6] ŘEZÁČ, Kamil. Krokové motory (Robotika.cz > Články): [online]. 28.10.2002 [cit. 2017-01-05]. Dostupný z WWW: <<http://robotika.cz/articles/steppers/cs>>.
- [7] Pololu A4988 – modul pro řízení krokových motorů. *RoboDoupe* [online]. Praha, 2013 [cit. 2016-12-25]. Dostupné z: <<http://robodoupe.cz/2013/pololu-a4988-modul-pro-rizeni-krokovych-motoru/>>
- [8] 8SMC1-USBhF - Microstep Driver (USB Interface). *www.standa.lt* [online]. Vilnius: Standa [cit. 2017-01-02]. Dostupné z: <[http://www.standa.lt/products/catalog/motorised\\_positioners?item=175](http://www.standa.lt/products/catalog/motorised_positioners?item=175)>
- [9] 8SMC1-USBhF User Manual. Standa Ltd [online]. 2011. [cit. 2016-12-30] Dostupné z: <<http://www.standa.lt/files/usb/8SMC1-USBhF%20User%20Manual.pdf>>
- [10] 8SMC1-USBhF Datasheet.pdf Standa Ltd [online]. 2011 [2016-12-30] <<http://www.standa.lt/files/usb/8SMC1-USBhF%20Datasheet.pdf>>

## **8 SEZNAM PŘÍLOH**

**Příloha 1.** CD/DVD obsahující původní ovladače výrobce, upravené ovladače výrobce, nově přidané funkce, vzorové popsané příklady, elektronický text