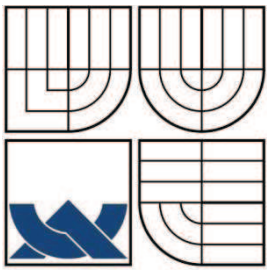
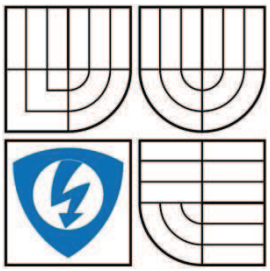


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

FUNKČNÍ BLOKY PRO SIMATIC S7-300 S SINAMICS S120 V RYCHLOSTNÍCH APLIKACÍCH

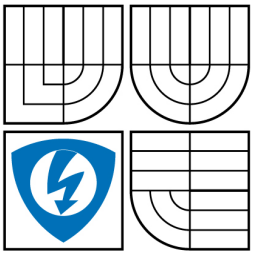
TITLE

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE — JAKUB VACEK
AUTHOR

VEDOUCÍ PRÁCE — Ing. RADEK ŠTOHL, Ph.D.
SUPERVISOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student:	Jakub Vacek	***ID:***	98151
Ročník:	3	***Akademický rok:***	2008/2009

NÁZEV TÉMATU:

Funkční bloky pro Simatic S7-300 s Sinamics S120 v rychlostních aplikacích

POKYNY PRO VYPRACOVÁNÍ:

1. Naprogramujte ve vývojovém prostředí Simatic Step7 sadu funkčních bloků pro PLC Simatic S7, které utvoří knihovnu využitelnou programátorem pro rychlostní aplikace s frekvenčními měniči řady Sinamics S120.
2. Komunikace mezi PLC a měničem probíhá prostřednictvím moderní sběrnice PROFINET v RT režimu.
3. Ověřte funkčnost bloků na reálné sestavě Simatic S7-300 a pohonů s měničem Sinamics S120 v laboratoři kanceláře Siemens s.r.o. v Brně.

DOPORUČENÁ LITERATURA:

Dle vlastního literárního průzkumu a doporučení vedoucího práce.

Termín zadání:	9.2.2009	***Termín odevzdání:***	1.6.2009

Vedoucí práce: Ing. Radek Štohl, Ph.D.

prof. Ing. Pavel Jura, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Abstrakt

Dokument obsahuje základní informace o frekvenčním měniči SINAMICS S120, modulárním PLC SIMATIC S7-300 a programu SIMATIC STEP 7. Dále je součástí dokumentu funkční blok Standard telegram 9 a sada funkcí (Move Absolute, Move Relative, Move Velocity, Jog) napsaných v programu SIMATIC STEP 7, které slouží k ovládání frekvenčního měniče komunikujícího s PLC.

Abstract

This document contains basic information about the frequency converter SINAMICS S120, modular PLC S7-300 and SIMATIC STEP 7. The function block Standard telegram 9 and a set of functions (Move Absolute, Relative Move, Move Velocity, Jog) written in SIMATIC STEP 7, which is used to control the frequency converter communicating with PLC are also part of this document.

Klíčová slova

SINAMICS S120, SIMATIC STEP 7, SIMATIC S7-300, Standardní telegram 9, Funkce Move Absolute, Move Relative, Move Velocity, Jog

Keywords

SINAMICS S120, SIMATIC STEP 7, SIMATIC S7-300, Standard telegram 9, Function Move Absolute, Move Relative, Move Velocity, Jog

Bibliografická citace

VACEK, J. *Funkční bloky pro Simatic S7-300 s Sinamics S120 v rychlostních aplikacích.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 43 s. Vedoucí bakalářské práce Ing. Radek Štohl, Ph.D.

Prohlášení o původnosti

„Prohlašuji, že svou semestrální práci na téma " Funkční bloky pro Simatic S7-300 s Sinamics S120 v rychlostních aplikacích" jsem vypracoval samostatně pod vedením vedoucího semestrální 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 semestrální práce dále prohlašuji, že v souvislosti s vytvořením této semestrální 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 §152 trestního zákona č. 140/1961 Sb.“

V Brně dne : 1.6.2009 Podpis:

Poděkování

Děkuji pánům Ing. Karlovi Dočkalovi a Bc. Františkovi Šabatovi za příležitost, cenné rady a odbornou pomoc, při práci na mém projektu. Dále bych chtěl poděkovat svému vedoucímu bakalářské práce, panu Ing. Radkovi Štohlovi, Ph.D., za pedagogickou a odbornou pomoc při zpracování mé bakalářské práce.

V Brně dne: 1.6.2009 Podpis:

Obsah

Seznam obrázků

Seznam tabulek

1. Úvod

Dokument obsahuje základní informace o frekvenčním měniči SINAMICS S120, modulárním PLC SIMATIC S7-300 a programu SIMATIC STEP 7. Dále je součástí dokumentu funkční blok Standard telegram 9 a sada funkcí (Move Absolute, Move Relative, Move Velocity, Jog) napsaných v programu SIMATIC STEP 7, které slouží k ovládání frekvenčního měniče komunikujícího s PLC.

V první části dokument seznamuje s frekvenčním měničem SINAMICS S120. Jsou zde uvedeny jeho základní vlastnosti, praktické použití a popis součástí, ze kterých se měnič skládá. Ve druhé části se čtenář seznámí s PLC, programovatelným automatem, který bude frekvenční měnič ovládat. Dokument popisuje základní vlastnosti PLC SIMATIC S7-300, přehled modulů, které nabízí a přehled sítí, přes které může PLC komunikovat. Třetí část dokumentu obsahuje základní informace o programu SIMATIC STEP 7, který použijeme k naprogramování PLC. Je zde také uveden popis programových bloků a programovacích jazyků, které SIMATIC STEP 7 nabízí. Čtvrtá část dokumentu, obsahuje popis funkčního bloku Standard telegram 9, naprogramovaného v programu SIMATIC STEP 7, který se pomocí téhož programu nahraje do PLC SIMATIC S7-300 a umožní nám tak ovládat frekvenční měnič komunikující s PLC. V poslední, páté části dokumentu, je popsána sada naprogramovaných funkcí.

2. SINAMICS S120

SINAMICS S120 [1] je frekvenční měnič pro náročné aplikace pohonů především v oblasti Motion control na výrobních strojích. Je navržen pro vysoce dynamické polohování a synchronizaci více os. Pro náročné aplikace typu Motion control je SINAMICS S120 předurčen ve spojení s řídicím systémem pohonů SIMOTION D. Podobně je možno nadřadit i technologické CPU z řady SIMATIC. Tento měnič je k dispozici ve všech stavebních provedeních, takže pokrývá celé výkonové spektrum. Příklad konfigurace s měničem SINAMICS S120 je na Obr. 2.1.

Příklad typických aplikací frekvenčního měniče SINAMICS S120 – udávané společností Siemens s.r.o.:

- Balicí stroje
- Sklářské stroje
- Dřevoobráběcí stroje
- Stroje na výrobu plastických výrobků
- Textilní stroje
- List, děrovačky
- Tiskařské stroje
- Manipulátory a zdvihací zařízení
- Montážní a testovací linky

Za hlavní přednosti frekvenčního měniče SINAMICS S120 můžeme podle společnosti Siemens s.r.o. považovat:

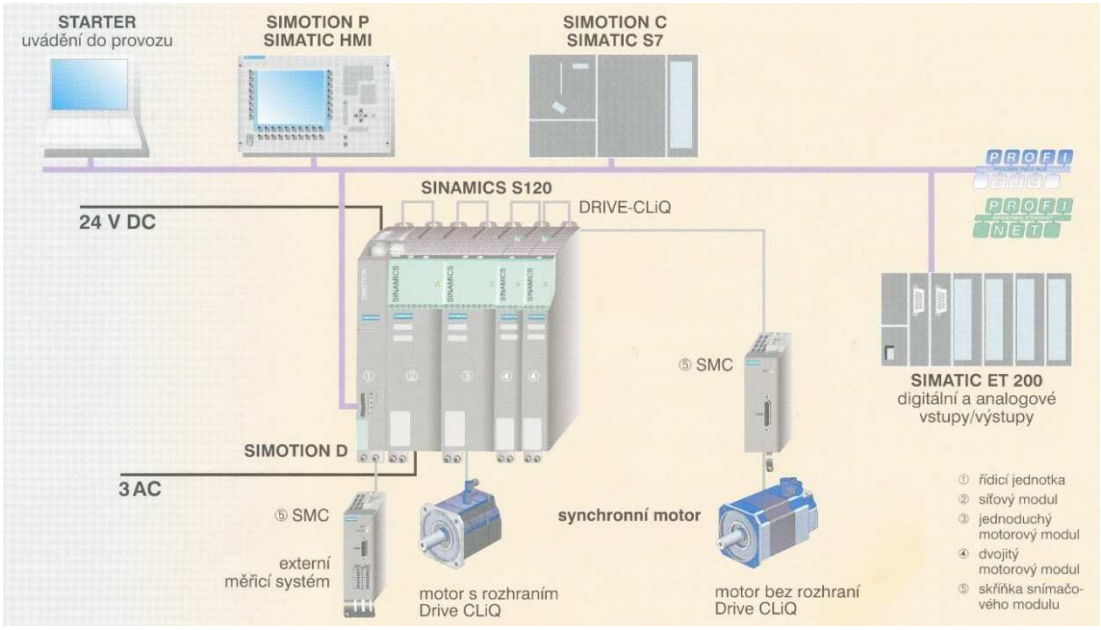
- Flexibilita daná modulární výstavbou
- Konfigurovatelná výkon, nabídka funkcí, počet os
- Jednoduché uvádění do provozu, autokonfigurace

- Spolupráce s asynchronními i synchronními motory

2.1 Struktura

- *Řídicí jednotka* – provádí výpočty pro řízení výkonové části měniče a současně poskytuje rozhraní k nadřízenému systému, s nímž komunikuje isochronním, tedy velmi rychlým a časově přesným protokolem.
- *Síťové moduly* – dodávají proud do stejnosměrného meziobvodu. Existují v provedení:
  - *Basic Line Module* – diodový můstek v napájecím směru, bez rekuperace.
  - *Smart Line Module* – diodový můstek v napájecím směru, IGBT (bipolární tranzistor s izolovaným hradlem – pro velký rozsah spínacích výkonů a vysokou pulzní frekvenci) ve zpětném směru synchronizováno s napájecí sítí.
  - *Active Line Module* – diodový můstek v napájecím směru, IGBT ve zpětném směru řízené pulsně šířkovou modulací; sinusový odběr i rekuperace proudu s definovaným cos φ.
- *Motorové moduly* – napájí připojený motor. Existuje jednomotorová a dvoumotorová jednotka se jmenovitým proudem 3 až 200 A v provedení „booksize“ a výkonem 75 až 1200 kW ve vestavném provedení.
- *Elektronické opce / snímačové a I/O moduly* – slouží k rozšíření funkčních možností, realizaci rychlých I/O, rozhraní pro enkodéry, atd.
- *Sběrnice Drive-CLiQ* – je uzavřenou sběrnicí firmy Siemens určenou k propojení jednotlivých výše uvedených modulů měniče.
  V návaznosti na použití této sběrnice na měniči vyrábí firma Siemens

i motory s tímto datovým rozhraním. Výhodou je čtení štítkových údajů přímo z motoru, včetně jedinečného čísla (ID) daného kusu motoru.



Obr. 2.1: *Elektrické pohony SIEMENS* – Příklad konfigurace s měničem SINAMICS S120 [1]

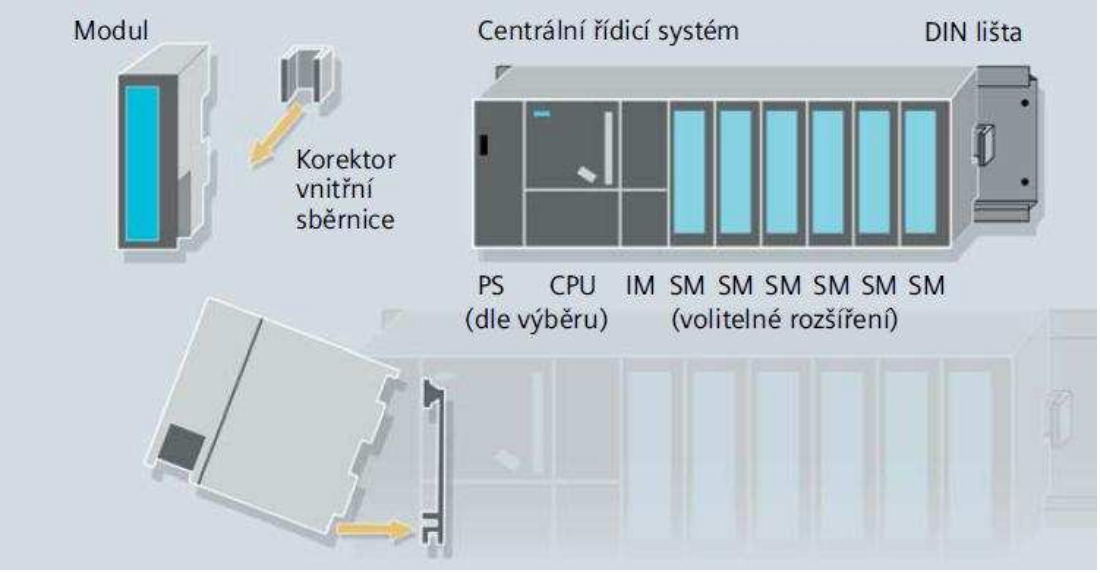
3. SIMATIC S7-300

SIMATIC S7-300 [2] je modulární mini PLC přístroj pro jednodušší a středně složité aplikace.

Základní vlastnosti PLC SIMATIC S7-300:

- Spektrum výkonově rozdílných CPU jednotek
- Rozsáhlé spektrum rozšiřujících periferních modulů
- Rozšiřitelnost přístroje až do konfigurace s 32 moduly
- Systémová sběrnice integrovaná dovnitř jednotek a modulů
- Možnost propojení do průmyslové sítě:
  - MPI (Multipoint interface)
  - PROFIBUS
  - Industrial Ethernet
- Jediné rozhranní pro připojení PG/PC přístroje s přístupem ke všem modulům konfigurace PLC přístroje
- Žádné omezení pro kombinaci jednotek a periferních modulů
- Parametrizace sestavy přístroje a nastavování parametrů pomocí programu – nástroje HWConfig z programu STEP 7

3.1 Moduly



Obr. 3.1: SIMATIC S7-300 SIEMENS – Konfigurace S7-300 [2]

- *Singulární moduly (SM)*
  - Digitální vstupní moduly (24V DC, 120/230V AC)
  - Digitální výstupní moduly (24V DC, relé)
  - Analogové vstupní moduly (běžné průmyslové rozsahy napětí a proudu, odporové snímače, termočlánky)
  - Analogové výstupní moduly (napětí, proud)
- *Propojovací moduly (IM)* – dvojice-pár propojovacích jednotek IM360/IM361 či dvě IM365 lze použít pro realizaci víceřadé konfigurace PLC přístroje. Osazení těchto jednotek do konfigurace přístroje propojí všechny moduly jedinou sběrnicí.
- *Vyplňovací moduly (DM)* – vyplňovací modul DM 370 se osazuje do konfigurace PLC jako náhrada za jednotku, která bude do konfigurace osazena později. Například proto, že parametry potřebné pro její provoz nejsou dosud známy.

- *Funkční moduly (FM)* – samostatně realizují složité řídicí funkce jako:
  - Čítání impulsů
  - Polohování
  - Zpětnovazební regulace
- *Komunikační procesory (CP)* – slouží výhradně pro připojení PLC přístroje do některého z dostupných síťových datových systémů:
  - Propojení z bodu do bodu (Point-to-point connections)
  - PROFIBUS
  - Industrial Ethernet

3.2 Komunikace

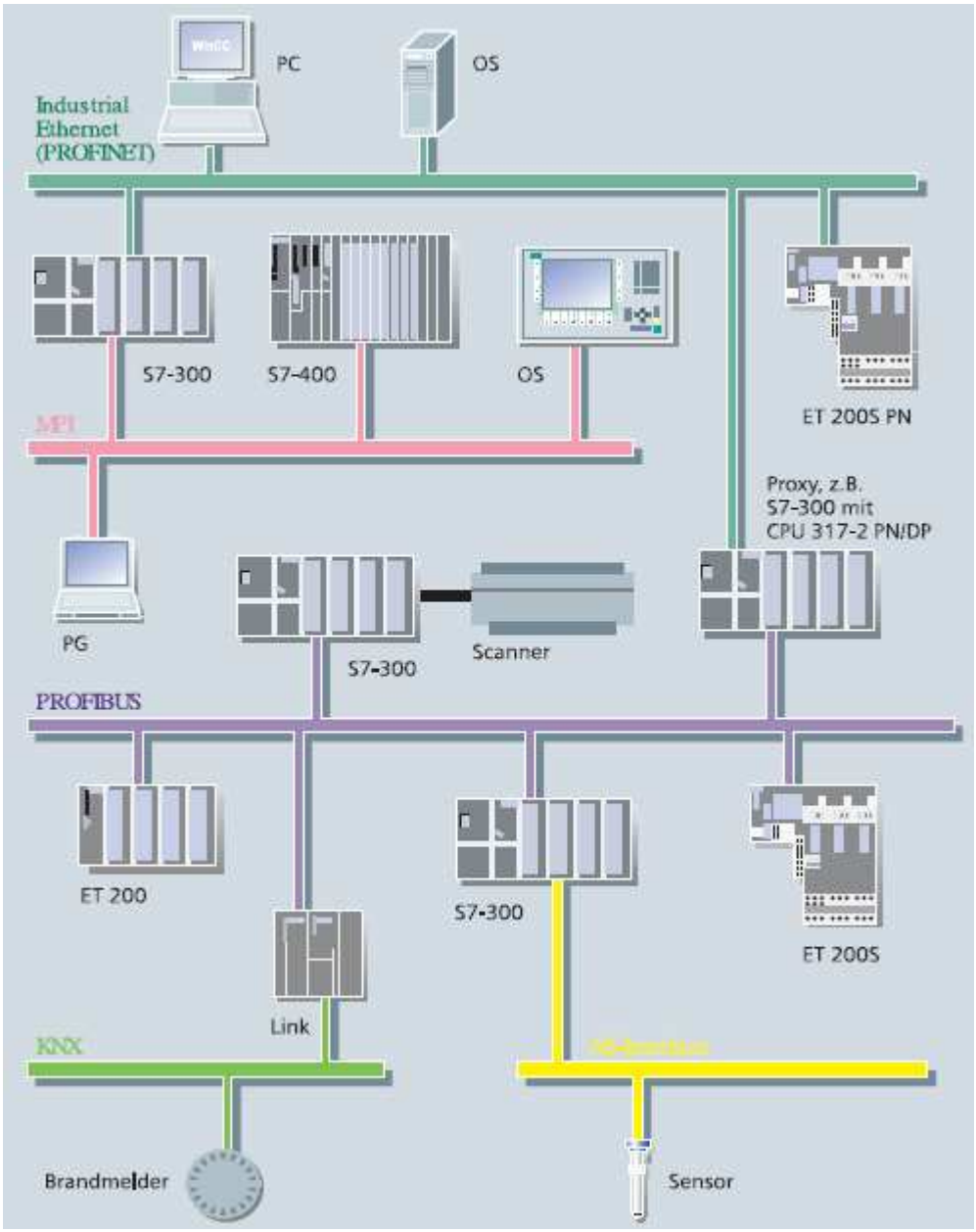
Nejdůležitější částí automatizačních systémů jsou komunikační sítě:

- *Průmyslový Ethernet (IEEE 802-3 a 802.3u)* – mezinárodní standard pro propojování rozsáhlých oblastí i jednotlivých řídicích systémů.
- *PROFIBUS (IE 61158/EN 50170)* – mezinárodní standard pro komunikaci jednotlivých řídicích systémů a polní instrumentace, stejně tak PROFIBUS PA pro jiskrově bezpečné aplikace v procesní automatizaci.
- *PROFINET (IEC 61158/EN 50170)* – je otevřený komunikační standard mezinárodní organizace PROFIBUS International (PI) založený na Ethernetu. Umožňuje jednotné a ucelené řešení pro veškeré požadavky průmyslové automatizace. Uživatelům poskytuje odstupňovanou komunikační architekturu, která pokrývá celý rozsah podnikové automatizace až ke specifickým požadavkům aplikací z oblasti řízení pohybu. PROFINET také podporuje koncepci automatizace založené na komponentech (CbA). Sdružením mechanických, elektronických a softwarových částí do jednotlivých komponent – tzv. technologických modulů, dojde k vytvoření

modulárních struktur, s jejichž pomocí lze ve specifickém softwarovém nástroji velmi jednoduše vybudovat příslušná řešení automatizačních úloh a zjednodušit tak inženýring celého výrobního procesu. Tímto se docílí vyšší míry standardizace, lepší rozšiřitelnosti řešení a možnost opětovného použití komponent v různých projektech.

- *AS-Interface (EN 50295)* – mezinárodní standard pro komunikaci mezi senzory a akčními členy.
- *EIB (EN 50090, ANSI EIA 776)* – celosvětově standardizovaný instalační systém pro nasazení při automatizaci budov.
- *MPI* – Multi point interface, pro komunikaci mezi CPUs, PG/PC a TD/OP.
- *Připojení Point-to-point* – pro komunikaci mezi dvěma uzly prostřednictvím speciálních protokolů. Point-to-point struktura představuje nejjednodušší formu komunikace. Jsou používány různé protokoly (např. RK 512, 3964(R) a ASCII).

Na Obr. 3.2 je příklad zapojení PLC SIMATIC S7-300 do všech sítí.



Obr. 3.2: SIMATIC S7-300 SIEMENS – SIMATIC S7-300 připojený do všech sítí [2]

4. SIMATIC STEP 7

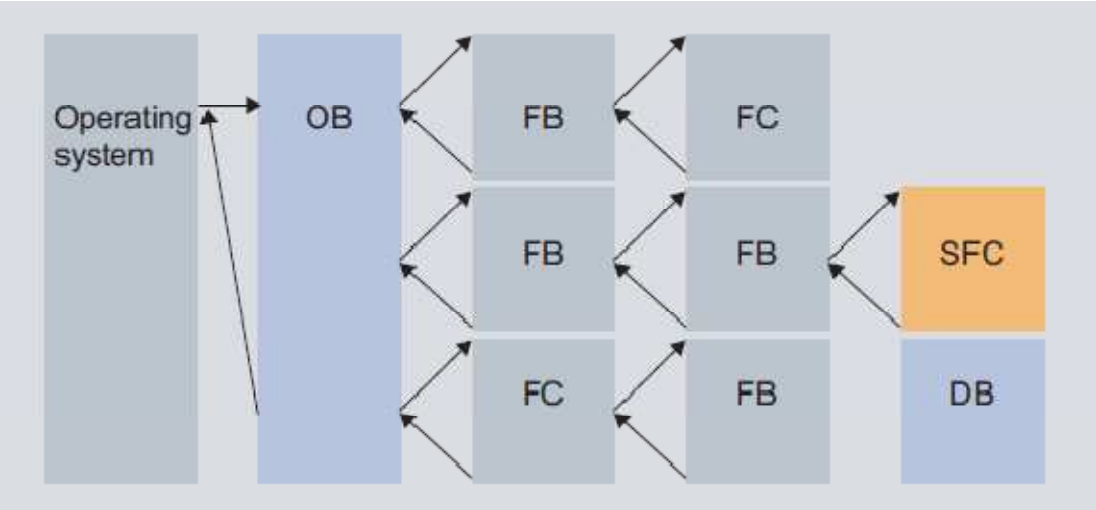
SIMATIC STEP 7 [3] je program, obsahující řadu nástrojů a funkcí pro většinu úkolů v rámci automatizačního projektu. STEP 7 Professional nabízí více programovacích jazyků než standardní balíček.

Základní nástroje a funkce, bez kterých se při tvorbě projektu neobejdeme:

- SIMATIC Manager pro správu všech nástrojů a údajů o projektu – je používán k vytváření, kopírování, stahování a archivaci projektů
- Hardware Config pro konfiguraci a parametrizaci hardware
- NetPro pro nastavení přenosu dat přes MPI nebo PROFIBUS/PROFINET

4.1 Struktura uživatelského programu

STEP 7 umožňuje strukturalizovat uživatelský program, tj. rozčlenit jej do samostatných sekcí – tzv. programových bloků. Příklad rozčlenění uživatelského programu je na Obr. 4.1.



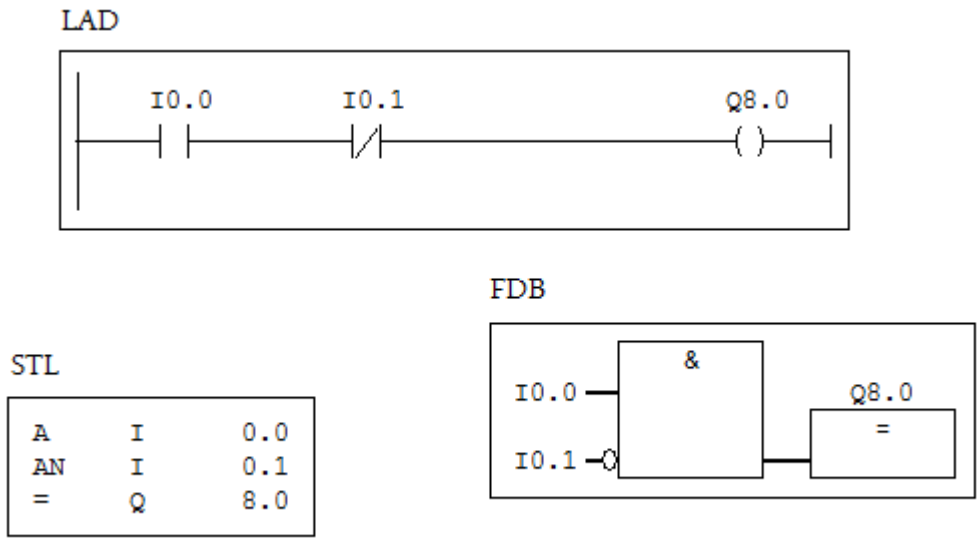
Obr. 4.1: SIMATIC Software SIEMENS – Struktura uživatelského programu [3]

- *Organizační bloky (OB)* – Organizační bloky jsou volány výhradně operačním systémem CPU. Tvoří rozhranní mezi uživatelským programem a operačním systémem CPU jednotky. V principu řídí a zajišťují:
  - Chování PLC systému při náběhu
  - Cyklické zpracování uživatelského programu
  - Zpracování přerušení
  - Správu detekovaných poruch a chybových stavů
- *Funkce (FC) a funkční bloky (FB)* – Jsou základními kameny, ze kterých se skládá uživatelský program a které umožňují rozdělit komplexní program do dílčích, vzájemně propojených částí.
- *Datové bloky (DB)* – Datové bloky jsou používány pro úschovu uživatelských dat potřebných pro řízení procesu. Na rozdíl od FC a FB neobsahují instrukce.
- *Systémové funkce a funkční bloky (SFC/SFB)* – Bloky vytvořené výrobcem PLC systému, které jsou integrovány do paměti CPU. Jsou určeny pro řešení složitých standardních dílčích úloh. Jsou uživatelem needitovatelné.
- *Systémové datové bloky (SDB)* – Slouží pro úschovu konfiguračních dat a parametrů. Jsou uživatelem needitovatelné (jejich obsah nelze zobrazit).

4.2 LAD/STL/FDB editor

Na Obr. 4.2 je příklad zobrazení jednoduchého zdrojového kódu ve všech jazycích. Pomocí volby *View* lze kdykoliv během práce v LAD/STL/FDB editoru změnit programovací jazyk, pomocí kterého bude obsah bloku zobrazen, respektive v kterém bude blok editován.

- LAD (schéma kontaktů)
- FDB (funkční schéma)
- STL (seznam instrukcí)



Obr. 4.2: LAD/STL/FDB editor

Převod zobrazení LAD/FDB => STL – Každá část programu sestavená v jazycích LAD/FDB je bezezbytku převeditelná do jazyka STL. Samozřejmě výsledkem této konverze je obvykle ne příliš efektivní zápis algoritmu v instrukcích STL.

Převod zobrazení STL => LAD/FDB – Obecně platí, že algoritmus obsažený v segmentu programu nemusí být vždy převeditelný do jazyků LAD/FDB. K tomu, aby bylo obsah segmentu možno převést do tvaru některého z vyšších programovacích jazyků, je nutné, aby program v STL byl sestaven s dodržením určitých syntaktických pravidel.

5. Standard telegram 9

Ve společnosti Siemens s.r.o. jsem měl za celý rok k dispozici více přístrojů. Vyzkoušel jsem si tak různé možnosti sestavení automatizačního projektu. Výsledné sestavy se od sebe ovšem moc nelišily. Vždy obsahovaly CPU SIMATIC S7-300 podporující sběrnici PROFIBUS, v letním semestru jsem měl již k dispozici i CPU podporující modernější PROFINET. Testovací kufry obsahovaly frekvenční měnič SINAMICS S120 a dva servomotory, které jsem pomocí mých programů ovládal.

5.1 Konfigurace projektu

Než se mohu pustit do programování, musím zajistit komunikaci mezi frekvenčním měničem, PLC a notebookem. Nebudu se zde zabývat možnostmi konfigurace projektu, kterých je opravdu hodně a pro různé přístroje se i trochu liší. Uvedu zde pouze příklad.

5.1.1 Nastavení pohonu SINAMICS S120 – STARTER

Nejdříve pár slov k programu STARTER. Je to nástroj k parametrování a monitorování všech pohonů rodiny Sinamics a Micromaster. Program umožňuje kompletní monitoring, uvádění pohonů do provozu. Všechny funkce lze editovat pomocí takzvaných expert listů. Vytvořím si nový projekt a pomocí volby Insert single drive unit si vložím do projektu měnič SINAMICS S120, který používám. Nyní je třeba vložit přes záložku Drives – Insert drive servomotor, který chci měničem ovládat. V záložce Configuration musím projít Configure DDS, kde si konfiguraci nastavím. Hned v prvním kroku si zvolím možnost Basic positioner, pro základní polohování. Dále vyberu typ Power unit a Motoru, který používám. V záložce Mechanics, si nastavím hodnotu LU per load revolution na 10000. Tato hodnota mi udává na kolik kroků se rozdělí otočení motoru o 360˚. Důležité je nezapomenout v záložce PROFIBUS process data exchange (drive) vybrat PROFIdrive message

frame: Standard telegram 9, PZD-10/5 (9), aby si motor s funkčním blokem rozuměl. Na závěr u měniče, v záložce Configuration, vyberu u již nastaveného motoru Message frame type: Standard telegram 9, PZD-10/5. Pomocí tlačítka Load project to target system nahraji vytvořenou konfiguraci do frekvenčního měniče.

5.1.2 Nastavení hardwarové konfigurace – HW Config

Otevřu si nástroj programu SIMATIC STEP 7, HW Config. Pomocí Catalogu si nastavím používanou konfiguraci. Začnu výběrem SIMATIC 300 – RACK-300 – Rail, kam vložím všechny používané moduly SIMATIC S7-300, konkrétně zdroj, CPU, vstupy a výstupy. K SIMATICU S7-300 připojím přes PROFIBUS SINAMICS S120. Nyní se vyplatí, z důvodů větší přehlednosti, nastavit v SINAMICS S120 vstupní a výstupní adresu Standardního telegramu 9 na stejnou hodnotu. Já zvolil vstup i výstup začínající na adrese 260. Připravenou konfiguraci nahrajeme pomocí tlačítka Download to Module.

Program SIMATIC STEP 7 nabízí více nástrojů, které uživateli pomohou v nastavení konfigurace projektu. Za zmínku stojí určitě i nástroj NetPro, který slouží ke grafickému zobrazení a nastavení konfigurace.

Když mám komunikaci zajištěnou, mohu začít s programováním.

5.2 Programování funkčního bloku

Pustím se tedy do programování samotného funkčního bloku pro ovládání frekvenčního měniče, který využívá Standardního telegramu 9. Nejprve je potřeba zjistit počet vstupů a výstupů Standardního telegramu 9. Tuto informaci mohu najít buď v programu STARTER (Drives – Communication – PROFIBUS), nebo v manuálu od společnosti Siemens AG, SINAMICS S List Manual (LH1). V Tab. 5.1 je výtah z tabulky standardních telegramů z manuálu společnosti Siemens AG.

Telegram	9	
Appl.- Class	3	
PZD 1	STW1	ZSW1
PZD 2	SATZANW	AKTSATZ
PZD 3	STW2	ZSW2
PZD 4	MDIPos	XIST_A
PZD 5		
PZD 6	MDIVel	
PZD 7		
PZD 8	MDIAcc	
PZD 9	MDIDec	
PZD 10	MDIMod	

Tab. 5.1: SINAMICS S List Manual(LH1) SIEMENS AG – Standard telegram 9 [4]

Podle tabulky 5.1 je zřejmé, že Standard telegram 9 bude mít velikost vstupů v paměti 10 Wordů (Word = 2 Byte) a velikost výstupů 5 Wordů.

5.2.1 Popis vstupů a výstupů Standardního telegramu 9

Vstupy:

- Řídicí slova STW1 a STW2
- Řídicí slovo SATZANW
- MDIPos – nastavení pozice měniče
- MDIVel – nastavení rychlosti měniče
- MDIAcc – nastavení zpomalení měniče
- MDIMod – nastavení módu měniče

Výstupy:

- Stavová slova ZSW1 a ZSW2
- Stavové slovo AKSATZ
- XIST_A – aktuální pozice měniče

5.2.2 Postup při programování

Při programování použiji již naprogramované komunikační bloky SFC14 a SFC15. SFC14 slouží k načtení dat z měniče do CPU - TEMP. Na vstup bloku LADDR dám adresu, kde mi začínají vstupy (260). Blok má dva výstupy. RET_VAL, kde musí být hodnota 0, pokud proběhl přesun v pořádku a výstup RECORD, kde napíši, kam se budou hodnoty ukládat a počet hodnot, které chci uložit. Konkrétně tedy P#L 0.0 BYTE 10 – načti 10 bajtů do TEMP proměnných od adresy 0.0. SFC15 slouží k nahrání dat z CPU – TEMP do měniče. SCF15 má dva vstupy. LADDR, kam zadám, kde mi začínají výstupy (260) a RECORD, kde napíši od které adresy a kolik bajtů budu přesouvat. Konkrétně P#L 10.0 BYTE 20. 20 bajtů od adresy 10.0 v TEMP proměnných. Výstup RET_VAL mi opět kontroluje, zda proběhl přenos v pořádku.

- Nejprve naplním Interface všemi potřebnými proměnnými:
  - IN : Vstupni_adresa; Vystupni_adresa; výpis řídicích slov STW1, STW2 a SATZANW po bitech (výpis jednotlivých Wordů po bitech najdu například ve STARTERU – Drives – Communication – PROFIBUS); MDIPos; MDIVel; MDIAcc; MDIDec; MDIMod
  - OUT: výpis stavových slov ZSW1, ZSW2, AKSATZ; XIST_A
  - TEMP: zde musím vypsat vstupy a výstupy přesně v takovém pořadí, s jakým budou SFC14 a SFC15 pracovat – ZSW1, AKSATZ, ZSW2, XIST_A, STW1, SATZANW, STW2, MDIPos, MDIVel, MDIAcc, MDIDec, MDIMod; u ostatních TEMP proměnných již na pořadí nezáleží, slouží pouze jako pomocné
- Program začíná vyprázdněním TEMP pro telegram jdoucí z měniče do CPU – 5 Wordů a pro telegram jdoucí z CPU do měniče – 10 Wordů
- Načtení dat z měniče do CPU – TEMP pomocí SFC14

- Přesun stavových slov na výstup (z TEMP do OUT)
- Přesun řídicích slov na vstup (z IN do TEMP)
- Nastavení pozice, rychlosti, zrychlení, zpomalení, módu a ošetření proti chybnému zadání
- Výpis aktuální pozice
- Nahrání dat z CPU – TEMP do měniče pomocí SFC15
- Vytvořím si v TEMP pole typu BOOL o velikosti osm bitů, do kterého si budu ukládat záznam o chybách – pokud bude mít některý z bitů velikost 1, došlo při zadávání parametrů k chybě:
  - ERROR[0] = 1, špatně zadaná rychlost
  - ERROR[1] = 1, špatně zadané zrychlení
  - ERROR[2] = 1, špatně zadané zpomalení
  - ERROR[3] = 1, špatně zadaný mód

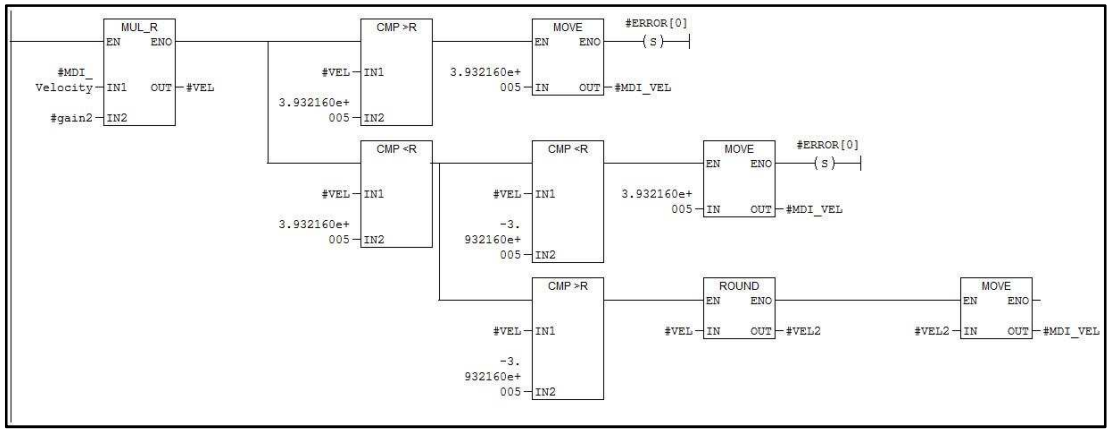
5.2.3 Popis vybraného zdrojového kódu

- *Přesun stavového slova ZSW1 na vstup v jazyce STL* - při psaní kódu v STL musím dávat pozor na to, že bity 0-7 musím psát jako bity 8-15. A bity 8-15 píši jako bity 0-7 (problematika Big-endian vs. Little-endian). Když se podíváte na Obr. 5.1, tak první příkaz znamená, že první bit stavového slova ZSW1 (TEMP/Array) přesunu do proměnné Ready_to_power_up (OUT/Bool).

```
A       #ZSW1[8]
=       #Ready_to_power_up

A       #ZSW1[9]
=       #Ready

A       #ZSW1[10]
=       #Operation_enabled

A       #ZSW1[11]
=       #Fault_present
```

Obr. 5.1: Přesun prvních čtyř bitů stavového slova ZSW1 na vstup v jazyce STL

- *Nastavení rychlosti v jazyce LAD* – na Obr. 5.2 je ošetření zadávání rychlosti v jazyce LAD. V manuálu – SINAMICS S List Manual (LH1) – se můžu dočíst, že standardní normalizace v měničích je 1 hex = 1000 LU/min. V programu STARTER, v Expert listu si najdu parametr p2000, kde si zjistím maximální rychlost servomotoru, tj. 6000 rev/min. Při mém nastavení 10000 LU per load revolution je tedy 100% = 60000 hex. Rychlost se bude zadávat v procentech. První blok, MUL_R, slouží k násobení dvou reálných čísel. Násobíme mezi sebou MDI_Velocity – rychlost zadaná v hodnotách 0-100(%) a gain2 – statická proměnná o velikosti 3932,16 (1% maximální rychlosti servomotoru). Nyní musím pomocí podmínek zajistit, aby nebyla rychlost zadaná špatně. V první větvi kontroluji, zda není rychlost zadaná větší než 100%. Pokud ano, nastavím rychlost na 100% a první bit v poli ERROR na 1. V druhé větvi kontroluji, zda nezadávám rychlost záporně. Pokud ano, tak nastavím automaticky 100% a opět nastavím první bit v poli ERROR na 1. Ve třetí větvi byla rychlost zadaná správně, tj. mezi 0-100%. Hodnotu pouze zaokrouhlím na celé číslo a rychlost nastavím.



Obr. 5.2: Nastavení rychlosti v jazyce LAD

Všechny potřebné údaje, které jsem při programování potřeboval, se mohu dočíst přímo v programu STARTER nebo v manuálu SINAMICS S List Manual (LH1). Tento manuál je trochu obsáhlejší a ze začátku je potřeba trochu trpělivosti, než se v něm člověk vyzná, ale našel jsem tam úplně vše, co jsem k práci se standardním telegramem potřeboval. Obsáhlejší dokumentace neexistuje.

5.3 Testování programu

Funkční blok mám již hotový. Musím si vytvořit organizační blok, do kterého si svůj funkční blok vložím. K funkčnímu bloku přidám název datového bloku. Program se mě zeptá, jestli chci vytvořit datový blok automaticky. S volbou souhlasím a program mi ho vytvoří. Funkční blok musím vložit do organizačního bloku proto, že je volaný cyklicky – jede neustále ve smyčce. V organizačním bloku vidím pěkně vstupy i výstupy a mohu si je libovolně nastavit. Nastavil jsem si bity Vstupni_adresa a Vystupni_adresa na 260, ONOFF1 na I0.0 (digitální vstup na PLC 0.0). Všechny bloky (funkční, organizační a datový blok, SFC14 i SFC15) nyní musím stáhnout do PLC pomocí tlačítka Download.

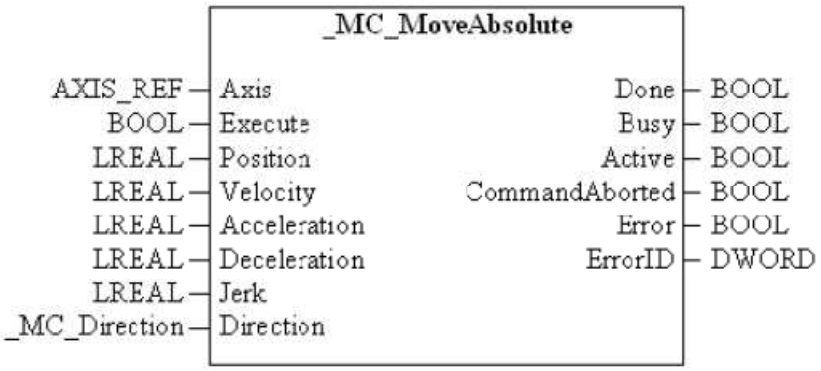
Pro otestování funkčnosti telegramu, si vyberu například funkci Jog, která je popsána v programu STARTER v záložce Basic Positioner. V SIMATIC Manageru jsem si vytvořil Variable Table (viz Tab. 5.2), kde jsem si nastavil potřebné bity k zprovoznění této funkce. Podle STARTERU si musím nastavit bit EPOS_activate a EPOS_jog_1 na hodnotu true. Najdu si tedy v datovém bloku adresy, na kterých se tyto bity nacházejí a pomocí Variable Table si je nastavím. Ještě bych chtěl nastavit rychlost a tak si MDI_Velocity nastavím třeba na 10. Servomotor se mi otáčí rychlostí 10% z jeho maximální rychlosti.

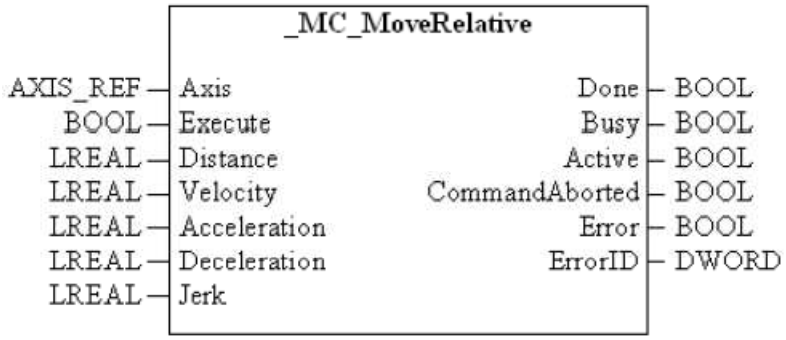
	Address	Symbol	Display format	Status value	Modify value
1	DB1.DBX 4.6	"DB pro ST9".EPOS_activate	BOOL		true
2	DB1.DBX 5.0	"DB pro ST9".EPOS_jog_1	BOOL		true
3	DB1.DBD 12	"DB pro ST9".MDI_Velocity	FLOATING_POI...		10.0
4					

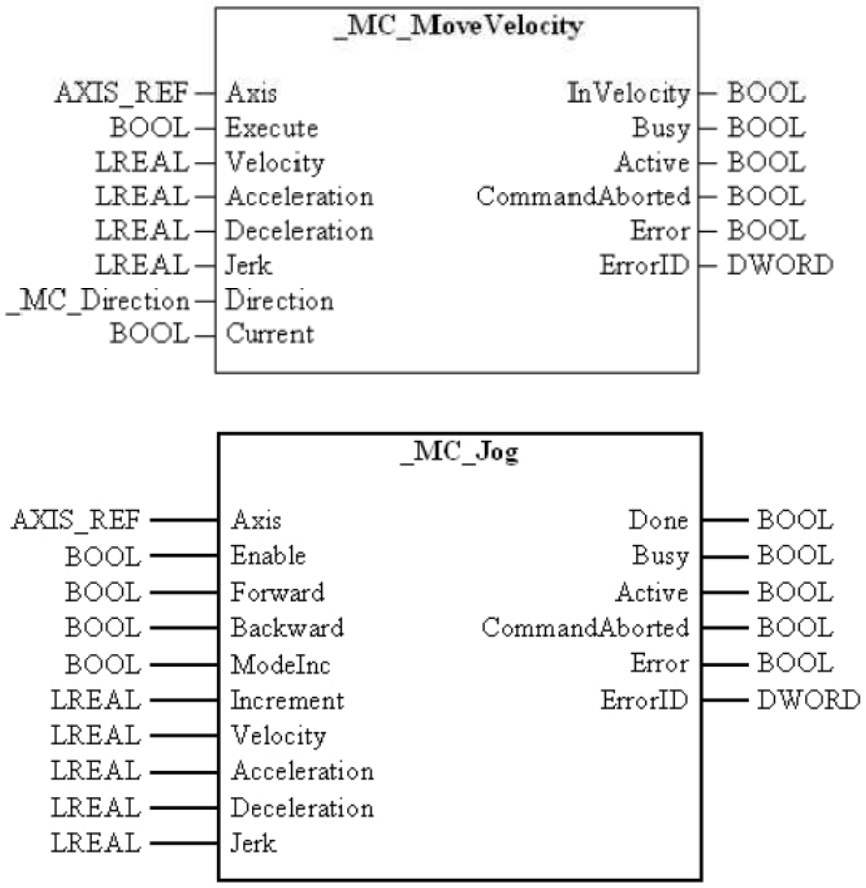
Tab. 5.2: Variable Table

6. Sada funkcí pro rychlostní aplikace

Společnost Siemens nabízí PLCopen bloky, které jsou určeny pro použití v cyklických programech a umožňují Motion Control (řízení polohování jednotlivých os nebo systémů s větším počtem os v rámci jednoho zařízení nebo stroje) v PLC. Je preferováno jejich použití v programovacích jazycích LAD a FDB. PLCopen bloky jsou k dispozici jako standardní funkce. Slouží však jen pro řízení v řídicích systémech SIMOTION. V kanceláři Siemens s.r.o. jsem dostal za úkol naprogramovat vybrané bloky v programu SIMATIC STEP 7, aby bylo možné funkce použít pro řídicí systémy SIMATIC. Na Obr. 6.1 jsou schémata vybraných funkcí.







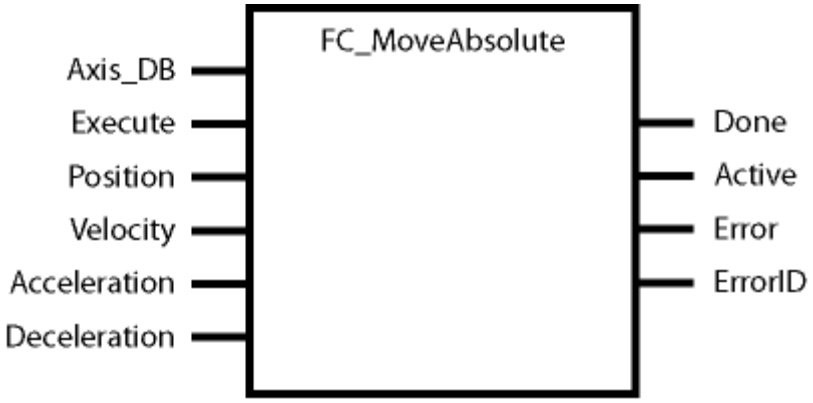
Obr. 6.1: SIMOTION PLCopen Blocks – Schémata vybraných funkcí [5]

- *Move Absolute* – startuje nastavení polohy osy podle absolutní pozice. Uživatel definuje parametry pohybu. Osa zastaví po dosažení absolutní pozice.
- *Move Relative* - startuje nastavení polohy osy podle relativní pozice. Uživatel definuje parametry pohybu. Osa zastaví po dosažení relativní pozice.
- *Move Velocity* – zrychluje nebo zpomaluje osu na požadovanou rychlost. Uživatel definuje parametry pohybu.
- *Jog* – startuje funkci Jog.

Jednotlivé vstupy a výstupy PLCopen bloků jsem upravil tak, aby vyhovovaly řízení pomocí řídicích systémů SIMATIC a požadavkům společnosti Siemens s.r.o. V následujících podkapitolách jsou jednotlivé funkce podrobněji rozebrány.

6.1 Move Absolute

- Nastavení polohy osy podle absolutní pozice
- Uživatel definuje parametry pohybu:
  - Absolutní pozice – je pozice, na kterou se má osa nastavit
  - Rychlost
  - Zrychlení
  - Zpomalení
- Osa zastaví po dosažení absolutní pozice



Obr. 6.2: Schéma funkce Move Absolute

Vstupní parametry

Jméno	**Datový typ**	**Komentář**
Axis_DB	Block_DB	Otevře datový blok pro požadovaný standardní telegram
Execute	Bool	Funkce povolena. Startuje nastavení polohy osy podle absolutní pozice
Position	DInt	Specifikuje absolutní, cílovou pozici pohybu
Velocity	Real	Specifikuje maximální rychlost pohybu. Rychlost je dosažena v závislosti na hodnotách Position a Acceleration
Acceleration	Real	Specifikuje maximální zrychlení
Deceleration	Real	Specifikuje maximální zpomalení

Výstupní parametry

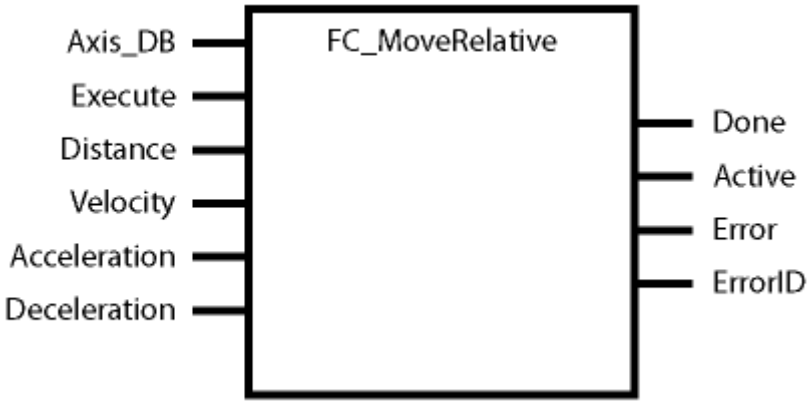
Jméno	**Datový typ**	**Komentář**
Done	Bool	Pokud má bit Done hodnotu true, dosažena cílová pozice
Active	Bool	Pokud má bit Active hodnotu true, funkce je aktivní
Error	Bool	Indikace chyby
ErrorID	Int	Identifikace chyby

Tab. 6.1: Vstupní a výstupní parametry funkce Move Absolute

Na Obr. 6.2 je schéma funkce Move Absolute, kde jsou vidět vstupy a výstupy funkce. Popis funkcí jednotlivých vstupů a výstupů je potom zobrazen v Tab. 6.1. Funkce Move Absolute je kompatibilní se Standardními telegramy 9, 110 a 111.

6.2 Move Relative

- Nastavení polohy osy podle relativní pozice
- Uživatel definuje parametry pohybu:
  - Relativní pozice – je vzdálenost, o kterou se má osa otočit
  - Rychlost
  - Zrychlení
  - Zpomalení
  - Směr – definuje, na kterou stranu se má osa otočit
- Osa zastaví po dosažení relativní pozice



Obr. 6.3: Schéma funkce Move Relative

Vstupní parametry

Jméno	**Datový typ**	**Komentář**
Axis_DB	Block_DB	Otevře datový blok pro požadovaný standardní telegram
Execute	Bool	Funkce povolena. Startuje nastavení polohy osy podle
		relativní pozice
Distance	DInt	Specifikuje relativní, cílovou pozici pohybu
Velocity	Real	Specifikuje maximální rychlost pohybu. Rychlost je
		dosažena v závislosti na hodnotách Position a Acceleration
Acceleration	Real	Specifikuje maximalní zrychlení
Deceleration	Real	Specifikuje maximalní zpomalení
Direction	Bool	Specifikuje směr pohybu

Výstupní parametry

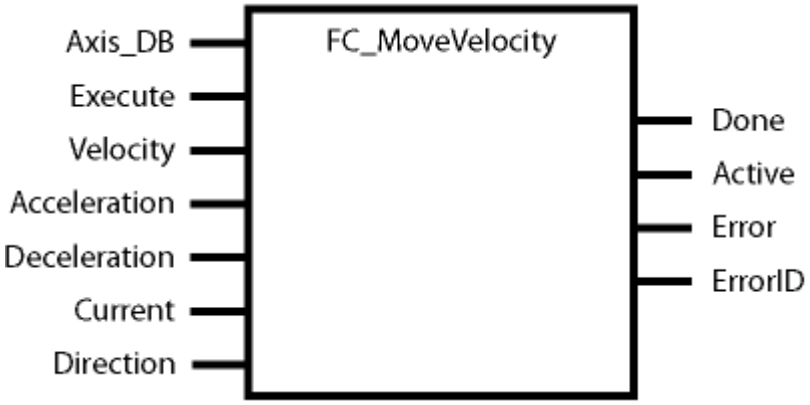
Jméno	**Datový typ**	**Komentář**
Done	Bool	Pokud má bit Done hodnotu true, dosažena cílová pozice
Active	Bool	Pokud má bit Active hodnotu true, funkce je aktivní
Error	Bool	Indikace chyby
ErrorID	Int	Identifikace chyby

Tab. 6.2: Vstupní a výstupní parametry funkce Move Relative

Na Obr. 6.3 je schéma funkce Move Relative, kde jsou vidět vstupy a výstupy funkce. Popis funkcí jednotlivých vstupů a výstupů je potom zobrazen v Tab. 6.2. Funkce Move Relative je kompatibilní se Standardními telegramy 9, 110 a 111.

6.3 Move Velocity

- Zrychluje nebo zpomaluje osu na požadovanou rychlost
- Uživatel definuje parametry pohybu:
  - Rychlost
  - Zrychlení
  - Zpomalení
  - Směr – definuje, na kterou stranu se má osa otáčet



Obr. 6.4: Schéma funkce Move Velocity

Vstupní parametry

Jméno	**Datový typ**	**Komentář**
Axis_DB	Block_DB	Otevře datový blok pro požadovaný standardní telegram
Execute	Bool	Funkce povolena.
Velocity	Real	Specifikuje maximální rychlost pohybu. Rychlost je
		dosažena v závislosti na hodnotě Acceleration
Acceleration	Real	Specifikuje maximalní zrychlení
Deceleration	Real	Specifikuje maximalní zpomalení
Current	Bool	Pokud má bit Current hodnotu true, bude použita aktuální,
		posledně nastavená rychlost osy. Pokud má bit hodnotu
		false, nastaví se rychlost vstupem Velocity
Direction	Bool	Specifikuje směr pohybu

Výstupní parametry

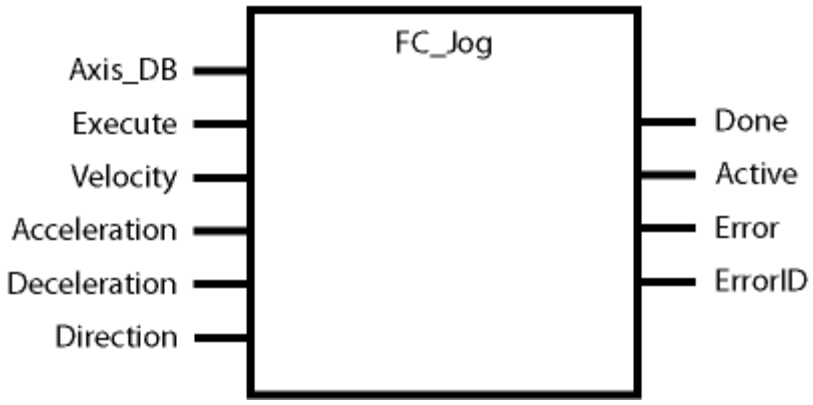
Jméno	**Datový typ**	**Komentář**
Done	Bool	Pokud má bit Done hodnotu true, rychlost nastavena
Active	Bool	Pokud má bit Active hodnotu true, funkce je aktivní
Error	Bool	Indikace chyby
ErrorID	Int	Identifikace chyby

Tab. 6.3: Vstupní a výstupní parametry funkce Move Velocity

Na Obr. 6.4 je schéma funkce Move Velocity, kde jsou vidět vstupy a výstupy funkce. Popis funkcí jednotlivých vstupů a výstupů je potom zobrazen v Tab. 6.3. Funkce Move Velocity je kompatibilní se Standardními telegramy 1, 2, 9, 110 a 111.

6.4 Jog

- Startuje funkci Jog
- Uživatel definuje parametry pohybu:
  - Rychlost
  - Zrychlení
  - Zpomalení
  - Směr – definuje, na kterou stranu se má osa otáčet



Obr. 6.5: Schéma funkce Jog

Vstupní parametry

Jméno	**Datový typ**	**Komentář**
Axis_DB	Block_DB	Otevře datový blok pro požadovaný standardní telegram
Execute	Bool	Funkce povolena
Velocity	Real	Specifikuje maximální rychlost
Acceleration	Real	Specifikuje maximální zrychlení
Deceleration	Real	Specifikuje maximální zpomalení
Direction	Bool	Specifikuje směr pohybu

Výstupní parametry

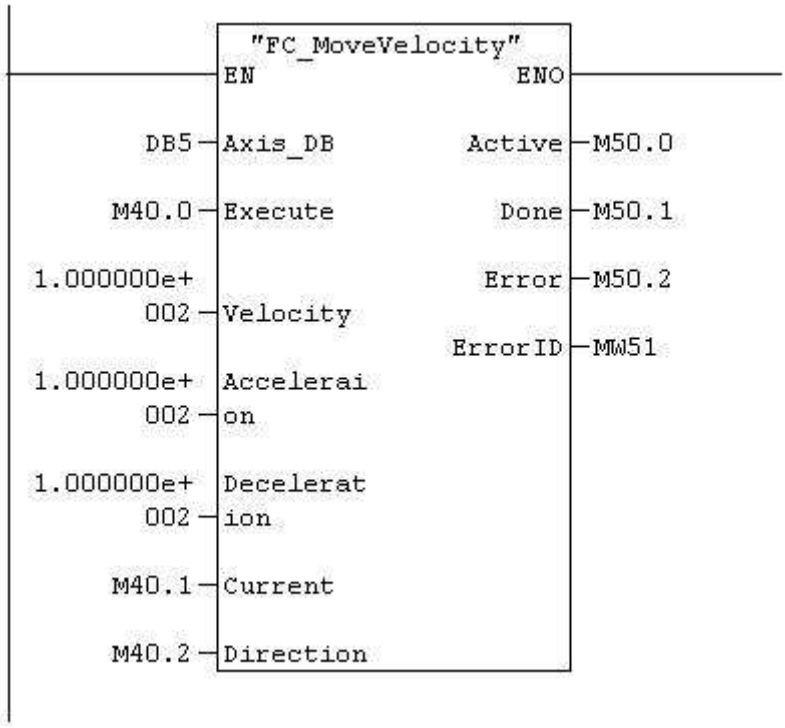
Jméno	**Datový typ**	**Komentář**
Done	Bool	Pokud má bit Done hodnotu true, funkce nastavena
Active	Bool	Pokud má bit Active hodnotu true, funkce je aktivní
Error	Bool	Indikace chyby
ErrorID	Int	Identifikace chyby

Tab. 6.4: Vstupní a výstupní parametry funkce Jog

Na Obr. 6.5 je schéma funkce Jog, kde jsou vidět vstupy a výstupy funkce. Popis funkcí jednotlivých vstupů a výstupů je potom zobrazen v Tab. 6.4. Funkce Jog je kompatibilní se Standardními telegramy 9, 110 a 111.

6.5 Použití funkcí pro rychlostní aplikace

Konfigurace automatizačního projektu je hotová. Stačí tedy nahrát všechny bloky do PLC. Do OB1 vložím Standardní telegram, pomocí kterého chci frekvenční měnič ovládat a funkci, kterou chci použít. Na vstup funkce Axis_DB napíši název datového bloku, který patří k vybranému telegramu. Potom už jen na vstup přivedu nějaké hodnoty, buď přímo nebo pomocí merkerů. Výstupům také přiřadím nějaké merkery, které mohu například ve VAT tabulce sledovat. Na Obr. 6.6 je příklad funkce Move Velocity vložené do OB1. Funkce má nastavené vstupy a výstupy. Použití funkcí uživateli velmi usnadní ovládání frekvenčního měniče.



Obr. 6.6: Funkce Move Velocity vložená do OB1

7. Závěr

Ve třetím ročníku bakalářského studia na fakultě elektrotechniky a komunikačních technologií VUT v Brně jsem navštěvoval kancelář společnosti Siemens s.r.o., kde jsem měl možnost pracovat s produkty této firmy. Seznámil jsem se s frekvenčními měniči, motory, programovatelnými automaty a hlavně s programem SIMATIC STEP 7, ve kterém jsem pracoval s již hotovými projekty na ovládání frekvenčních měničů a později napsal i své vlastní programy, které v této práci prezentuji. Dokument popisuje základní informace potřebné k vytvoření automatizačního projektu s použitím frekvenčního měniče SINAMICS S120, komunikujícím s modulárním PLC SIMATIC S7-300. Komunikace mezi PLC a měničem probíhá prostřednictvím moderní sběrnice PROFINET v RT režimu.

Společností Siemens s.r.o. bude udělána revize, mnou naprogramovaných funkčních bloků a funkcí. Poté budou nabízeny zákazníkům společnosti. Funkční blok Standard telegram 9 jsem otestoval a všechny testované funkce fungují v pořádku. Sadu funkcí za otestovanou zatím považovat nemohu.

Seznam použité literatury

[1] *Elektrické pohony SIEMENS*. Brno: Siemens s.r.o., vydání září 2006, oprava duben 2007. 19 s.

[2] *SIMATIC S7-300 SIEMENS*. Praha: Siemens s.r.o., 2005. 19 s.

[3] *SIMATIC Software SIEMENS*. Nürnberg: Siemens AG, duben 2008. 27 s.

[4] *SINAMICS S List Manual (LH1) SIEMENS*. Erlagen: Siemens AG, červenec 2007. 1816 s.

[5] *SIMOTION PLCopen Blocks Function Manual*. Nürnberg: Siemens AG, srpen 2008. 118 s.

Seznam zkratek

CP – Komunikační procesor

CPU – Procesor

DB – Datový blok

DM – Vyplňovací modul

FB – Funkční blok

FC – Funkce

FDB – Programovací jazyk, funkční schéma

FM – Funkční modul

IM – Propojovací modul

LAD – Programovací jazyk, schéma kontaktů

MPI – Multi point interface

OB – Organizační blok

PLC – Programovatelný automat

SDB – Systémový datový blok

SFB – Systémový funkční blok

SFC – Systémová funkce

SM – Singulární modul

STL – Programovací jazyk, seznam instrukcí

Seznam příloh

Zdrojový kód

A.1 Funkční blok - Standard telegram 9

FB9 - <offline>

"FB_ST9"

Name: **Family:**
Author: Vacek **Version:** 0.1
Block version: 2
Time stamp Code: 05/26/2009 05:55:11 PM
Interface: 05/26/2009 05:55:11 PM
Lengths (block/logic/data): 02120 01668 00080

Name	Data Type	Address	Initial Value	Comment
IN		0.0		
Vstupni_adresa	Int	0.0	256	Vstupni adresa
Vystupni_adresa	Int	2.0	256	Vystupni adresa
ONOFF1	Bool	4.0	FALSE	ON/OFF1
OFF2	Bool	4.1	TRUE	1.OFF2
OFF3	Bool	4.2	TRUE	1.OFF3
Enable_operation	Bool	4.3	TRUE	Enable operation
EPOS_reject	Bool	4.4	FALSE	EPOS reject traversing task
EPOS_stop	Bool	4.5	FALSE	EPOS intermediate stop
EPOS_activate	Bool	4.6	FALSE	EPOS activate traversing task
Acknowledge_faults	Bool	4.7	FALSE	1.Acknowledge faults
EPOS_jog_1	Bool	5.0	FALSE	EPOS jog 1 signal source
EPOS_jog_2	Bool	5.1	FALSE	EPOS jog 2 signal source
Master_ctrl_by_PLC	Bool	5.2	TRUE	Master ctrl by PLC
EPOS_start	Bool	5.3	FALSE	EPOS referencing start
EPOS_bit_0	Bool	5.4	FALSE	EPOS traversing block selection, bit 0
EPOS_bit_1	Bool	5.5	FALSE	EPOS traversing block selection, bit 1
EPOS_bit_2	Bool	5.6	FALSE	EPOS traversing block selection, bit 2
EPOS_bit_3	Bool	5.7	FALSE	EPOS traversing block selection, bit 3

Name	Data Type	Address	Initial Value	Comment
EPOS_bit_4	Bool	6.0	FALSE	EPOS traversing block selection, bit 4
EPOS_bit_5	Bool	6.1	FALSE	EPOS traversing block selection, bit 5
EPOS_direct_setpoint_in	Bool	6.2	FALSE	EPOS direct setpoint input/MDI selection
DDS_bit_0	Bool	6.3	FALSE	Drive data set selection DDS bit 0
DDS_bit_1	Bool	6.4	FALSE	Drive data set selection DDS bit 1
DDS_bit_2	Bool	6.5	FALSE	Drive data set selection DDS bit 2
DDS_bit_3	Bool	6.6	FALSE	Drive data set selection DDS bit 3
DDS_bit_4	Bool	6.7	FALSE	Drive data set selection DDS bit 4
Parking_axis_selection	Bool	7.0	FALSE	Parking axis selection
Activates_travel	Bool	7.1	FALSE	Activates travel to a fixed stop
Motor_changeover	Bool	7.2	FALSE	Motor changeover, feedback signal
Master_bit_0	Bool	7.3	FALSE	Master sign of life bit 0
Master_bit_1	Bool	7.4	FALSE	Master sign of life bit 1
Master_bit_2	Bool	7.5	FALSE	Master sign of life bit 2
Master_bit_3	Bool	7.6	FALSE	Master sign of life bit 3
MDI_Position	Real	8.0	0.000000e+000	MDI_TARPOS
MDI_Velocity	Real	12.0	0.000000e+000	MDI_VELOCITY
MDI_Acceleration	Real	16.0	0.000000e+000	MDI_ACC
MDI_Deceleration	Real	20.0	0.000000e+000	MDI_DEC
MDI_Mode	Real	24.0	0.000000e+000	MDI_MOD
OUT		0.0		
Ready_to_power_up	Bool	28.0	FALSE	Ready to power up

Name	Data Type	Address	Initial Value	Comment
Ready	Bool	28.1	FALSE	Ready
Operation_enabled	Bool	28.2	FALSE	Operation enabled
Fault_present	Bool	28.3	FALSE	Fault present
No_coasting_active	Bool	28.4	FALSE	No coasting active
No_quick_stop_active	Bool	28.5	FALSE	No Quick Stop active
Power_on_inhibit	Bool	28.6	FALSE	Power-on inhibit active
Alarm_present	Bool	28.7	FALSE	Alarm present
Error_in_tolerance	Bool	29.0	FALSE	Following error in tolerance
Control_requested	Bool	29.1	FALSE	Control requested
Target_position	Bool	29.2	FALSE	Target position reached
Reference_point_set	Bool	29.3	FALSE	Reference point set
Traversing_activated	Bool	29.4	FALSE	Acknowledgement, traversing block activated
n_act_speed_threshold	Bool	29.5	FALSE	|n_act| < speed threshold value 3
Traversing_bit_0	Bool	29.6	FALSE	Active traversing block, bit 0
Traversing_bit_1	Bool	29.7	FALSE	Active traversing block, bit 1
Traversing_bit_2	Bool	30.0	FALSE	Active traversing block, bit 2
Traversing_bit_3	Bool	30.1	FALSE	Active traversing block, bit 3
Traversing_bit_4	Bool	30.2	FALSE	Active traversing block, bit 4
Traversing_bit_5	Bool	30.3	FALSE	Active traversing block, bit 5
MDI_active	Bool	30.4	FALSE	MDI active
DDS_eff_bit_0	Bool	30.5	FALSE	DDS eff., bit 0
DDS_eff_bit_1	Bool	30.6	FALSE	DDS eff., bit 1
DDS_eff_bit_2	Bool	30.7	FALSE	DDS eff., bit 2
DDS_eff_bit_3	Bool	31.0	FALSE	DDS eff., bit 3
DDS_eff_bit_4	Bool	31.1	FALSE	DDS eff., bit 4

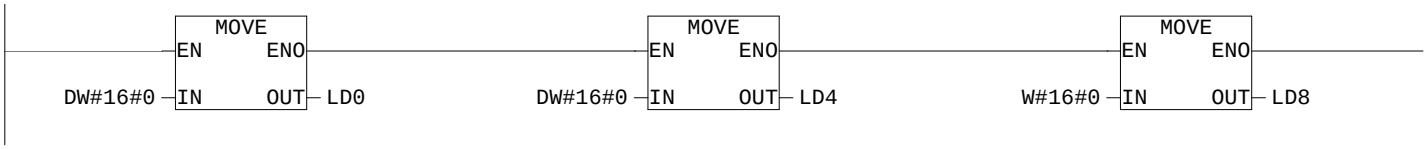
Name	Data Type	Address	Initial Value	Comment
Parking_axes_active	Bool	31.2	FALSE	Parking axis active
Travel_to_fixed_stop	Bool	31.3	FALSE	Travel to fixed stop active
Motor_changeover_set	Bool	31.4	FALSE	Motor data set changeover active
Slave_bit_0	Bool	31.5	FALSE	Slave sing-of-life bit 0
Slave_bit_1	Bool	31.6	FALSE	Slave sing-of-life bit 1
Slave_bit_2	Bool	31.7	FALSE	Slave sing-of-life bit 2
Slave_bit_3	Bool	32.0	FALSE	Slave sing-of-life bit 3
Ret_val_SFC14	Word	34.0	W#16#0	Navratova hodnota pri cteni dat z menice (0=OK)
Ret_val_SFC15	Word	36.0	W#16#0	Navratova hodnota pri zapisu dat do menice (0=OK)
Actual_position	DInt	38.0	L#0	XIST_A
IN_OUT		0.0		
STAT		0.0		
Adresa_SFC14	Word	42.0	W#16#0	
Adresa_SFC15	Word	44.0	W#16#0	
gain	Real	46.0	1.638400e+002	100%=16384
gain2	Real	50.0	3.932160e+003	100%=393216 (pri 10000 LU per load revolution)
TEMP		0.0		
ZSW1	Array [0..15] Of Bool	0.0		ZSW1
AKTSATZ	Array [0..15] Of Bool	2.0		AKTSATZ
ZSW2	Array [0..15] Of Bool	4.0		ZSW2
Actual_position_1	DWord	6.0		XIST_A
STW1	Array [0..15] Of Bool	10.0		STW1
SATZANW	Array [0..15] Of Bool	12.0		SATZANW
STW2	Array [0..15] Of Bool	14.0		STW2
MDI_POS	DWord	16.0		MDI_TARPOS
MDI_VEL	DWord	20.0		MDI_VELOCITY
MDI_ACC	Word	24.0		MDI_ACC

Name	Data Type	Address	Initial Value	Comment
MDI_DEC	Word	26.0		MDI_DEC
MDI_MOD	Array [0..15] Of Bool	28.0		MDI_MOD
Return_SFC14	Int	30.0		Navratova hodnota SFC14
Return_SFC15	Int	32.0		Navratova hodnota SFC15
VEL	Real	34.0		
VEL2	DInt	38.0		
ACC	Real	42.0		
ACC2	DInt	46.0		
DEC	Real	50.0		
DEC2	DInt	54.0		
MDI	Int	58.0		
ERROR	Array [0..7] Of Bool	60.0		Kontrola zadanych hodnot, je-li bit pole = 1, hodnota byla zadana spatne

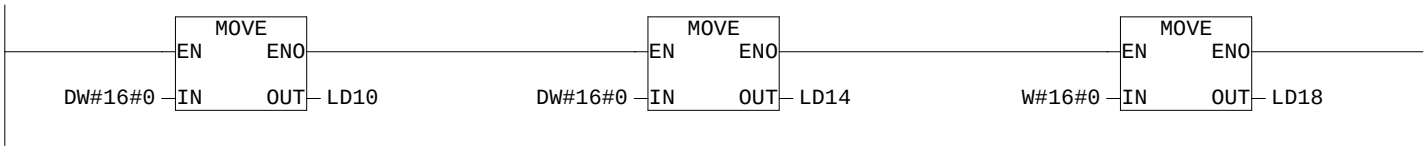
Block: FB9 FB pro ovladani menice, ktery vyuziva standardniho telegramu 9

```
Created:
date 12.12.2008 version 1.0 - Jakub Vacek
-----------------------------------------------------
Tento funkcni blok umi ovladat menic, u ktereho
byl v zalozce PROFIBUS navolen Standard Telegram 9.
-----------------------------------------------------
Vypis chybovych hlaseni do pole ERROR
ERROR[0] = 1, spatne zadana rychlost
ERROR[1] = 1, spatne zadane zrychleni
ERROR[2] = 1, spatne zadane zpomaleni
ERROR[3] = 1, spatne zadany mod
```

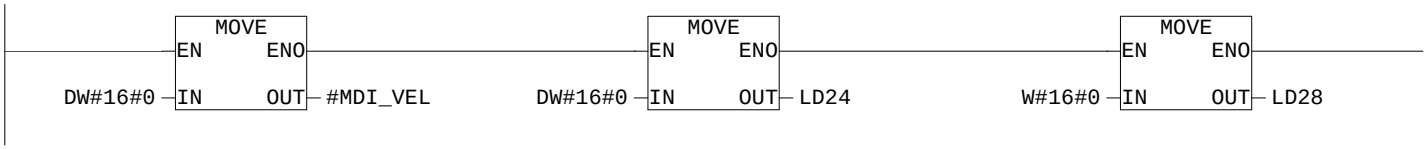
Network: 1 Vyprazdneni TEMP pro telegram jdouci z menice do CPU (5 wordu)



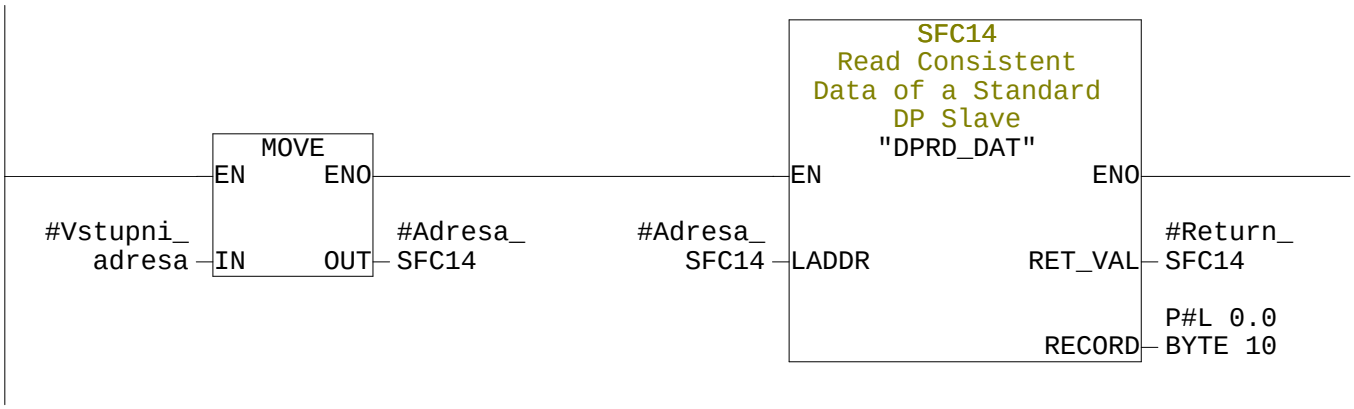
Network: 2 Vyprazdneni TEMP pro telegram jdouci z CPU do menice(10 wordu)



Network: 3 Vyprazdneni TEMP pro telegram jdouci z CPU do menice(10 wordu)

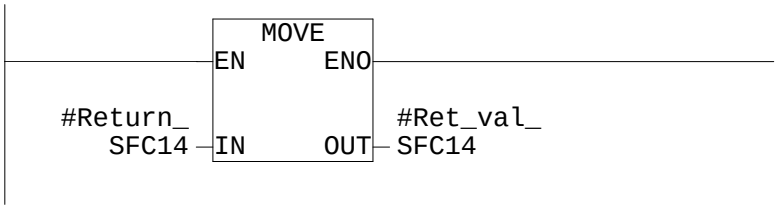


Network: 4 Nacteni dat z menice do CPU (TEMP)



Network: 5 Presun navratove hodnoty SFC14 na vystup (cteni dat z menice)

0 = v poradku, jinak chyba



Network: 6 Presun stavovych slov ZSW1 a ZSW2 na vystup

```
A     #ZSW1[8]
=     #Ready_to_power_up

A     #ZSW1[9]
=     #Ready

A     #ZSW1[10]
=     #Operation_enabled

A     #ZSW1[11]
=     #Fault_present

A     #ZSW1[12]
=     #No_coasting_active

A     #ZSW1[13]
=     #No_quick_stop_active

A     #ZSW1[14]
=     #Power_on_inhibit

A     #ZSW1[15]
=     #Alarm_present

A     #ZSW1[0]
=     #Error_in_tolerance

A     #ZSW1[1]
=     #Control_requested

A     #ZSW1[2]
=     #Target_position

A     #ZSW1[3]
=     #Reference_point_set

A     #ZSW1[4]
=     #Traversing_activated

A     #ZSW1[5]
=     #n_act_speed_threshold

A     #ZSW2[8]
=     #DDS_eff_bit_0

A     #ZSW2[9]
=     #DDS_eff_bit_1

A     #ZSW2[10]
=     #DDS_eff_bit_2

A     #ZSW2[11]
=     #DDS_eff_bit_3

A     #ZSW2[12]
=     #DDS_eff_bit_4

A     #ZSW2[15]
=     #Parking_axes_active
```

```
A     #ZSW2[0]
=     #Travel_to_fixed_stop

A     #ZSW2[3]
=     #Motor_changeover_set

A     #ZSW2[4]
=     #Slave_bit_0

A     #ZSW2[5]
=     #Slave_bit_1

A     #ZSW2[6]
=     #Slave_bit_2

A     #ZSW2[7]
=     #Slave_bit_3
```

Network: 7 Presun stavoveho slova AKTSATZ na vystup

```
A     #AKTSATZ[8]
=     #Traversing_bit_0

A     #AKTSATZ[9]
=     #Traversing_bit_1

A     #AKTSATZ[10]
=     #Traversing_bit_2

A     #AKTSATZ[11]
=     #Traversing_bit_3

A     #AKTSATZ[12]
=     #Traversing_bit_4

A     #AKTSATZ[13]
=     #Traversing_bit_5

A     #AKTSATZ[7]
=     #MDI_active
```

Network: 8 Presun ridicich slov STW1 a STW2 na vstup

```
A     #ONOFF1
=     #STW1[8]

A     #OFF2
=     #STW1[9]

A     #OFF3
=     #STW1[10]

A     #Enable_operation
=     #STW1[11]

A     #EPOS_reject
```

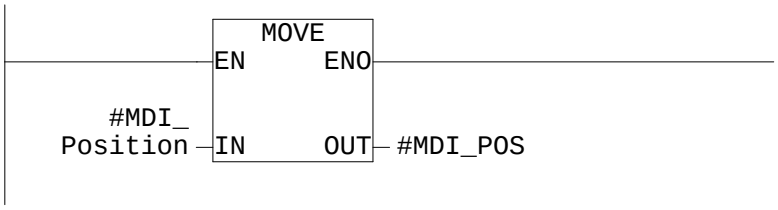
```
=     #STW1[12]

A     #EPOS_stop
=     #STW1[13]

A     #EPOS_activate
=     #STW1[14]

A     #Acknowledge_faults
=     #STW1[15]

A     #EPOS_jog_1
=     #STW1[0]

A     #EPOS_jog_2
=     #STW1[1]

A     #Master_ctrl_by_PLC
=     #STW1[2]

A     #EPOS_start
=     #STW1[3]

A     #DDS_bit_0
=     #STW2[8]

A     #DDS_bit_1
=     #STW2[9]

A     #DDS_bit_2
=     #STW2[10]

A     #DDS_bit_3
=     #STW2[11]

A     #DDS_bit_4
=     #STW2[12]

A     #Parking_axis_selection
=     #STW2[15]

A     #Activates_travel
=     #STW2[0]

A     #Motor_changeover
=     #STW2[3]

A     #Master_bit_0
=     #STW2[4]

A     #Master_bit_1
=     #STW2[5]

A     #Master_bit_2
=     #STW2[6]

A     #Master_bit_3
=     #STW2[7]
```

Network: 9 Presun ridiciho slova SATZANW na vstup

```
      A     #EPOS_bit_0
      =     #SATZANW[8]

      A     #EPOS_bit_1
      =     #SATZANW[9]

      A     #EPOS_bit_2
      =     #SATZANW[10]

      A     #EPOS_bit_3
      =     #SATZANW[11]

      A     #EPOS_bit_4
      =     #SATZANW[12]

      A     #EPOS_bit_5
      =     #SATZANW[13]

      A     #EPOS_direct_setpoint_in
      =     #SATZANW[7]
```

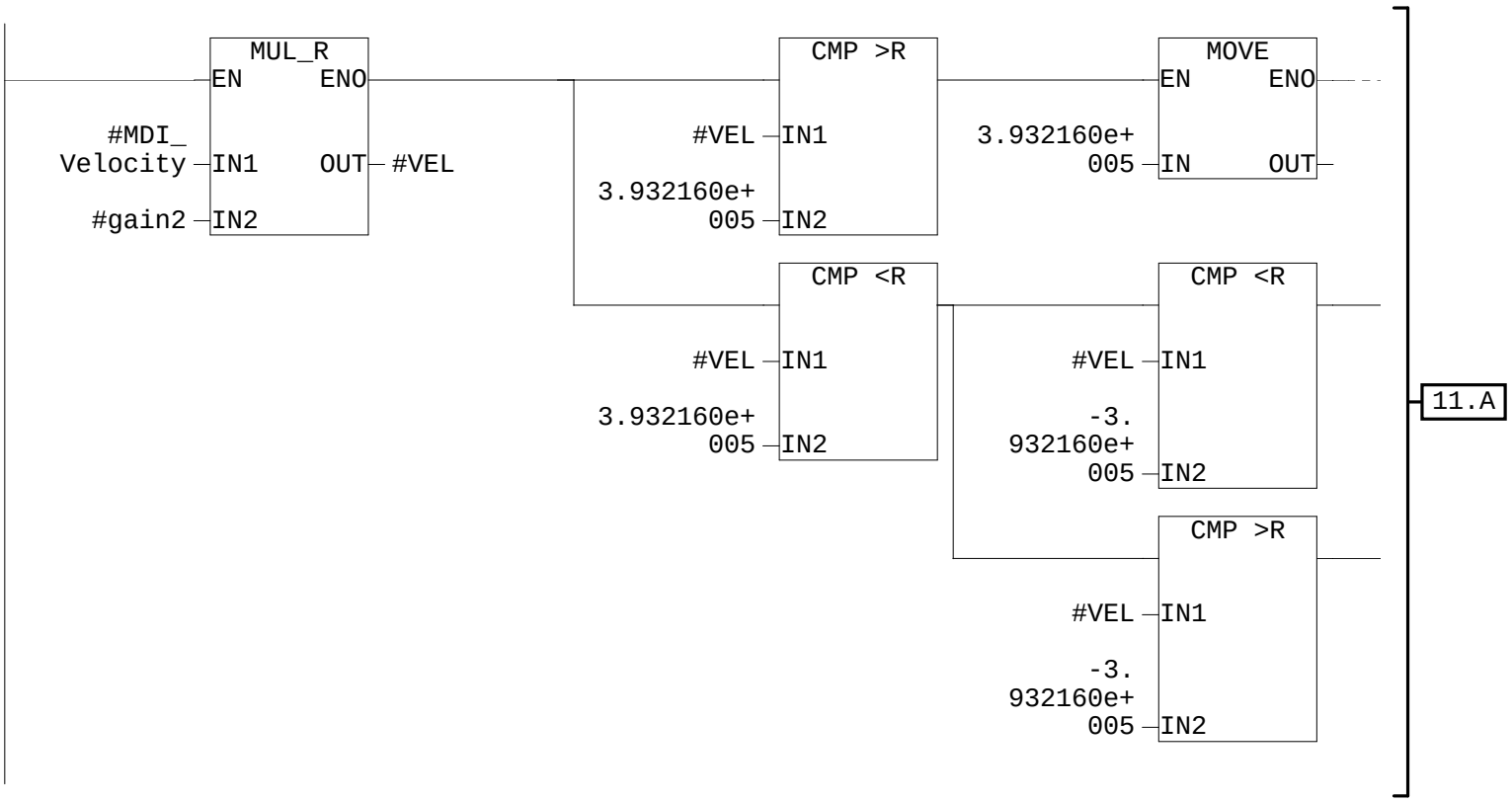
Network: 10 MDI Position

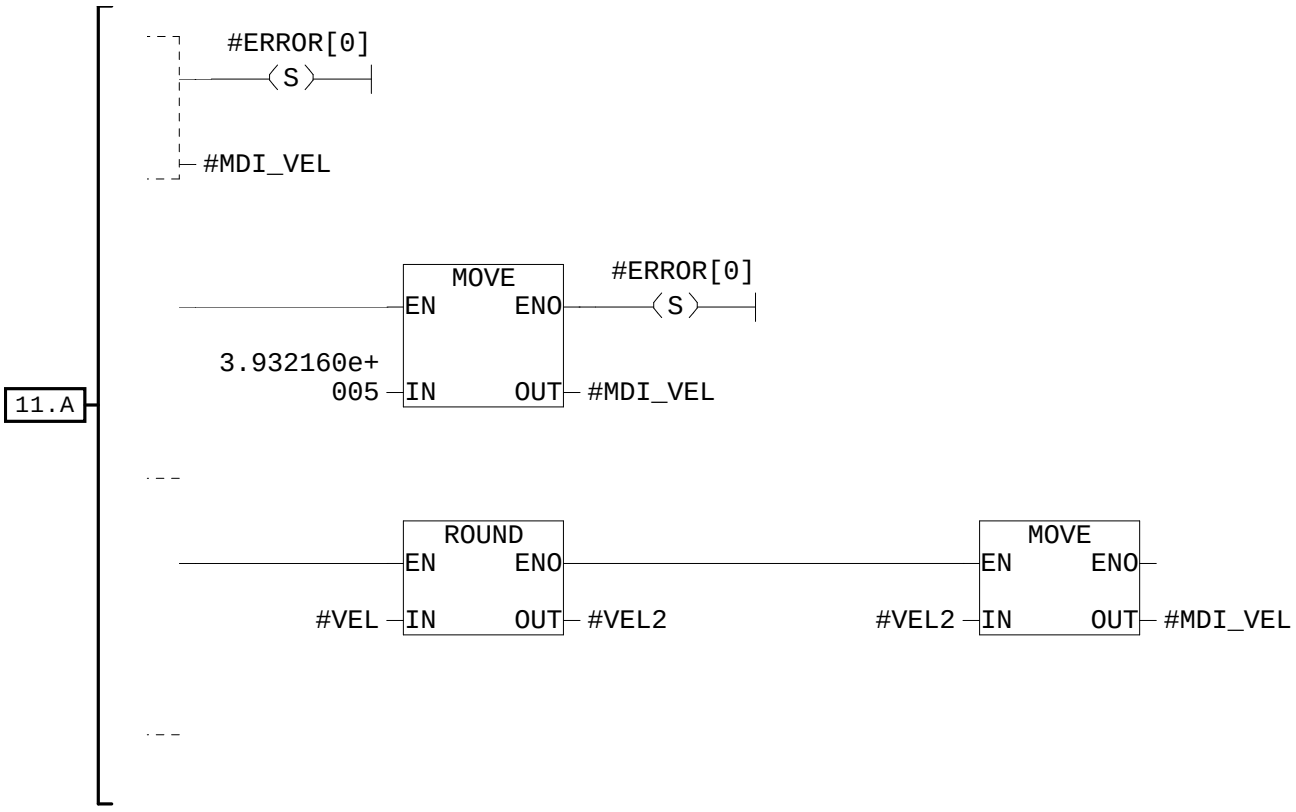
Nastaveni pozice
Standardni normalizace v menicich je 1 hex = 1 LU



Network: 11 MDI Velocity

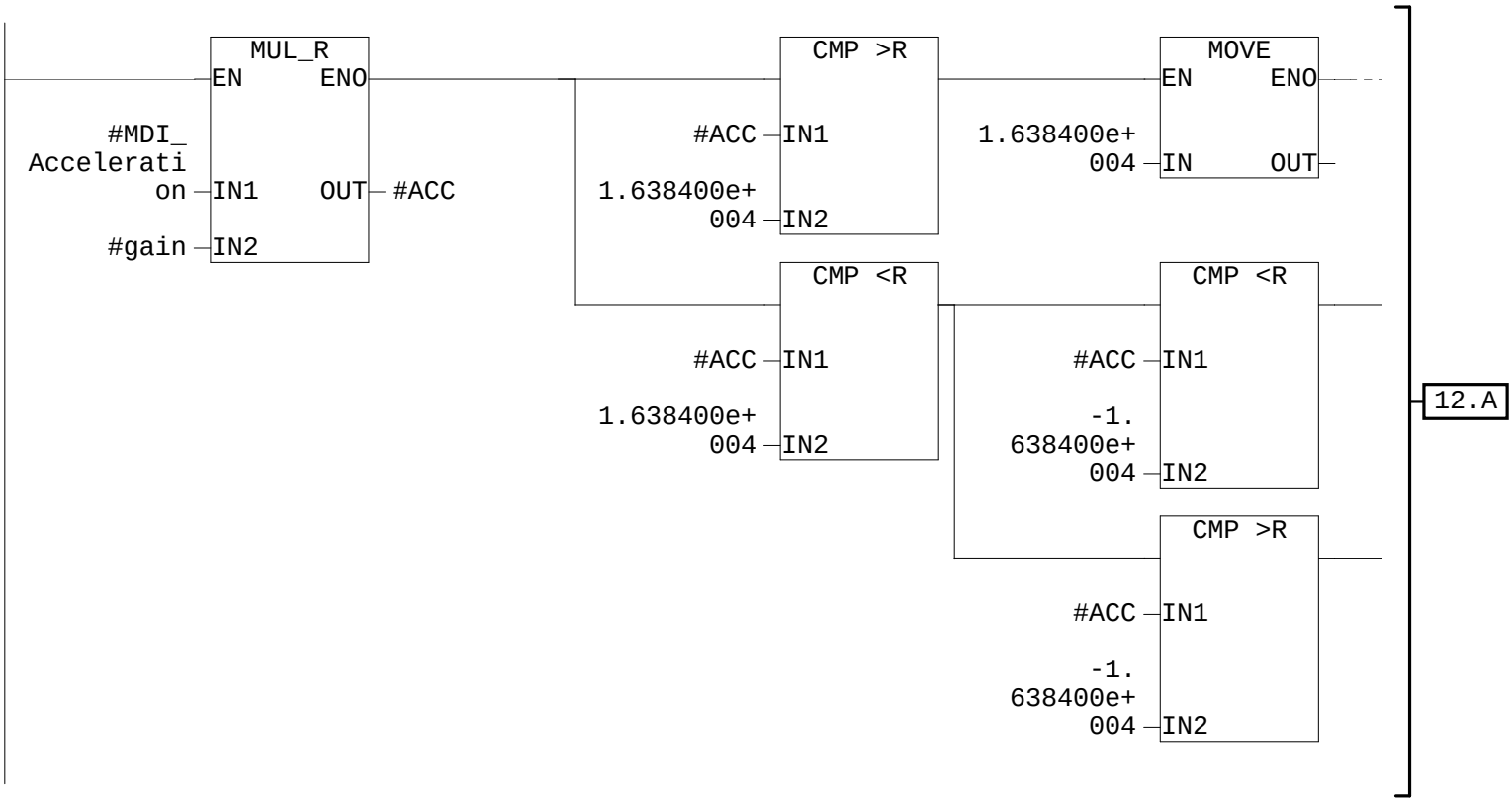
Nastaveni rychlosti v procentech (0-100)
Standardni normalizace v menicich je 1 hex = 1000 LU/min, maximální rychlost
udává parametr p2000 6000 rev/min
Pri nastaveni 10000 LU per load revolution je 60000 hex = 100%
Provadi se omezeni na hodnoty double integer

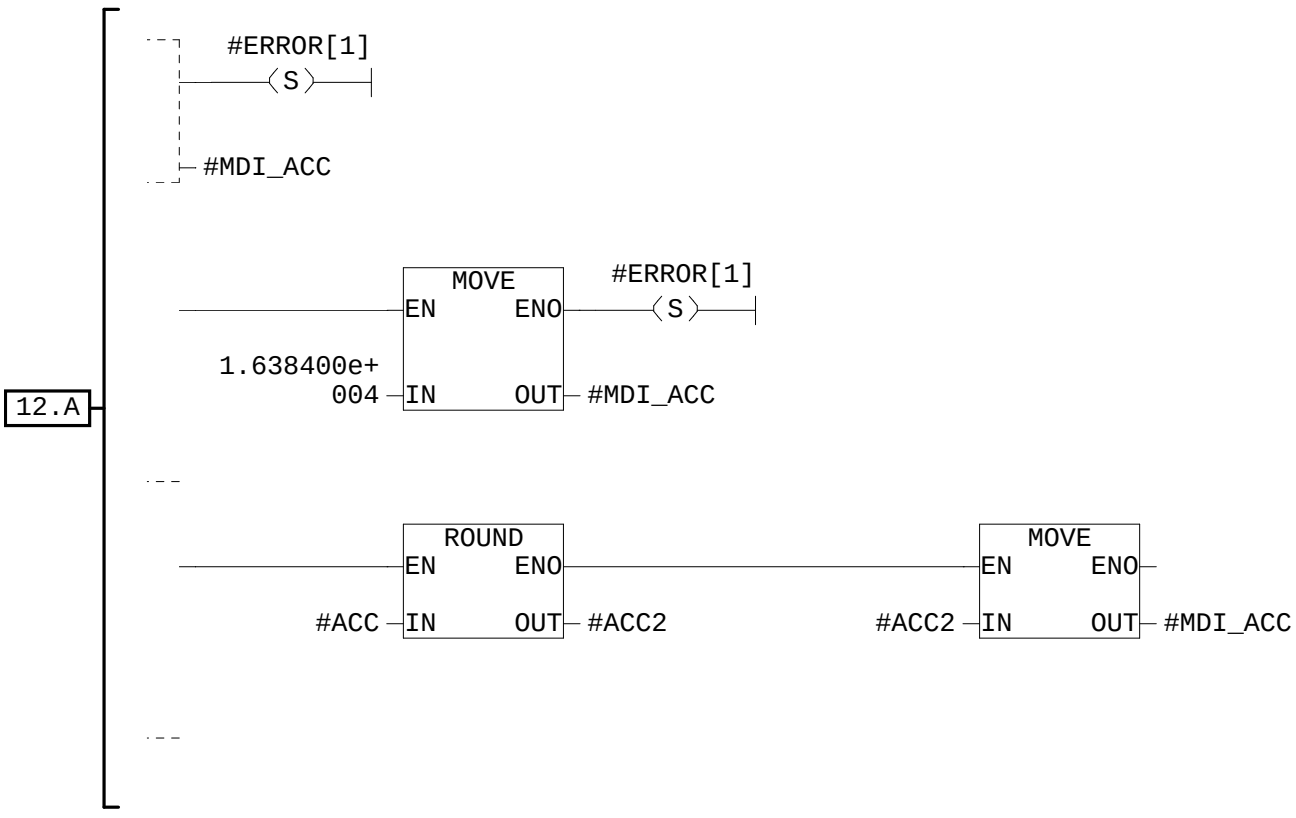



#ERROR[0]
S
#MDI_VEL
MOVE
EN
ENO
#ERROR[0]
S
3.932160e+
005
IN
OUT
#MDI_VEL
11.A
ROUND
EN
ENO
#VEL
IN
OUT
#VEL2
MOVE
EN
ENO
#VEL2
IN
OUT
#MDI_VEL

Network: 12 MDI Acceleration

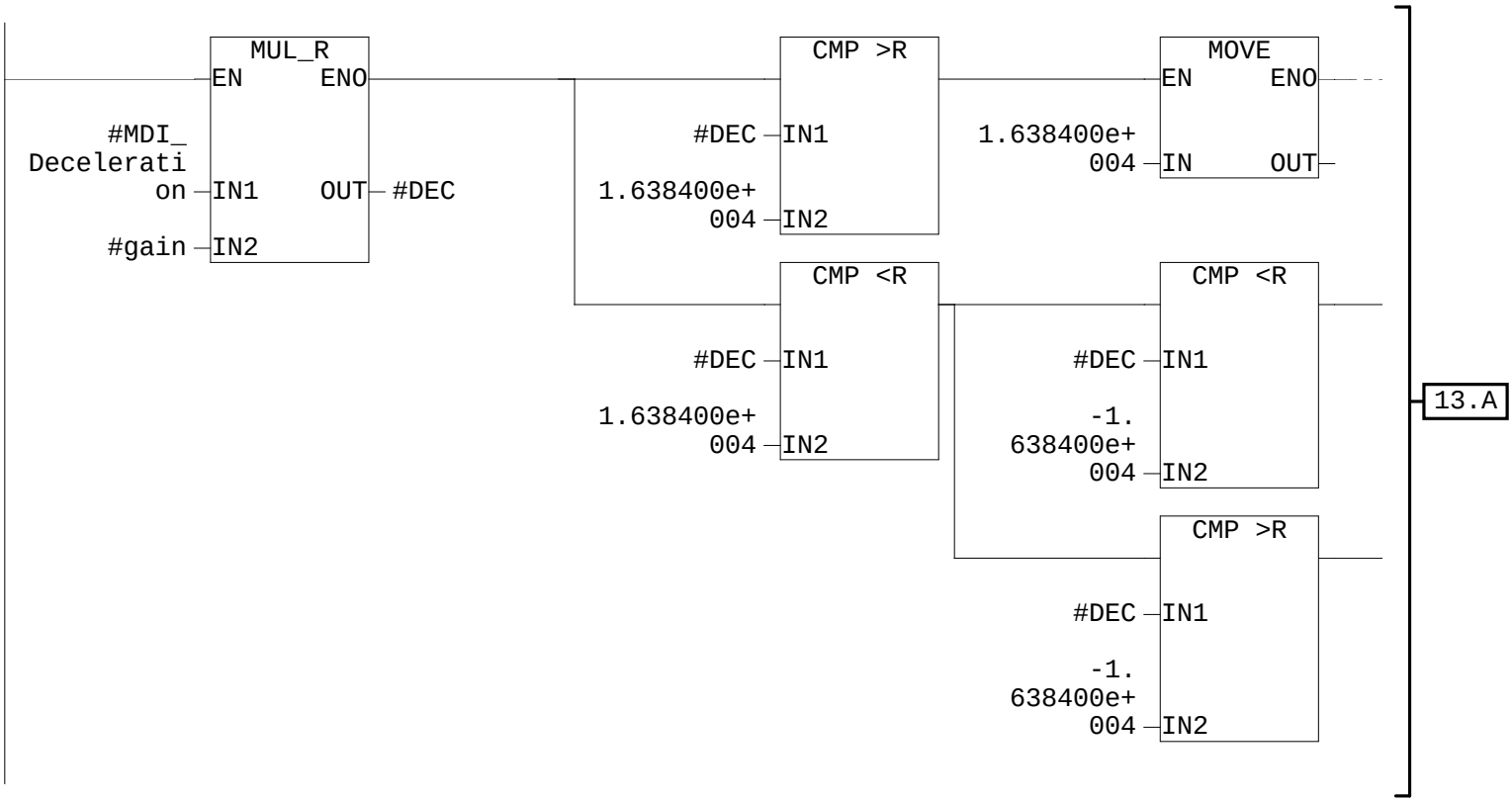
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

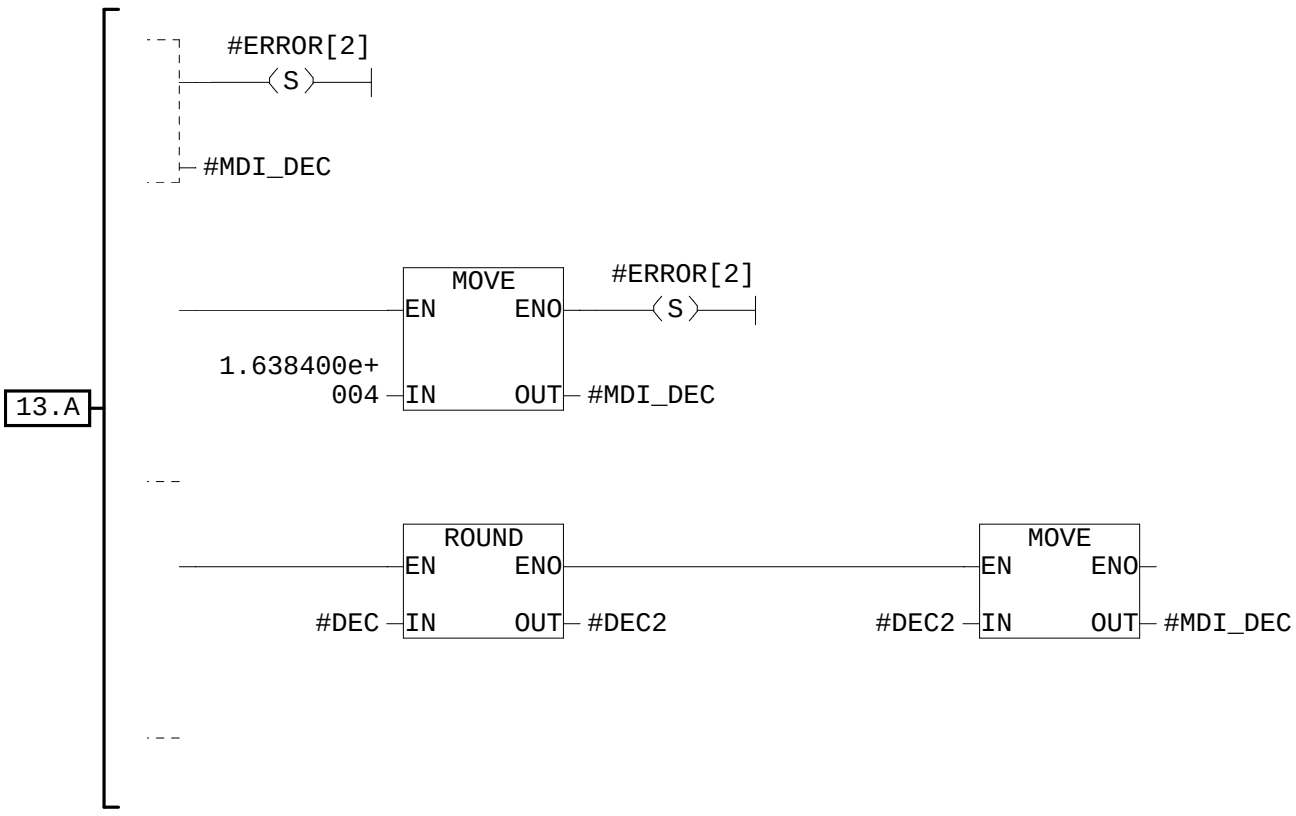




Network: 13 MDI Deceleration

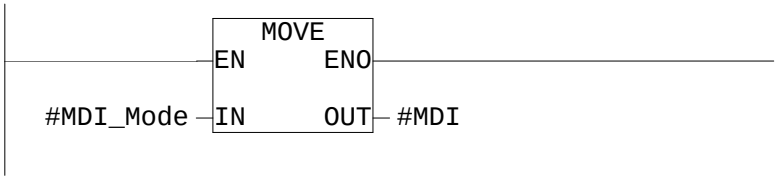
Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer





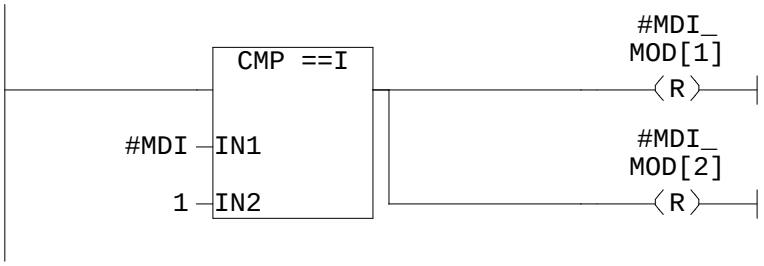
Network: 14 MDI Mode

Presun z IN do TEMP



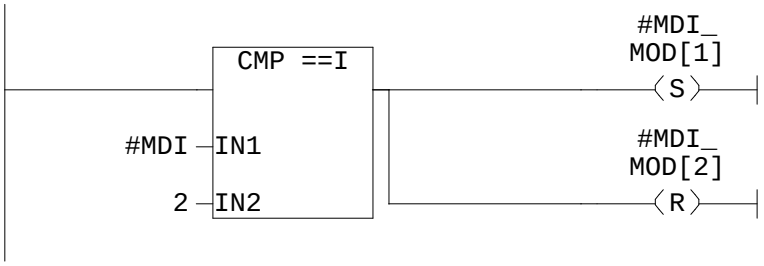
Network: 15 MDI Mode

Nastaveni modu
Absolute = 1



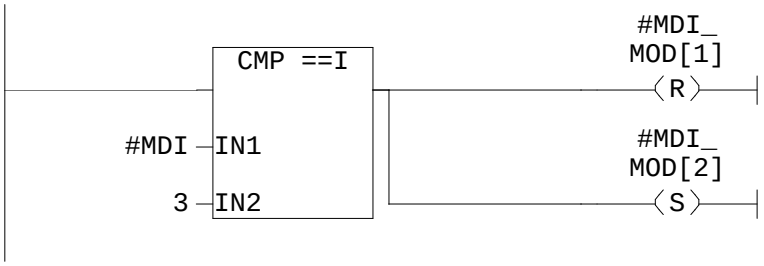
Network: 16 MDI Mode

Nastaveni modu
Relative = 2



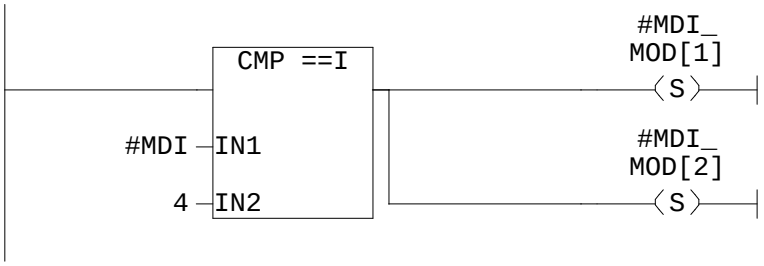
Network: 17 MDI Mode

Nastaveni modu
Absolute positive = 3



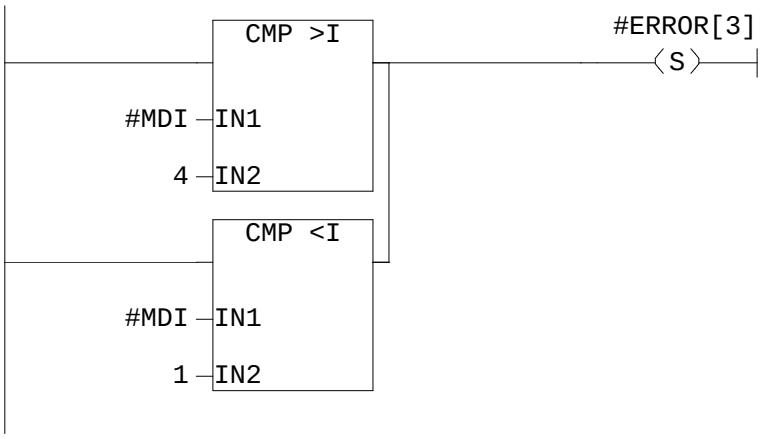
Network: 18 MDI Mode

Nastaveni modu
Absolute negative = 4



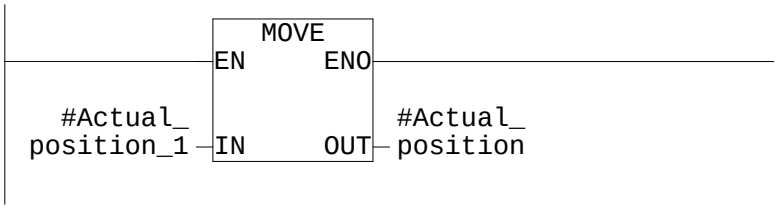
Network: 19 MDI Mode

Spatne nastaveny MDI Mode

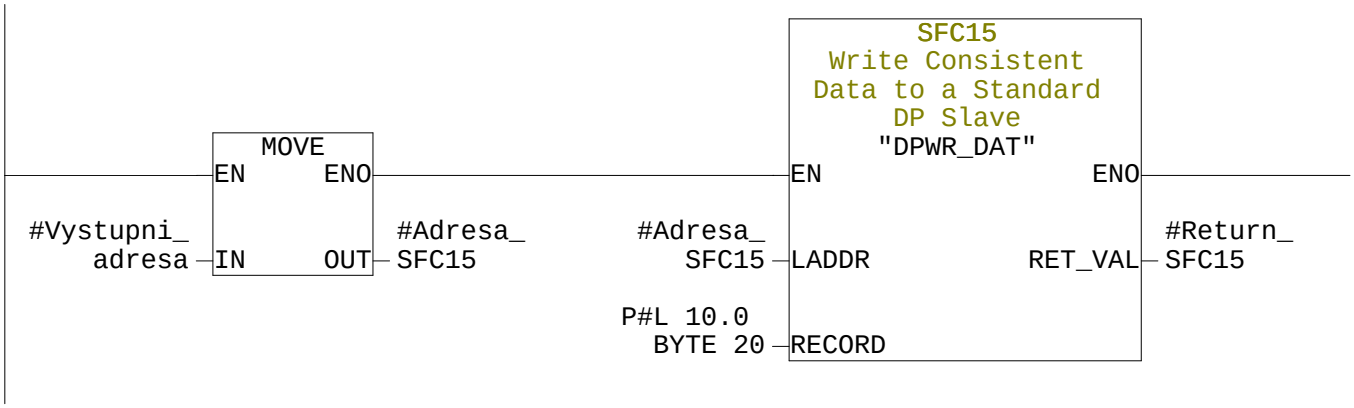


Network: 20 Actual position

Vypise aktualni pozici
Standardni normalizace v menicich je 1 hex = 1 LU

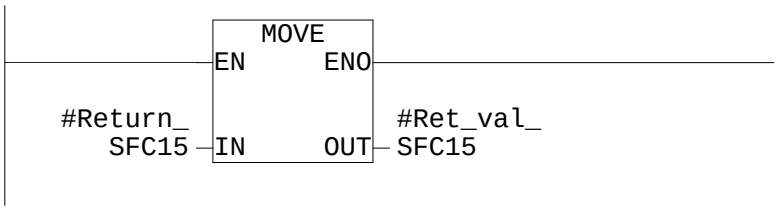


Network: 21 Nahrani dat z CPU (TEMP) do menice



Network: 22 Presun navratove hodnoty SFC15 (zapis dat do menice) na vystup

0 = v poradku, jinak chyba



A.2 Funkce – Move Absolute

FC1 - <offline>

"FC_MoveAbsolute"

Name: **Family:**
Author: Vacek **Version:** 0.1
Block version: 2
Time stamp Code: 05/25/2009 09:01:58 PM
Interface: 05/20/2009 10:39:40 AM
Lengths (block/logic/data): 03324 02952 00040

Name	Data Type	Address	Comment
IN		0.0	
Axis_DB	Block_DB	0.0	Otevre datovy blok pro pozadovany standardni telegram
Execute	Bool	2.0	Start
Position	DInt	4.0	Pozice
Velocity	Real	8.0	Rychlost
Acceleraion	Real	12.0	Zrychleni
Deceleration	Real	16.0	Zpomaleni
OUT		0.0	
Done	Bool	20.0	Dosazena cilova pozice
Active	Bool	20.1	Funkce je aktivni
Error	Bool	20.2	Chyba
ErrorID	Int	22.0	Identifikace chyb
IN_OUT		0.0	
TEMP		0.0	
HELP	Bool	0.0	
POS	DInt	2.0	
VEL	Real	6.0	
VEL2	DInt	10.0	
ACC	Real	14.0	
ACC2	DInt	18.0	
DEC	Real	22.0	
DEC2	DInt	26.0	
ID9	Bool	30.0	
ID110	Bool	30.1	
ID111	Bool	30.2	
Cislo	Word	32.0	Cislo DB
Delka	Word	34.0	Delka DB v bytech
Return	Int	36.0	Navratova hodnota (0=OK)
WrPr	Bool	38.0	DB chranen proti prepsani (1=ANO)
RETURN		0.0	
RET_VAL		0.0	

Block: FC1

```
Created:
date 17.04.2009 version 1.0 - Jakub Vacek
-----------------------------------------------------
Funkce MoveAbsolute startuje nastaveni polohy osy podle absolutni pozice
-----------------------------------------------------
Telegram: 9,110,111
-----------------------------------------------------
Identifikace chyby:
ErrorID = 1, spatne zadana rychlost
ErrorID = 2, spatne zadane zrychleni
ErrorID = 3, spatne zadane zpomaleni
ErrorID = 4, nebyl nalezen kompatibilni telegram
```

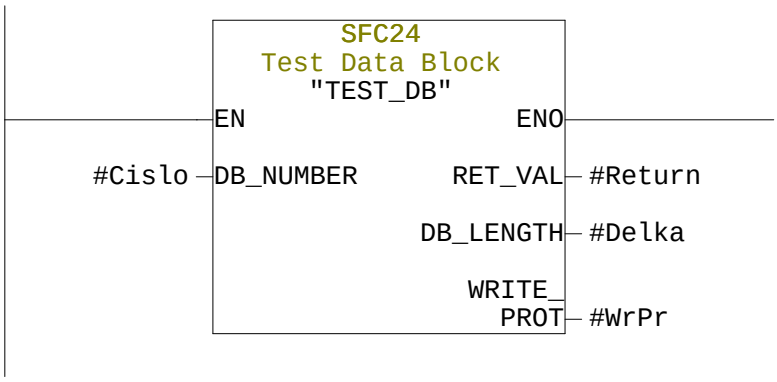
Network: 1 Otevri DB

Funkce otevre datovy blok pro pozadovany standardni telegram

Network: 2

```
      L     DBNO
      T     #Cislo
```

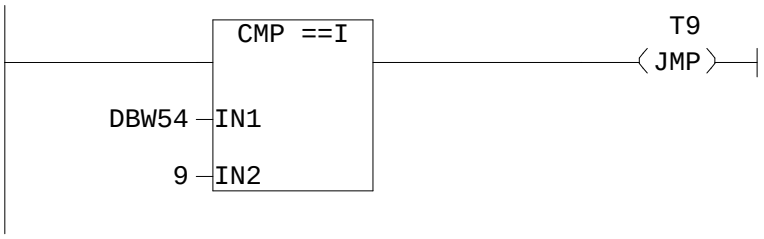
Network: 3



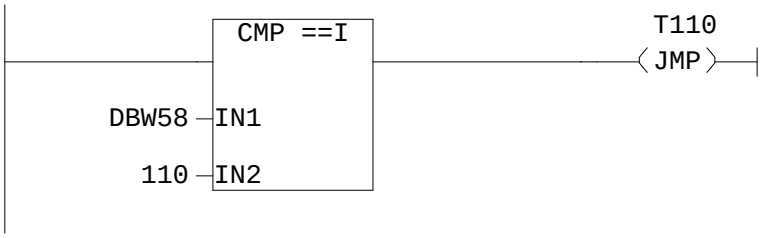
Network: 4

Provnavani wordu !!

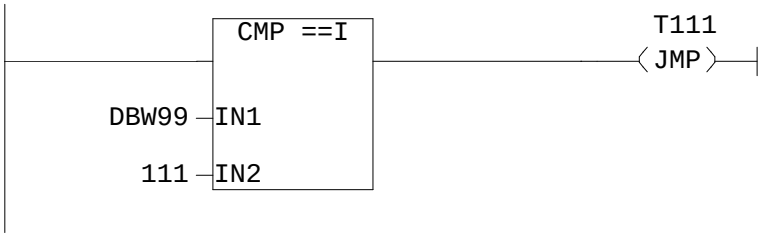
Network: 5 Je pouzit telegram 9?



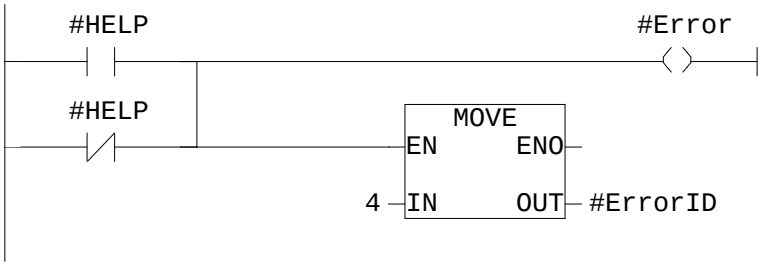
Network: 6 Je pouzit telegram 110?



Network: 7 Je pouzit telegram 111?

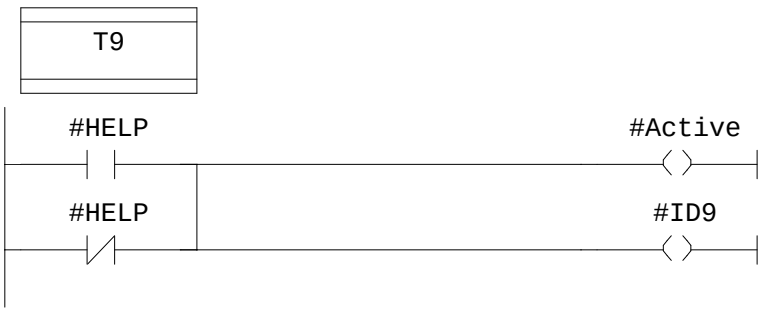


Network: 8 Nebyl pouzit vhodny telegram



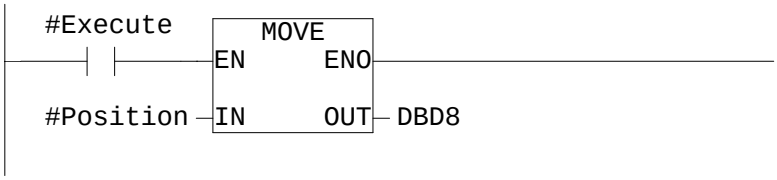
Network: 9 Konec

Network: 10 Telegram 9



Network: 11 Position

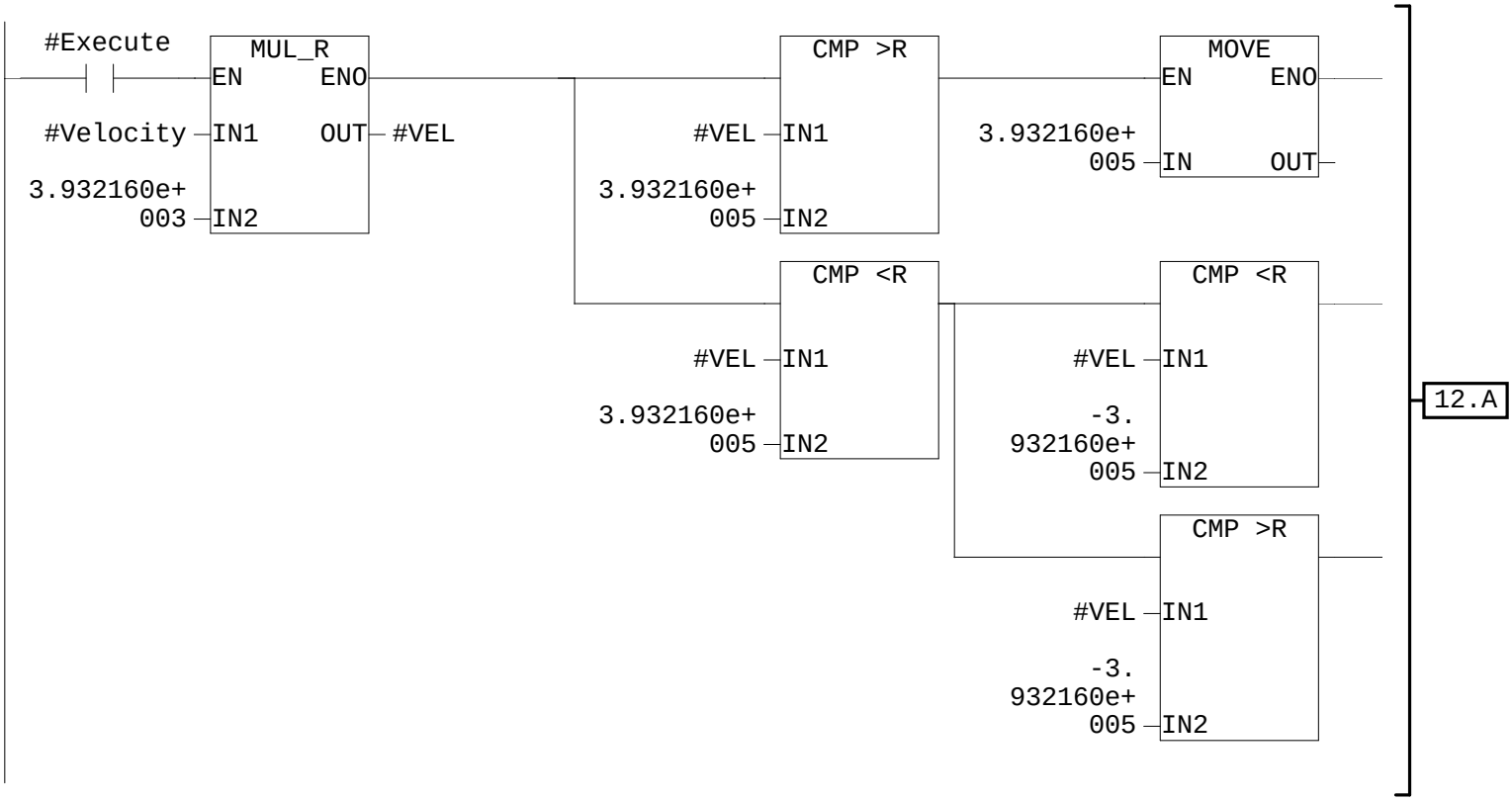
Nastaveni pozice
Standardni normalizace v menicich je 1 hex = 1 LU

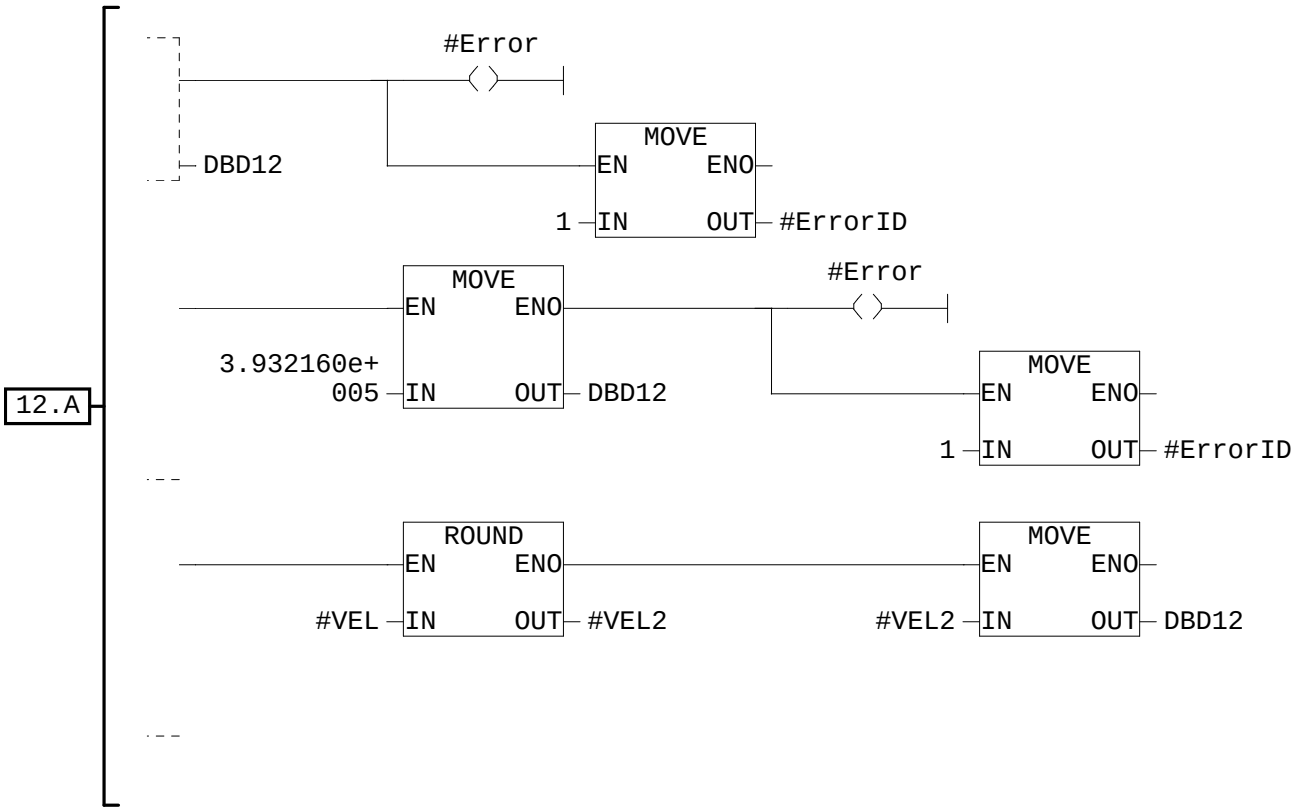


Network: 12 Velocity

Nastaveni rychlosti v procentech (0-100)
Standardni normalizace v menicich je 1 hex = 1000 LU/min, maximalni rychlost

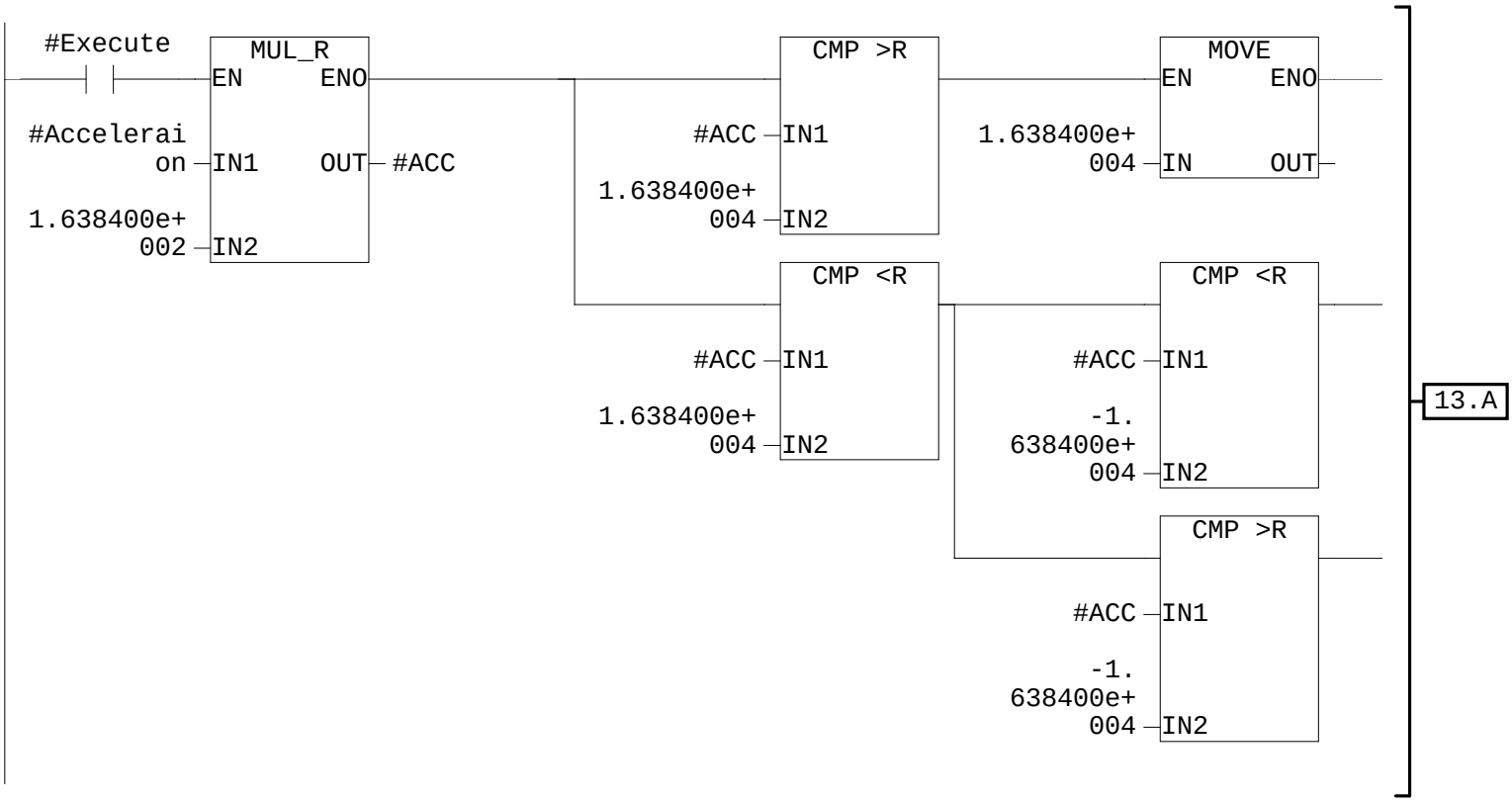
udava parametr p2000 (6000 rev/min)
Pri nastaveni 10000 LU per load revolution je 60000 hex = 100%
Provadi se omezeni na hodnoty double integer

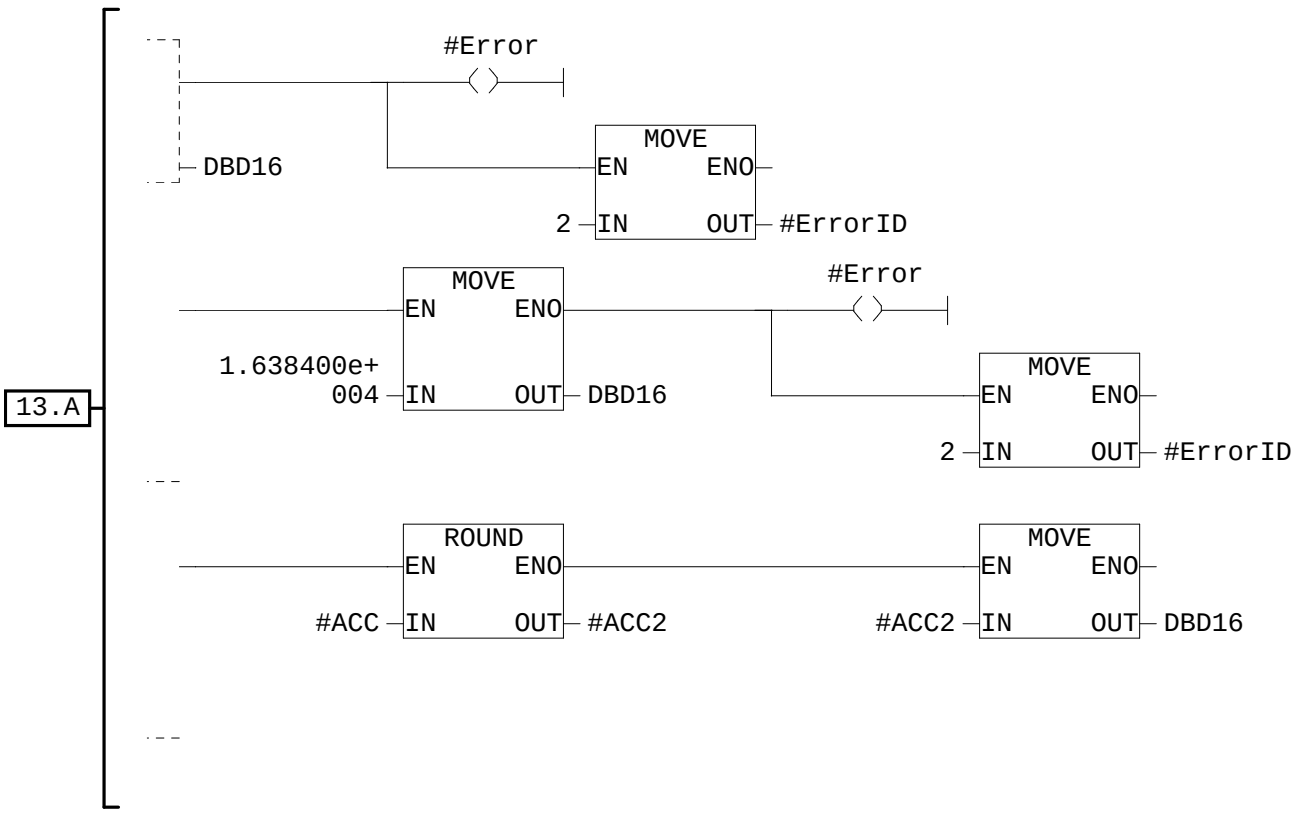




Network: 13 Acceleration

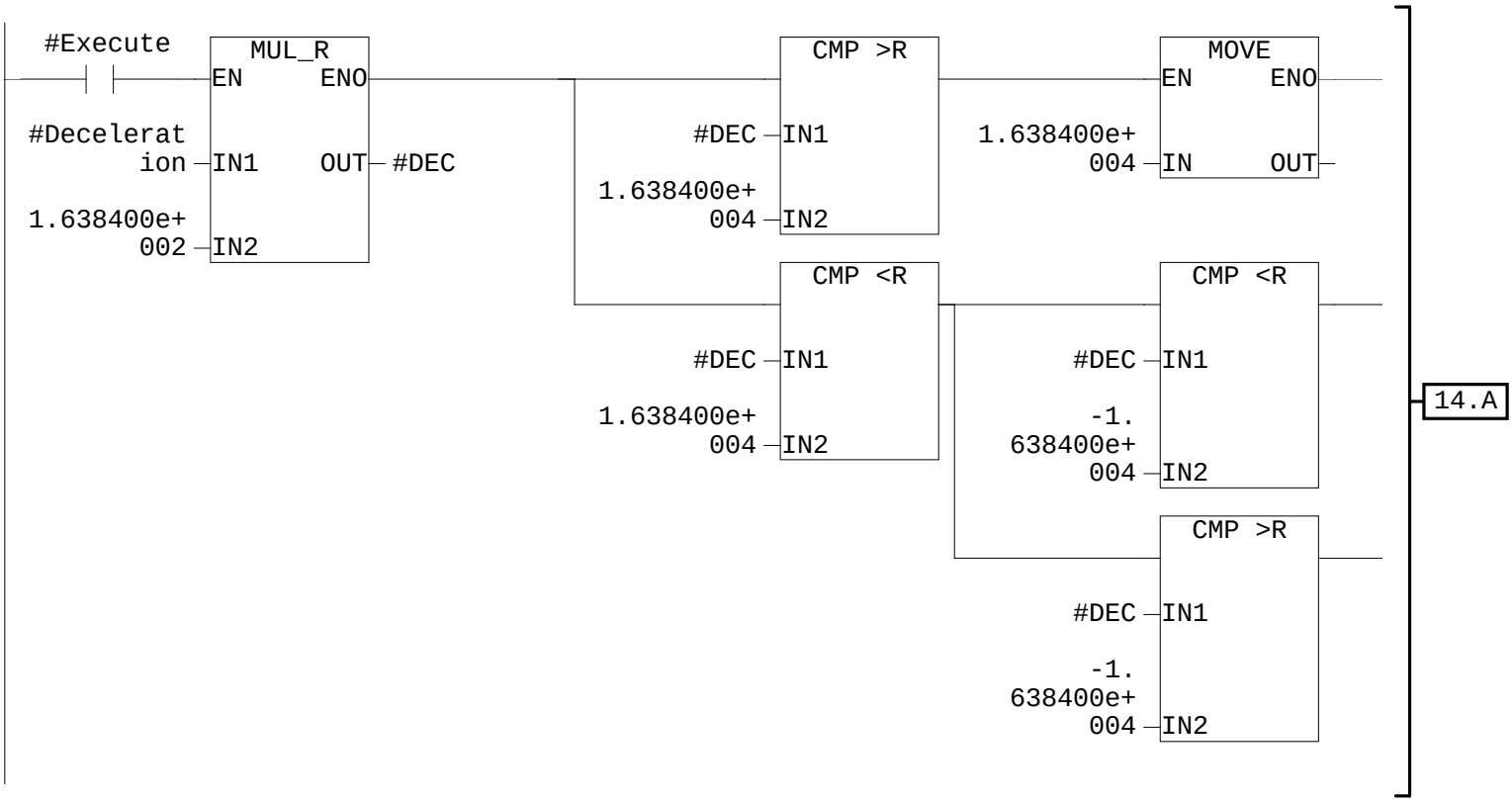
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

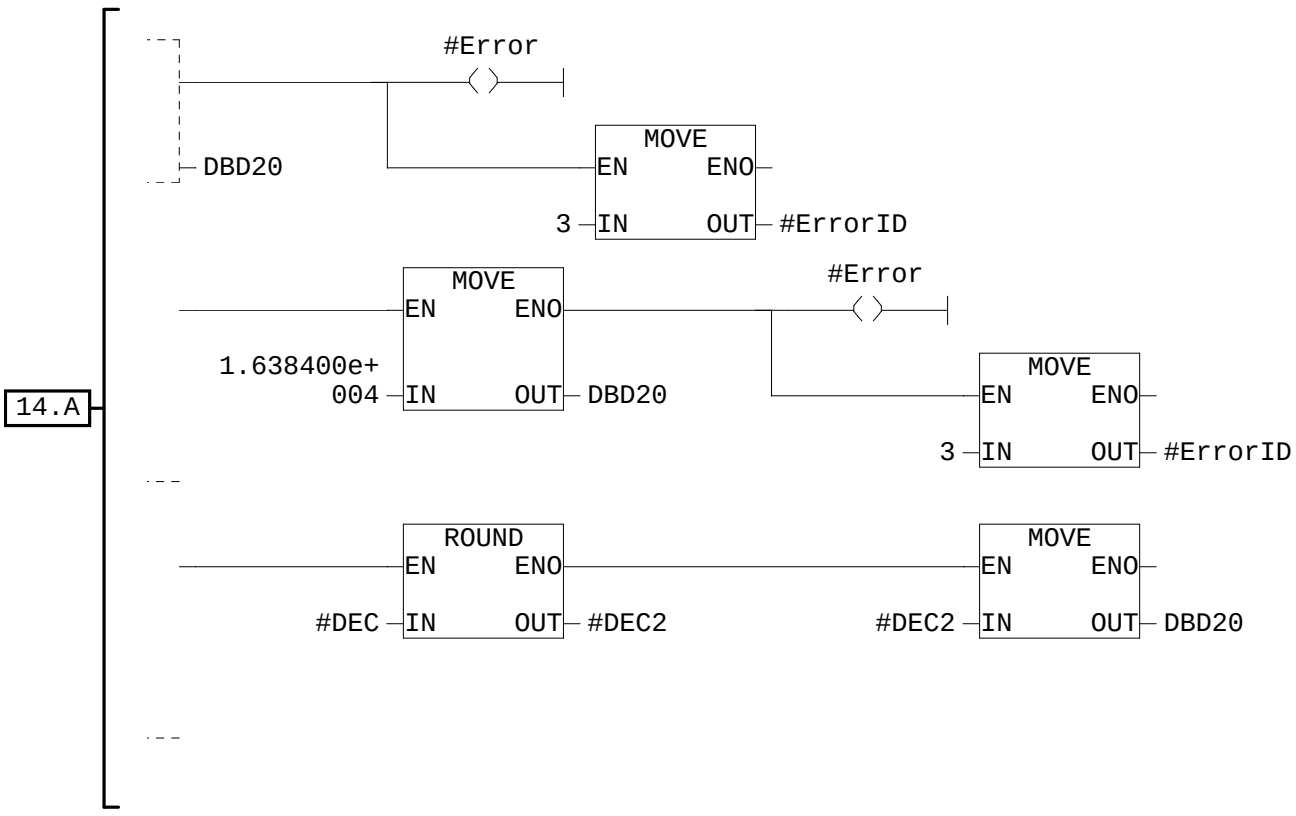




Network: 14 Deceleration

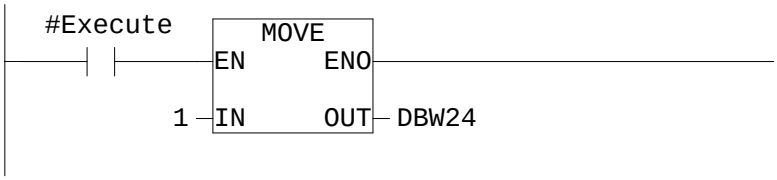
Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer





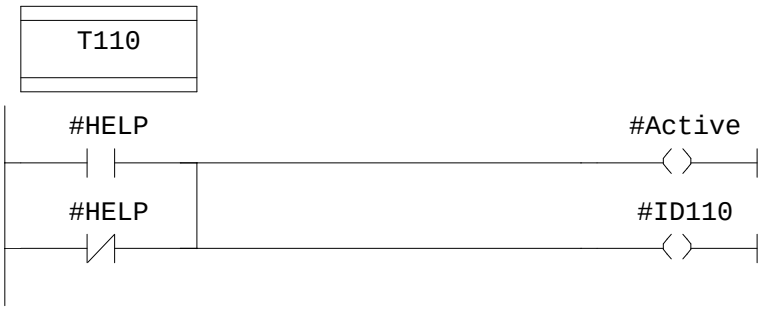
Network: 15 Mode

Nastaveni modu - nastaven na Absolute
Absolute = 1, Relative = 2, Absolute positive = 3, Absolute negative = 4



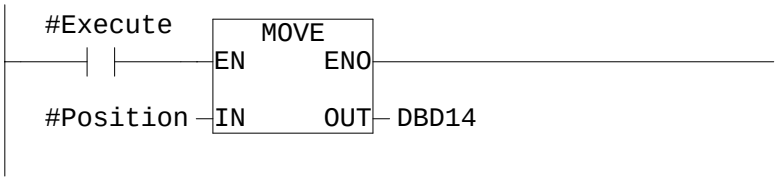
Network: 16 Konec

Network: 17 Telegram 110



Network: 18 Position

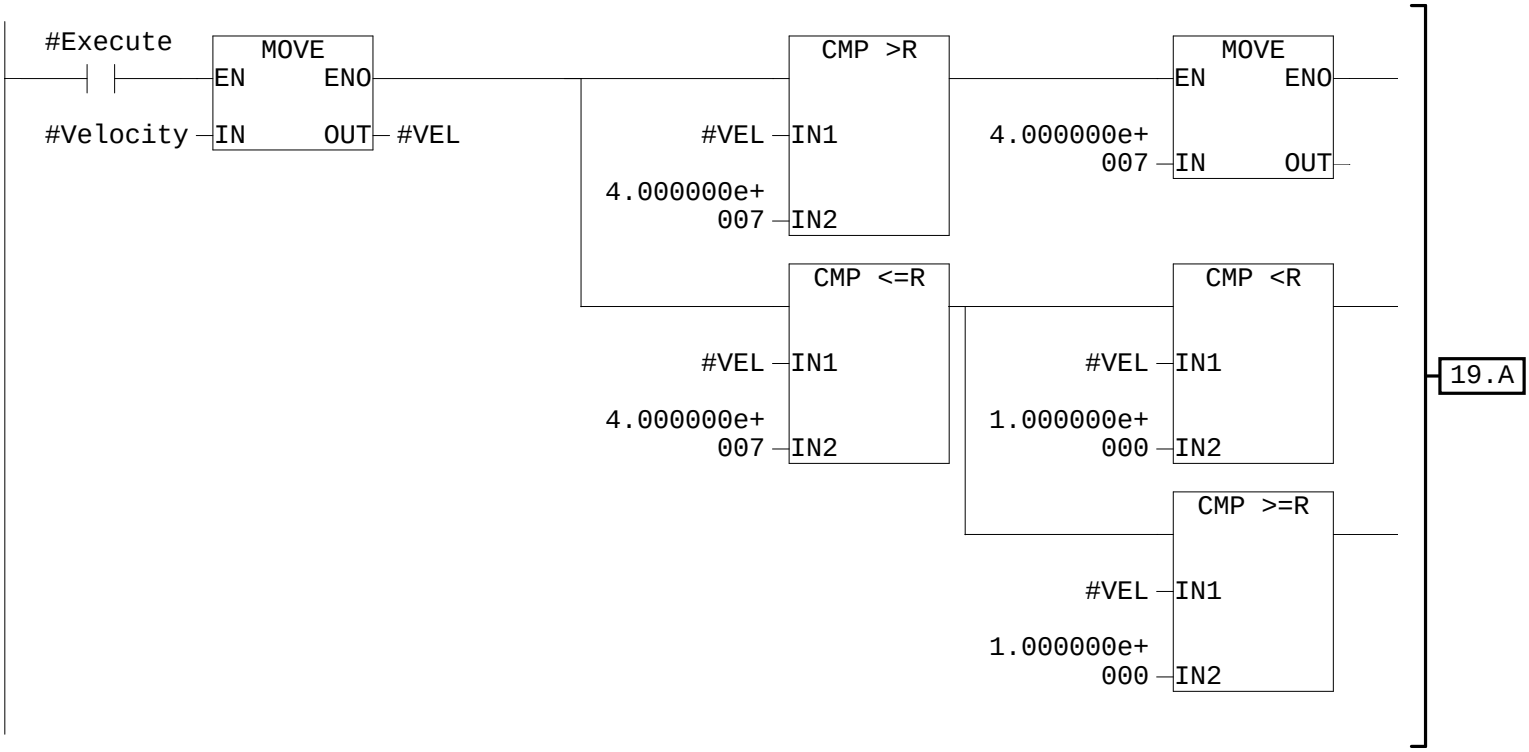
Nastaveni pozice
Standardni normalizace v menicich je 1 hex = 1 LU

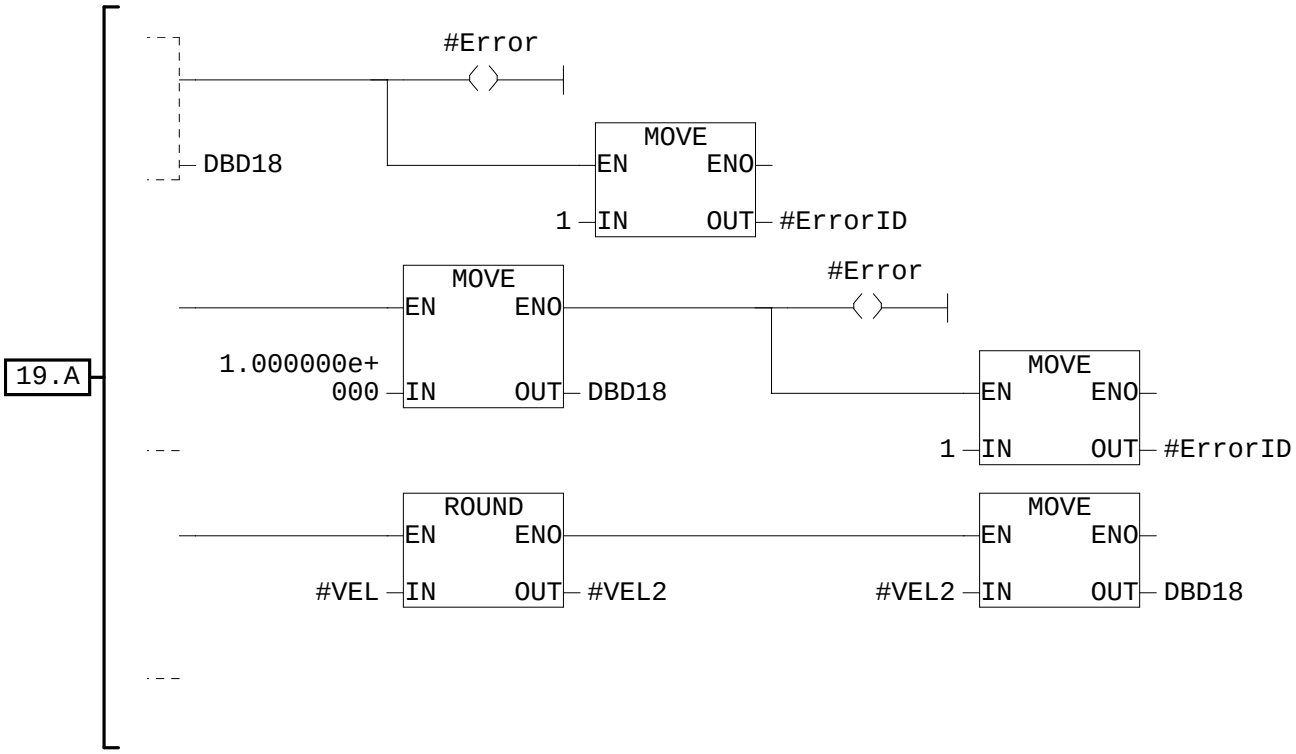


Network: 19 Velocity

Nastaveni rychlosti
Regulovatelna v rozmezi 1 - 40000000
Standardni normalizace v menicich je 1 hex = 1000 LU/min, maximalni rychlost

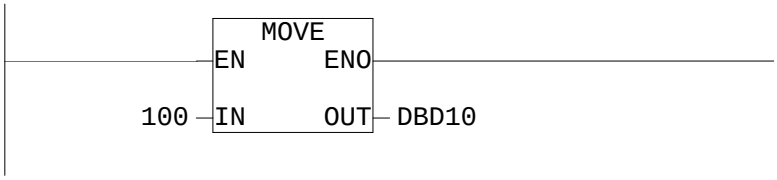
udava parametr p2571
Provadi se omezeni na hodnoty double integer





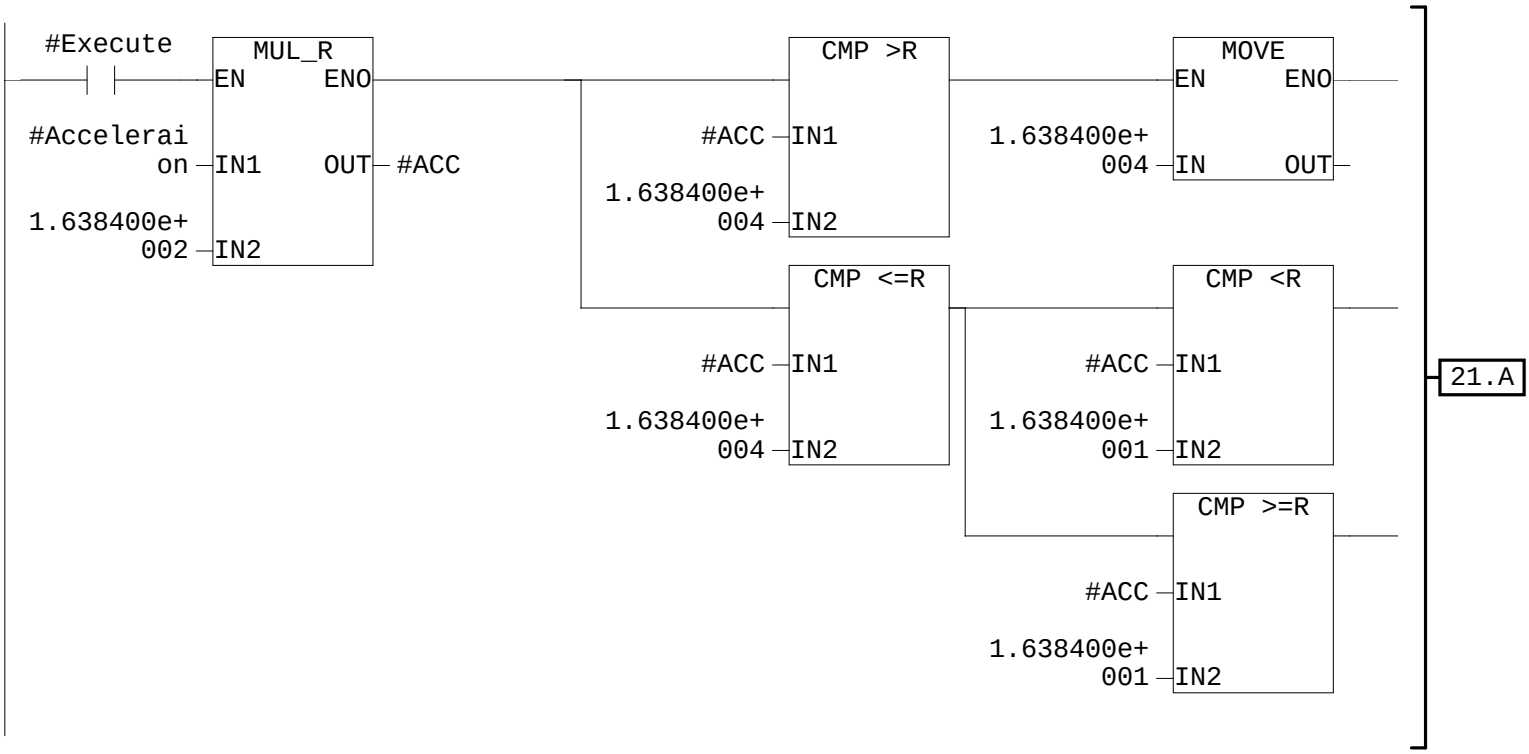
Network: 20 Override

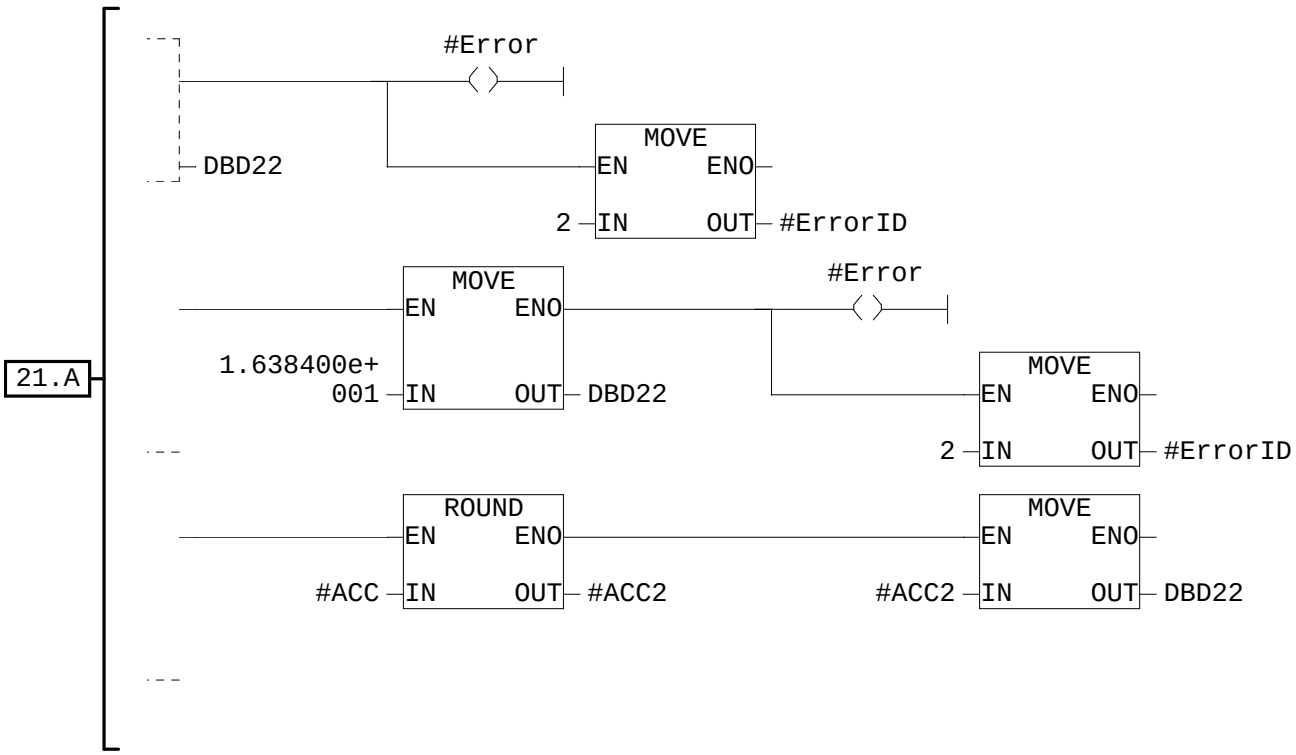
Nastaveni Override pro rychlost - 100%



Network: 21 Acceleration

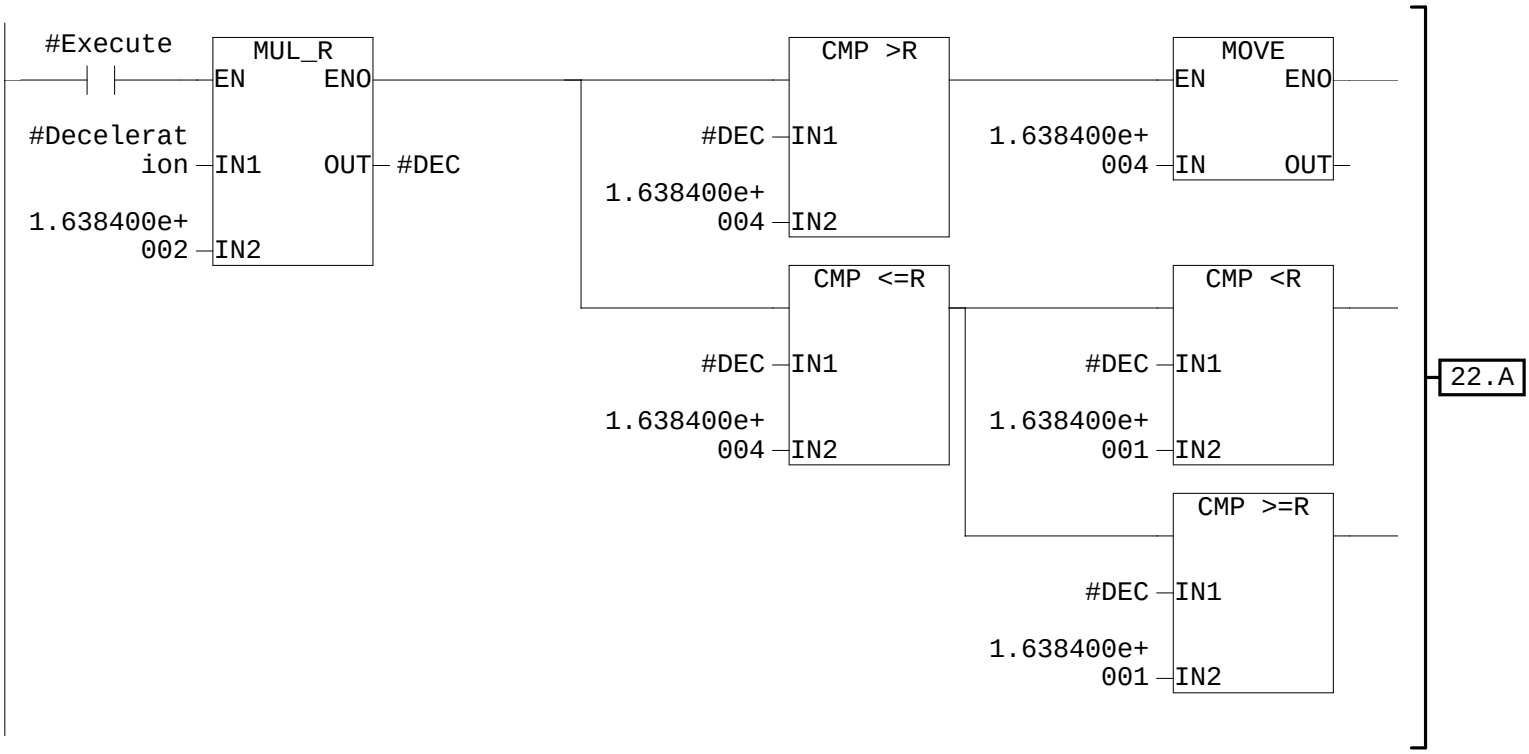
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

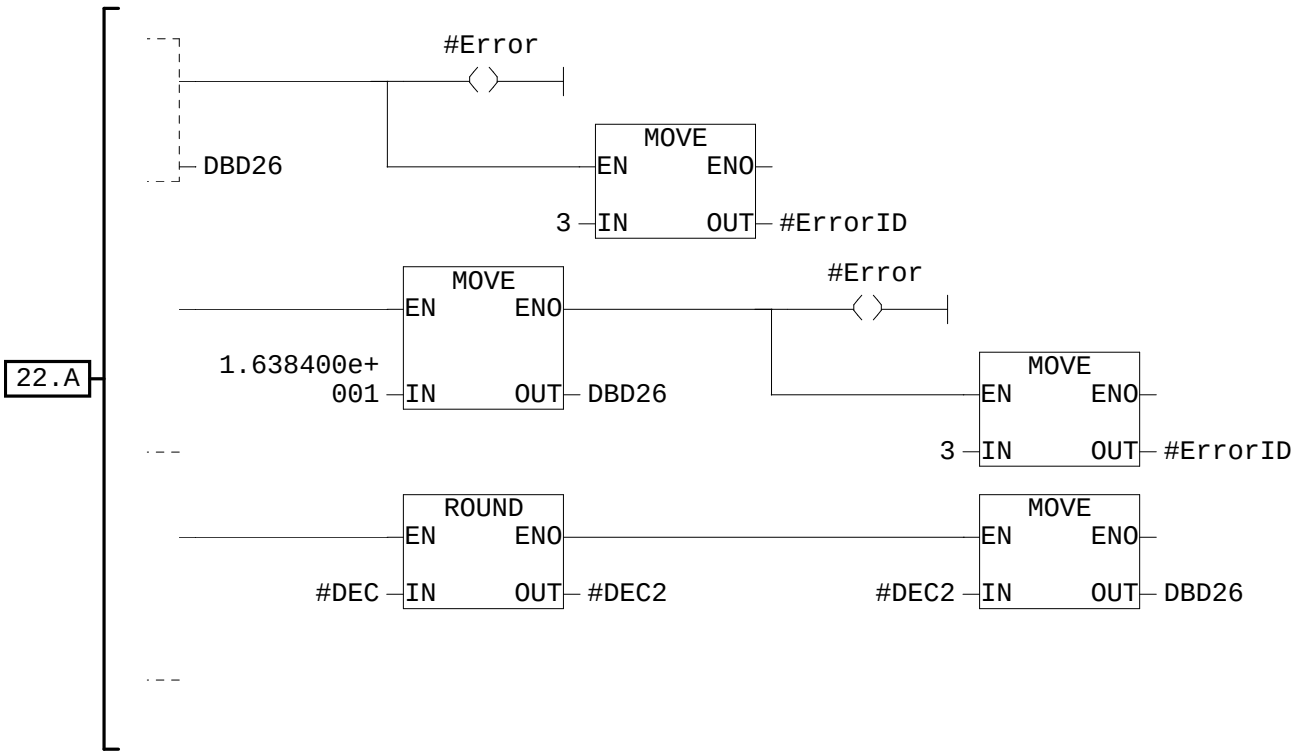




Network: 22 Deceleration

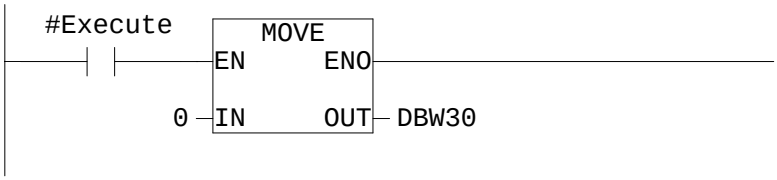
Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer





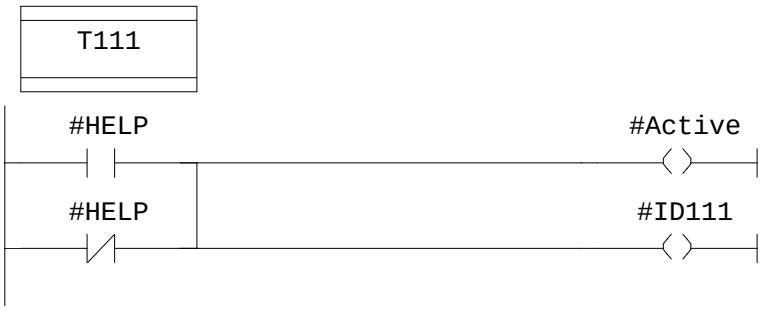
Network: 23 Mode

Nastaveni modu - nastaven na Absolute
Absolute = 0, Relative = 1, Absolute positive = 2, Absolute negative = 3



Network: 24 Konec

Network: 25 Telegram 111

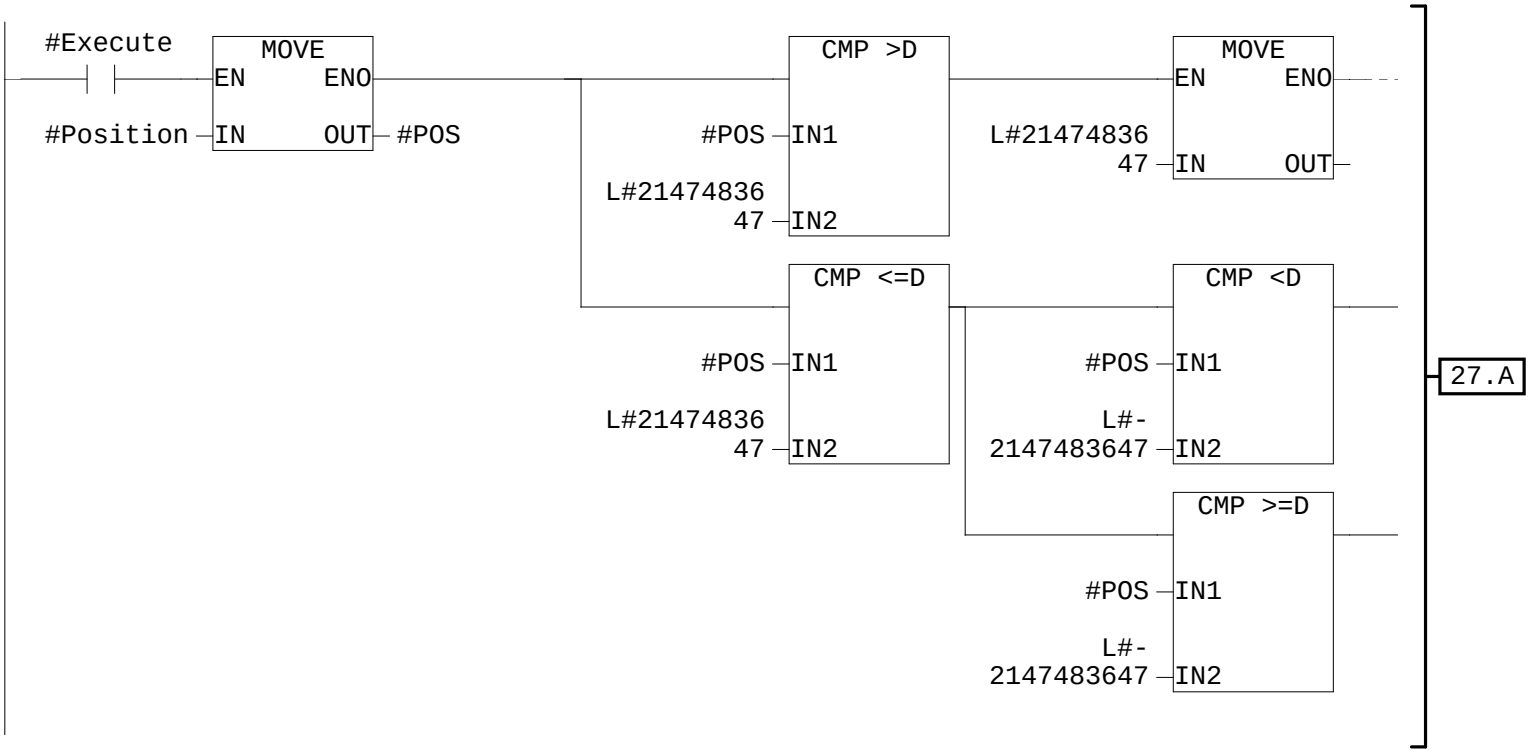


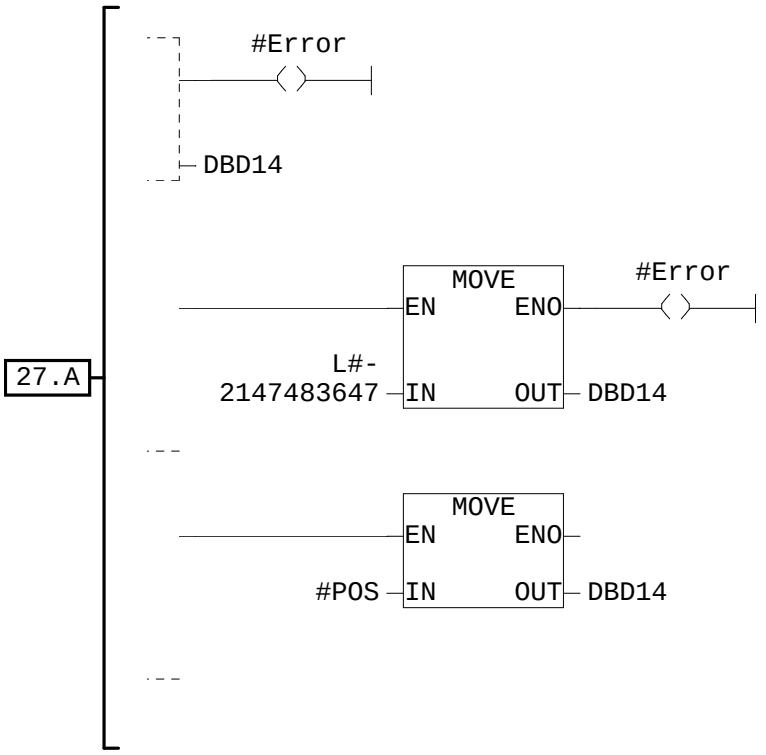
Network: 26 Mode

Nastaveni modu (EPOS_pos_type) - nastaven na Absolute
Absolute = 1, Relative = 0

Network: 27 Position

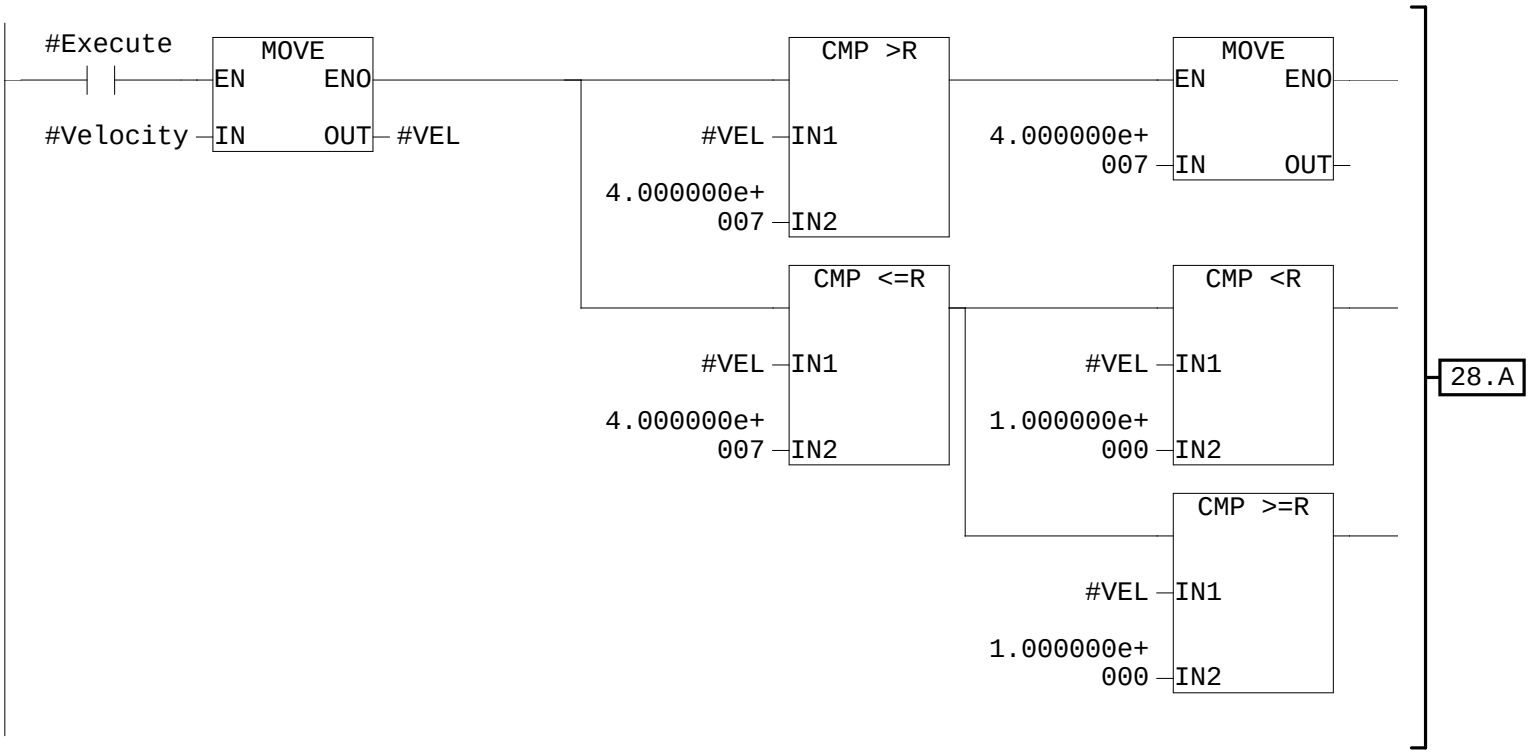
Rozsah nastaveni je od -2147483647 po 2147483647. Normalizace: 1 hex = 1 LU

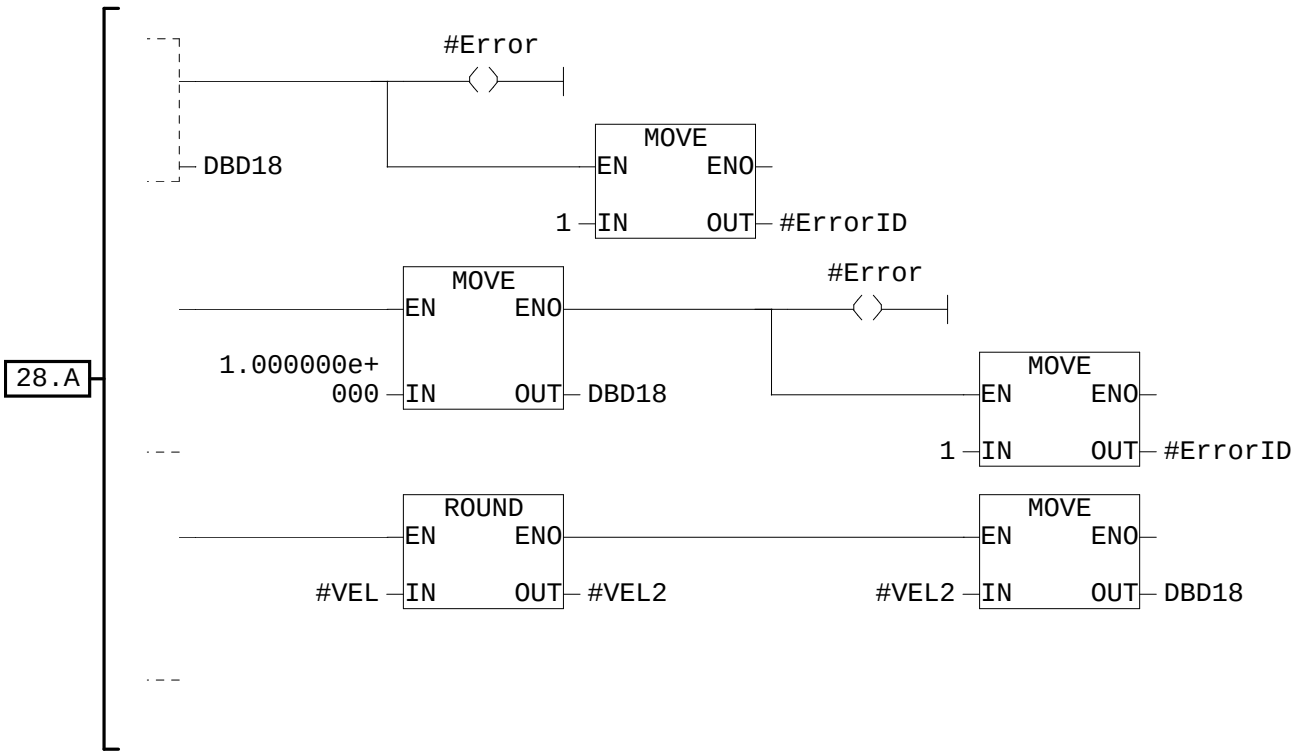




Network: 28 Velocity

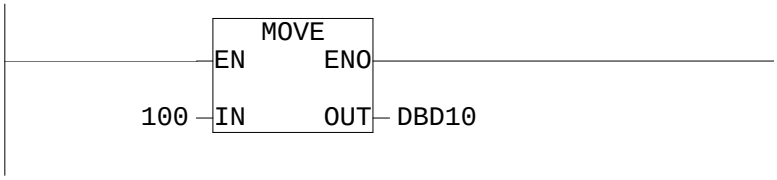
Regulovatelna v rozmezi 1 - 40000000. Co je ale maximalni rychlost, urcuje nastavitelny parametr p2571!
Normalizace: 1 hex = 1000 LU/min





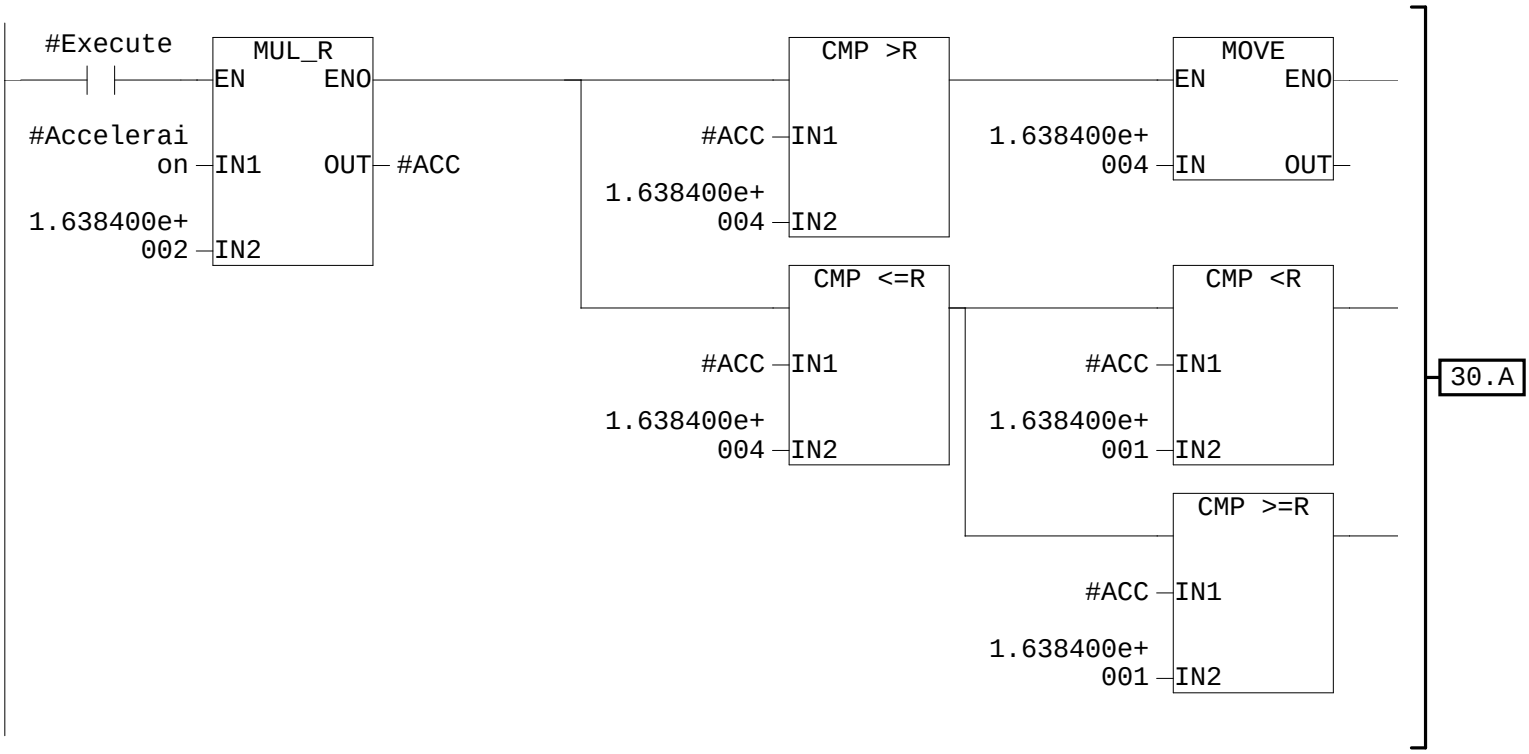
Network: 29 Override

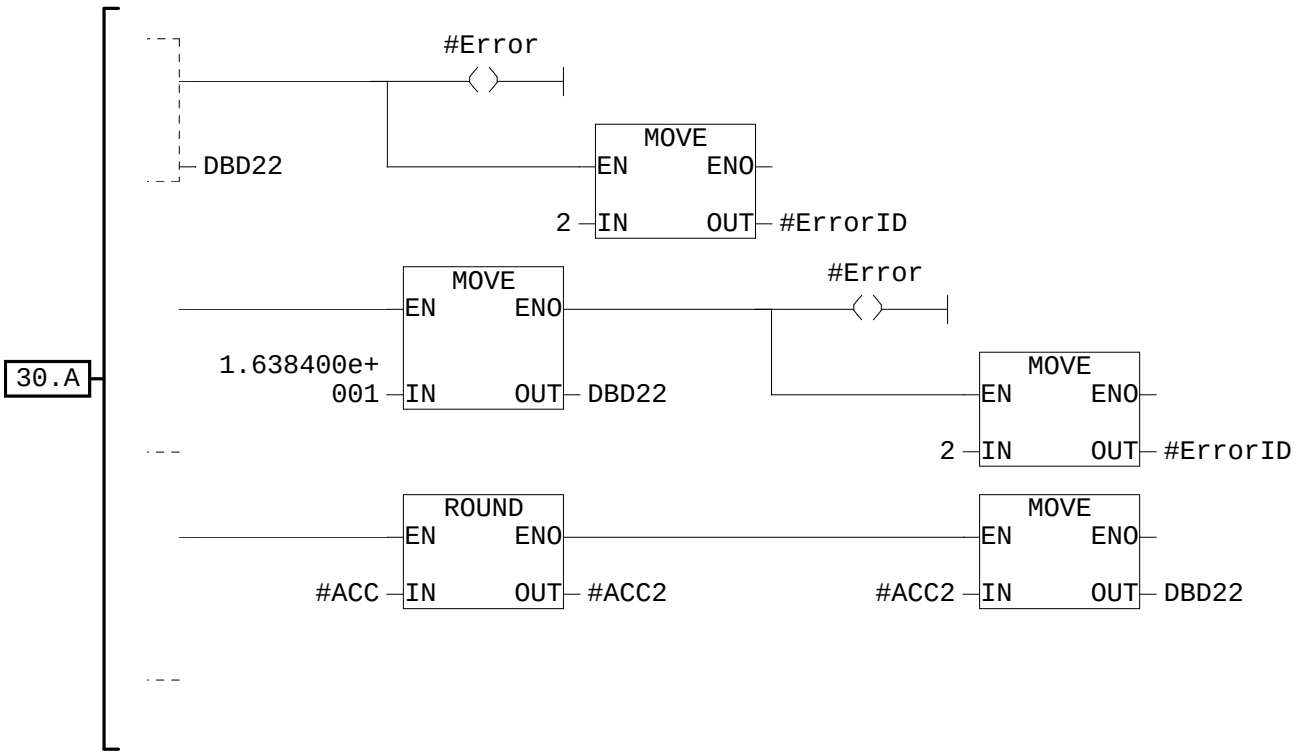
Nastaveni Override pro rychlost - 100%



Network: 30 Acceleration

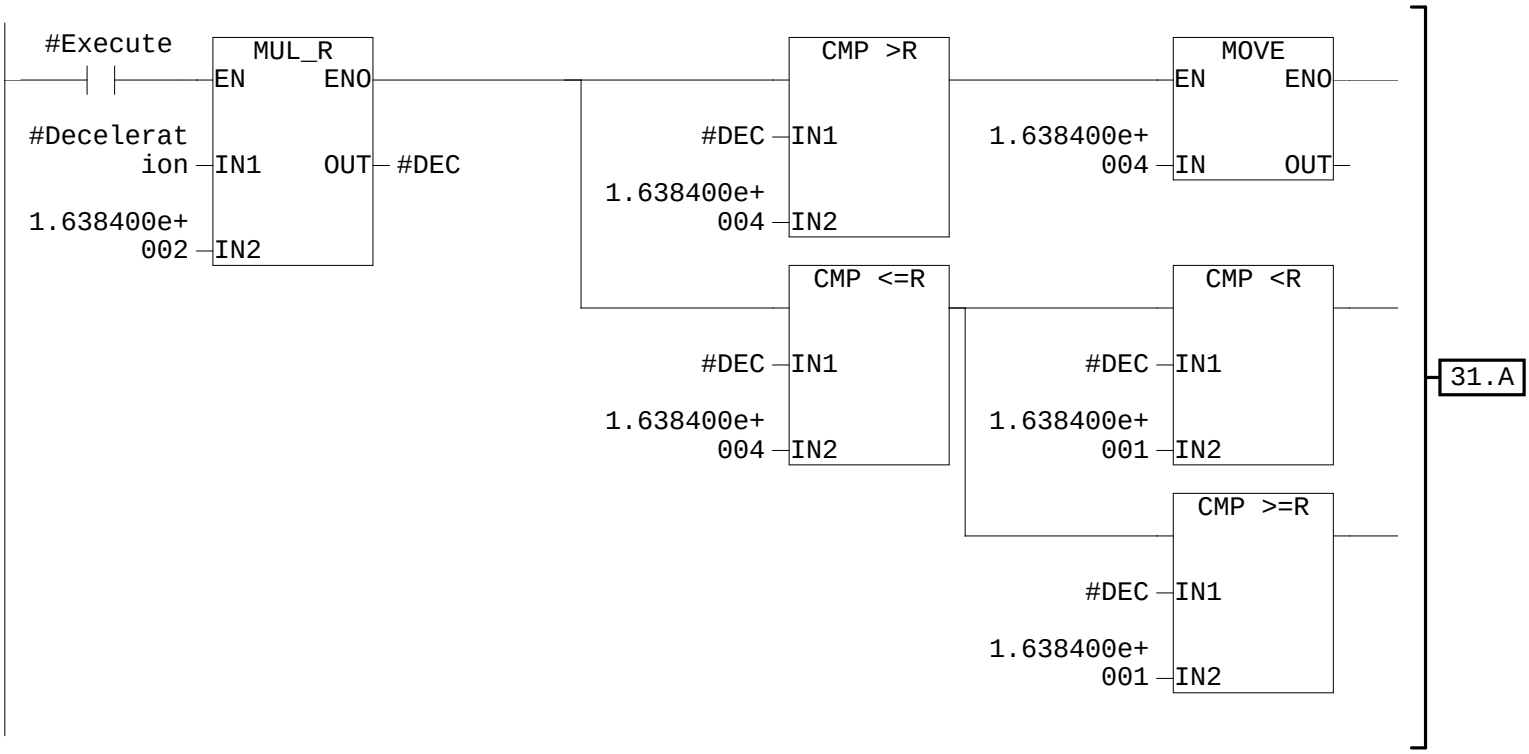
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

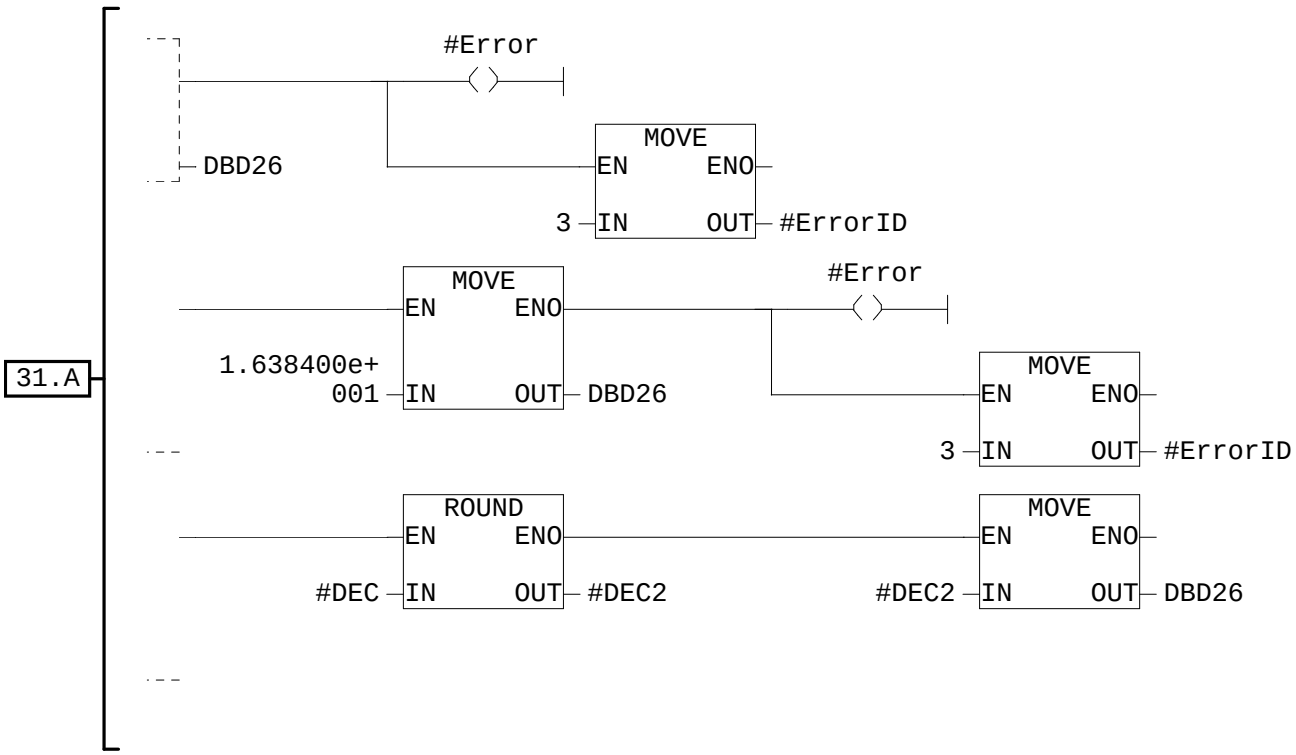




Network: 31 Deceleration

Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

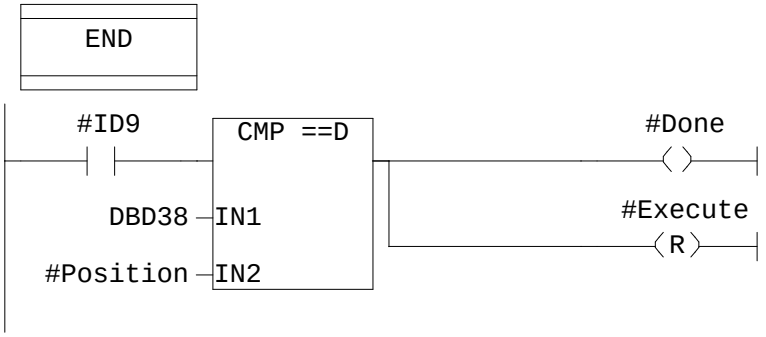




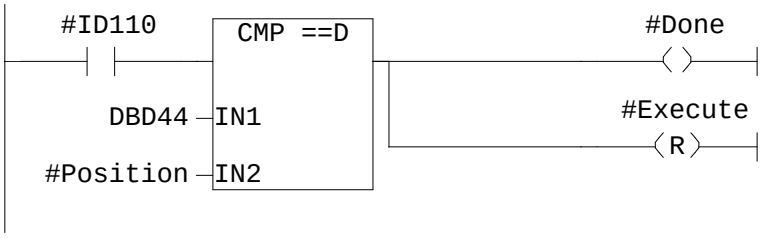
Network: 32 Konec

Network: 33 Konec

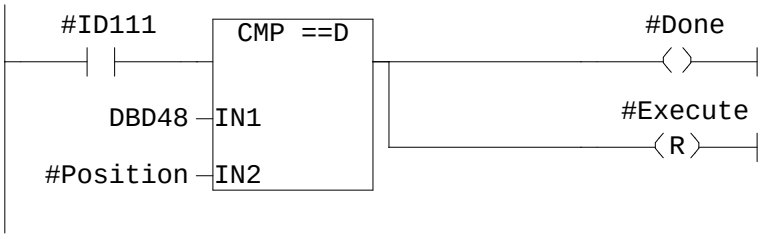
Pokud byla dosazena cilova pozice, je bit Done nastaven na hodnotu TRUE



Network: 34



Network: 35



Network: 36

Bit Active je nastaven na hodnotu FALSE

A.3 Funkce – Move Relative

FC2 - <offline>

"FC_MoveRelative"

Name: **Family:**
Author: Vacek **Version:** 0.1
Block version: 2
Time stamp Code: 05/26/2009 06:04:23 PM
Interface: 05/26/2009 06:04:23 PM
Lengths (block/logic/data): 04252 03818 00060

Name	Data Type	Address	Comment
IN		0.0	
Axis_DB	Block_DB	0.0	Otevre datovy blok pro pozadovany standardni telegram
Execute	Bool	2.0	Start
Distance	DInt	4.0	Vzdalenost
Velocity	Real	8.0	Rychlost
Acceleraion	Real	12.0	Zrychleni
Deceleration	Real	16.0	Zpomaleni
Direction	Bool	20.0	Smer
OUT		0.0	
Done	Bool	22.0	Dosazena cilova pozice
Active	Bool	22.1	Funkce je aktivni
Error	Bool	22.2	Chyba
ErrorID	Int	24.0	Identifikace chyb
IN_OUT		0.0	
TEMP		0.0	
HELP	Bool	0.0	
VEL	Real	2.0	
VEL2	DInt	6.0	
ACC	Real	10.0	
ACC2	DInt	14.0	
DEC	Real	18.0	
DEC2	DInt	22.0	
DIR	DInt	26.0	
POS	DInt	30.0	
ID9	Bool	34.0	
ID110	Bool	34.1	
ID111	Bool	34.2	
STARTPOS	DInt	36.0	
CILPOS	DInt	40.0	
CILPOS2	DInt	44.0	
Cislo	Word	48.0	Cislo DB
Delka	Word	50.0	Delka DB v bytech
Return	Int	52.0	Navratova hodnota (0=OK)

Name	Data Type	Address	Comment
WrPr	Bool	54.0	DB chranen proti prepsani (1=ANO)
Delka2	Int	56.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC2

```
Created:
date 17.04.2009 version 1.0 - Jakub Vacek
-------------------------------------------------------
Funkce MoveAbsolute startuje nastaveni polohy osy podle relativni pozice
-------------------------------------------------------
Telegram: 9,110,111
-------------------------------------------------------
Identifikace chyby:
ErrorID = 1, spatne zadana rychlost
ErrorID = 2, spatne zadane zrychleni
ErrorID = 3, spatne zadane zpomaleni
ErrorID = 4, nebyl nalezen kompatibilni telegram
```

Network: 1 Otevri DB

Funkce otevre datovy blok pro pozadovany standardni telegram

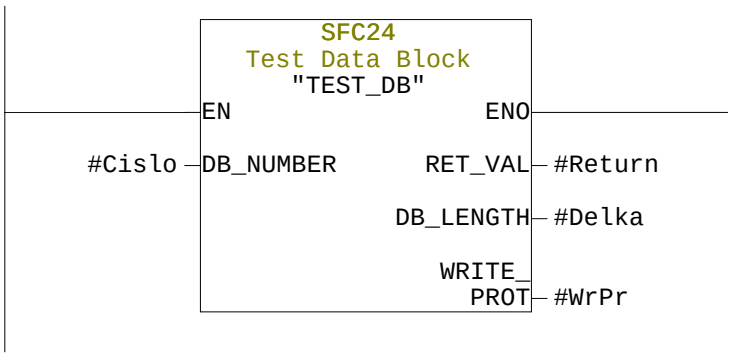
Network: 2

Do pomocne promenne Cislo ulozi cislo DB

```
      L     DBNO
      T     #Cislo
```

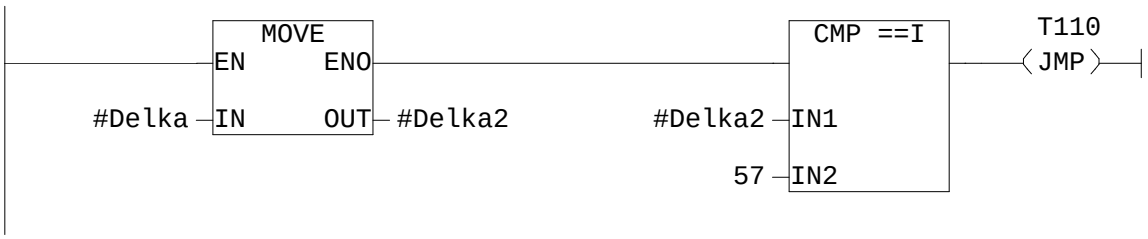
Network: 3

Zjisteni delky datoveho bloku



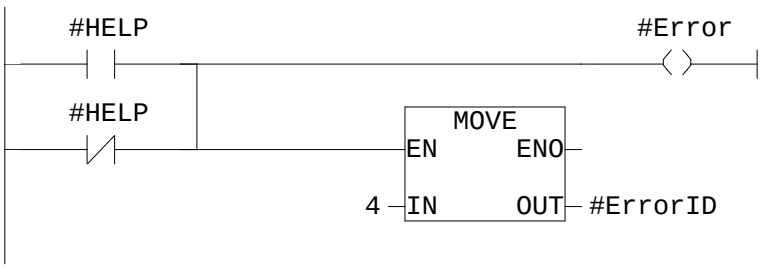
Network: 4 Je pouzit telegram 9?

Network: 5 Je pouzit telegram 110?



Network: 6 Je pouzit telegram 111?

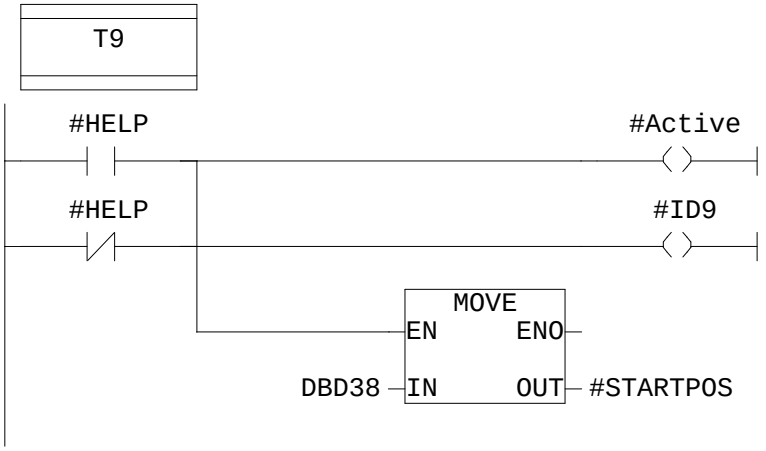
Network: 7 Nebyl pouzit vhodny telegram



Network: 8 Konec

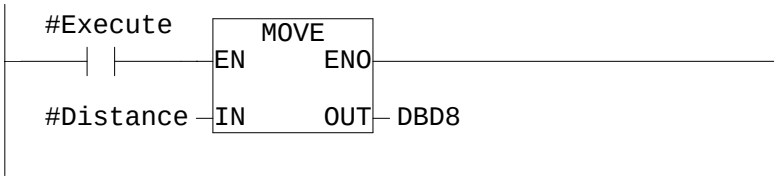
Network: 9 Telegram 9

Do pomocne promenne STARTPOS se ulozi aktualni pozice osy pred startem funkce



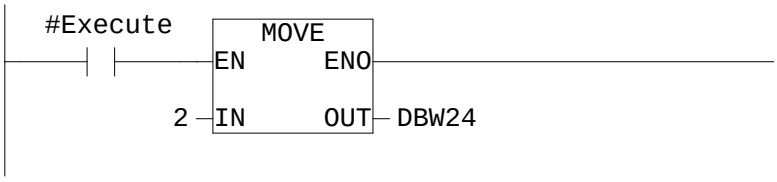
Network: 10 Distance

Nastaveni vzdalenosti
Standardni normalizace v menicich je 1 hex = 1 LU



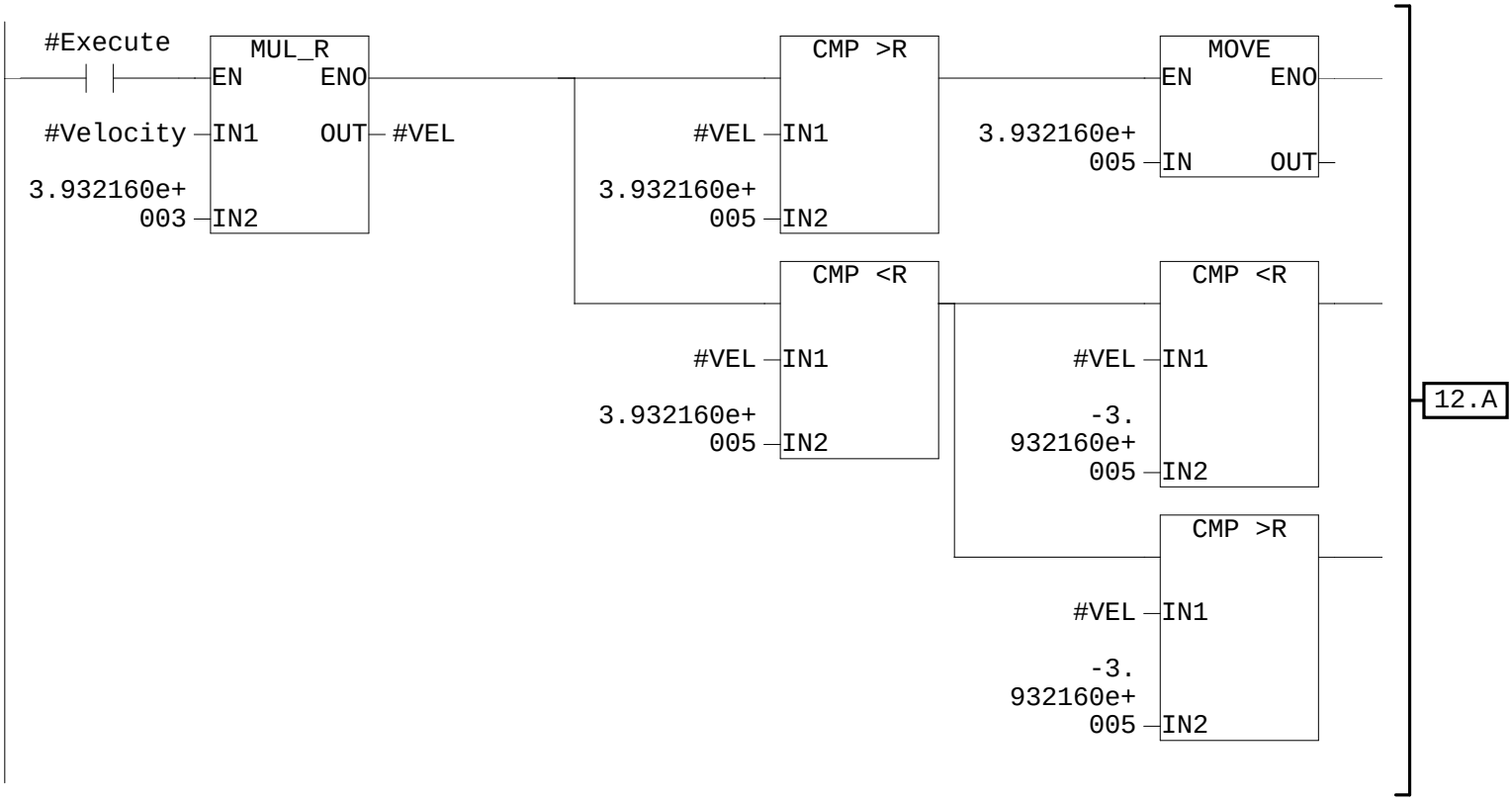
Network: 11 Mode

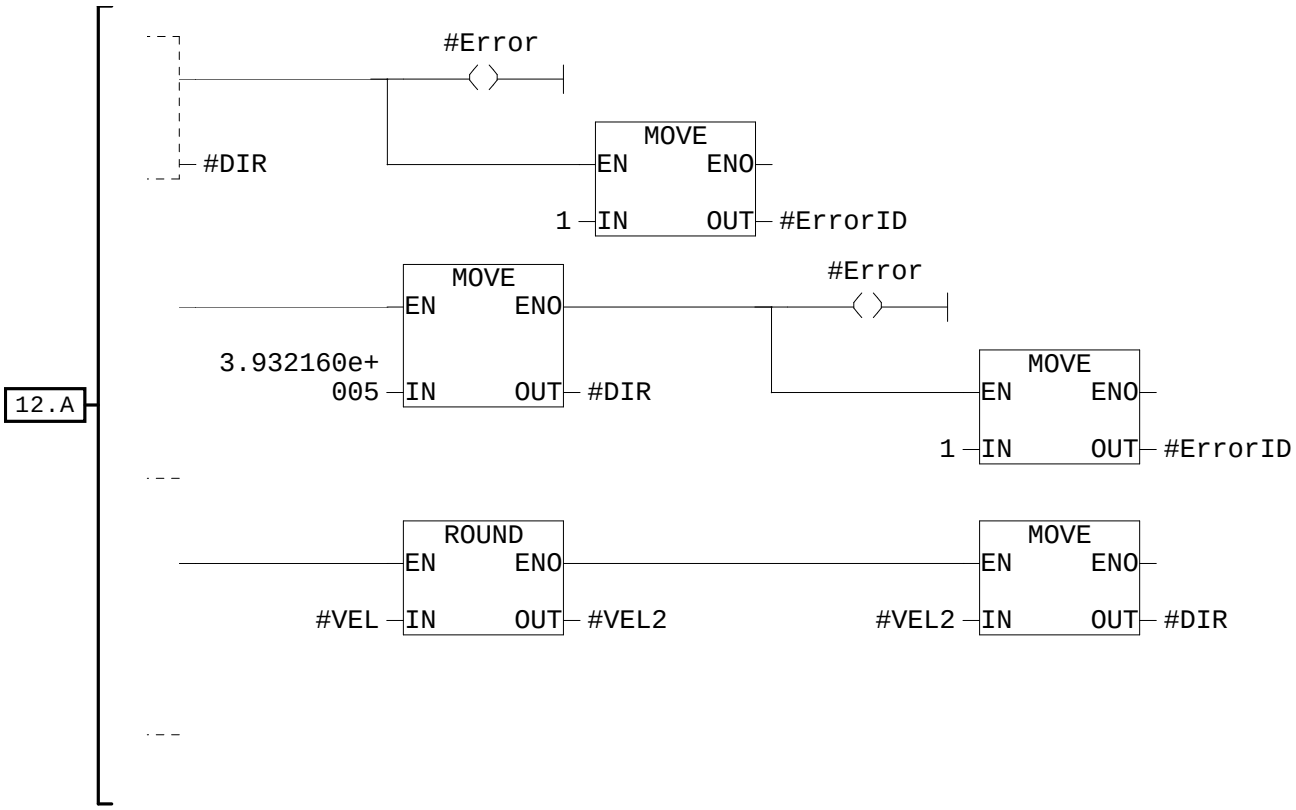
Nastaveni modu - nastaven na Relative
Absolute = 1, Relative = 2, Absolute positive = 3, Absolute negative = 4



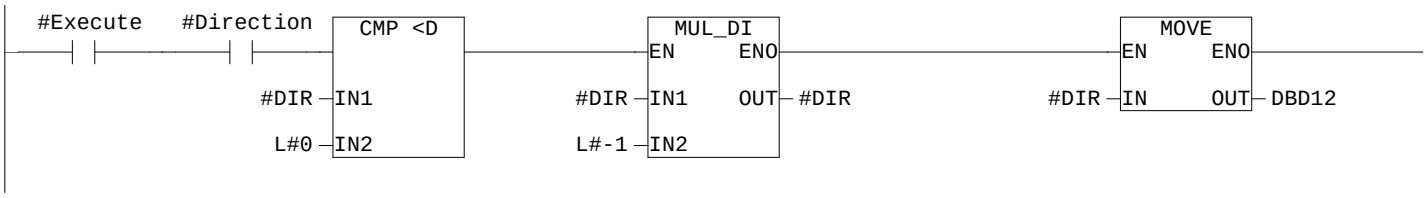
Network: 12 Velocity

Nastaveni rychlosti v procentech (0-100)
Standardni normalizace v menicich je 1 hex = 1000 LU/min, maximalni rychlost
udava parametr p2000 (6000 rev/min)
Pri nastaveni 10000 LU per load revolution je 60000 hex = 100%
Provadi se omezeni na hodnoty double integer

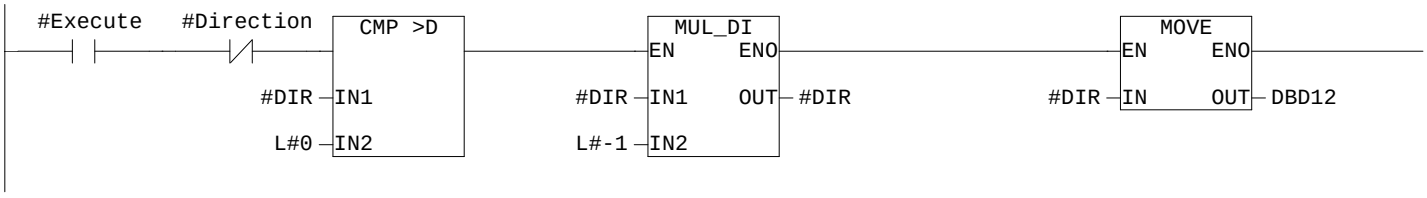




Network: 13 Direction
Smer otaceni nastaven na Forward Direction = True

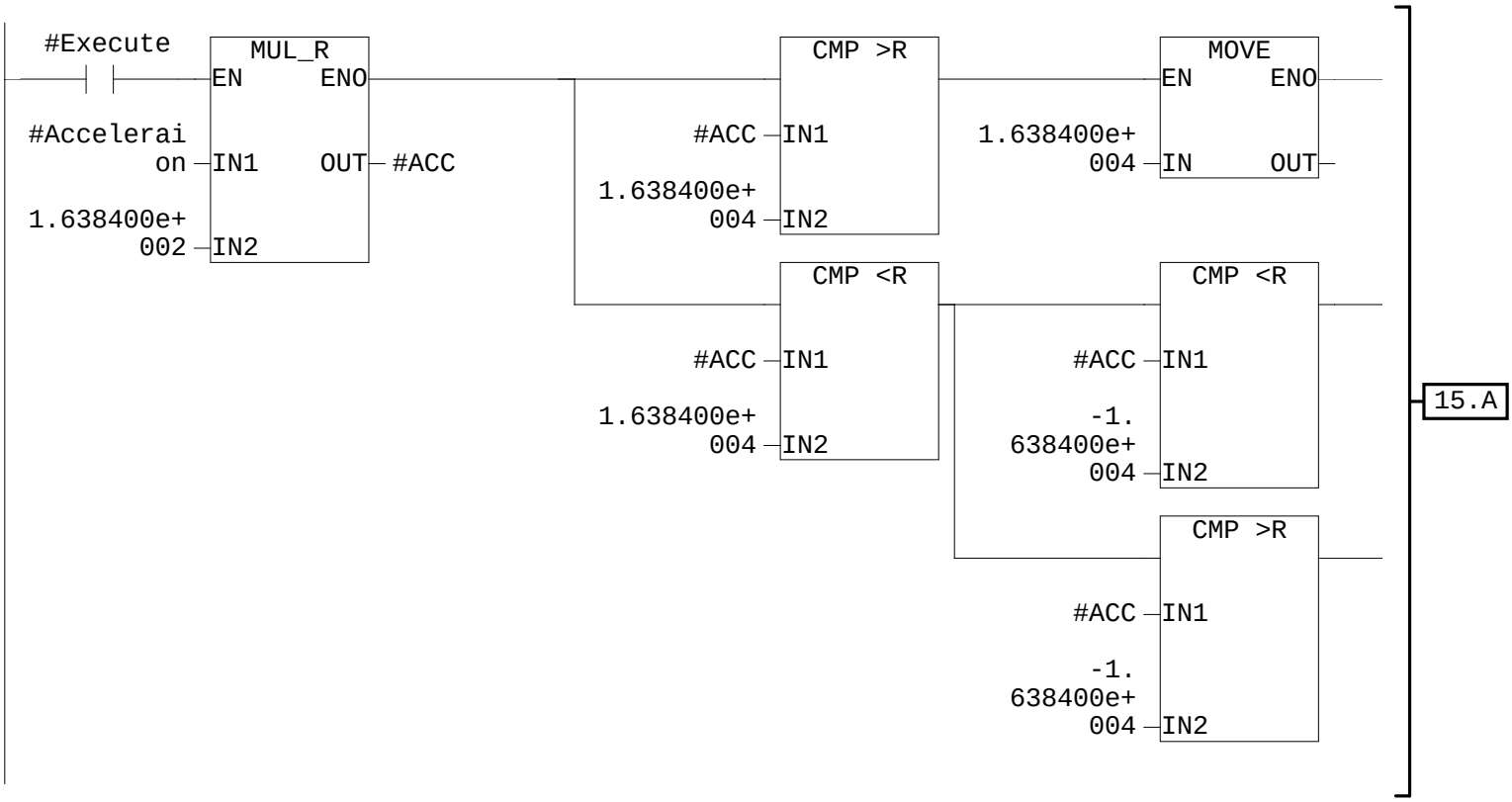


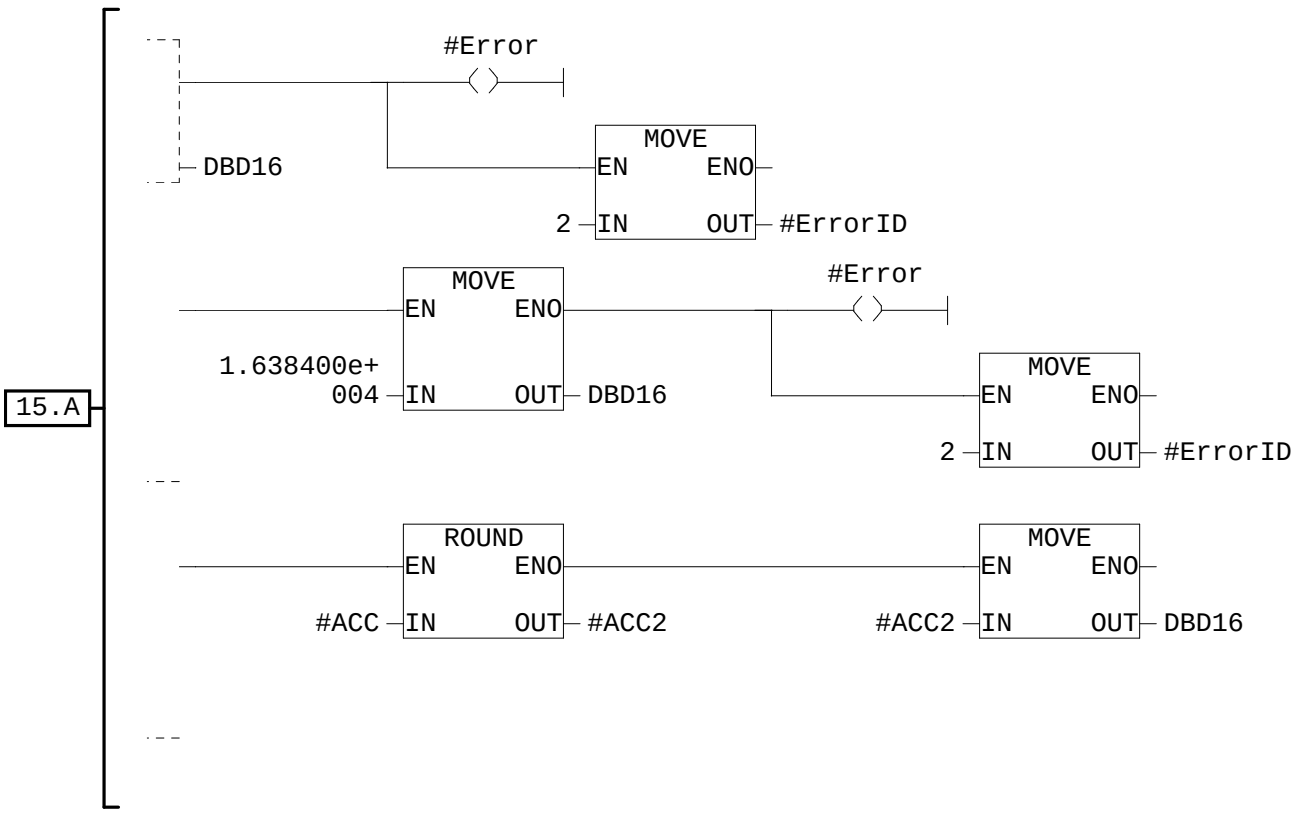
Network: 14 Direction
Smer otaceni nastaven na Backward Direction = False



Network: 15 Acceleration

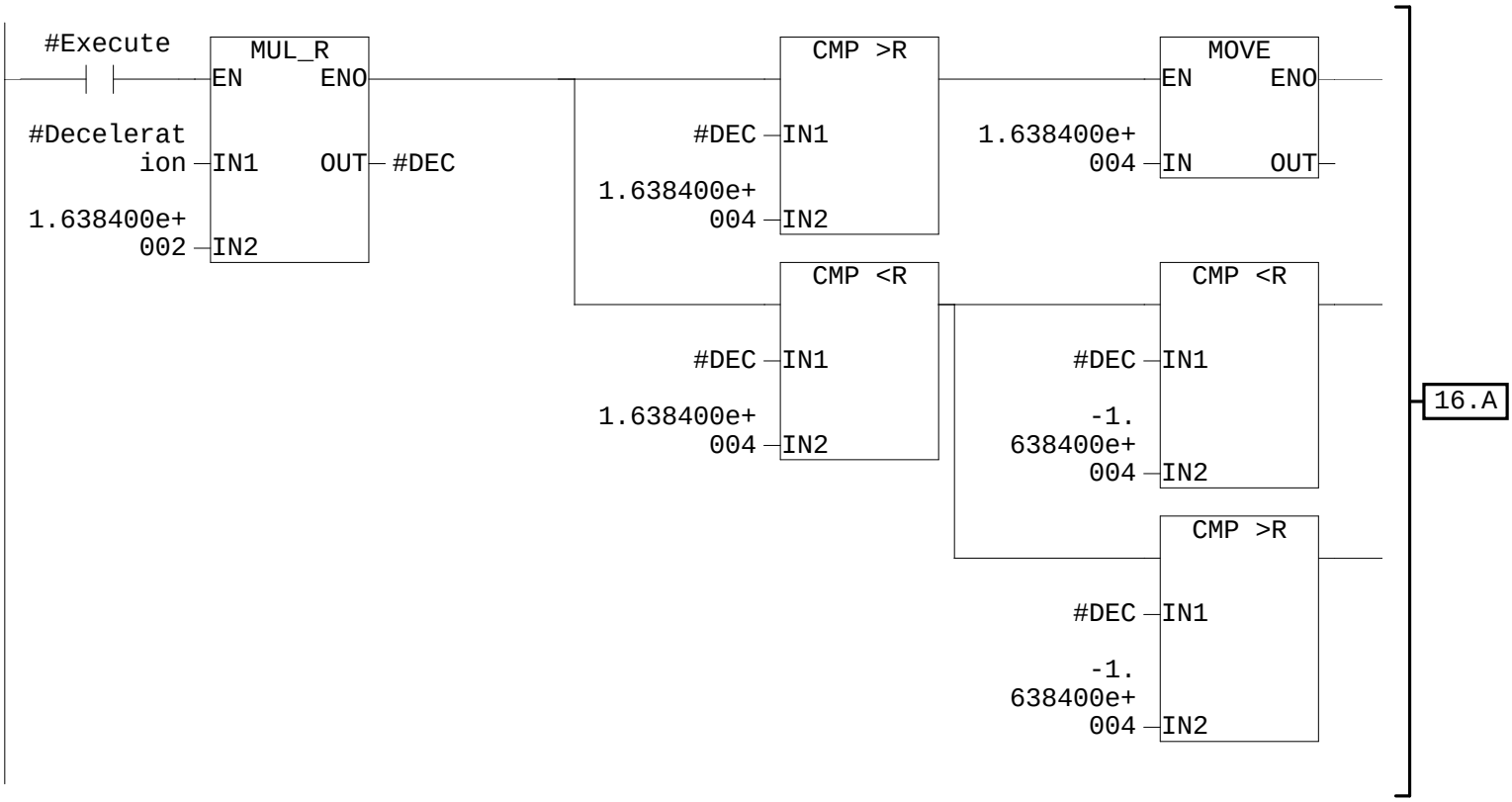
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

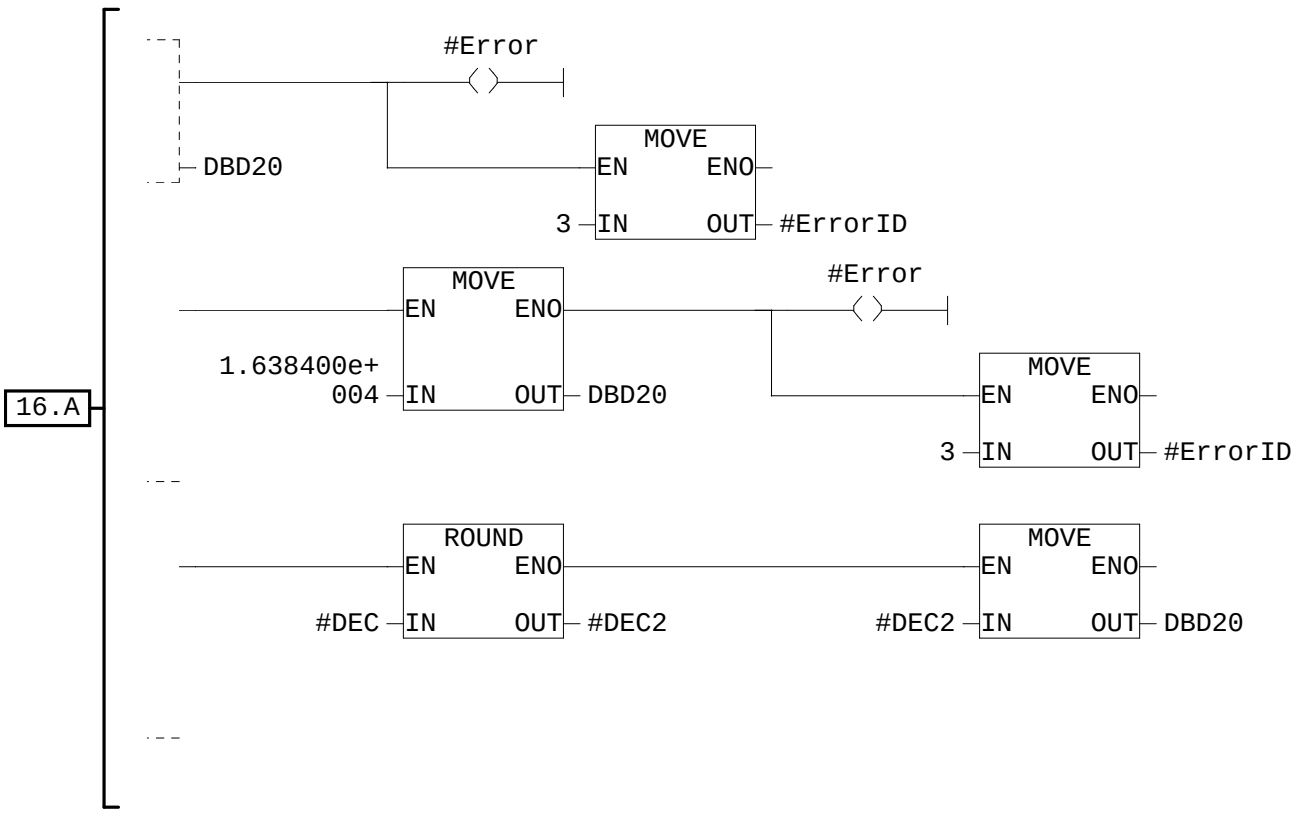




Network: 16 Deceleration

Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

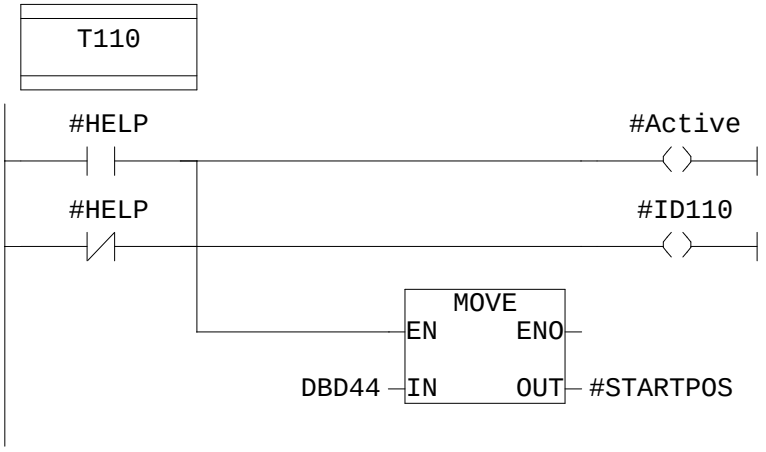




Network: 17 Konec

Network: 18 Telegram 110

Do pomocne promenne STARTPOS se ulozi aktualni pozice osy pred startem funkce

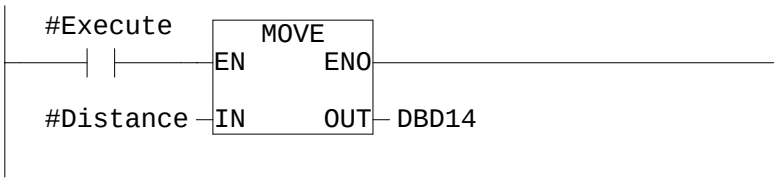


Network: 19 Mode

Nastaveni modu (EPOS_pos_type) - nastaven na Relative
Absolute = 1, Relative = 0

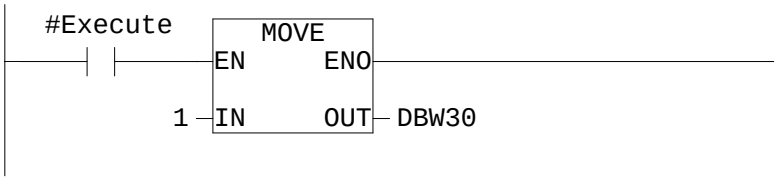
Network: 20 Distance

Nastaveni vzdalenosti
Standardni normalizace v menicich je 1 hex = 1 LU



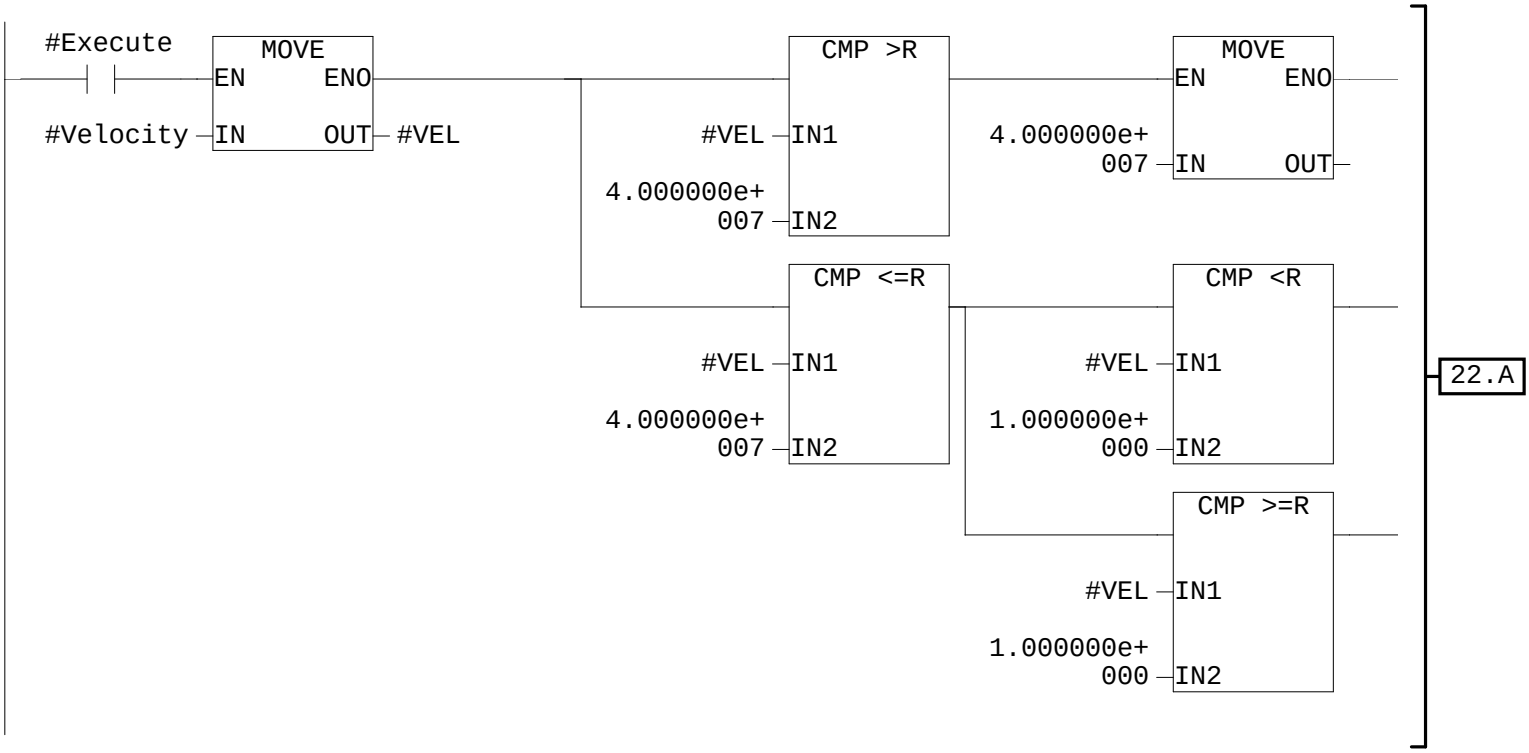
Network: 21 Mode

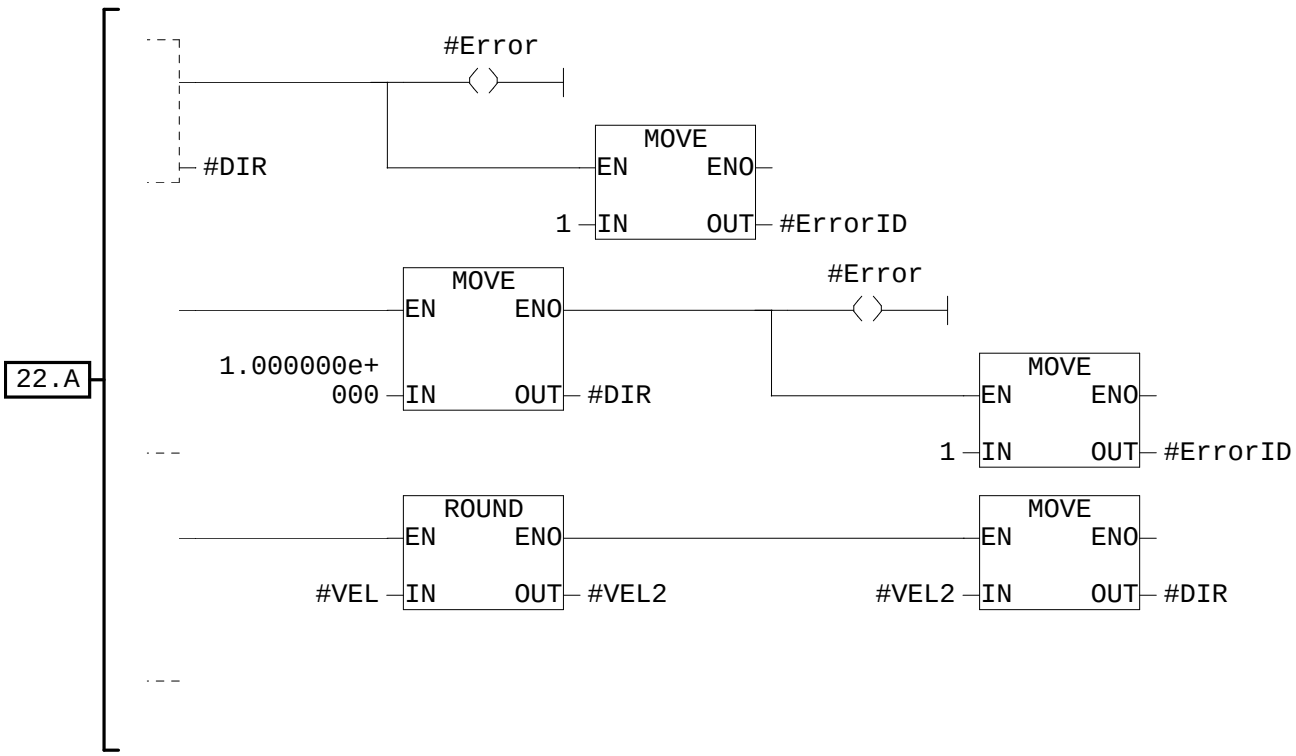
Nastaveni modu - nastaven na Relative
Absolute = 0, Relative = 1, Absolute positive = 2, Absolute negative = 3



Network: 22 Velocity

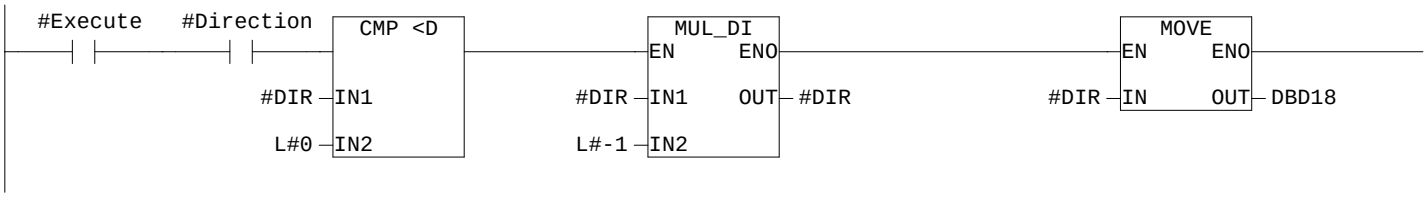
Nastaveni rychlosti
Regulovatelna v rozmezi 1 - 40000000
Standardni normalizace v menicich je 1 hex = 1000 LU/min, maximalni rychlost
udava parametr p2571
Provadi se omezeni na hodnoty double integer





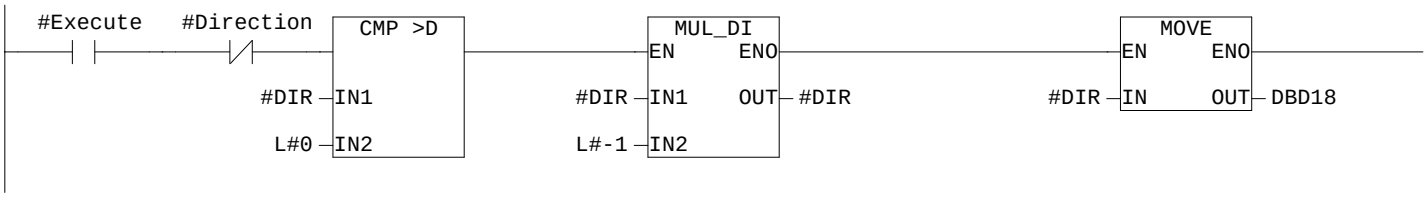
Network: 23 Direction

Smer otaceni nastaven na Forward
Direction = True



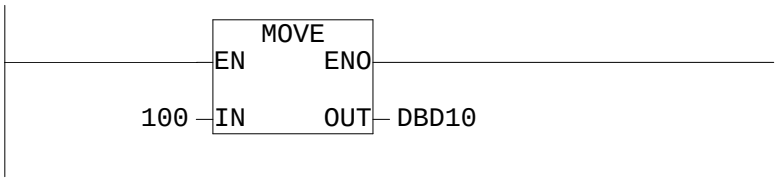
Network: 24 Direction

Smer otaceni nastaven na Backward
Direction = False



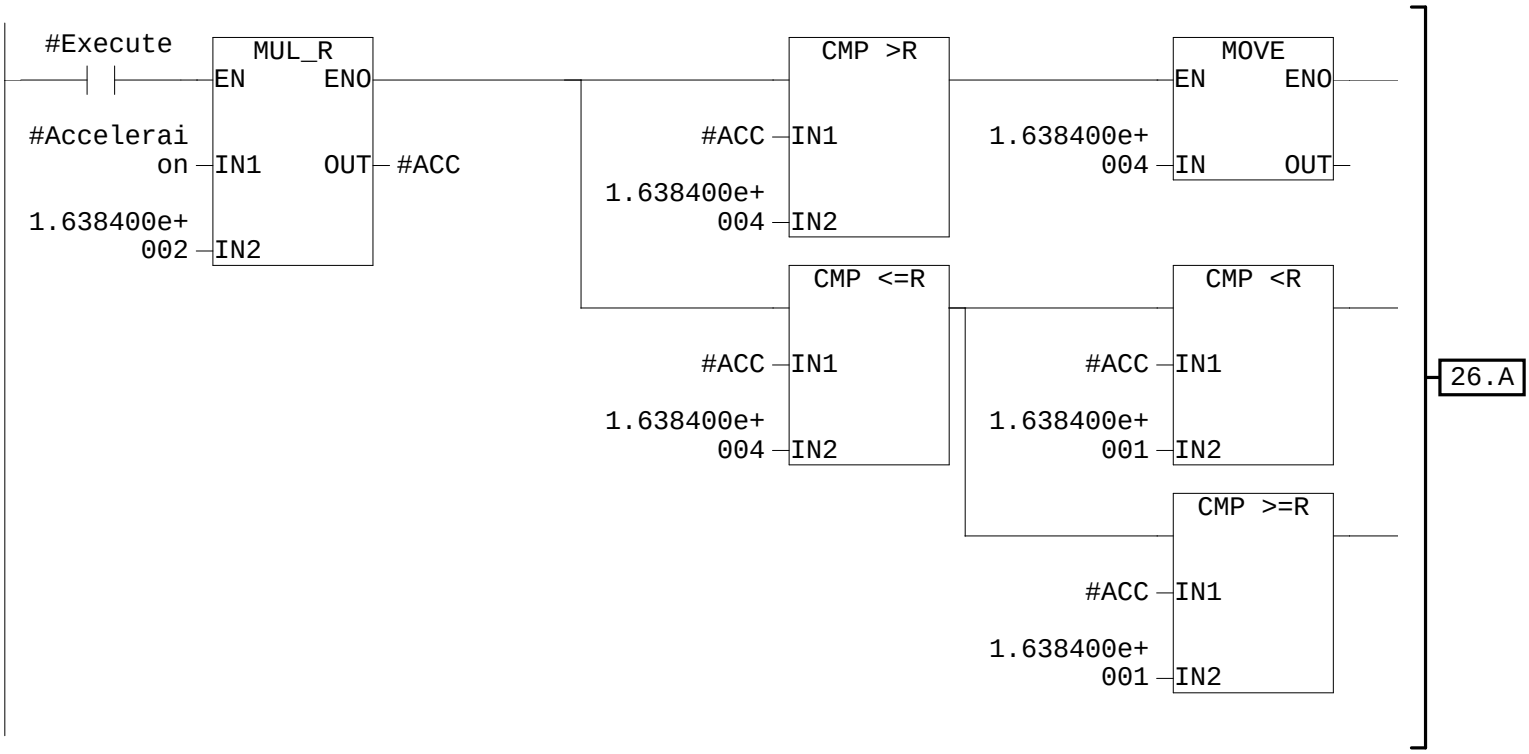
Network: 25 Override

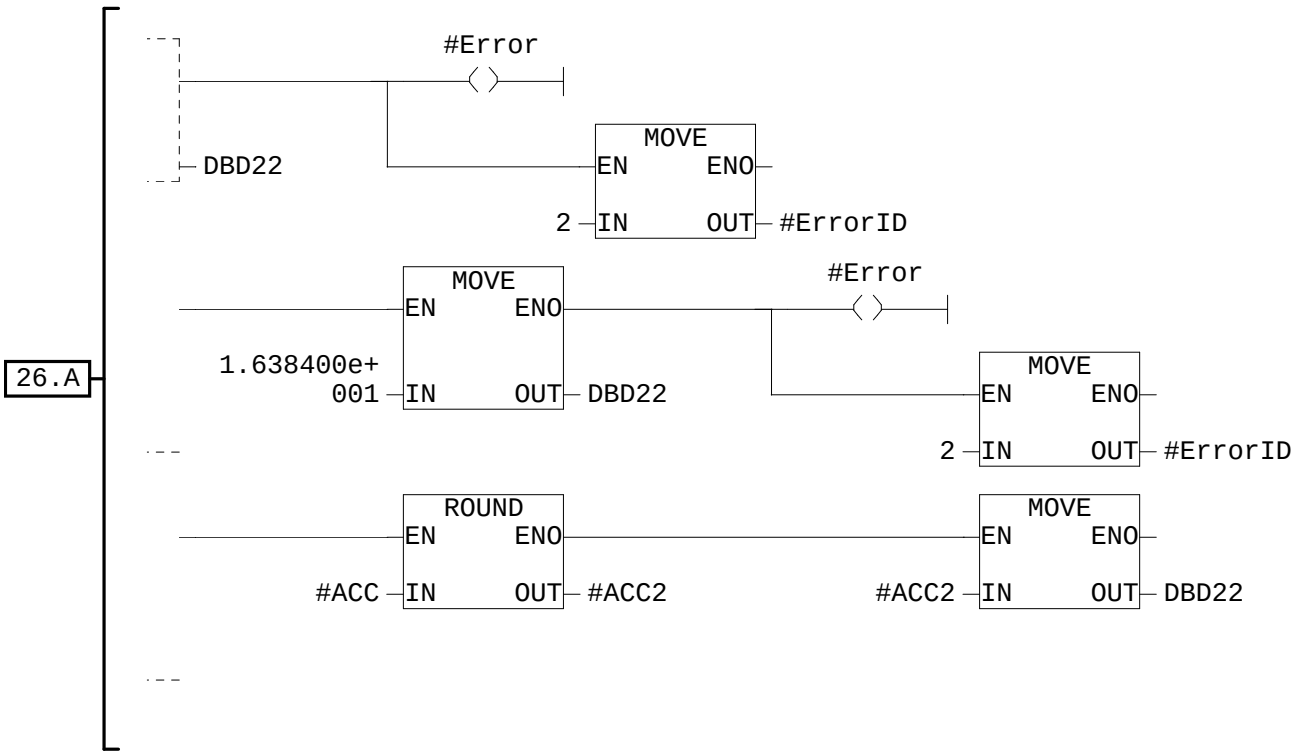
Nastaveni Override pro rychlost - 100%



Network: 26 Acceleration

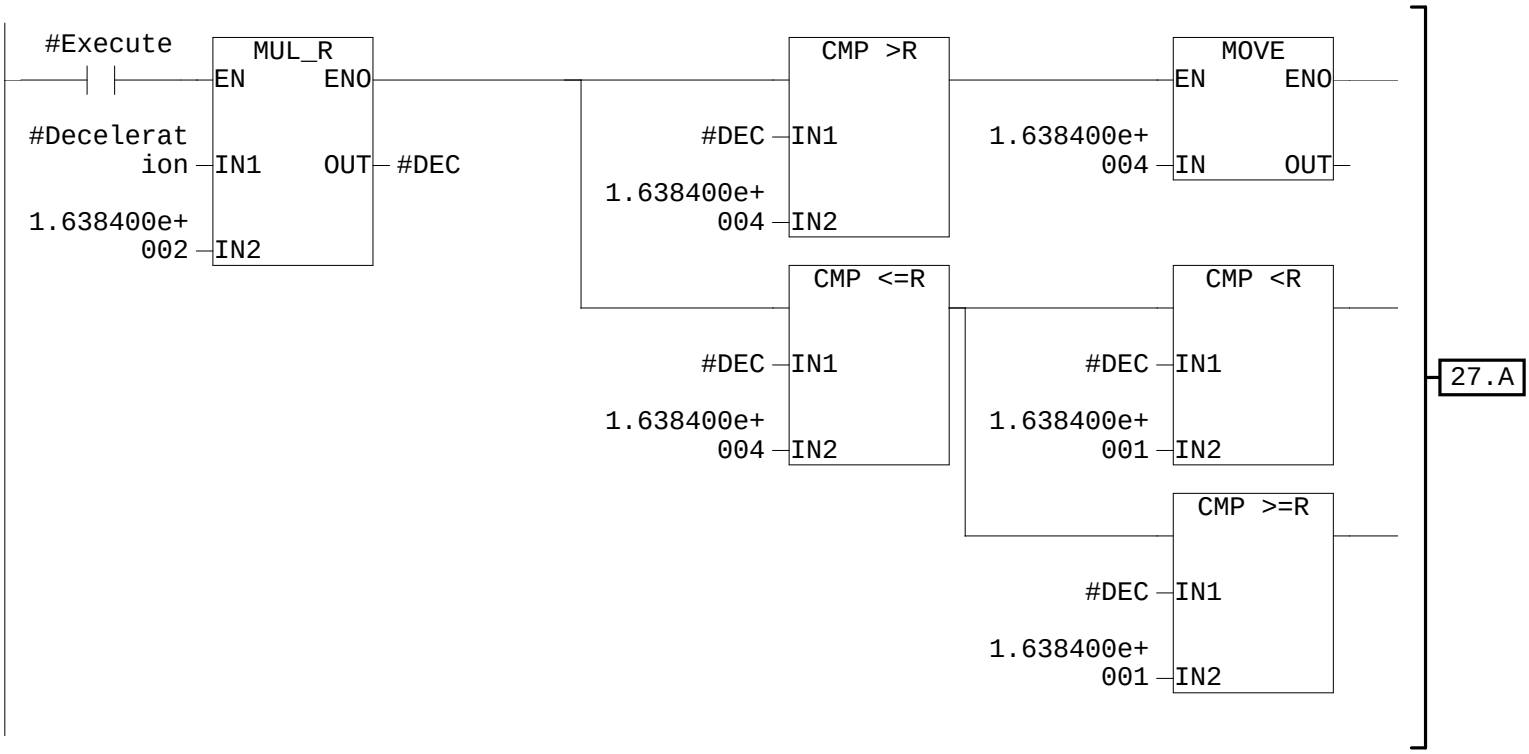
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

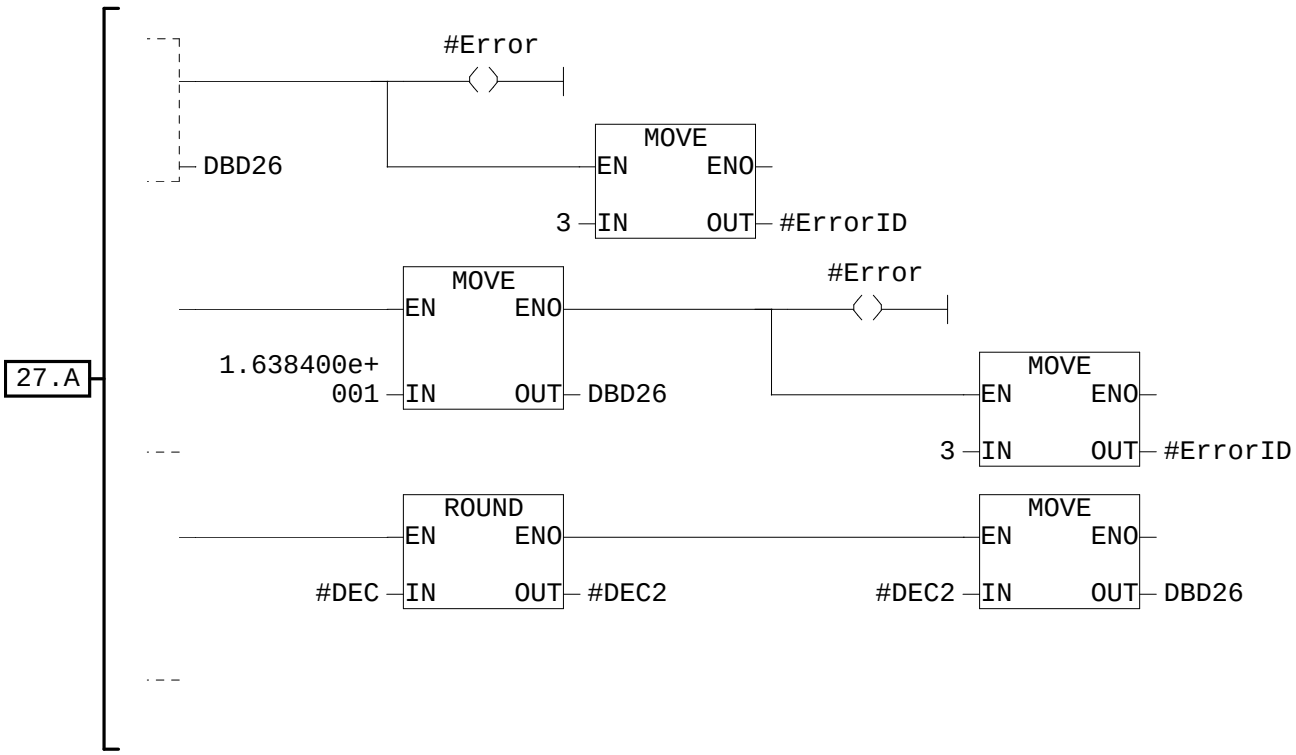




Network: 27 Deceleration

Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

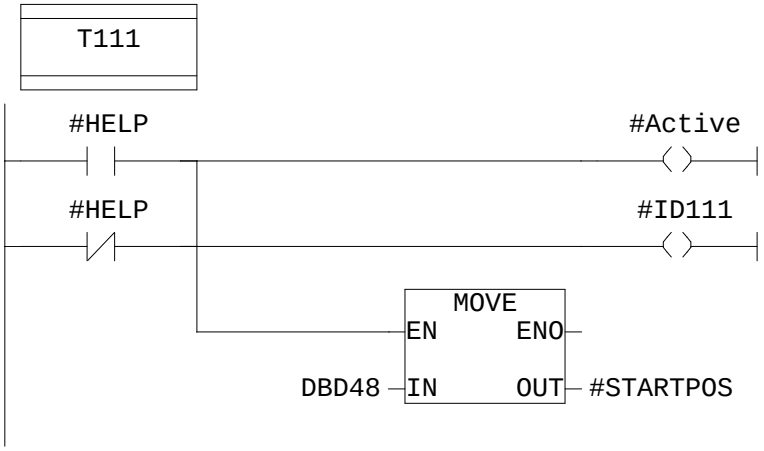




Network: 28 Konec

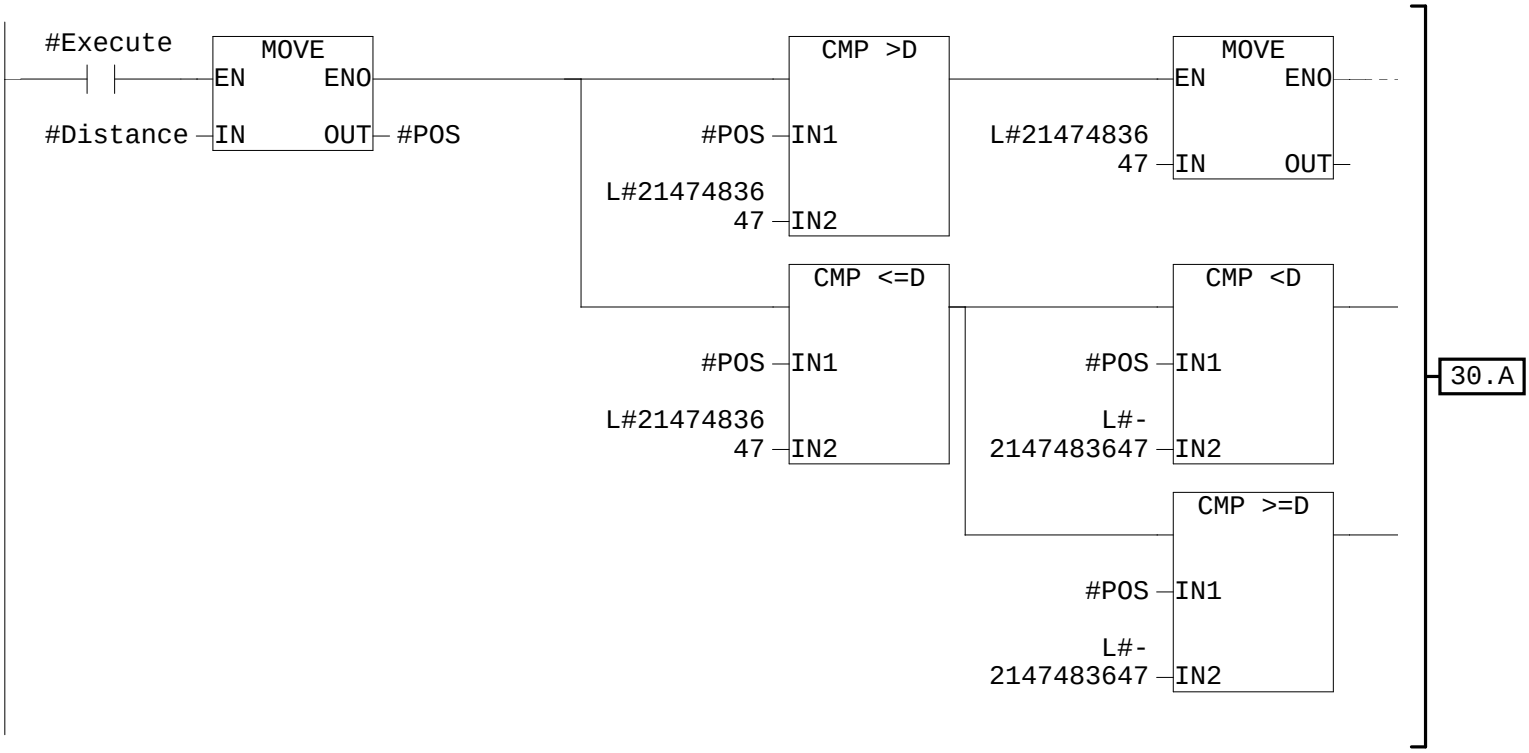
Network: 29 Telegram 111

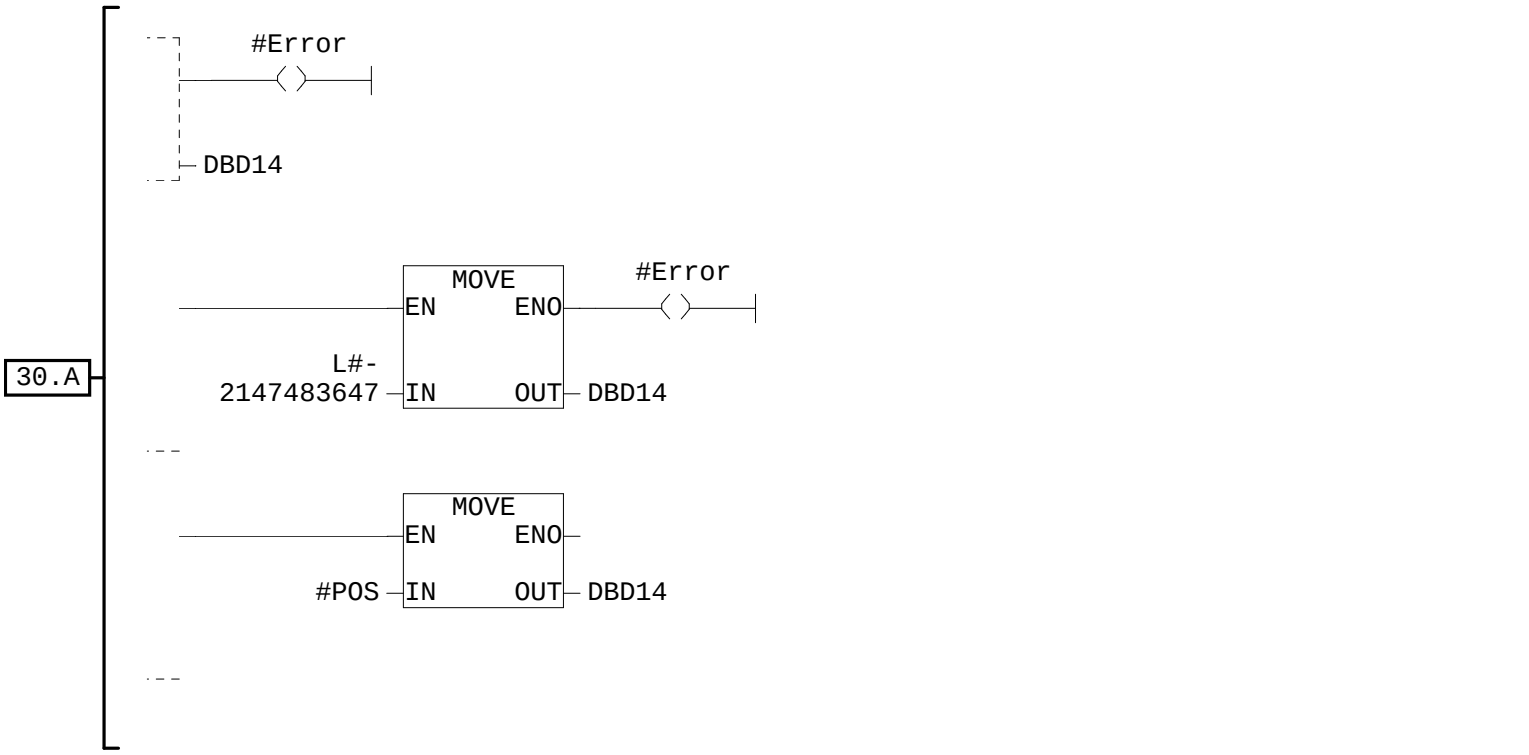
Do pomocne promenne STARTPOS se ulozi aktualni pozice osy pred startem funkce



Network: 30 Distance

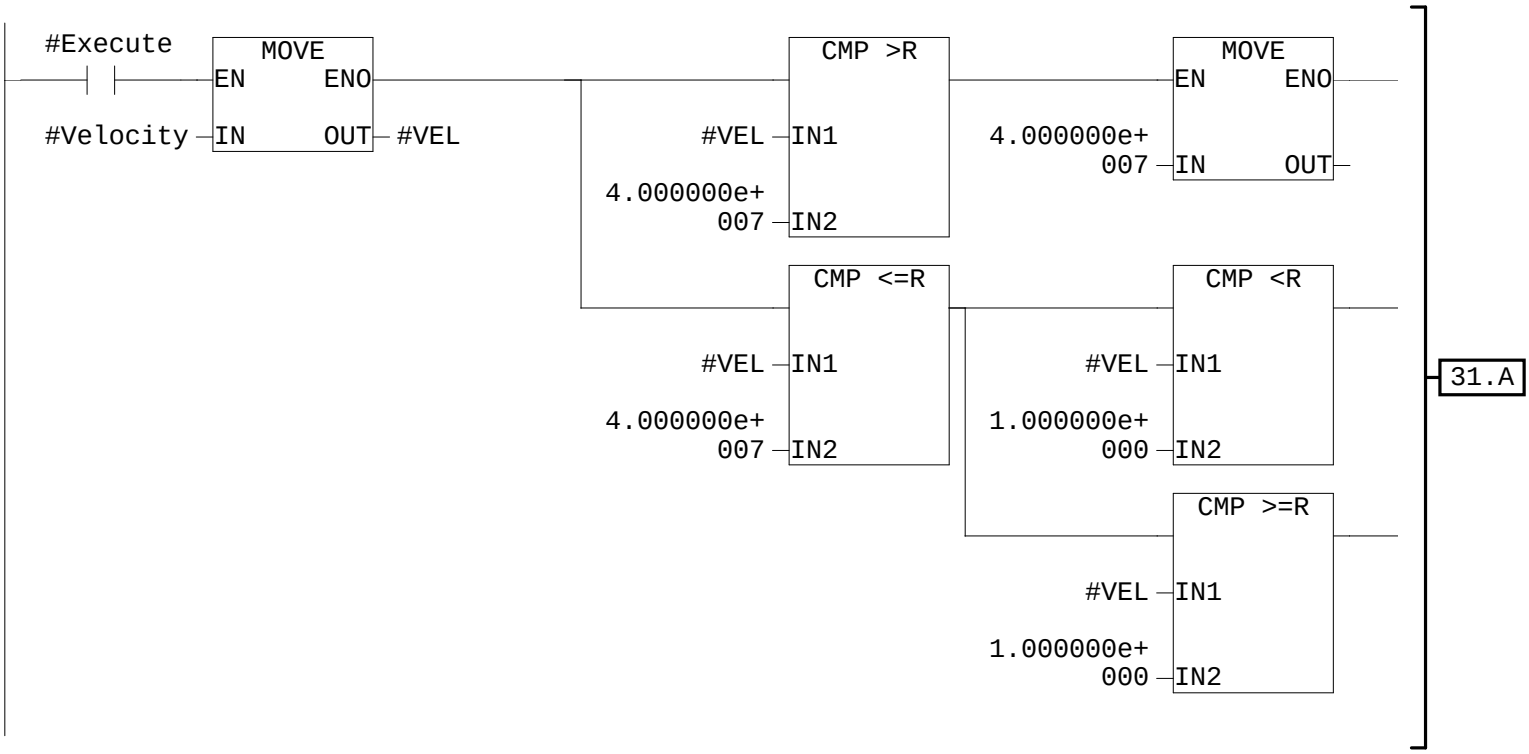
Normalizace: 1 hex = 1 LU

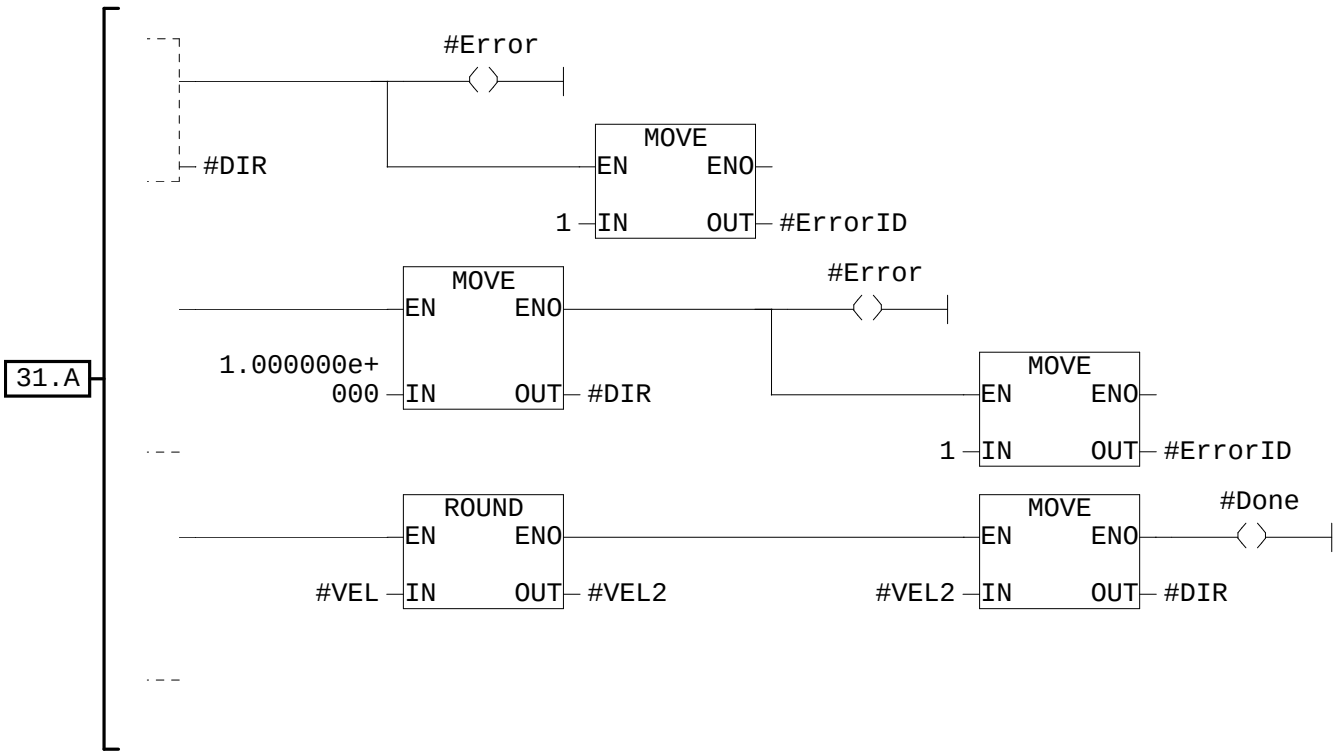




Network: 31 Velocity

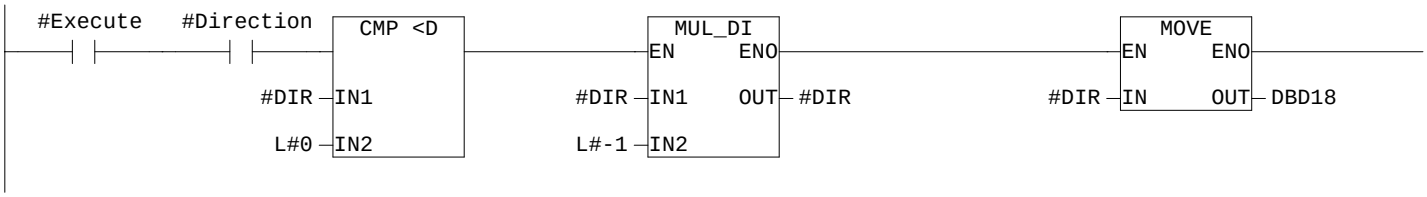
Regulovatelna v rozmezi 1 - 40000000. Co je ale maximalni rychlost, urcuje nastavitelny parametr p2571!
Normalizace: 1 hex = 1000 LU/min





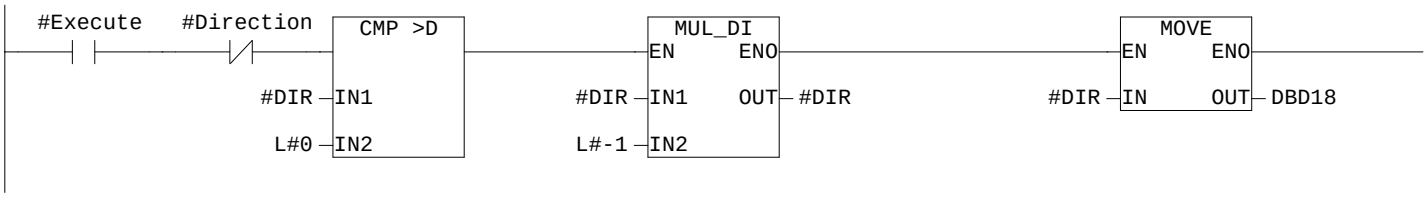
Network: 32 Direction

Smer otaceni nastaven na Forward
Direction = True



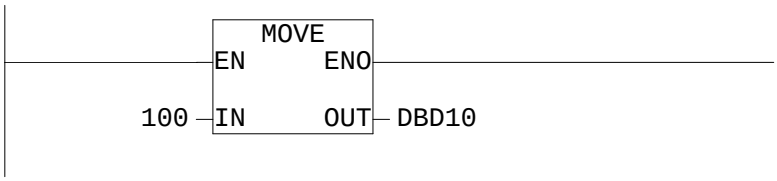
Network: 33 Direction

Smer otaceni nastaven na Backward
Direction = False



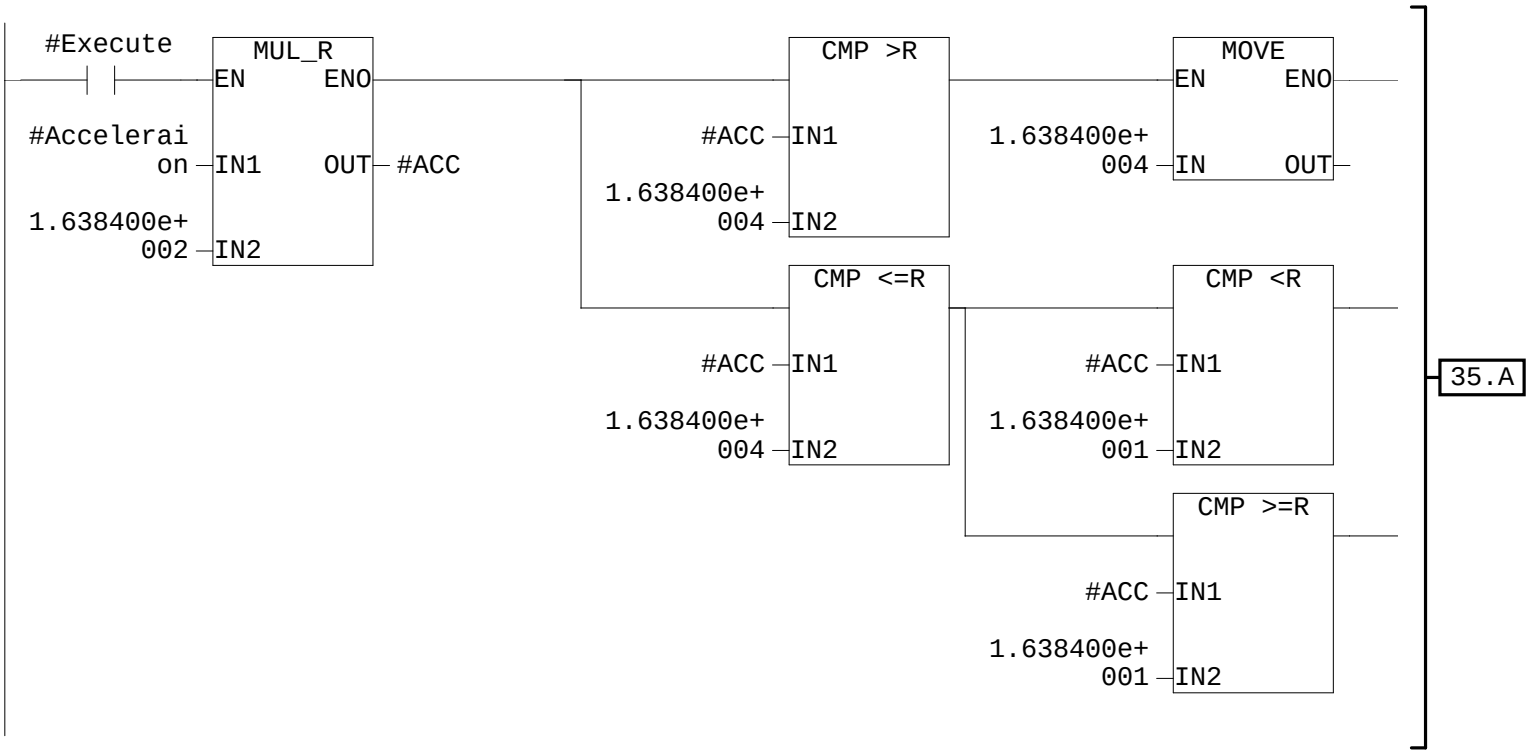
Network: 34 Override

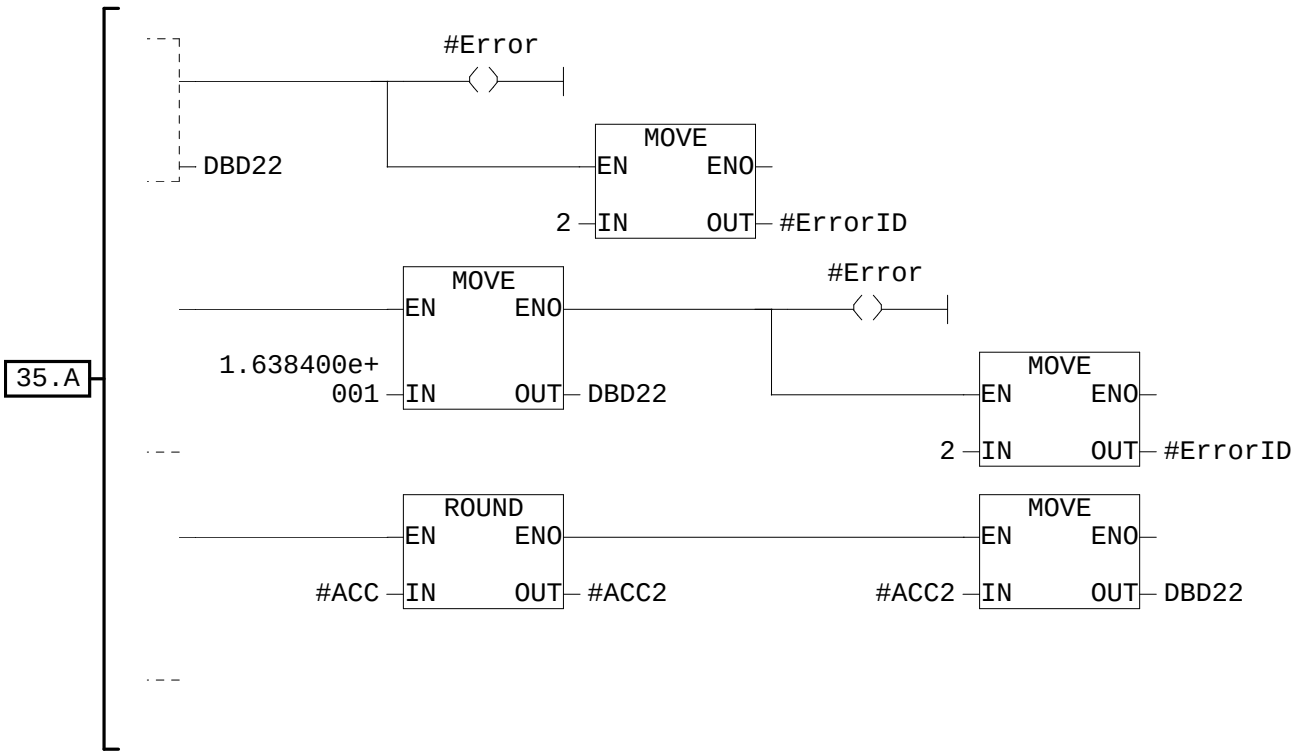
Nastaveni Override pro rychlost - 100%



Network: 35 Acceleration

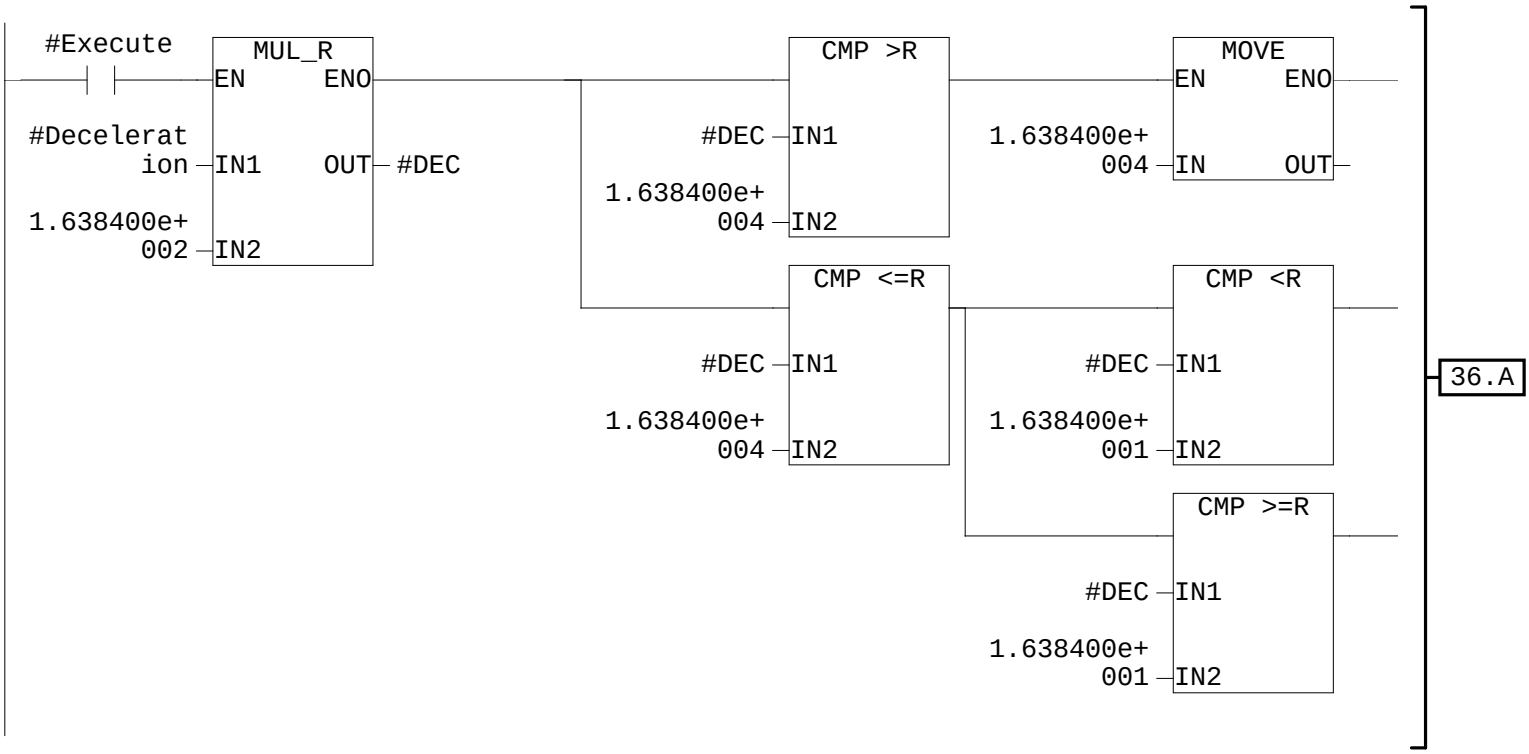
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

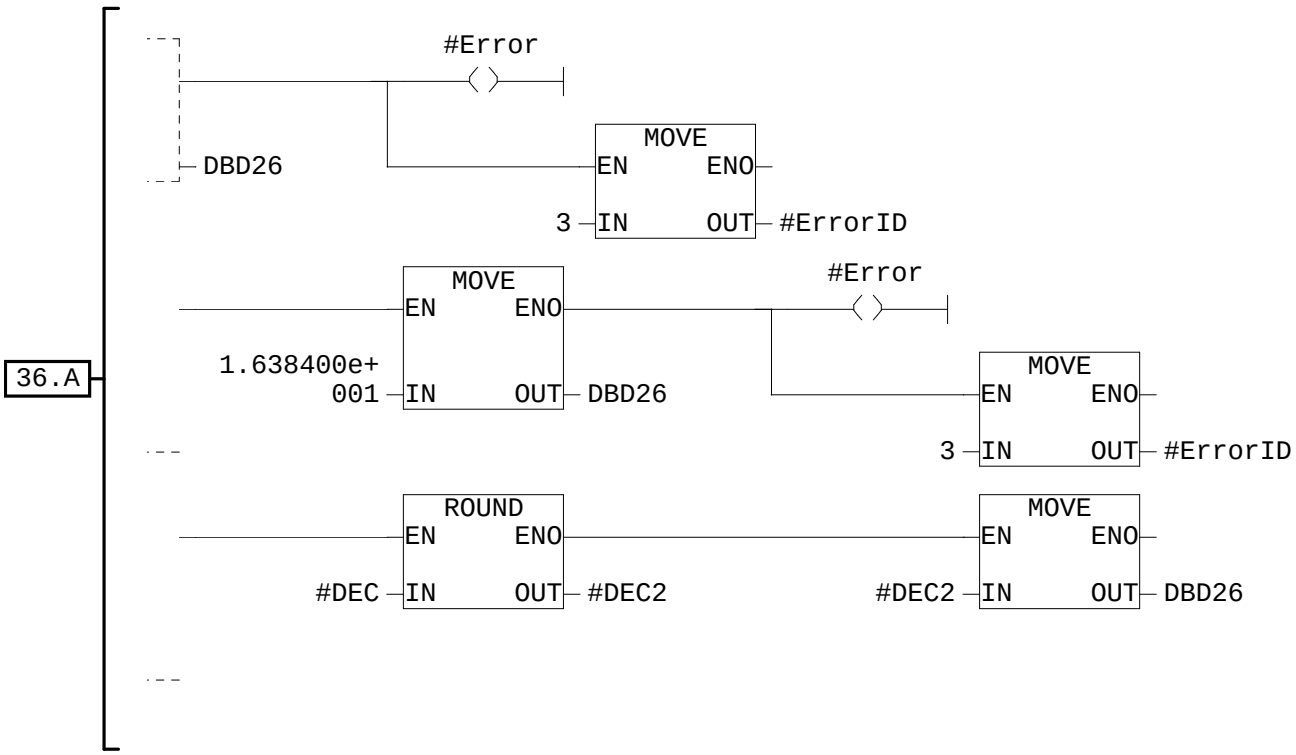




Network: 36 Deceleration

Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

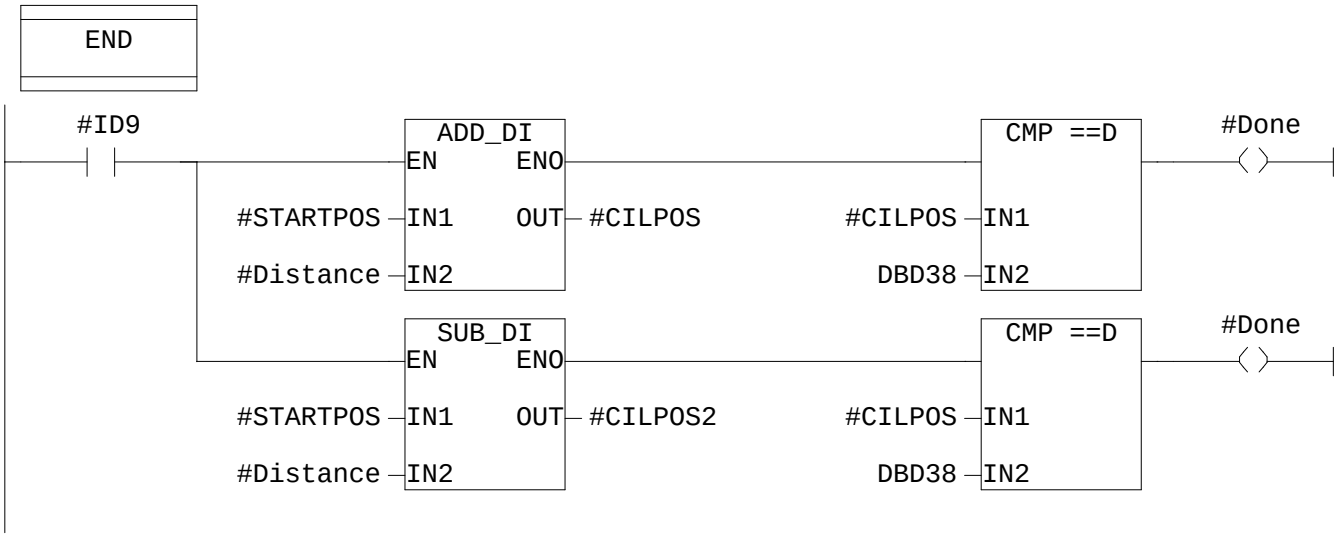




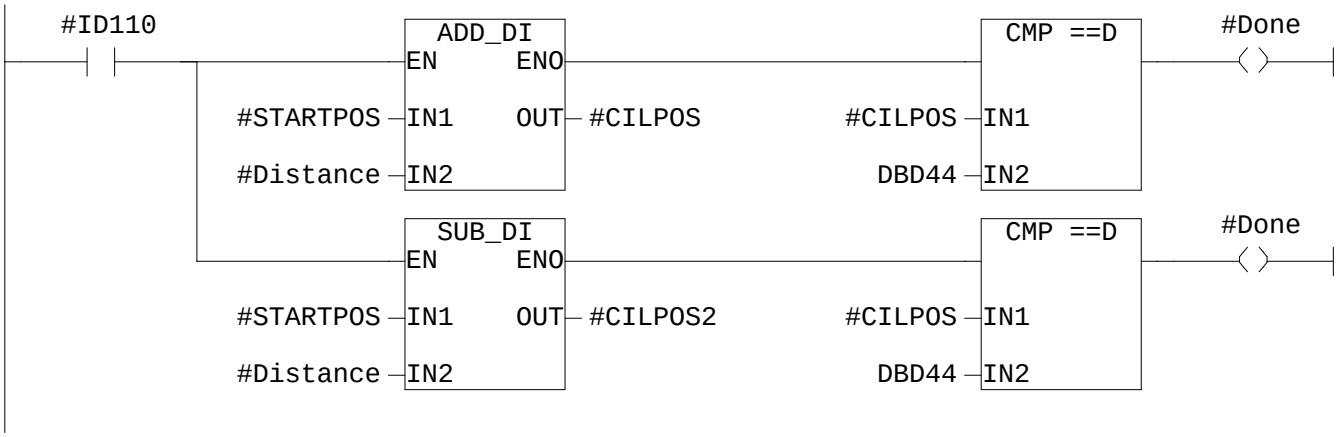
Network: 37 Konec

Network: 38

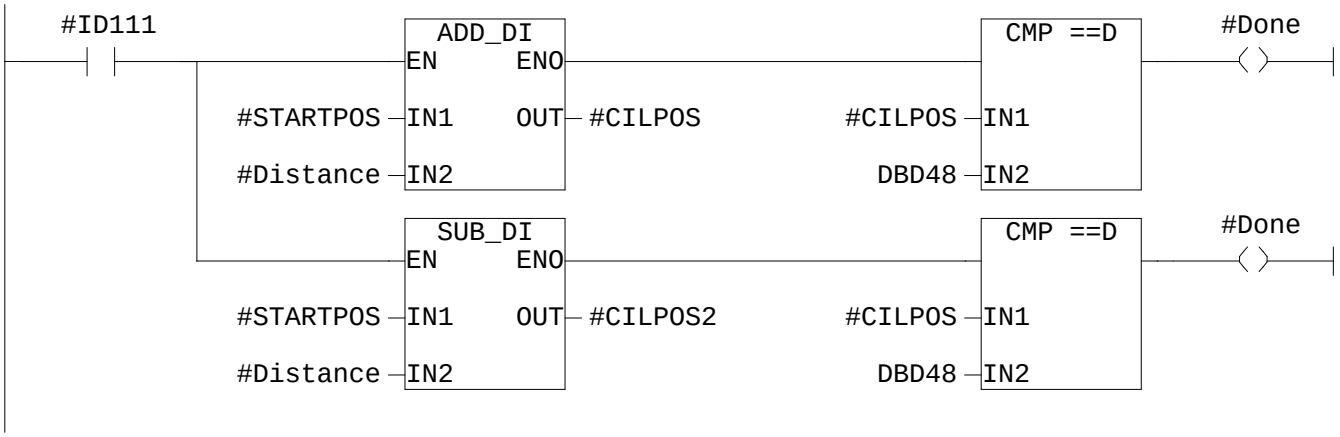
Pokud byla dosazena cilova pozice, je bit Done nastaven na hodnotu TRUE
Prvni vetev - osa se otacela v kladnem smeru
Druha vetev - osa se otacela v zapornem smeru



Network: 39



Network: 40



Network: 41 Konec

A.4 Funkce – Move Velocity

FC3 - <offline>

"FC_MoveVelocity"

Name: **Family:**
Author: Vacek **Version:** 0.1
Block version: 2
Time stamp Code: 05/26/2009 06:16:49 PM
Interface: 05/14/2009 11:50:18 AM
Lengths (block/logic/data): 04704 04240 00042

Name	Data Type	Address	Comment
IN		0.0	
Axis_DB	Block_DB	0.0	Otevre datovy blok pro pozadovany standardni telegram
Execute	Bool	2.0	Start
Velocity	Real	4.0	Rychlost
Acceleraion	Real	8.0	Zrychleni
Deceleration	Real	12.0	Zpomaleni
Current	Bool	16.0	TRUE - vezme se aktualni rychlost osy, FALSE - nastavi se pozadovana rychlost
Direction	Bool	16.1	Smer
OUT		0.0	
Active	Bool	18.0	Funkce je aktivni
Done	Bool	18.1	Rychlost nastavena
Error	Bool	18.2	Chyba
ErrorID	Int	20.0	Identifikace chyby
IN_OUT		0.0	
TEMP		0.0	
HELP	Bool	0.0	
VEL	Real	2.0	
VEL2	DInt	6.0	
ACC	Real	10.0	
ACC2	DInt	14.0	
DEC	Real	18.0	
DEC2	DInt	22.0	
DIR	DInt	26.0	
Cislo	Word	30.0	Cislo DB
Delka	Word	32.0	Delka DB v bytech
Return	Int	34.0	Navratova hodnota (0=OK)
WrPr	Bool	36.0	DB chranen proti prepsani (1=ANO)
Delka2	Int	38.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC3

```
Created:
date 17.04.2009 version 1.0 - Jakub Vacek
-----------------------------------------------------
Funkce MoveVelocity zrychluje nebo zpomaluje osu na pozadovanou rychlost
-----------------------------------------------------
Telegram: 1,2,9,110,111
-----------------------------------------------------
Identifikace chyby:
ErrorID = 1, spatne zadana rychlost
ErrorID = 2, spatne zadane zrychleni
ErrorID = 3, spatne zadane zpomaleni
ErrorID = 4, nebyl nalezen kompatibilni telegram
```

Network: 1 Otevri DB

Funkce otevre datovy blok pro pozadovany standardni telegram

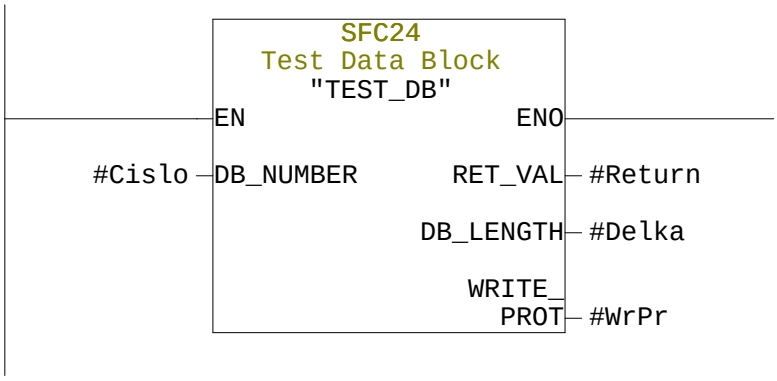
Network: 2

Do pomocne promenne Cislo ulozi cislo DB

```
      L     DBNO
      T     #Cislo
```

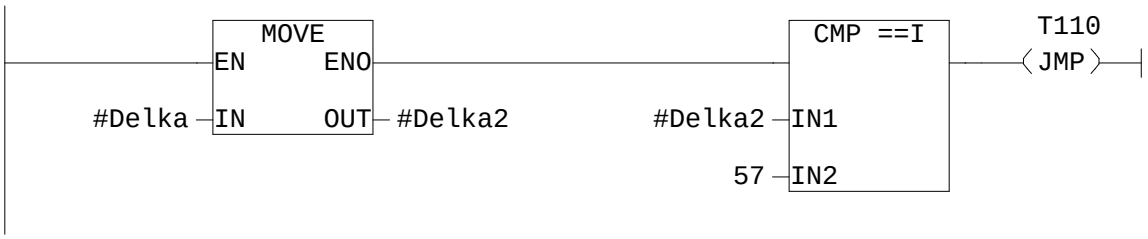
Network: 3

Zjisteni delky datoveho bloku

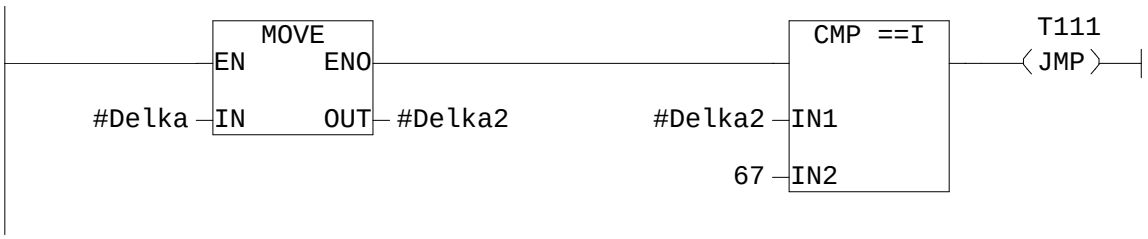


Network: 4 Je pouzit telegram 9?

Network: 5 Je pouzit telegram 110?



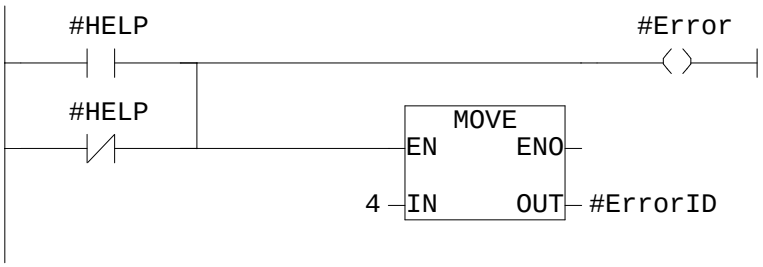
Network: 6 Je pouzit telegram 111?



Network: 7 Je pouzit telegram 1?

Network: 8 Je pouzit telegram 2?

Network: 9 Nebyl pouzit vhodny telegram



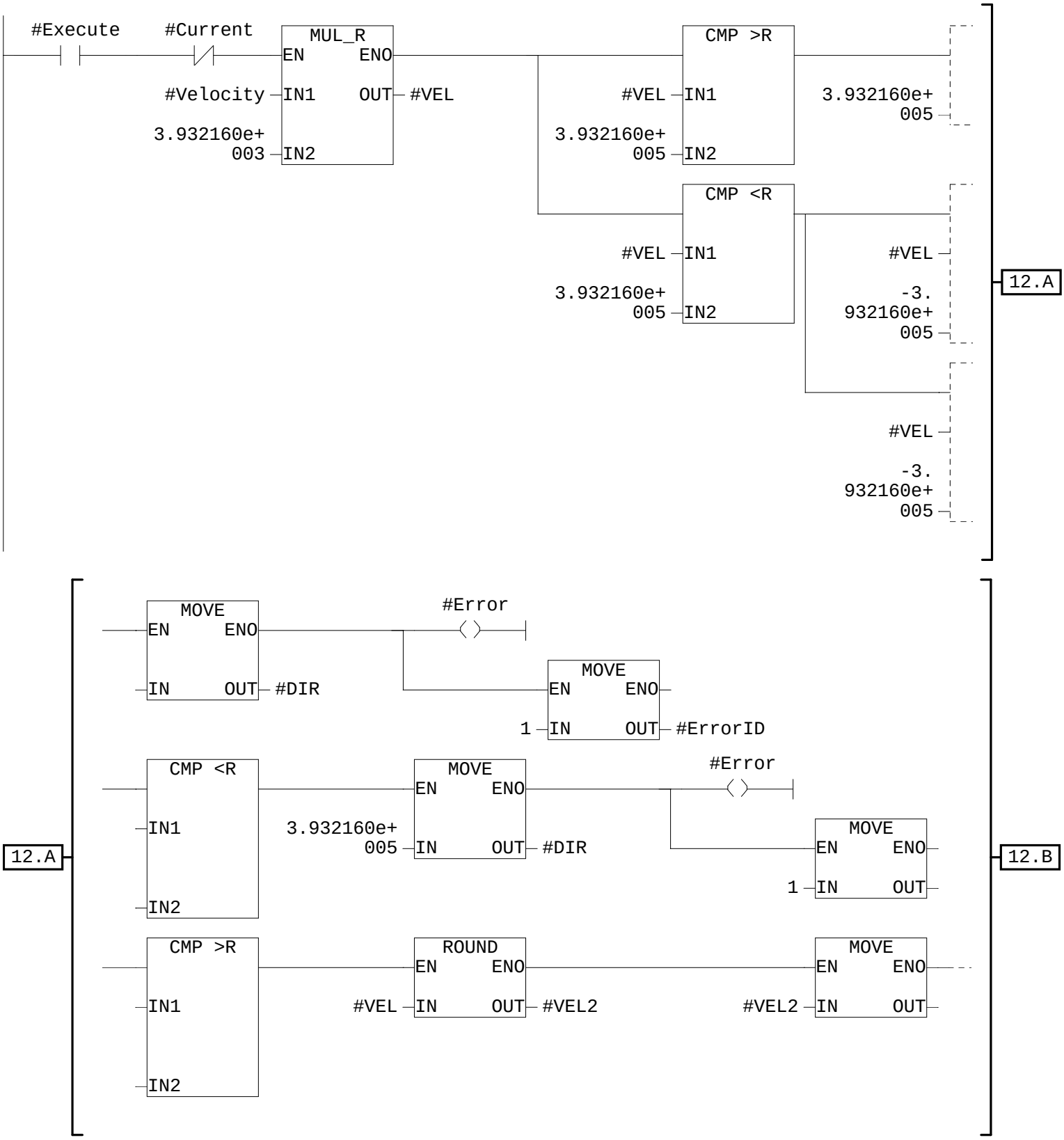
Network: 10 Konec

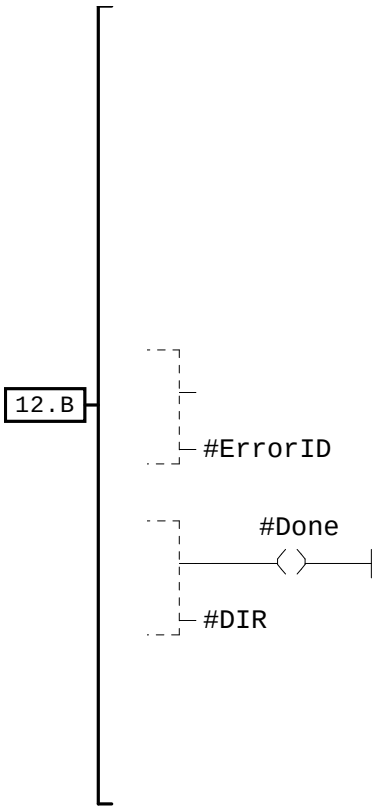
Network: 11 Telegram 9

Network: 12 Velocity

Nastaveni rychlosti v procentech (0-100)
Bit Current = false

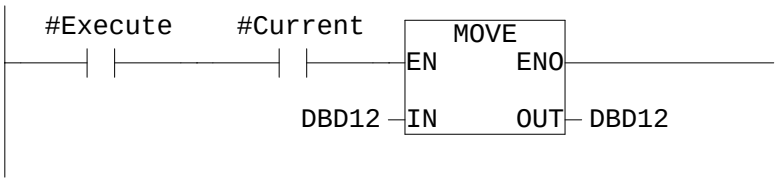
Standardni normalizace v menicich je 1 hex = 1000 LU/min, maximalni rychlost udava parametr p2000 (6000 rev/min)
Pri nastaveni 10000 LU per load revolution je 60000 hex = 100%
Provadi se omezeni na hodnoty double integer





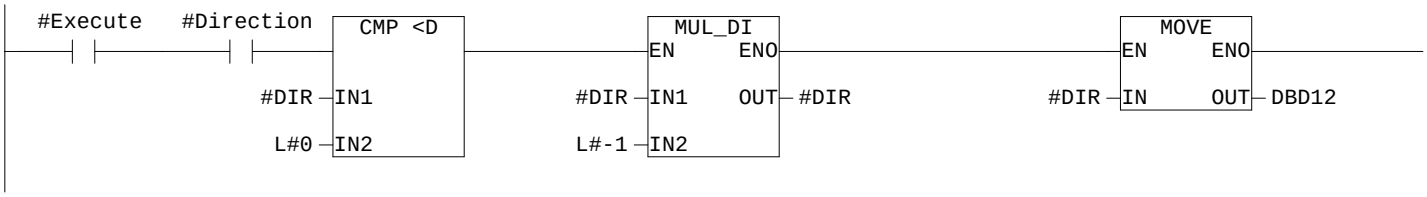
Network: 13 Velocity

Bit Current = true
Bude pouzita aktualni, posledne nastavena, rychlost osy



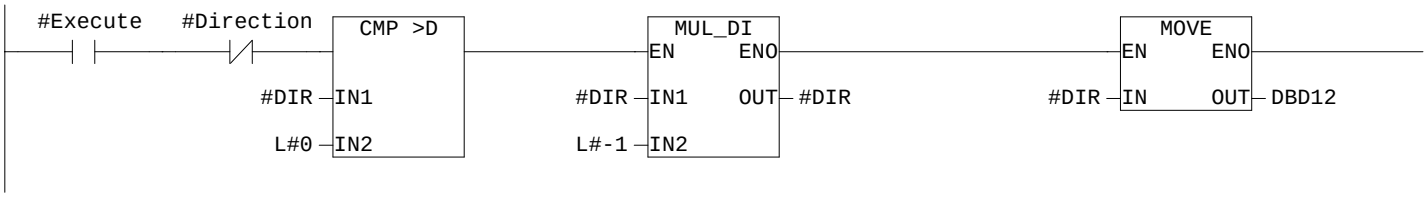
Network: 14 Direction

Smer otaceni nastaven na Forward
Direction = True



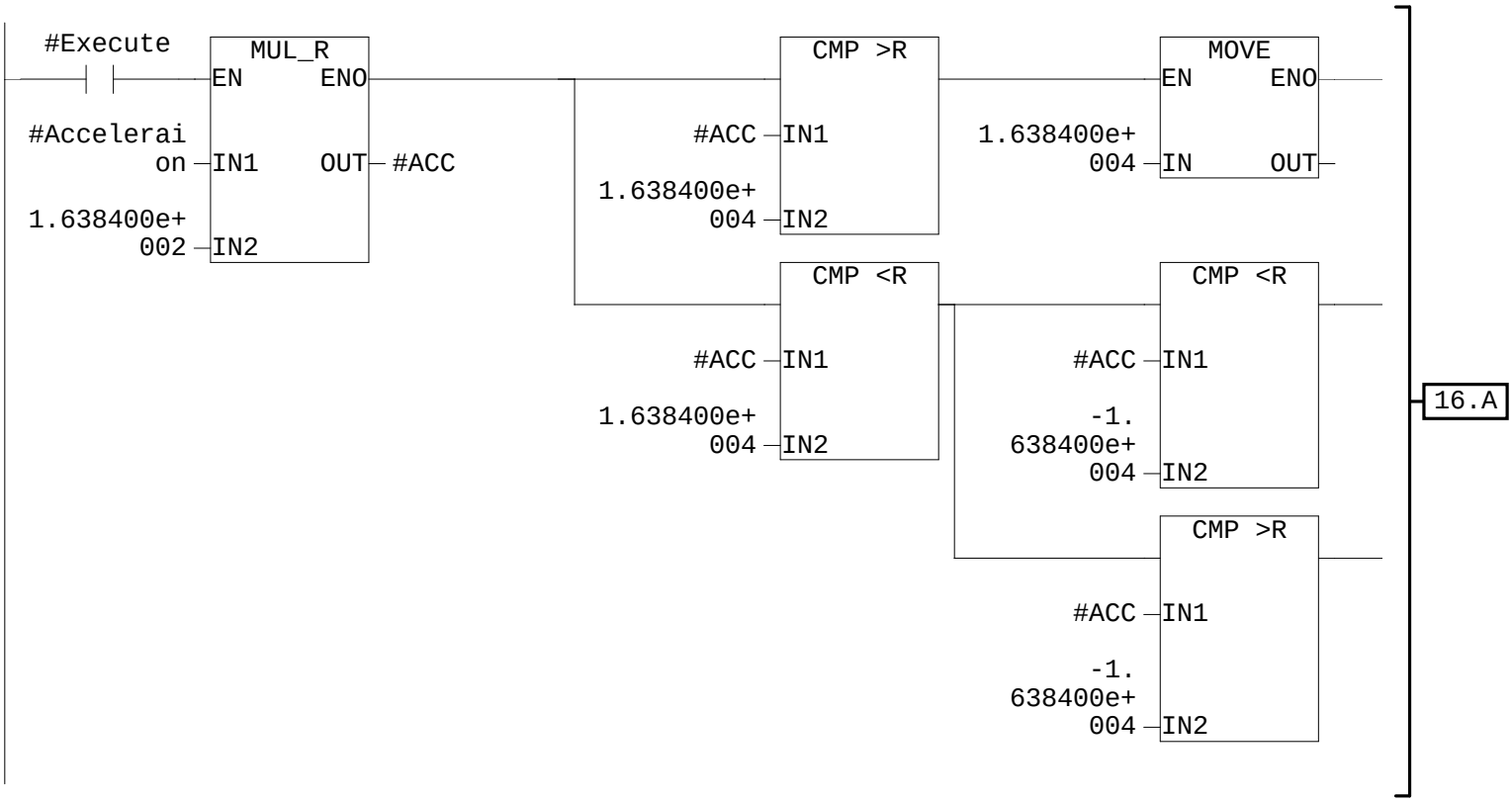
Network: 15 Direction

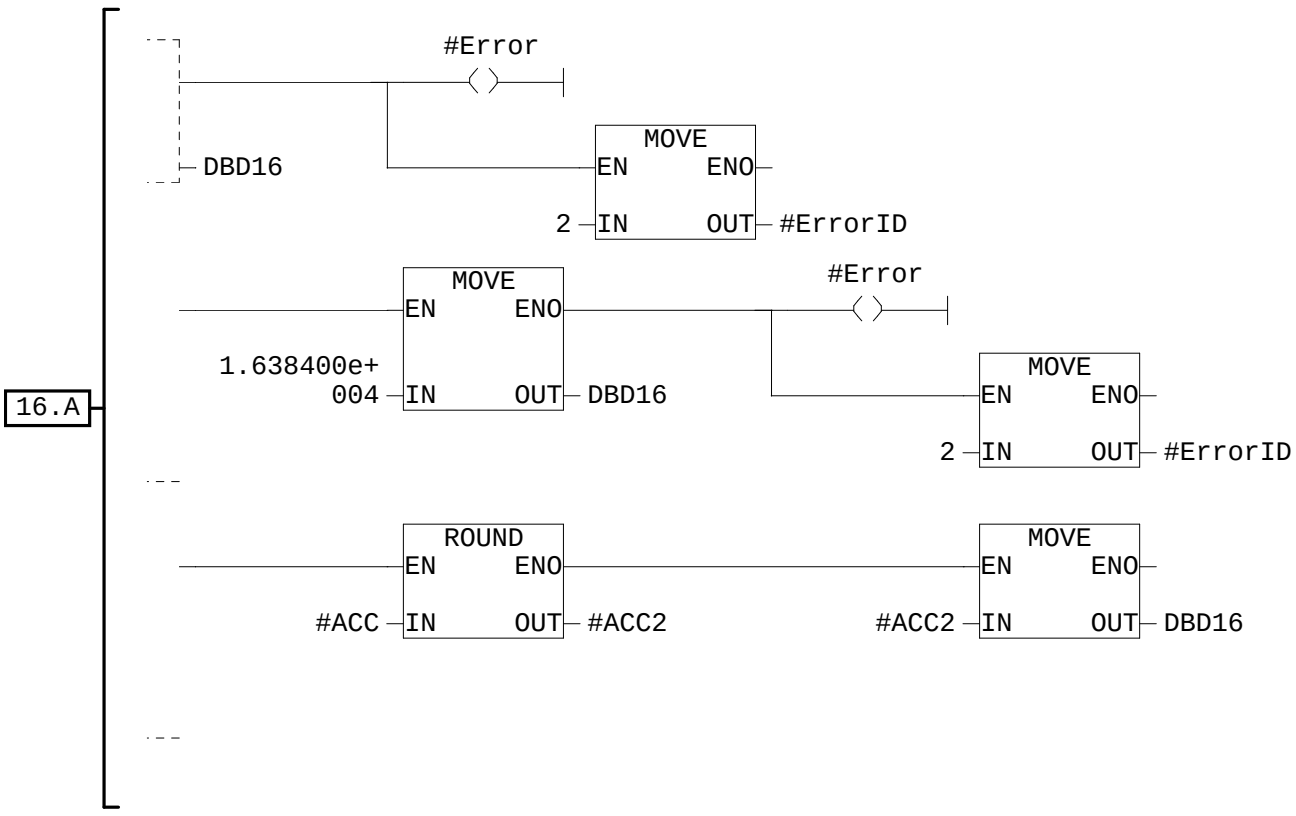
Smer otaceni nastaven na Backward
Direction = False



Network: 16 Acceleration

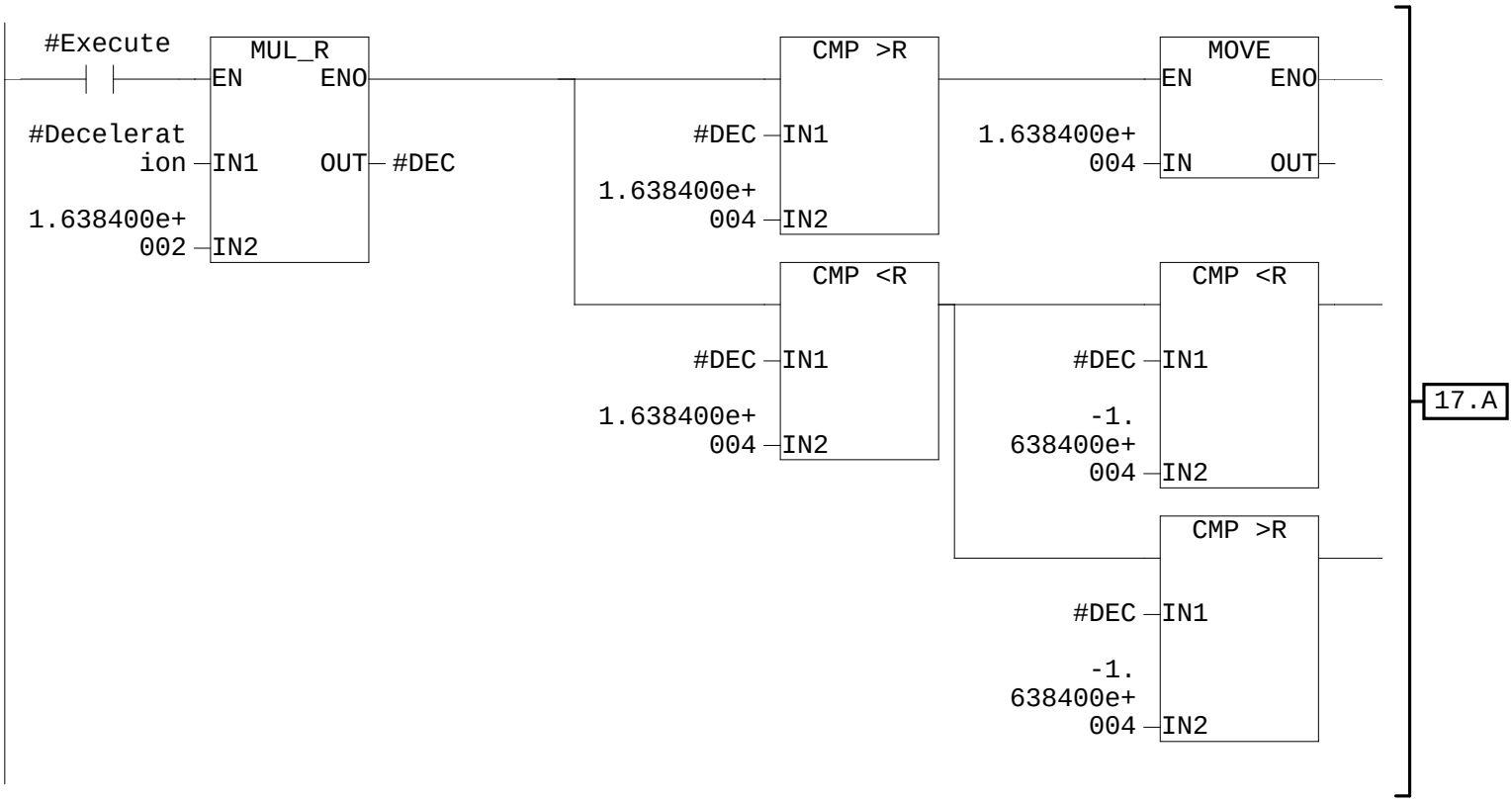
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

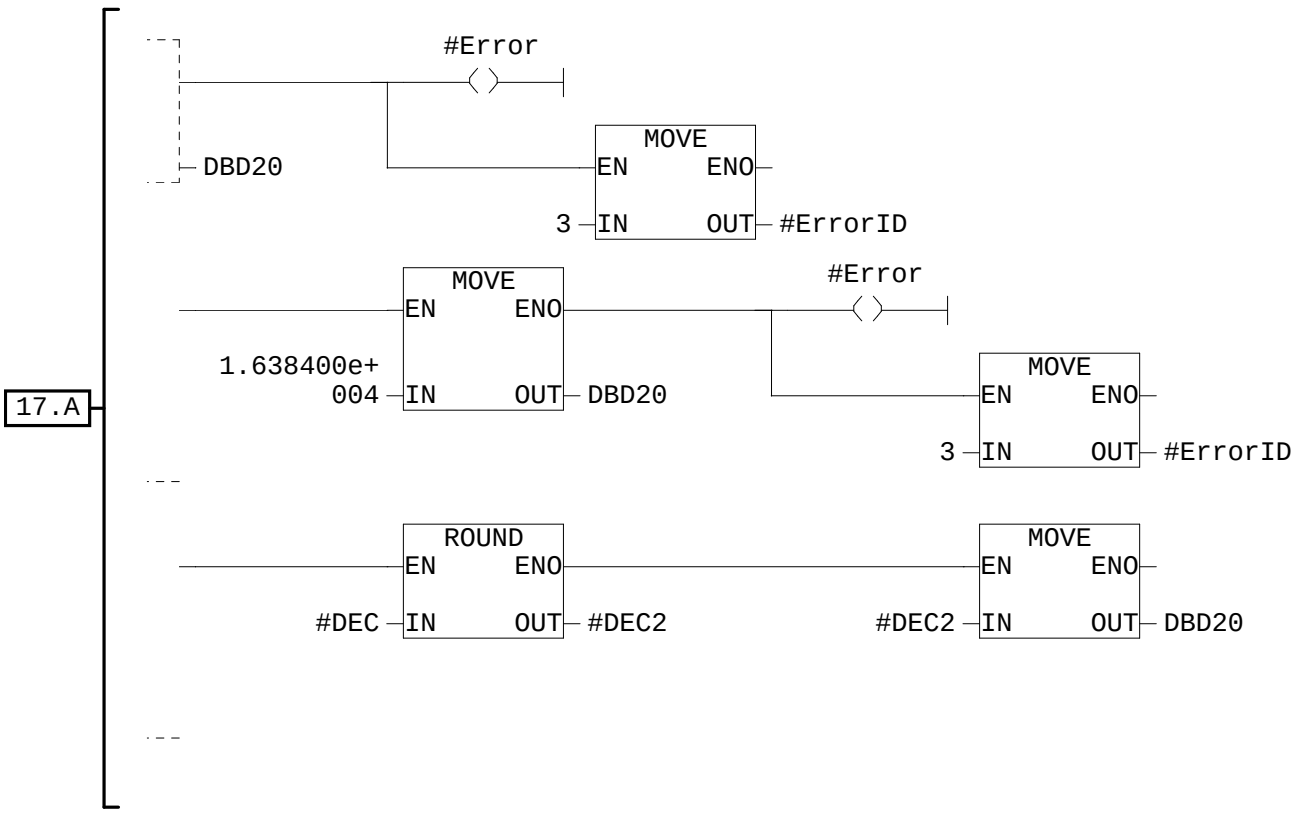




Network: 17 Deceleration

Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer



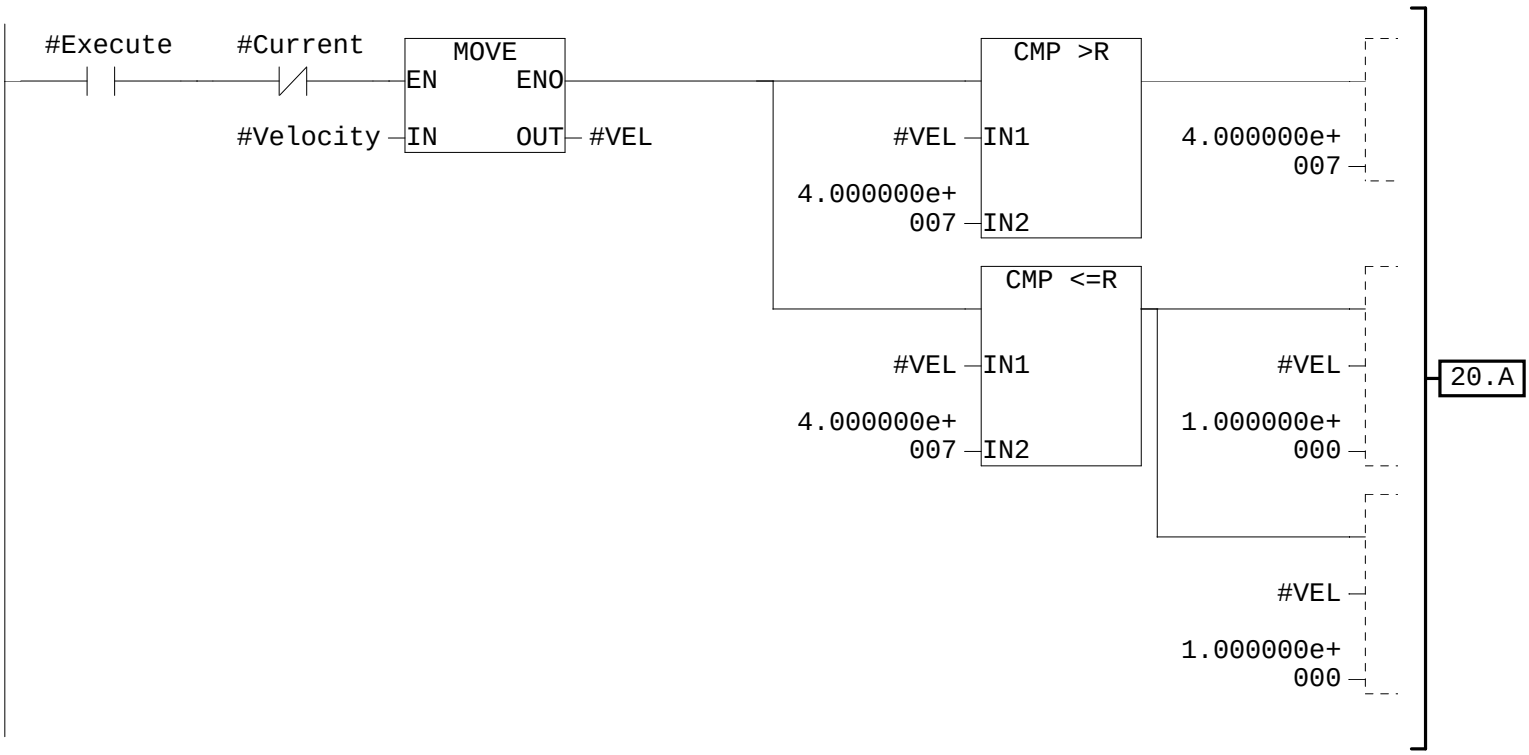


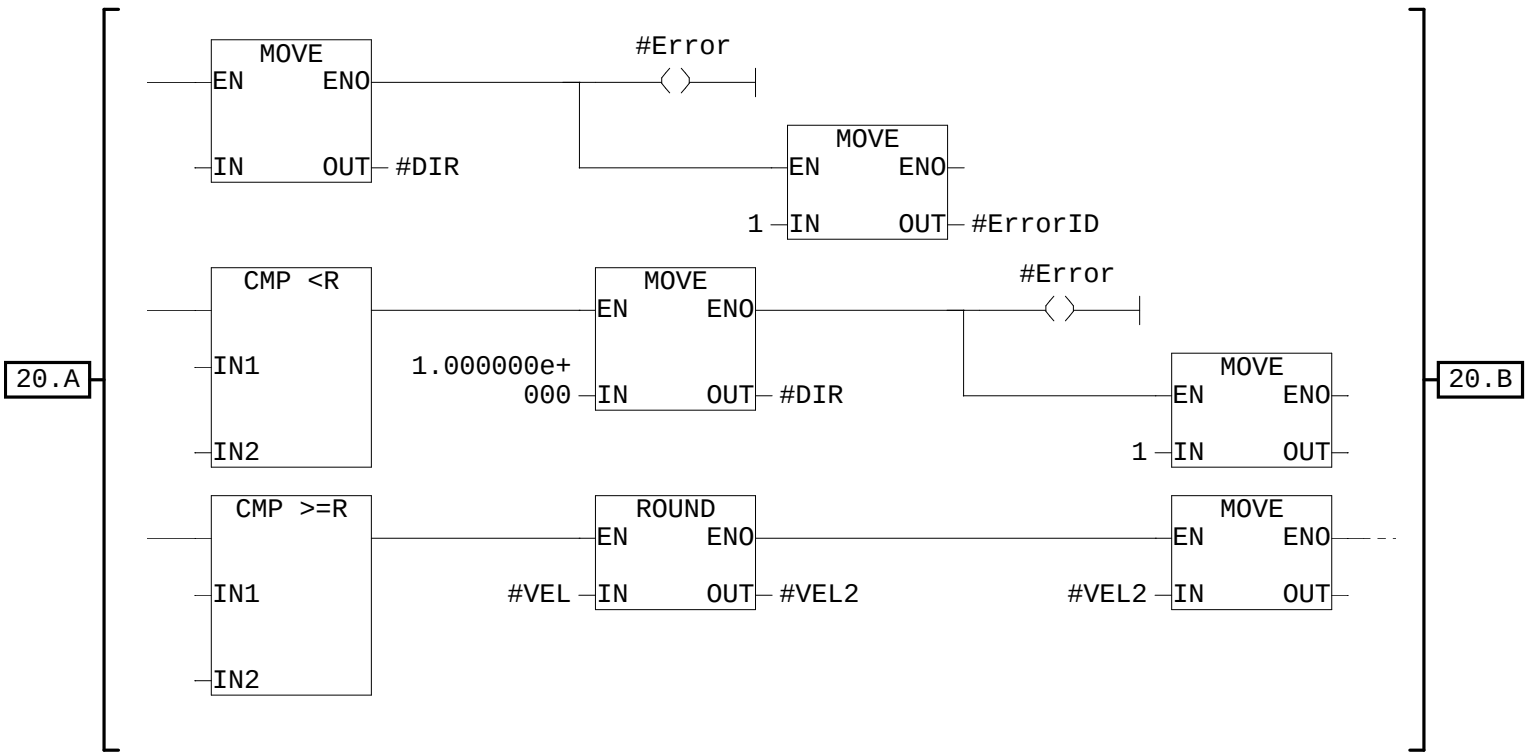
Network: 18 Konec

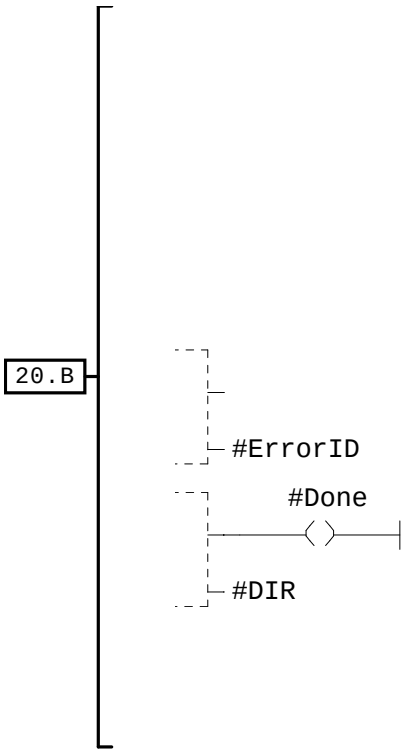
Network: 19 Telegram 110

Network: 20 Velocity

Nastaveni rychlosti
Regulovatelna v rozmezi 1 - 40000000
Standardni normalizace v menicich je 1 hex = 1000 LU/min, maximalni rychlost
udava parametr p2571
Provadi se omezeni na hodnoty double integer

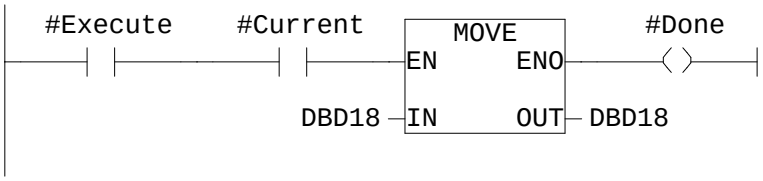






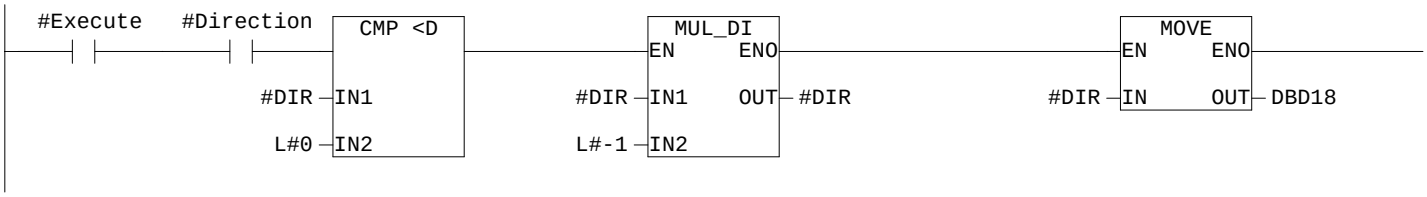
Network: 21 Velocity

Bit Current = true
Bude pouzita aktualni, posledne nastavena, rychlost osy



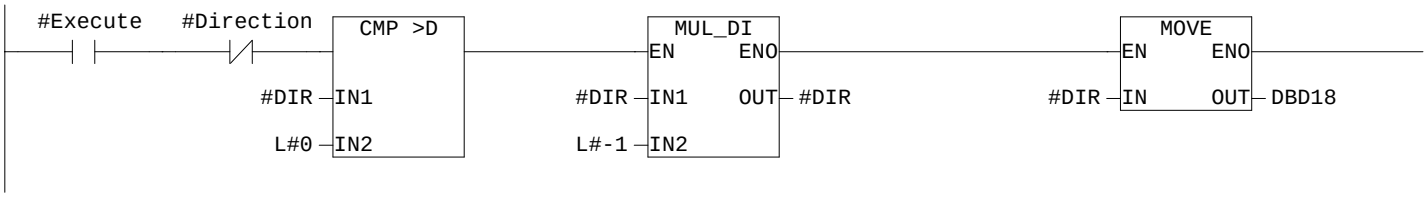
Network: 22 Direction

Smer otaceni nastaven na Forward
Direction = True



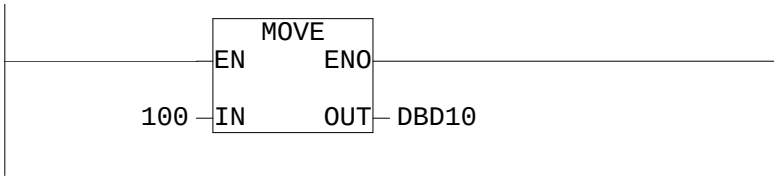
Network: 23 Direction

Smer otaceni nastaven na Backward
Direction = False



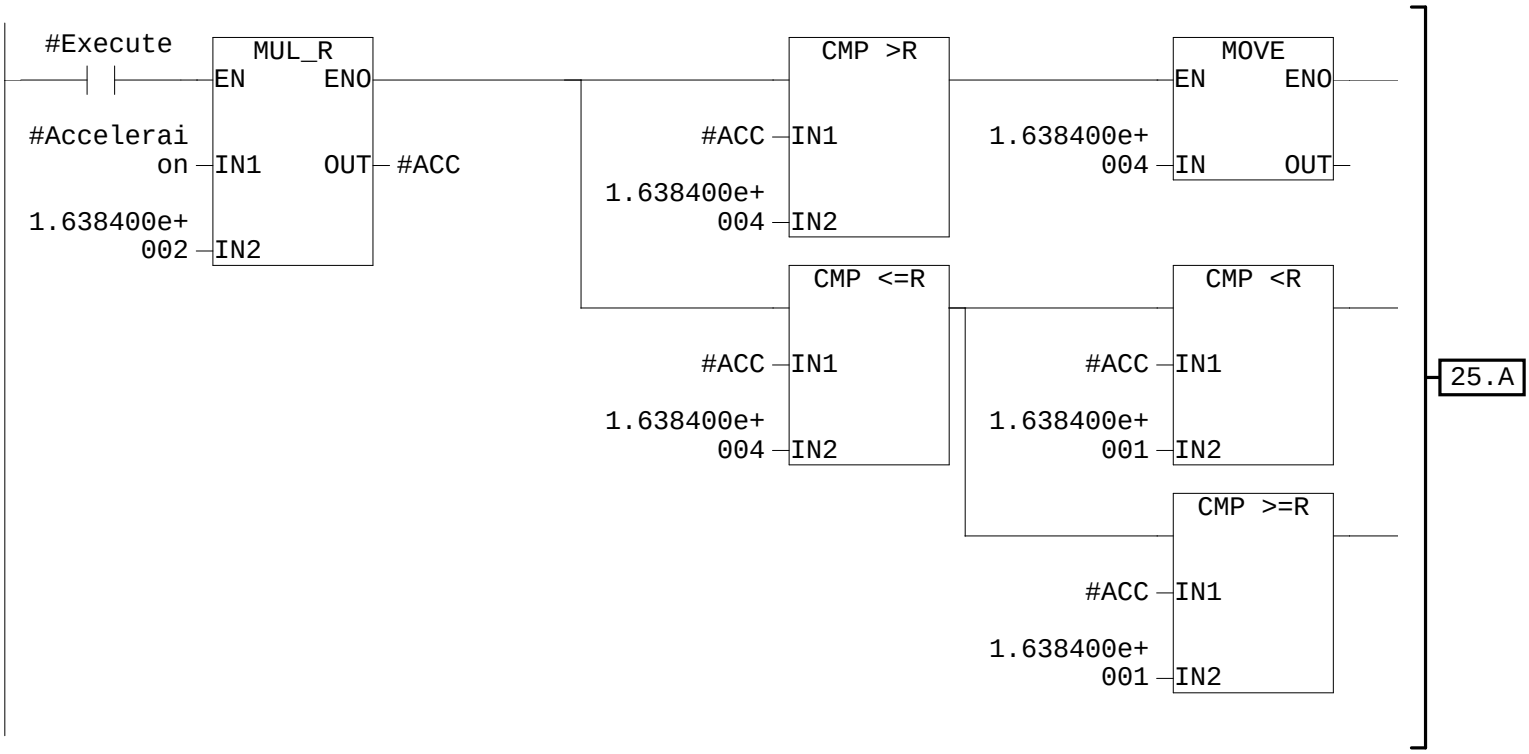
Network: 24 Override

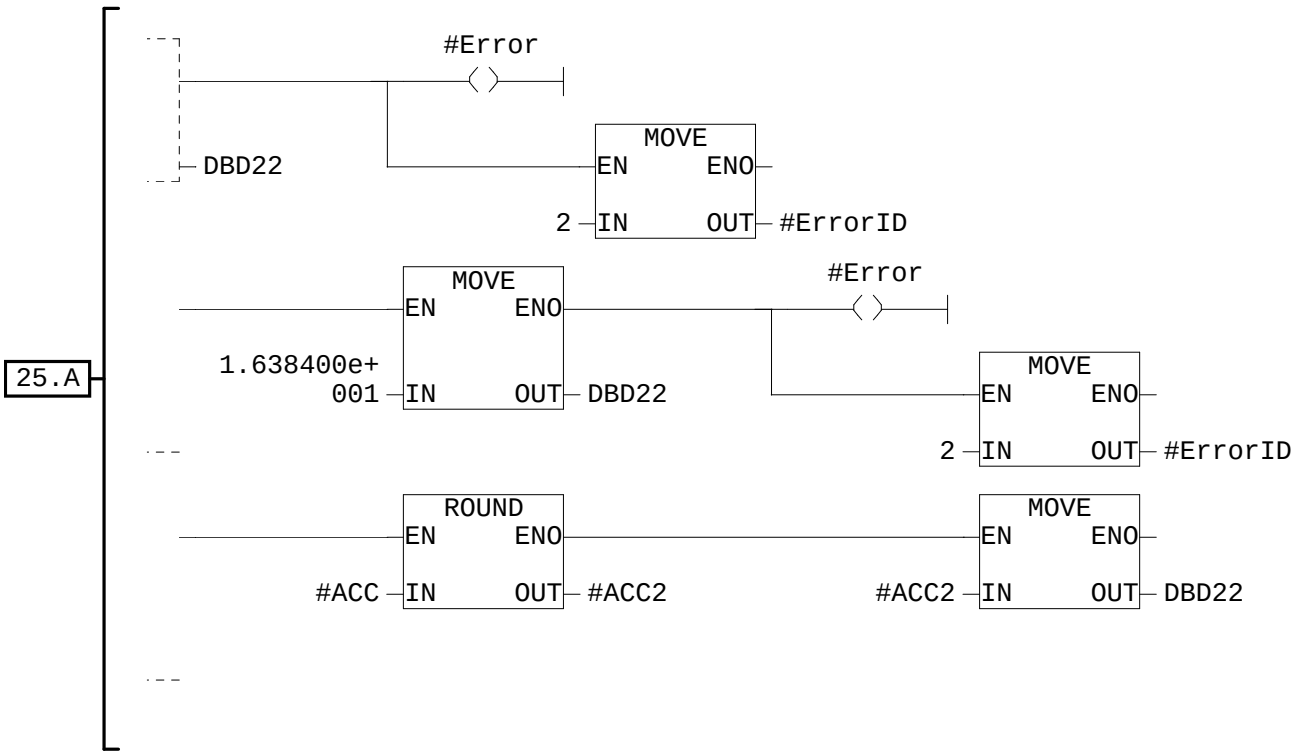
Nastaveni Override pro rychlost - 100%



Network: 25 Acceleration

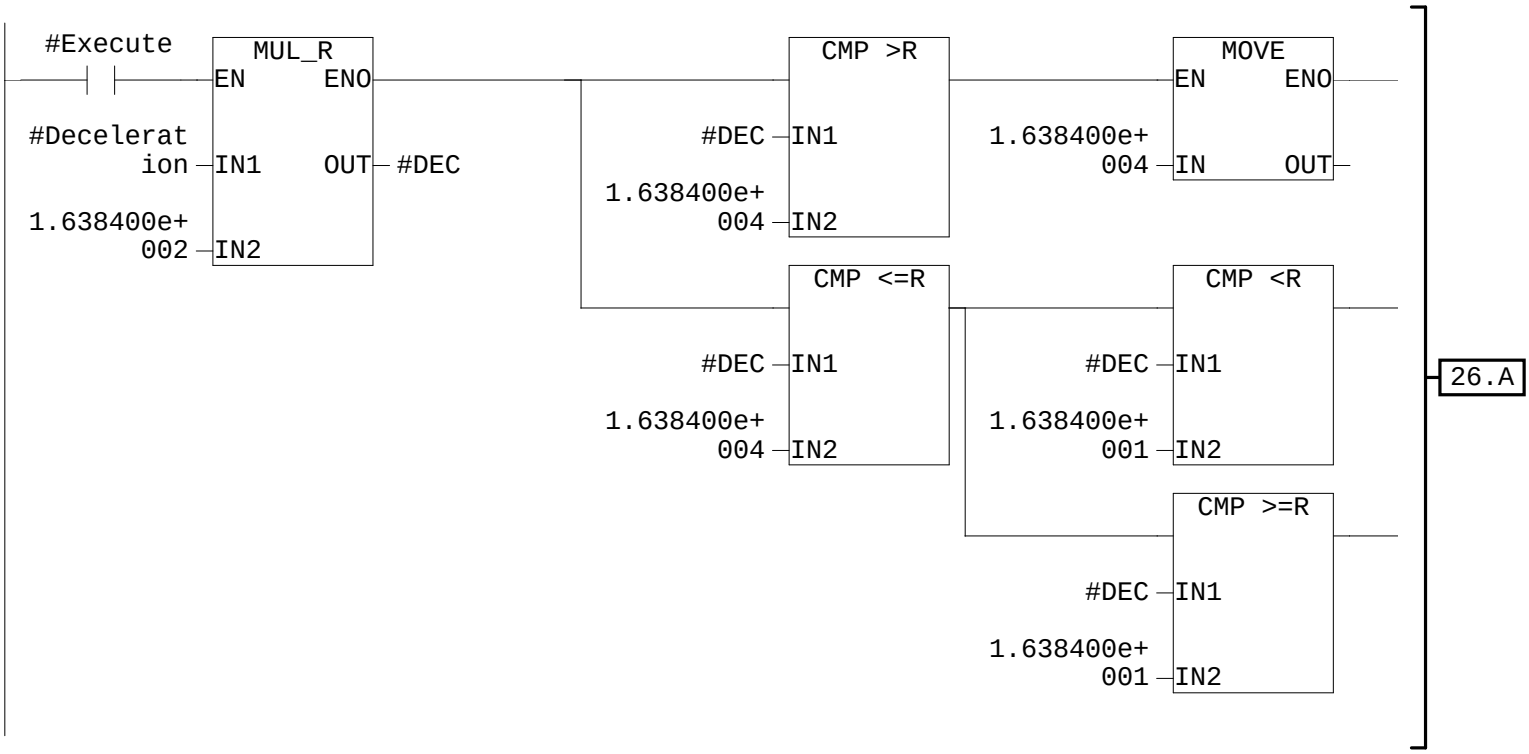
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

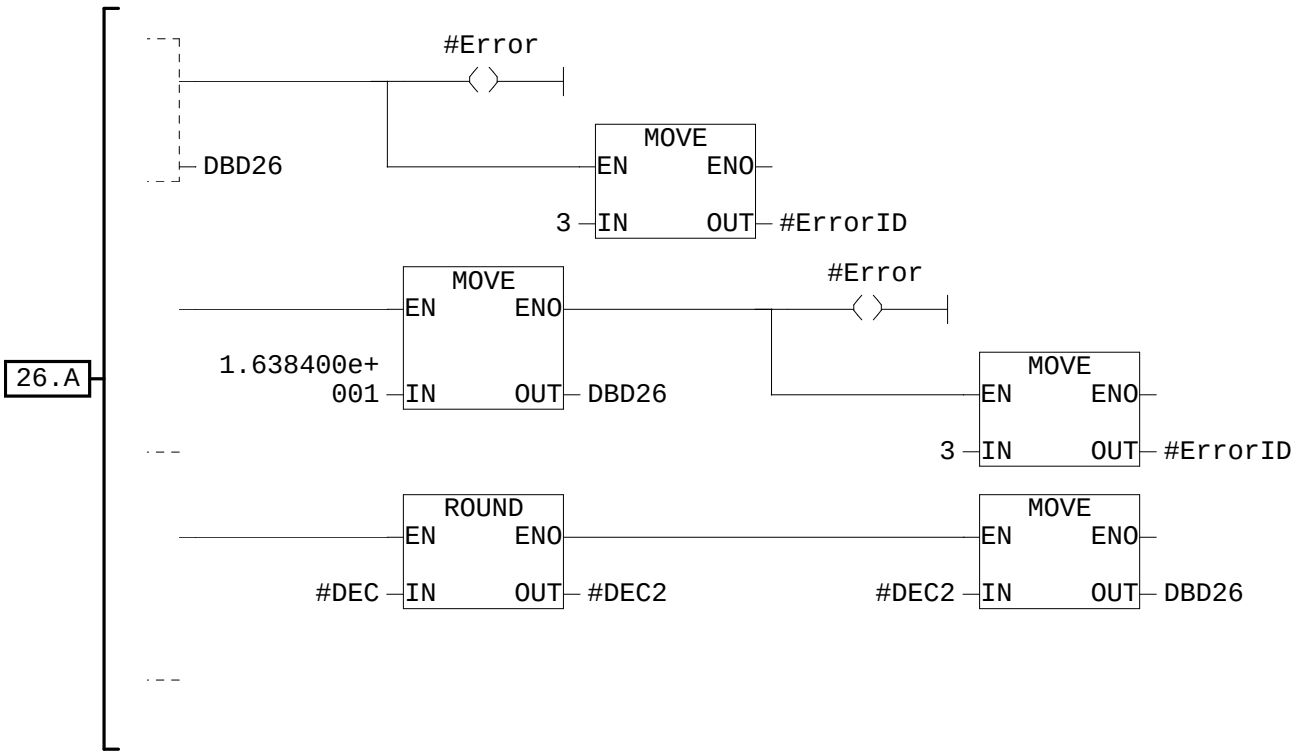




Network: 26 Deceleration

Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer



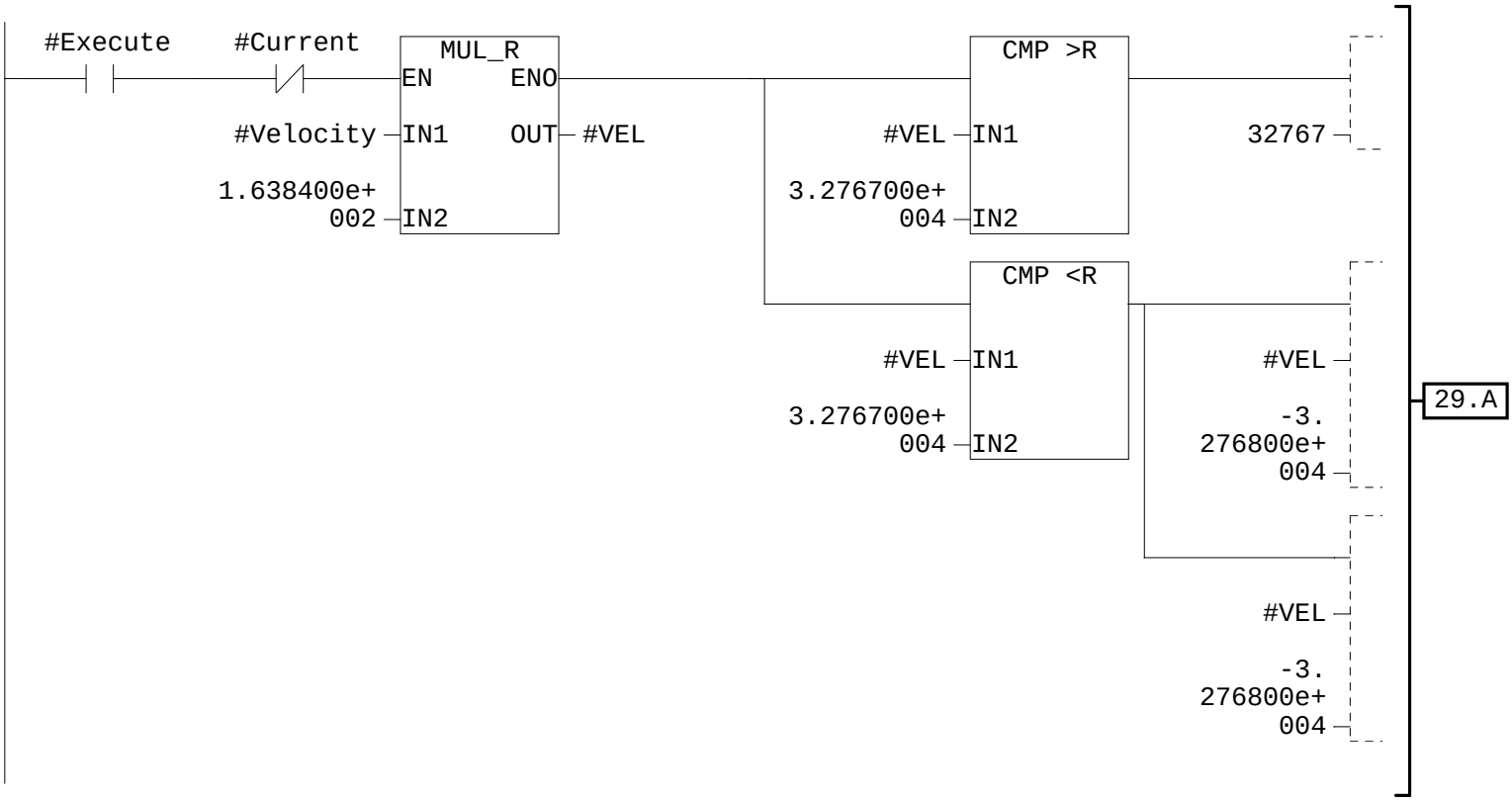


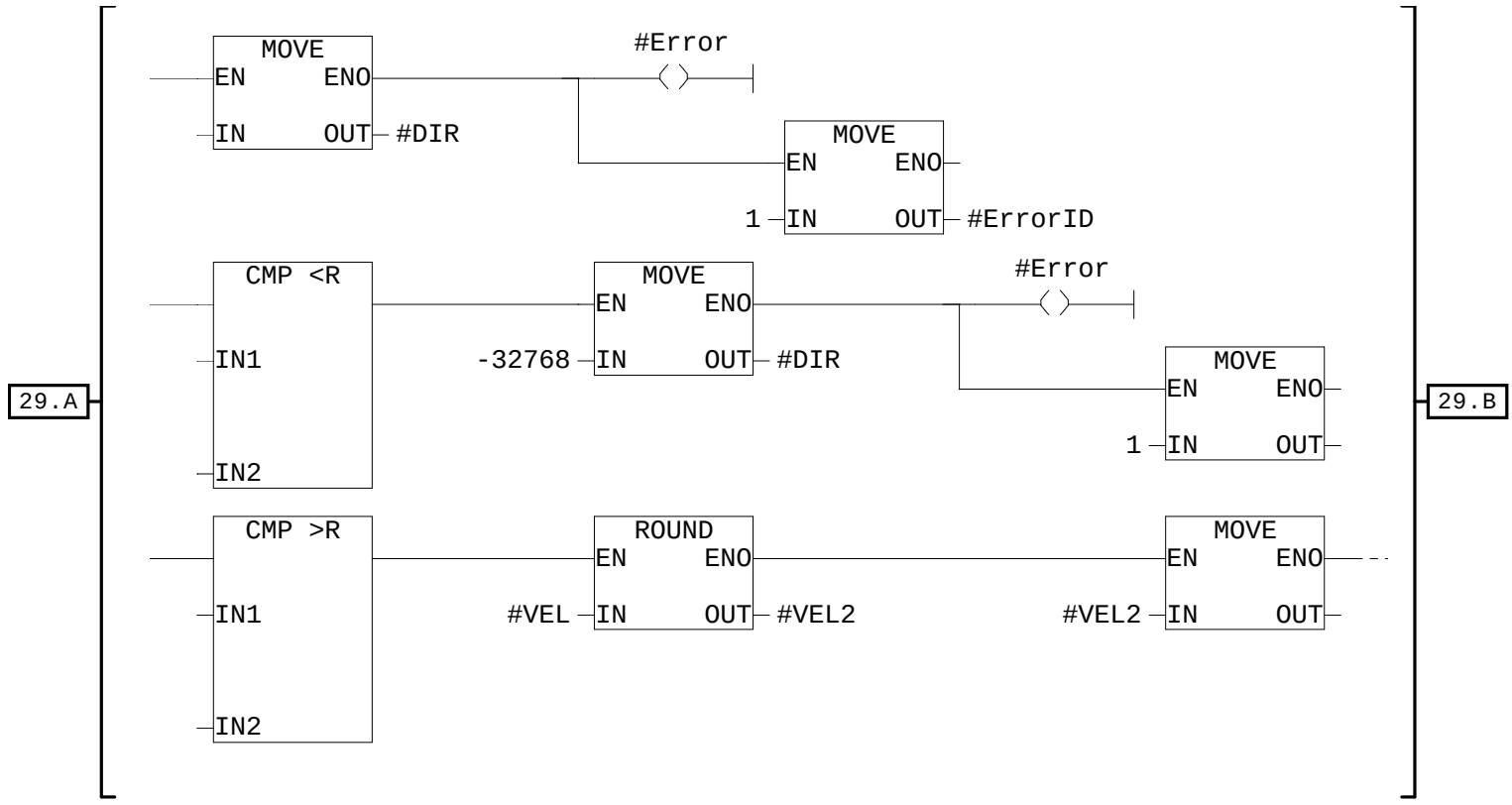
Network: 27 Konec

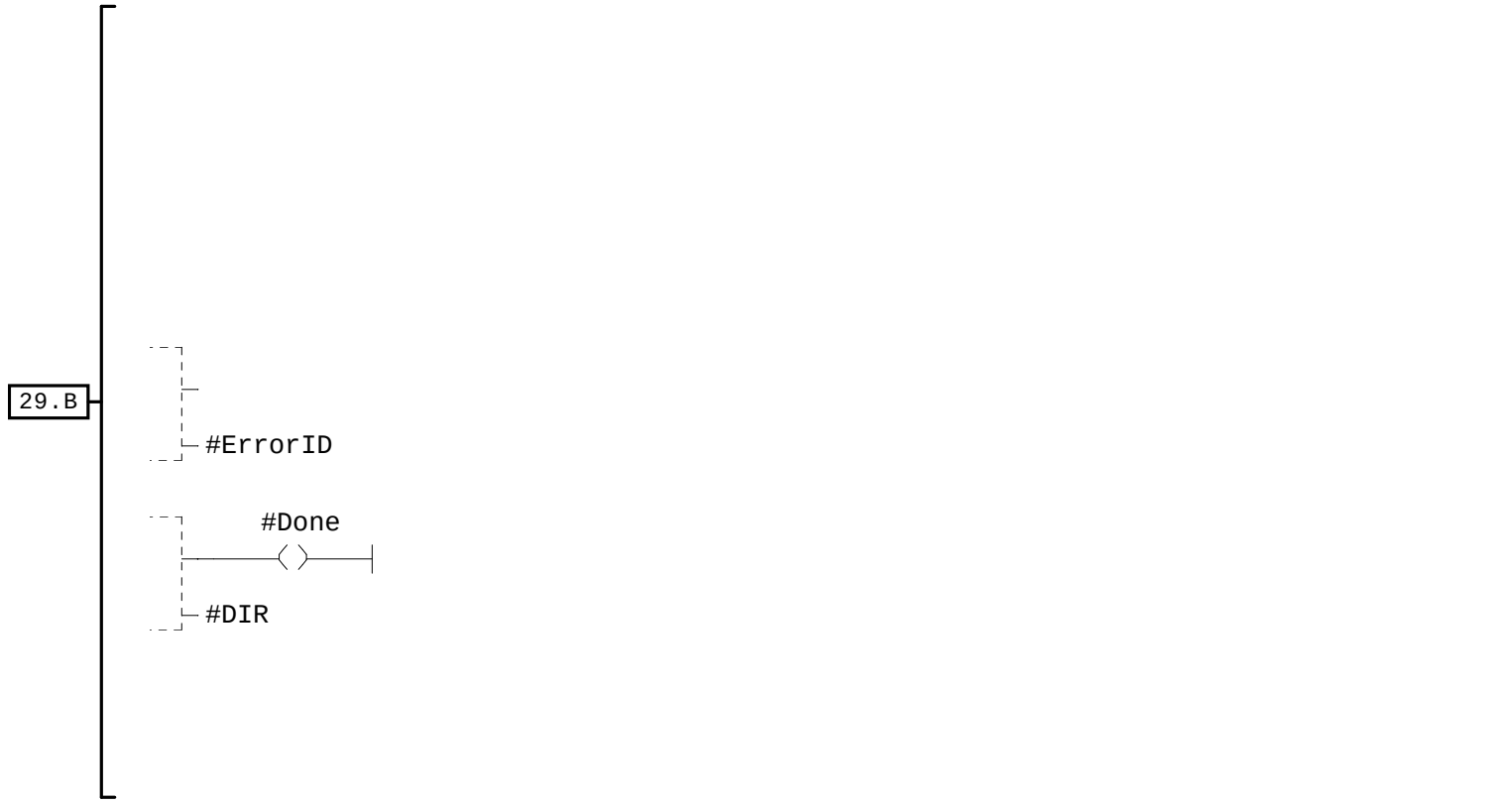
Network: 28 Telegram 1

Network: 29 Velocity

Nastaveni rychlosti
Standardni normalizace v menicich je 4000H (16384) => 100%, maximalni rychlost
udava parametr p2000
Provadi se omezeni na hodnoty double integer

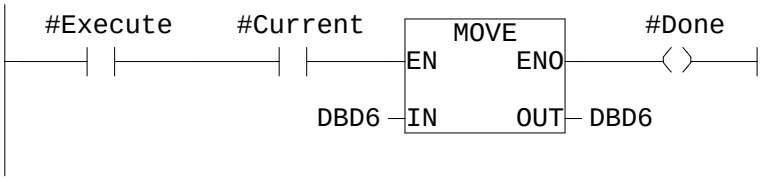



MOVE
EN
ENO
IN
OUT
#DIR
#Error
MOVE
EN
ENO
1
IN
OUT
#ErrorID
CMP <R
IN1
IN2
MOVE
EN
ENO
-32768
IN
OUT
#DIR
#Error
MOVE
EN
ENO
1
IN
OUT
29.A
29.B
CMP >R
IN1
IN2
ROUND
EN
ENO
#VEL
IN
OUT
#VEL2
MOVE
EN
ENO
#VEL2
IN
OUT


29.B
#ErrorID
#Done
#DIR

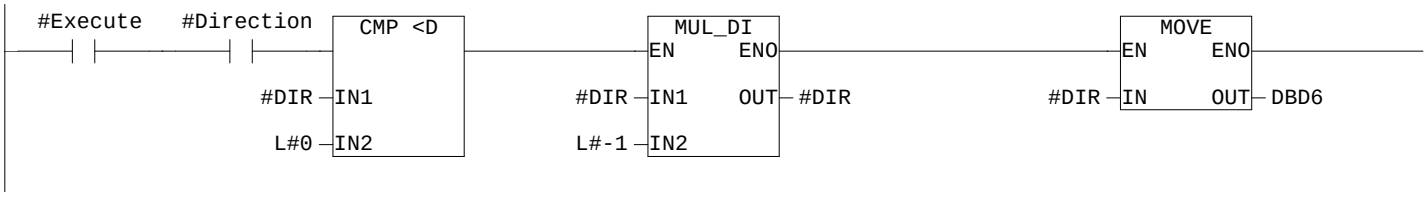
Network: 30 Velocity

Bit Current = true
Bude pouzita aktualni, posledne nastavena, rychlost osy



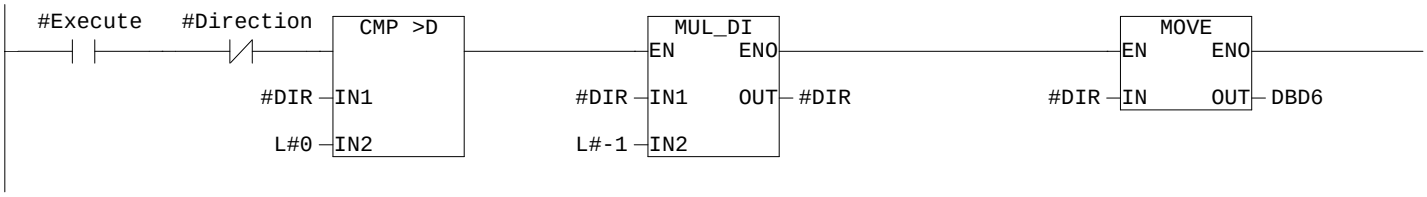
Network: 31 Direction

Smer otaceni nastaven na Forward
Direction = True



Network: 32 Direction

Smer otaceni nastaven na Backward
Direction = False

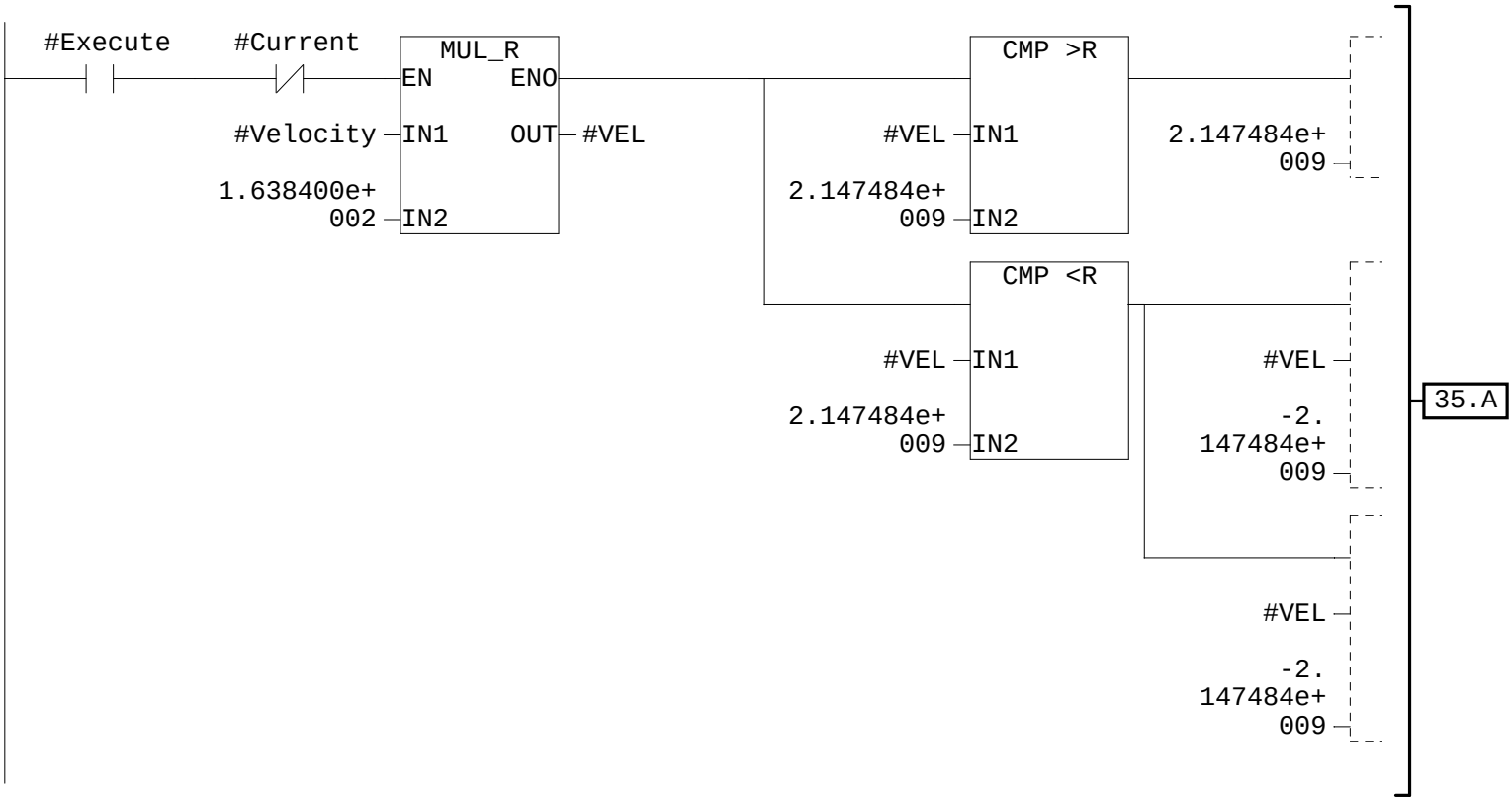


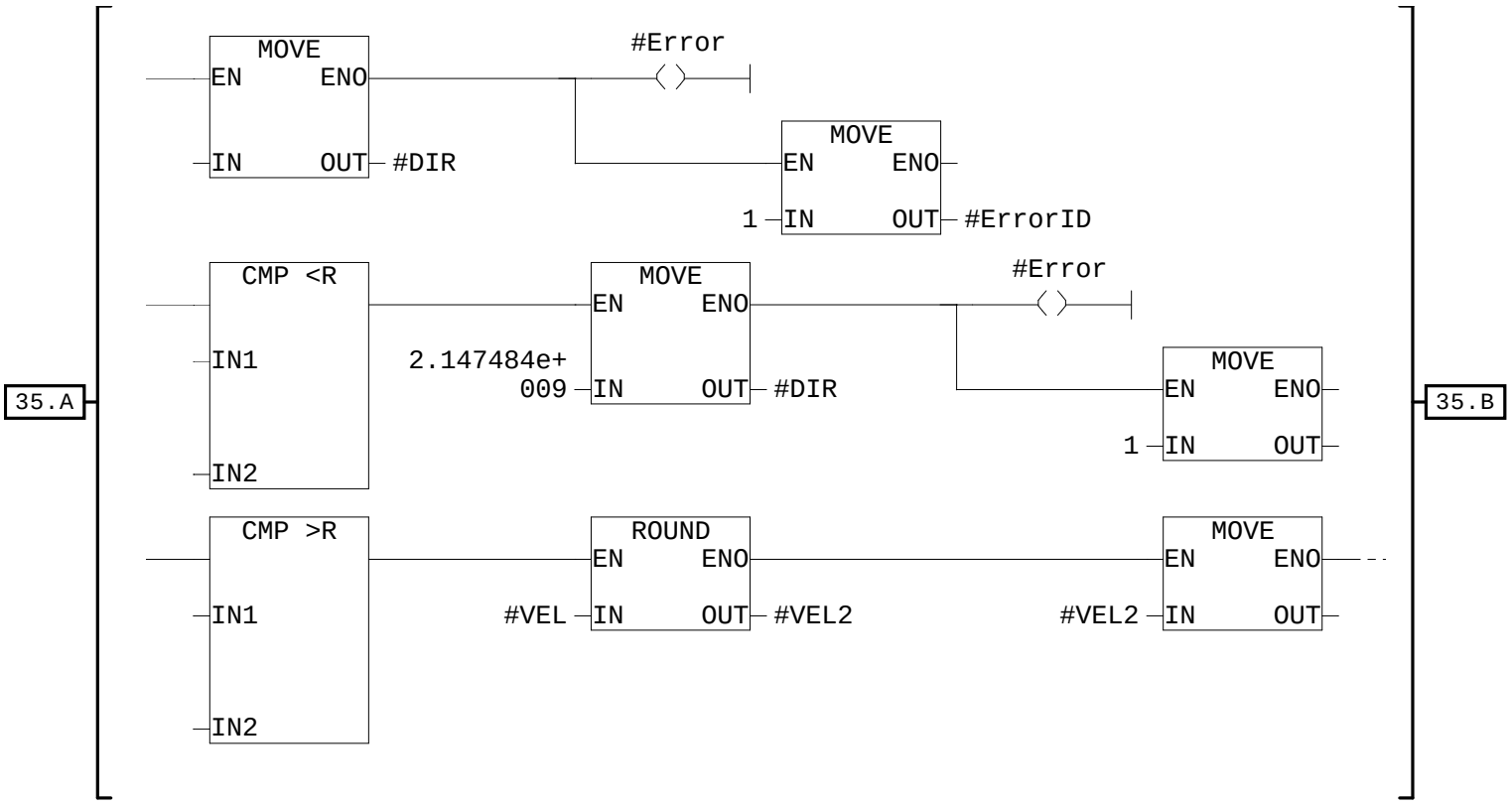
Network: 33 Konec

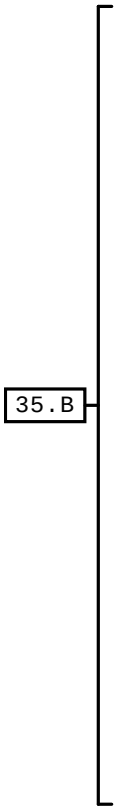
Network: 34 Telegram 2

Network: 35 Velocity

Nastaveni rychlosti
Standardni normalizace v menicich je 4000H (16384) => 100%, maximalni rychlost
udava parametr p2000
Provadi se omezeni na hodnoty double integer

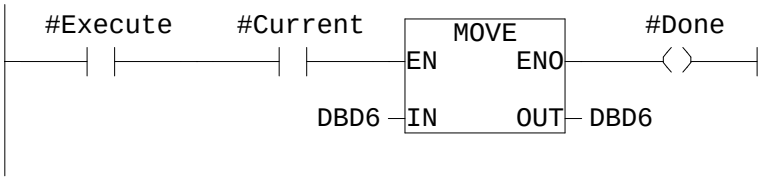



MOVE
EN
ENO
IN
OUT
#DIR
#Error
MOVE
EN
ENO
1
IN
OUT
#ErrorID
CMP <R
IN1
IN2
MOVE
EN
ENO
2.147484e+
009
IN
OUT
#DIR
#Error
MOVE
EN
ENO
1
IN
OUT
35.A
35.B
CMP >R
IN1
IN2
ROUND
EN
ENO
#VEL
IN
OUT
#VEL2
MOVE
EN
ENO
#VEL2
IN
OUT


35.B
#ErrorID
#Done
#DIR

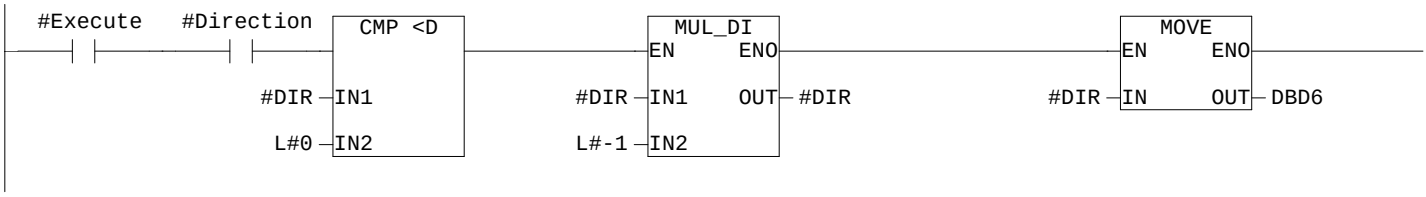
Network: 36 Velocity

Bit Current = true
Bude pouzita aktualni, posledne nastavena, rychlost osy



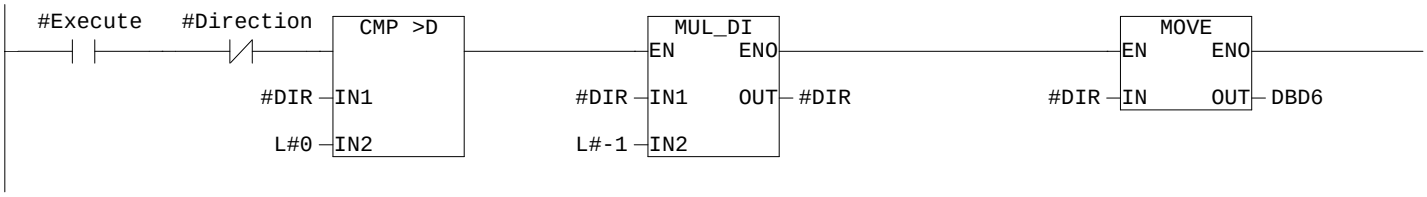
Network: 37 Direction

Smer otaceni nastaven na Forward
Direction = True



Network: 38 Direction

Smer otaceni nastaven na Backward
Direction = False

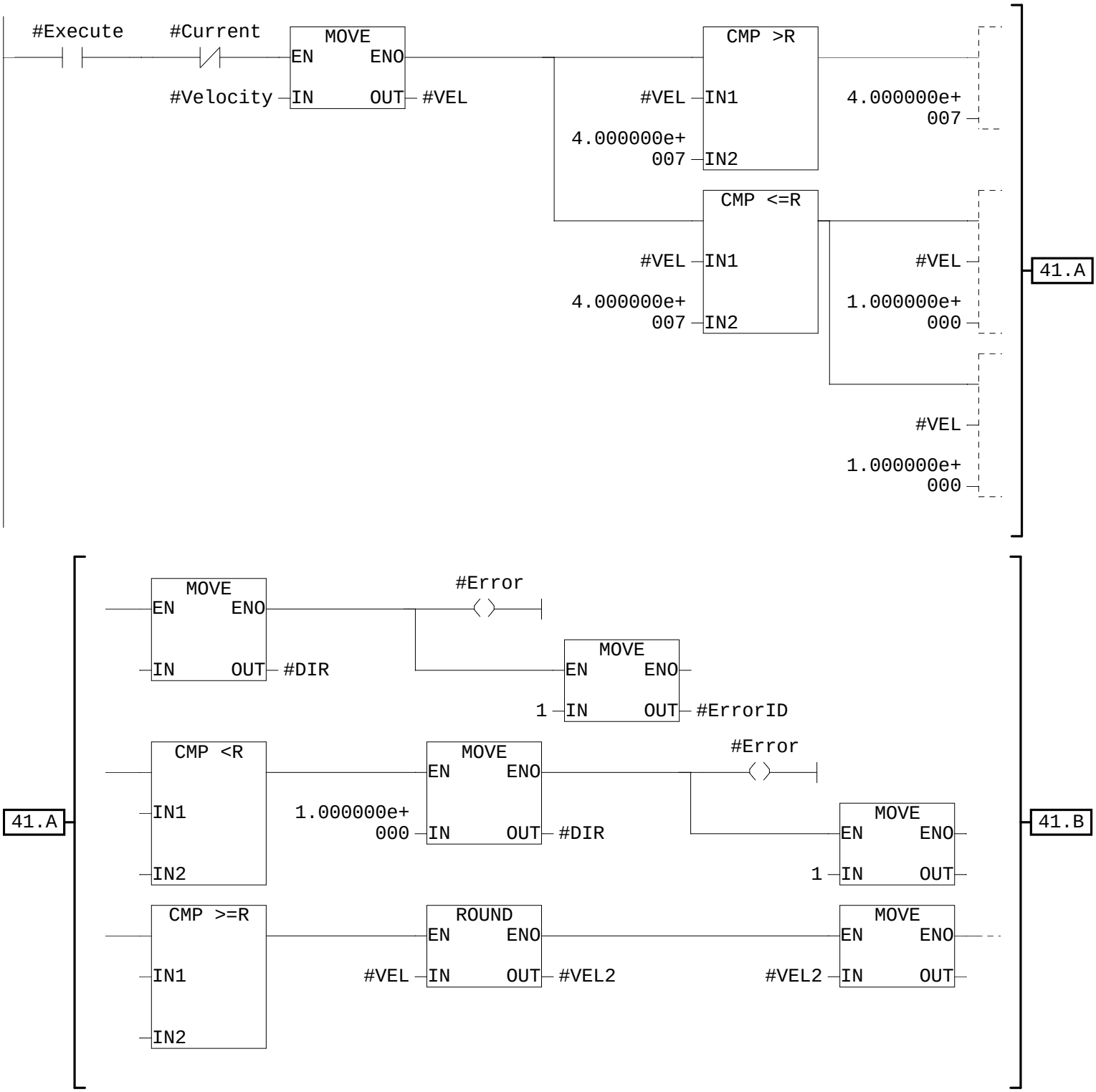


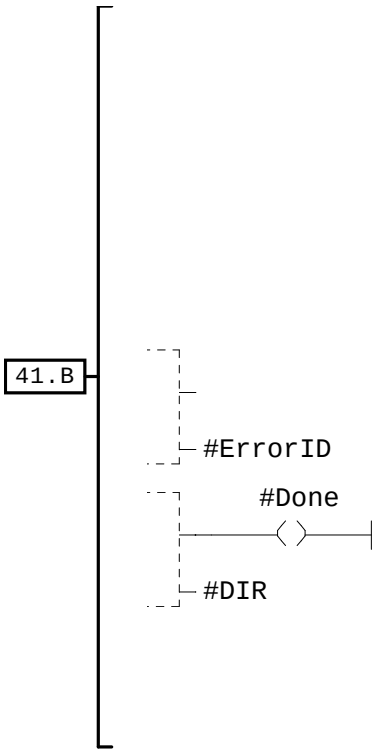
Network: 39 Konec

Network: 40 Telegram 111

Network: 41 Velocity

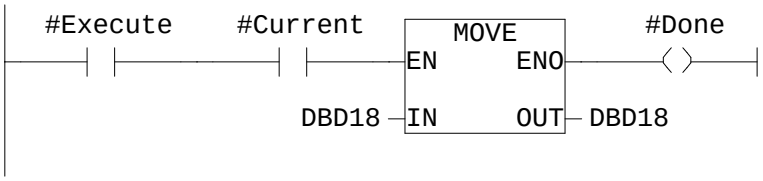
Regulovatelna v rozmezi 1 - 40000000. Co je ale maximalni rychlost, urcuje nastavitelny parametr p2571!
Normalizace: 1 hex = 1000 LU/min





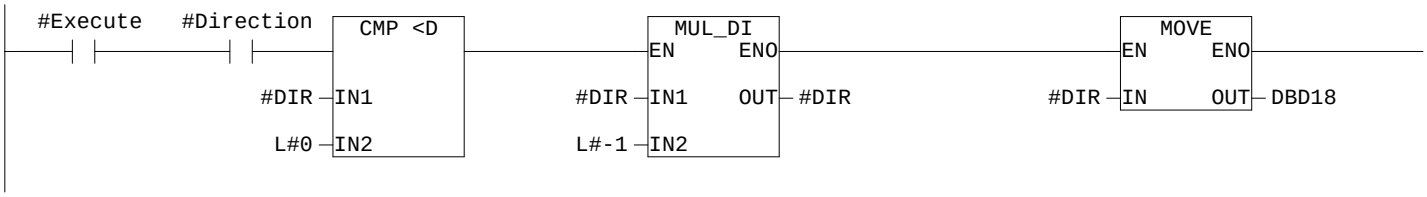
Network: 42 Velocity

Bit Current = true
Bude pouzita aktualni, posledne nastavena, rychlost osy



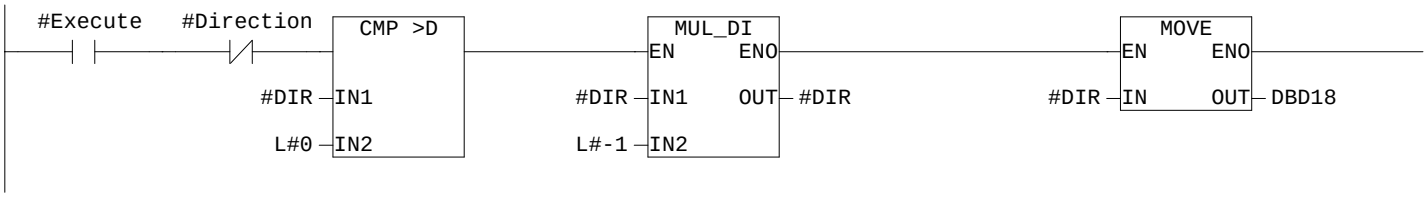
Network: 43 Direction

Smer otaceni nastaven na Forward
Direction = True



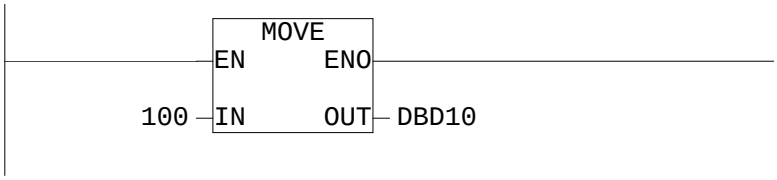
Network: 44 Direction

Smer otaceni nastaven na Backward
Direction = False



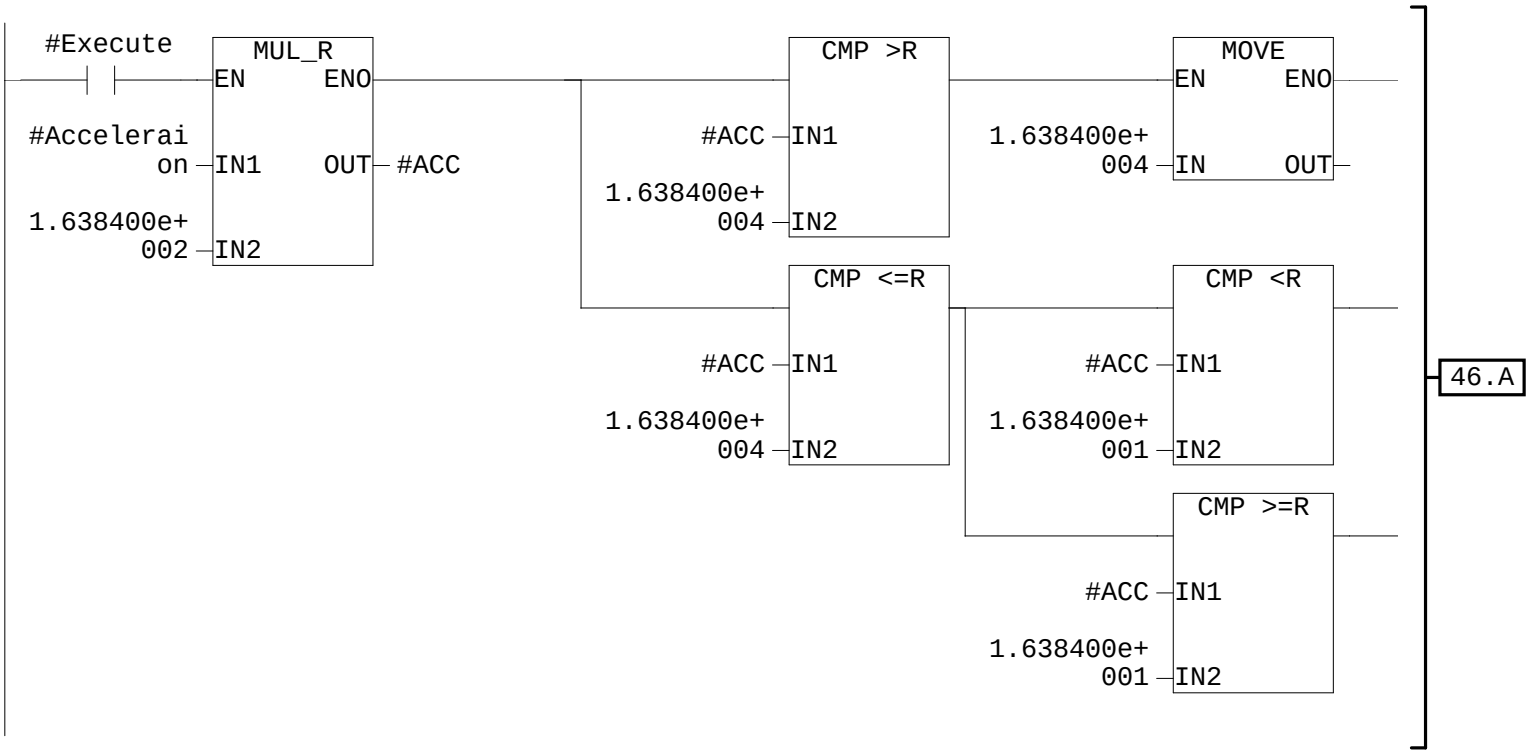
Network: 45 Override

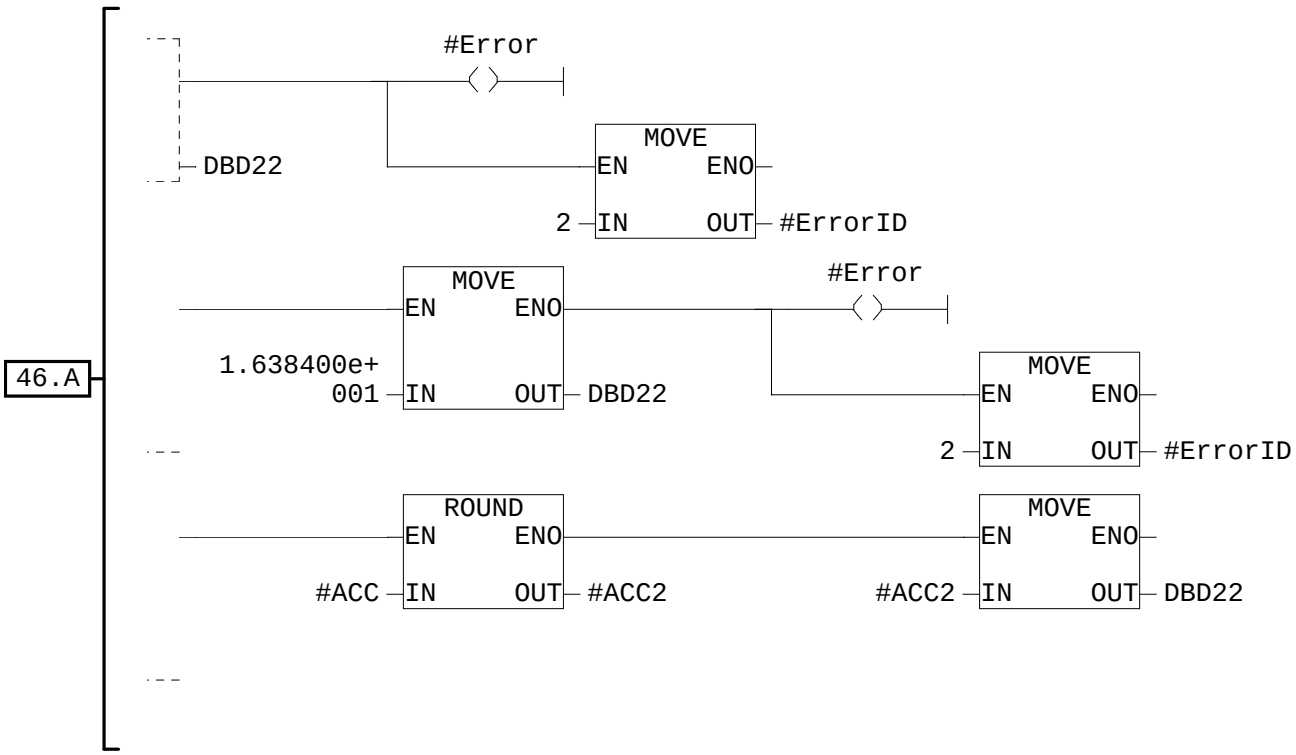
Nastaveni Override pro rychlost - 100%



Network: 46 Acceleration

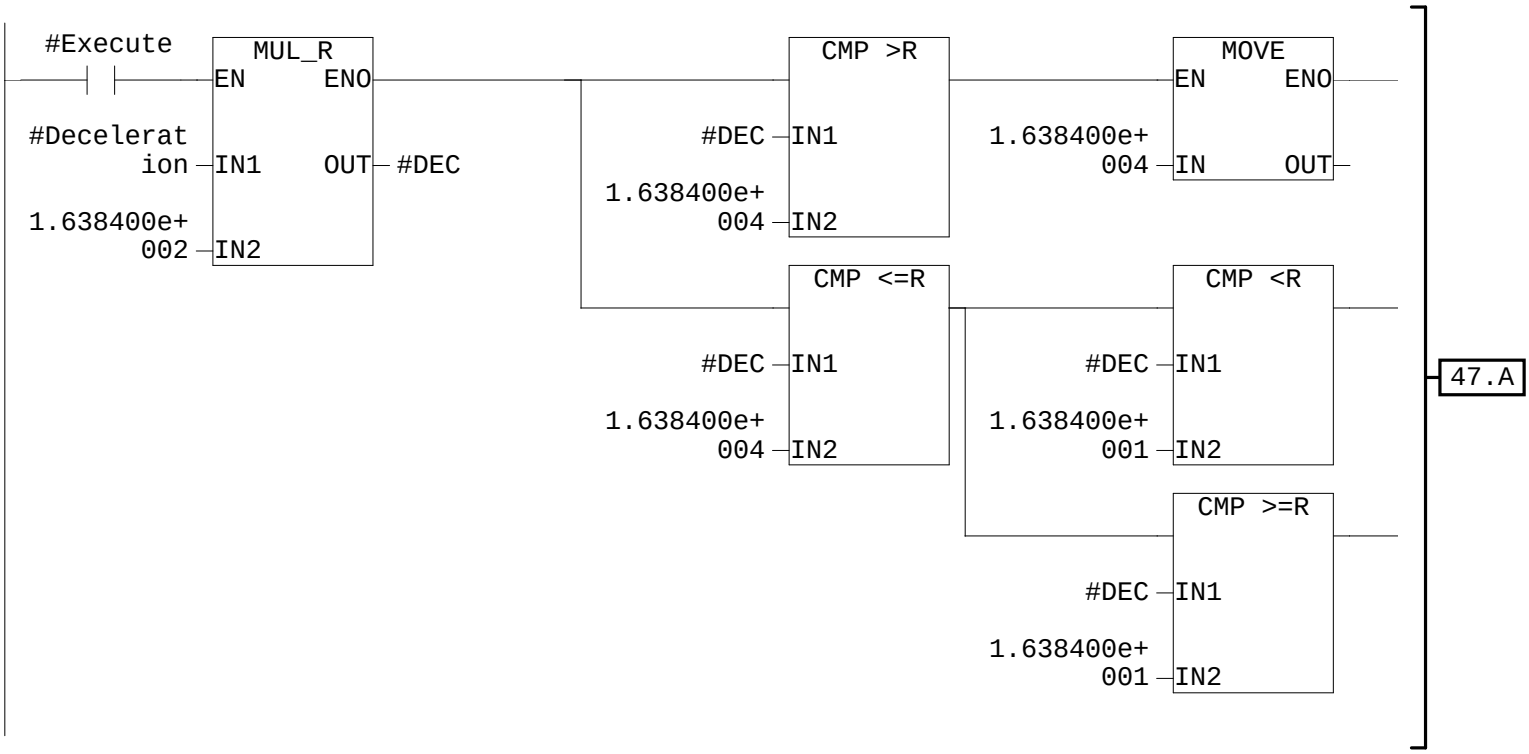
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

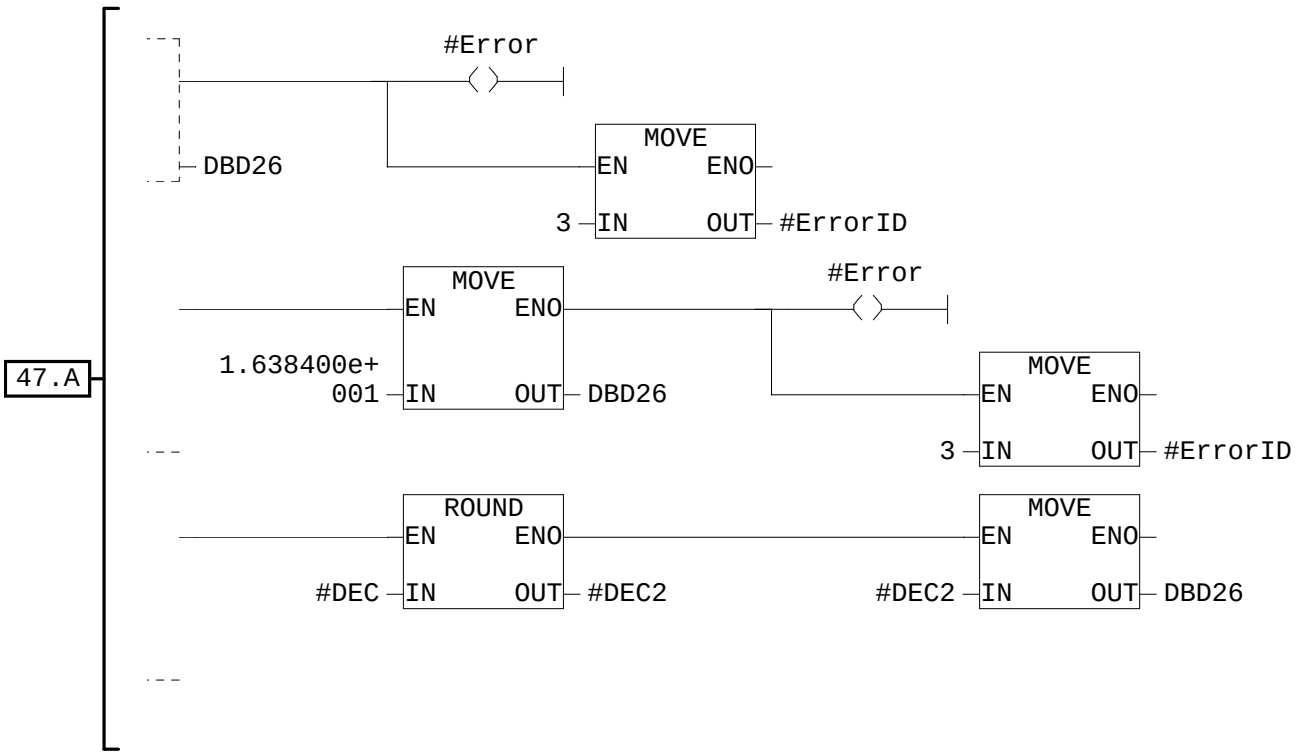




Network: 47 Deceleration

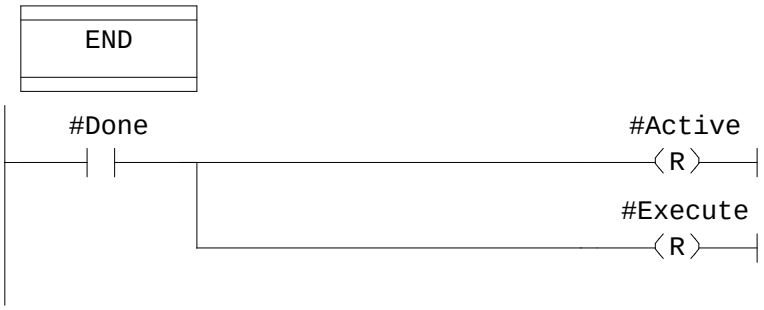
Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer





Network: 48 Konec

Network: 49 Konec



A.5 Funkce – Jog

FC4 - <offline>

"FC_Jog"

Name: **Family:**
Author: Vacek **Version:** 0.1
Block version: 2
Time stamp Code: 05/26/2009 06:17:53 PM
Interface: 05/20/2009 11:52:58 AM
Lengths (block/logic/data): 03514 03146 00042

Name	Data Type	Address	Comment
IN		0.0	
Axis_DB	Block_DB	0.0	Otevre datovy blok pro pozadovany standardni telegram
Execute	Bool	2.0	Start
Velocity	Real	4.0	Rychlost
Acceleraion	Real	8.0	Zrychleni
Deceleration	Real	12.0	Zpomaleni
Direction	Bool	16.0	Smer
OUT		0.0	
Done	Bool	18.0	Dosazena cilova pozice
Active	Bool	18.1	Funkce je aktivni
Error	Bool	18.2	Chyba
ErrorID	Int	20.0	Identifikace chyb
IN_OUT		0.0	
TEMP		0.0	
HELP	Bool	0.0	
VEL	Real	2.0	
VEL2	DInt	6.0	
ACC	Real	10.0	
ACC2	DInt	14.0	
DEC	Real	18.0	
DEC2	DInt	22.0	
DIR	DInt	26.0	
Cislo	Word	30.0	Cislo DB
Delka	Word	32.0	Delka DB v bytech
Return	Int	34.0	Navratova hodnota (0=OK)
WrPr	Bool	36.0	DB chranen proti prepsani (1=ANO)
Delka2	Int	38.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC4

```
Created:
date 17.04.2009 version 1.0 - Jakub Vacek
-----------------------------------------------------
Funkce Jog startuje funkci Jog
-----------------------------------------------------
Telegram: 9,110,111
-----------------------------------------------------
Vypis chybovych hlaseni do pole ErrorArray
ErrorArray[0] = 1, nebyl nalezen kompatibilni telegram
ErrorArray[1] = 1, spatne zadana rychlost
ErrorArray[2] = 1, spatne zadane zrychleni
ErrorArray[3] = 1, spatne zadane zpomaleni
```

Network: 1 Otevri DB

Funkce otevre datovy blok pro pozadovany standardni telegram

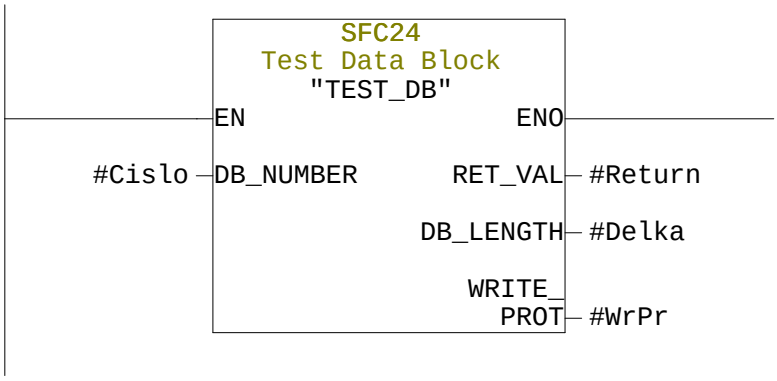
Network: 2

Do pomocne promenne Cislo ulozi cislo DB

```
      L     DBNO
      T     #Cislo
```

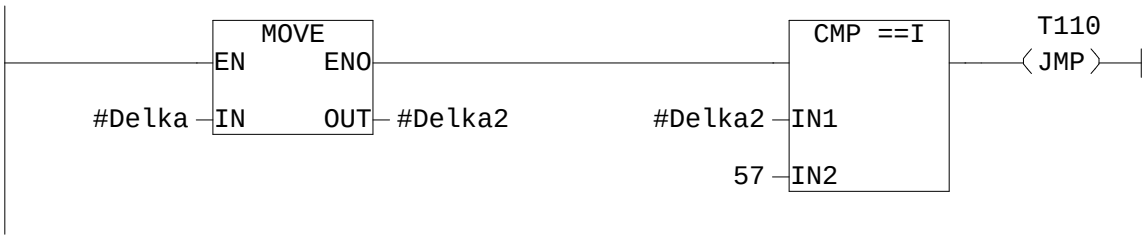
Network: 3

Zjisteni delky datoveho bloku

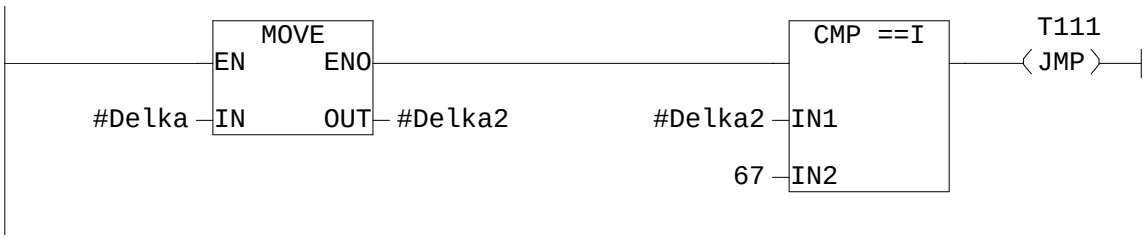


Network: 4 Je pouzit telegram 9?

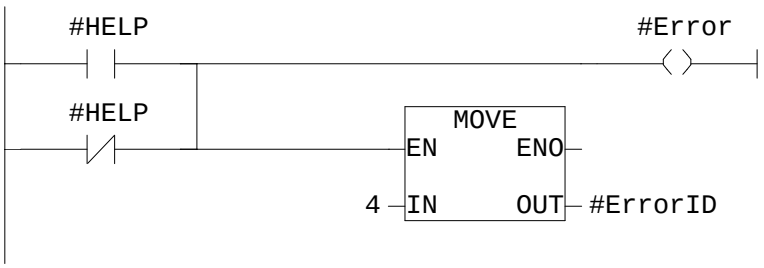
Network: 5 Je pouzit telegram 110?



Network: 6 Je pouzit telegram 111?



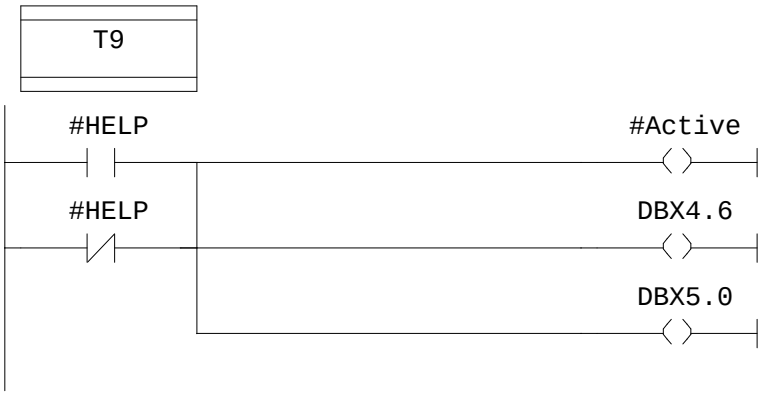
Network: 7 Nebyl pouzit vhodny telegram



Network: 8 Konec

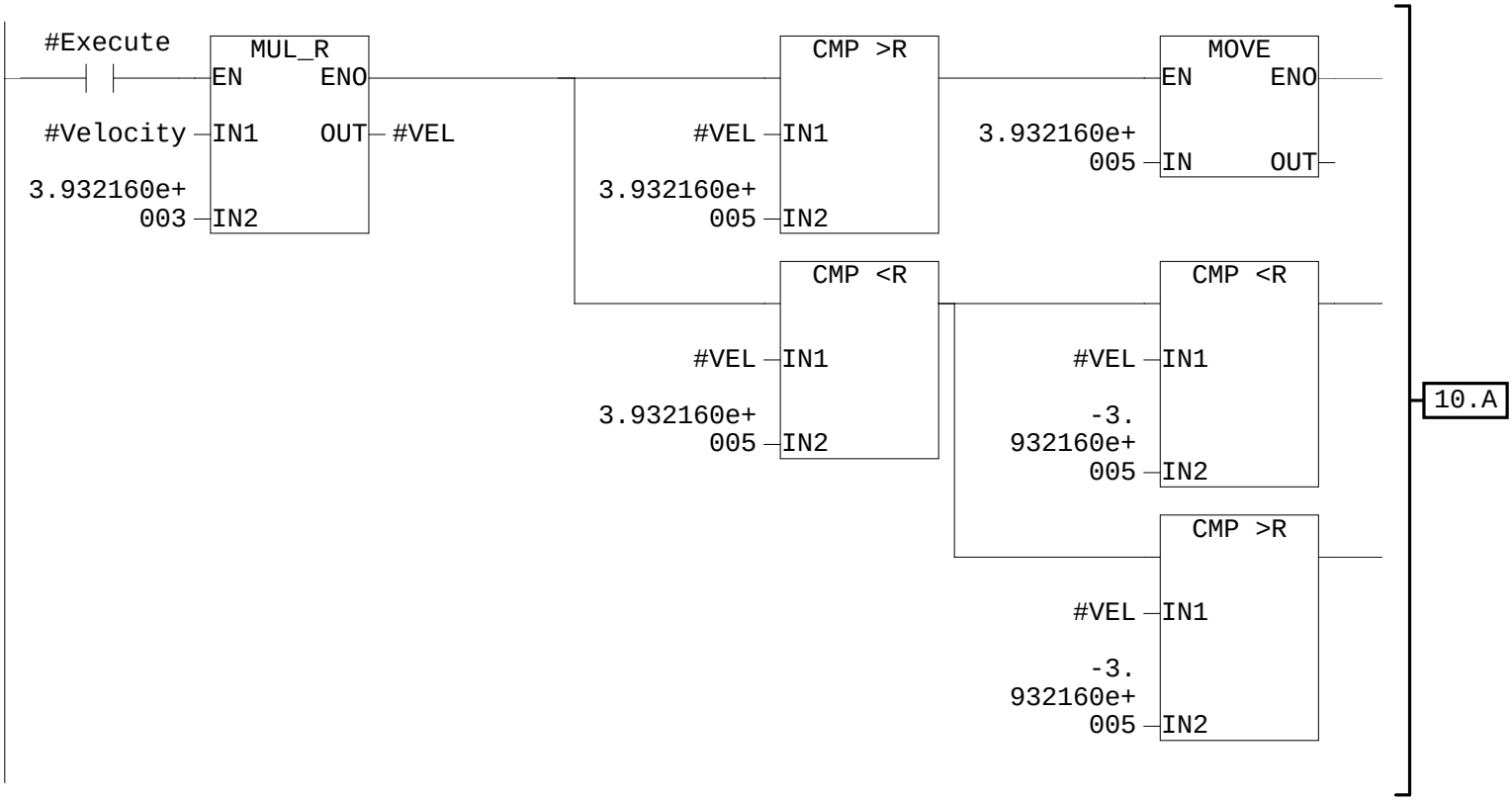
Network: 9 Telegram 9

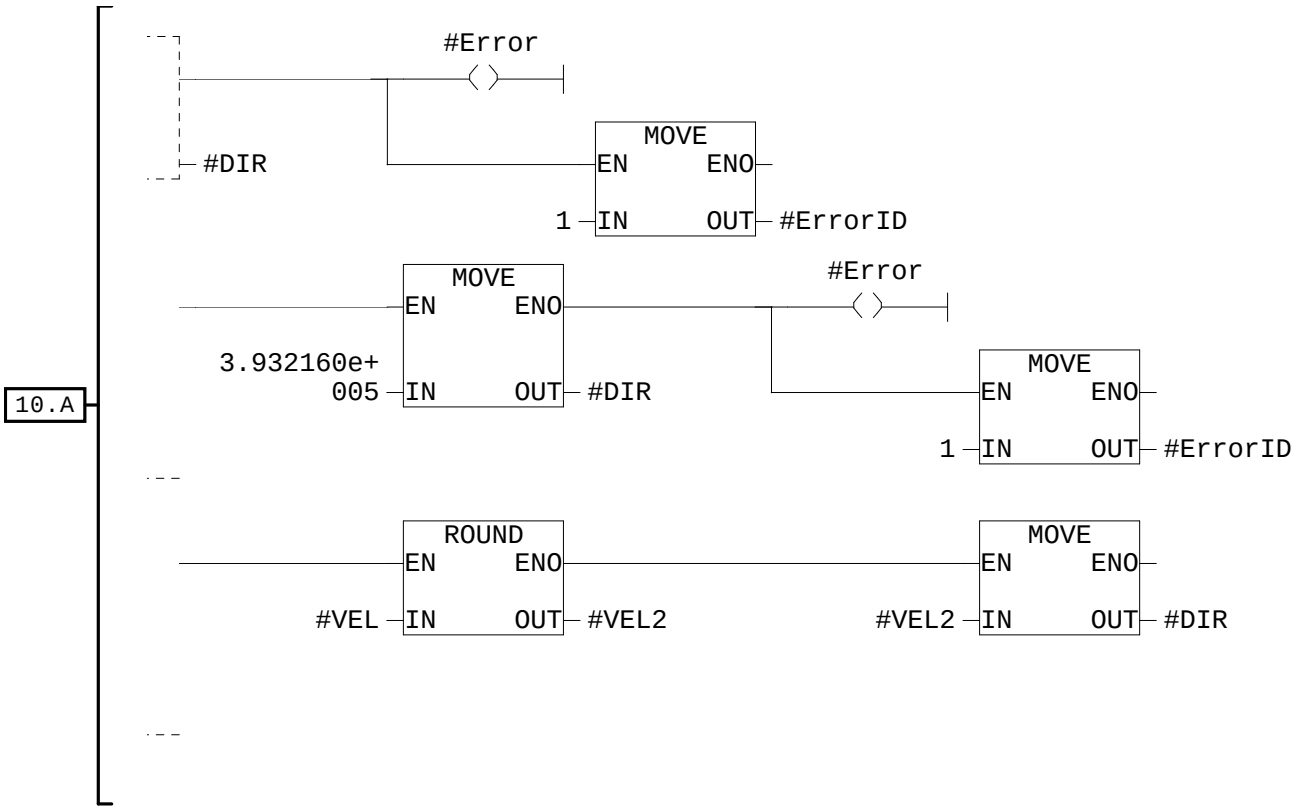
Bity EPOS_activate a EPOS_jog_1 jsou nastaveny na hodnotu true, coz je nezbytne pro povoleni funkce jog



Network: 10 Velocity

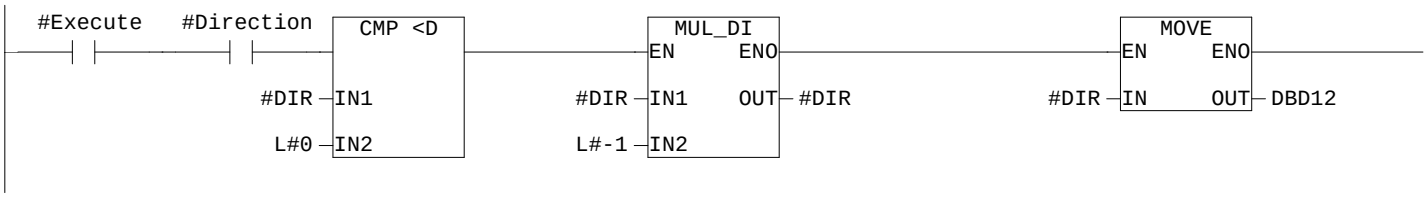
Nastaveni rychlosti v procentech (0-100)
Standardni normalizace v menicich je 1 hex = 1000 LU/min, maximalni rychlost
udava parametr p2000 (6000 rev/min)
Pri nastaveni 10000 LU per load revolution je 60000 hex = 100%
Provadi se omezeni na hodnoty double integer





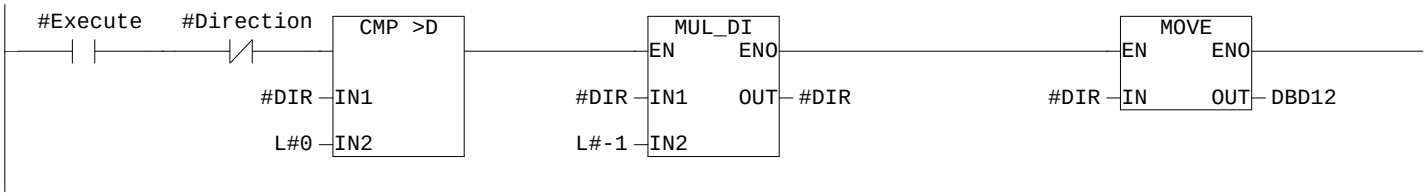
Network: 11 Direction

Smer otaceni nastaven na Forward
Direction = True



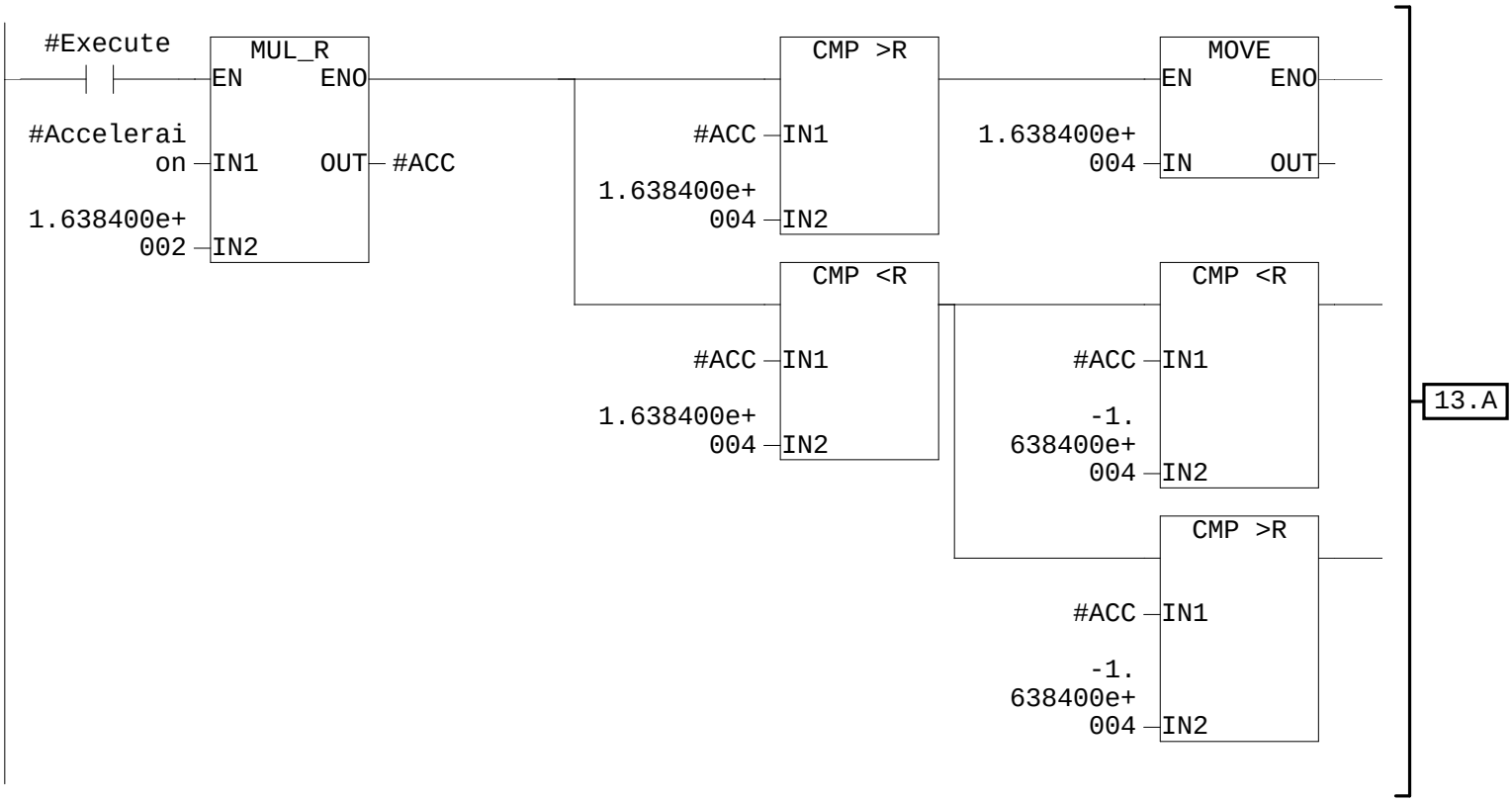
Network: 12 Direction

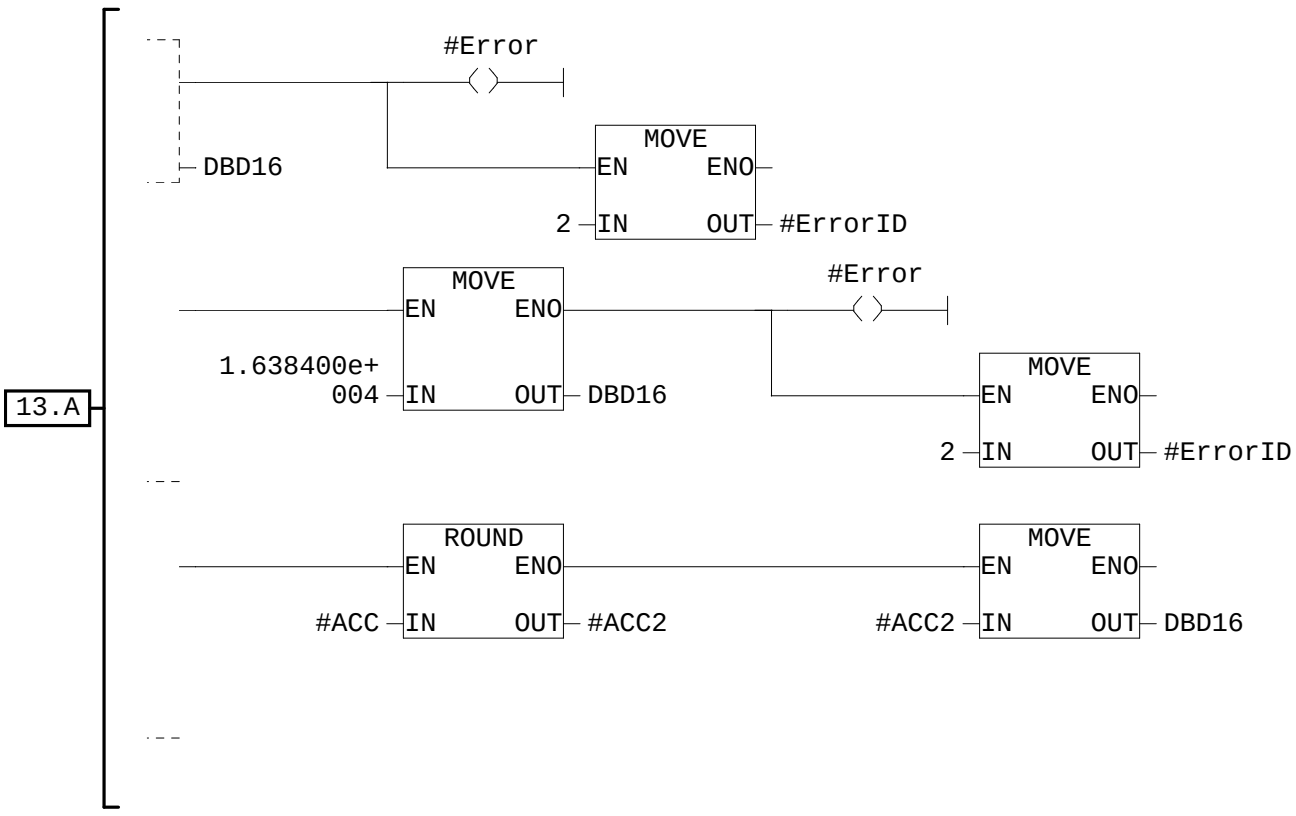
Smer otaceni nastaven na Backward
Direction = False



Network: 13 Acceleration

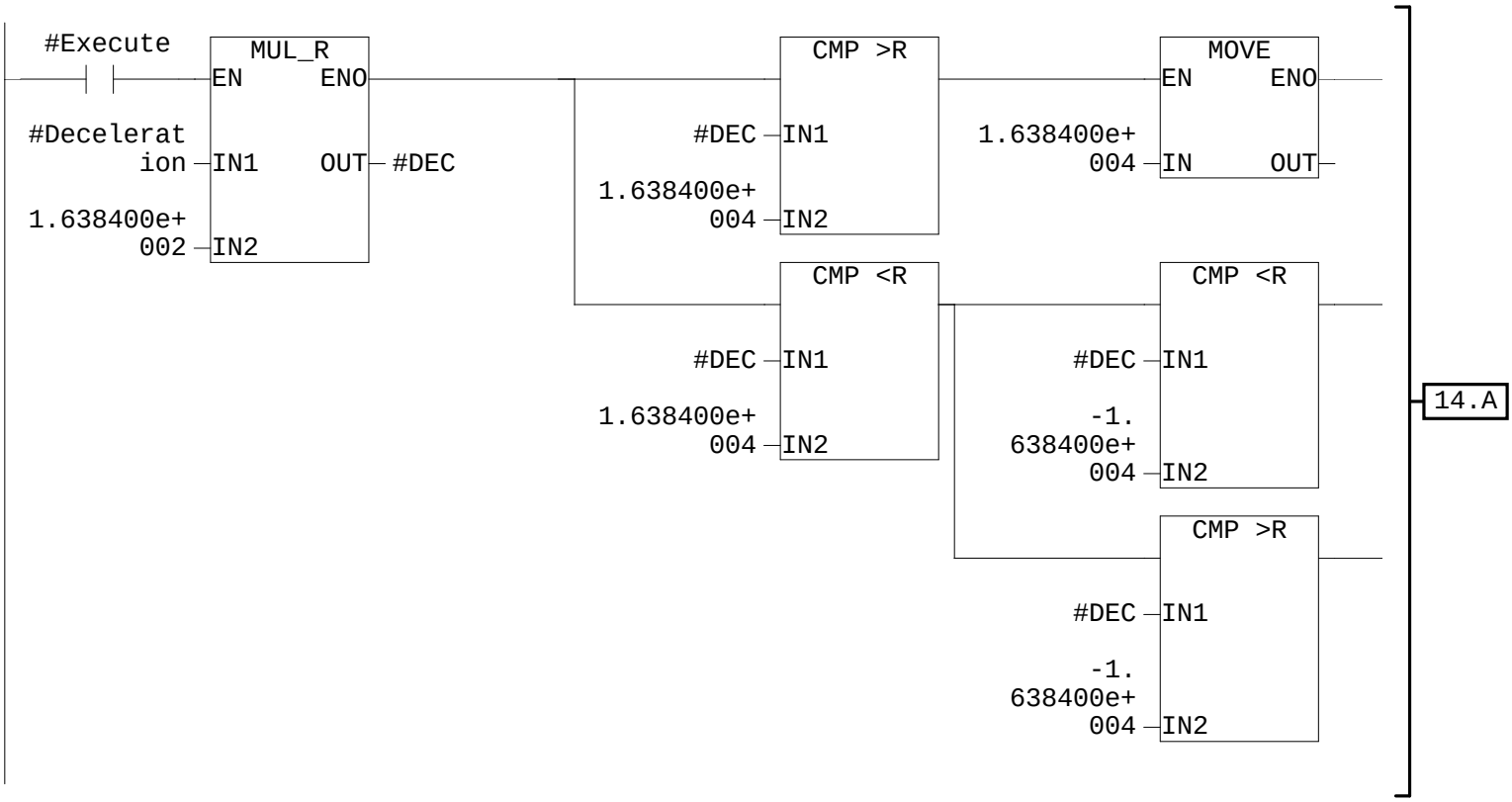
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

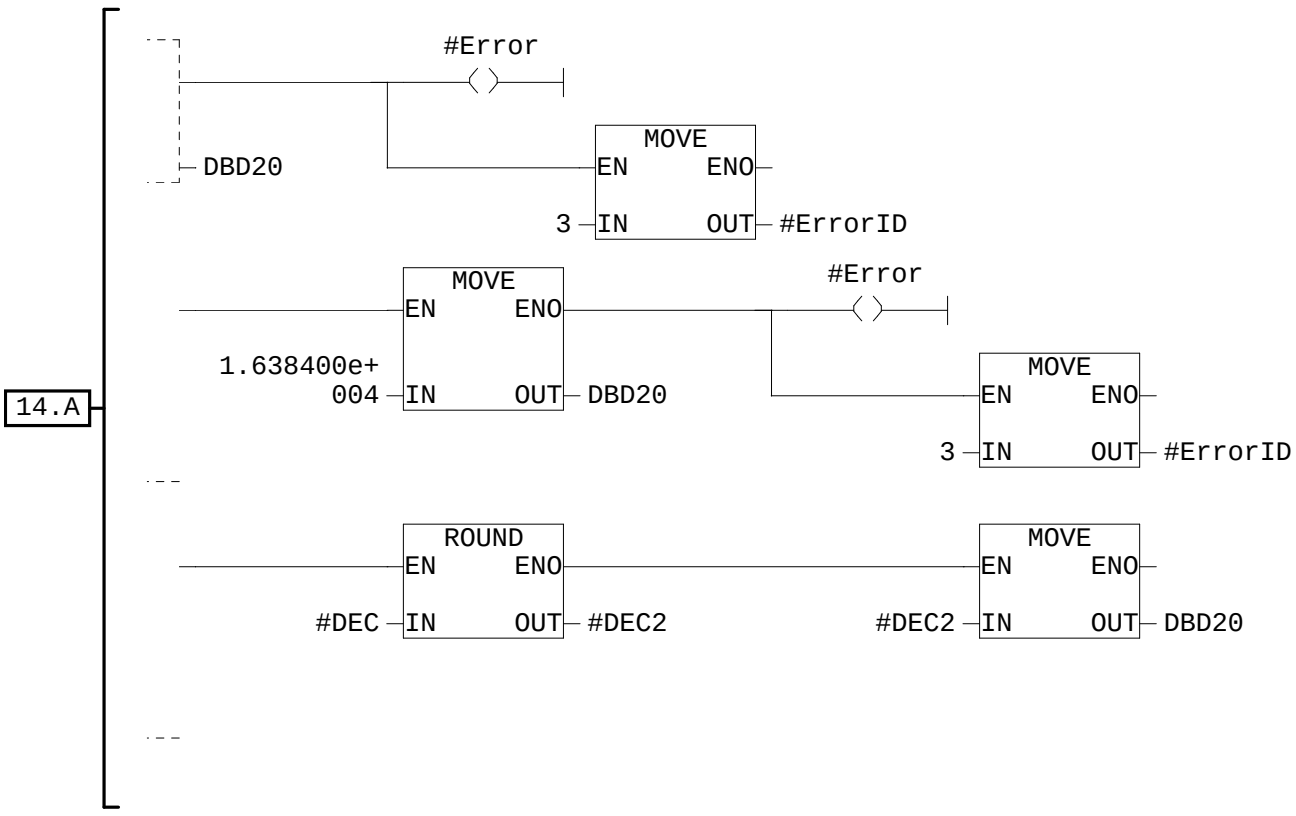




Network: 14 Deceleration

Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

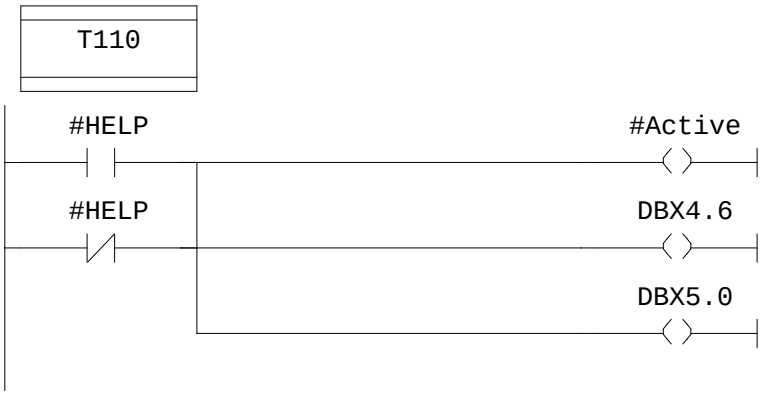




Network: 15 Konec

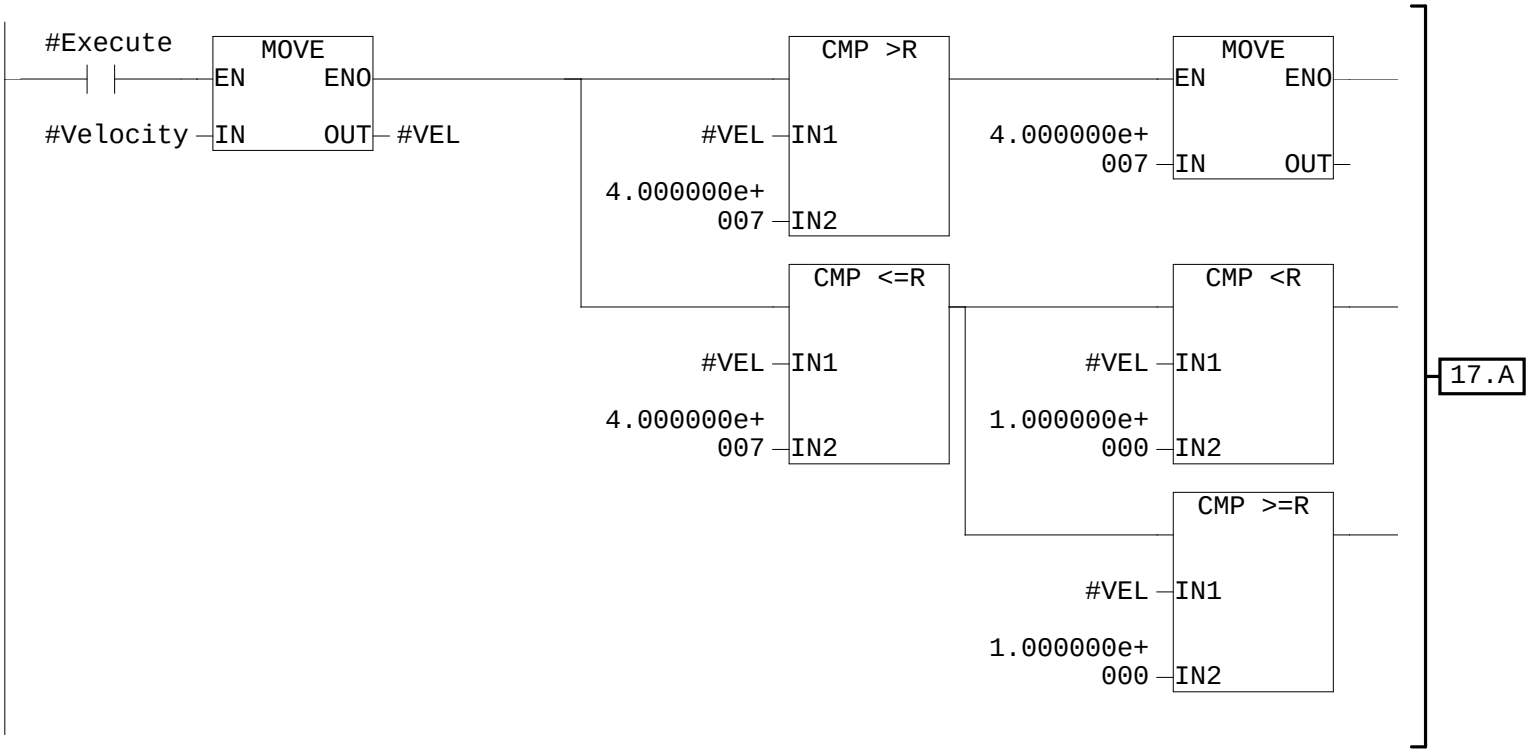
Network: 16 Telegram 110

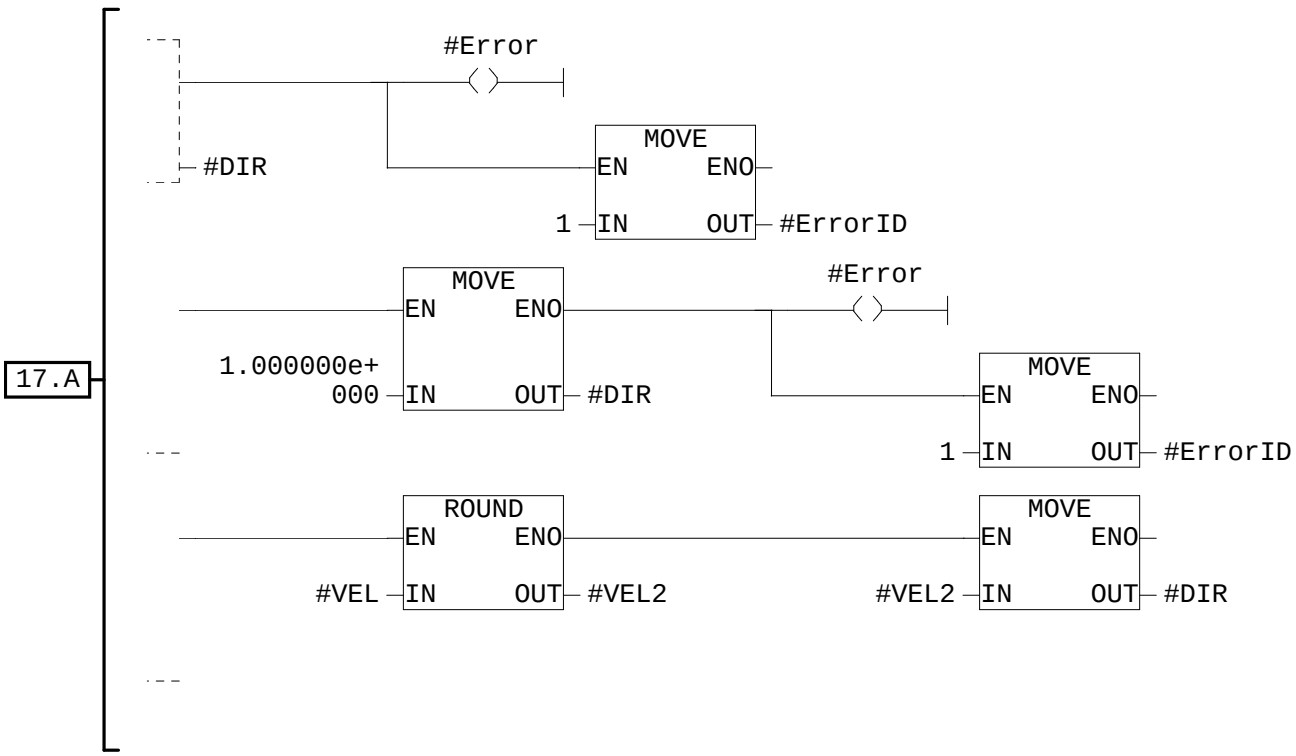
Bity EPOS_activate a EPOS_jog_1 jsou nastaveny na hodnotu true, coz je nezbytne pro povoleni funkce jog



Network: 17 Velocity

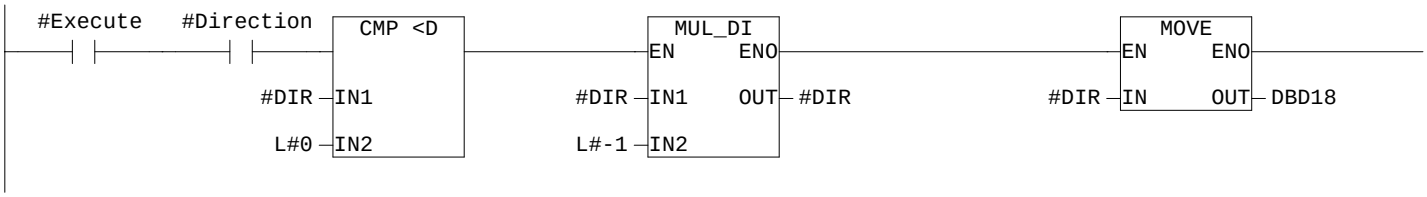
Nastaveni rychlosti
Regulovatelna v rozmezi 1 - 40000000
Standardni normalizace v menicich je 1 hex = 1000 LU/min, maximalni rychlost
udava parametr p2571
Provadi se omezeni na hodnoty double integer





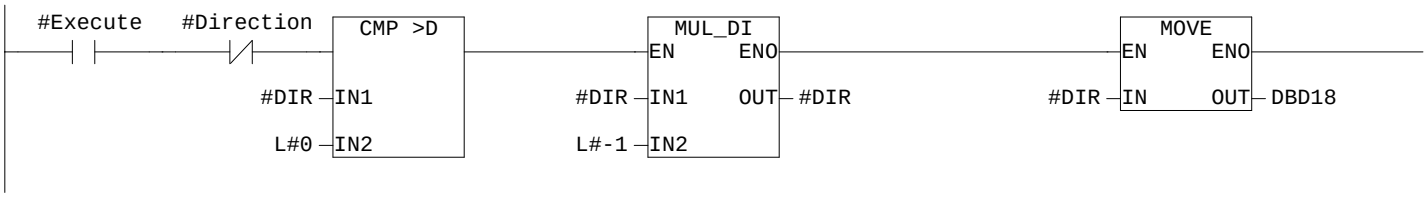
Network: 18 Direction

Smer otaceni nastaven na Forward
Direction = True



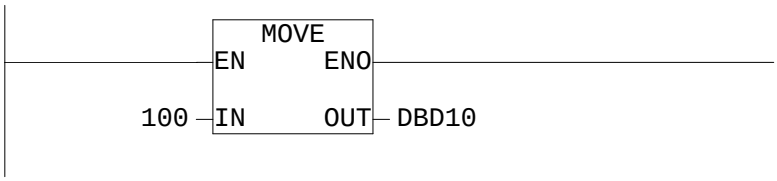
Network: 19 Direction

Smer otaceni nastaven na Backward
Direction = False



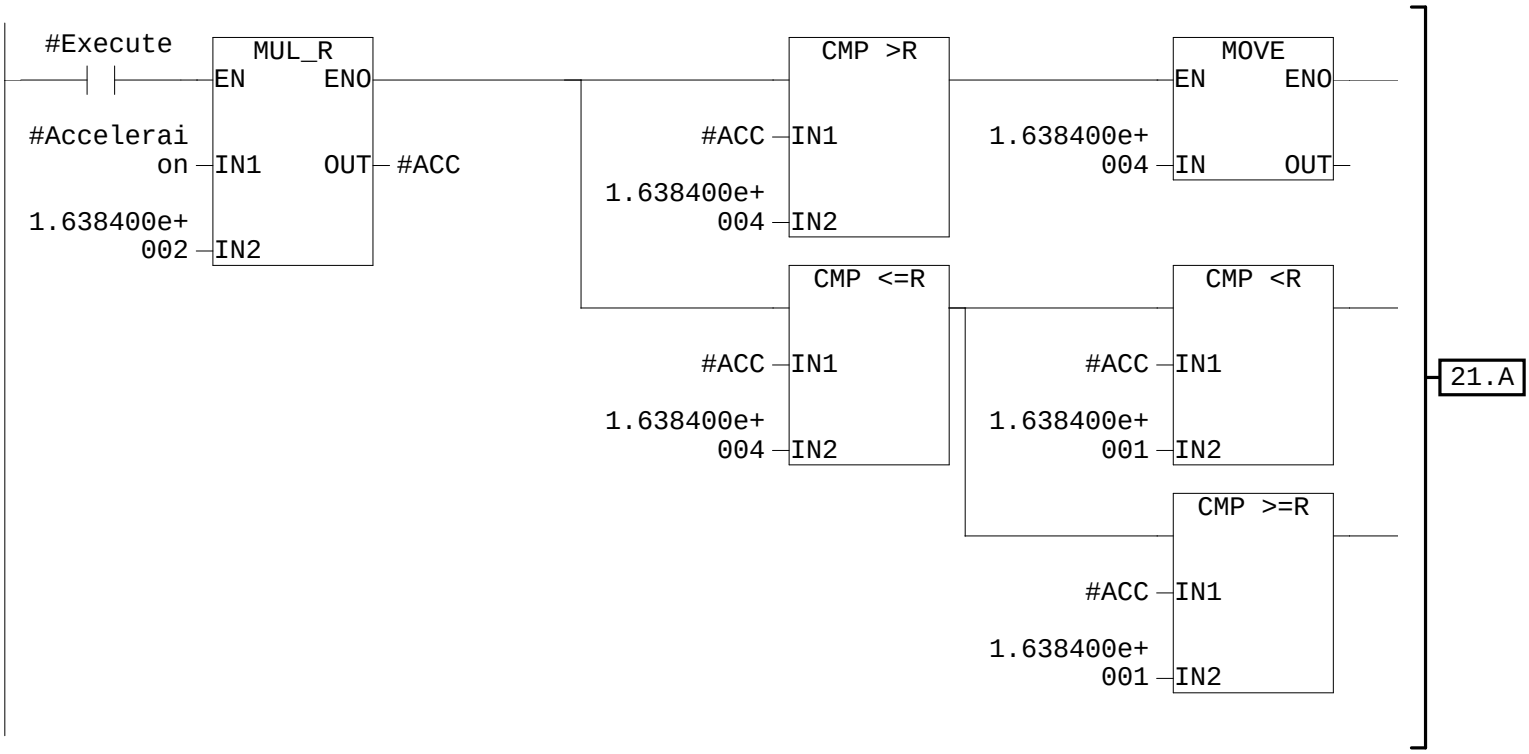
Network: 20 Override

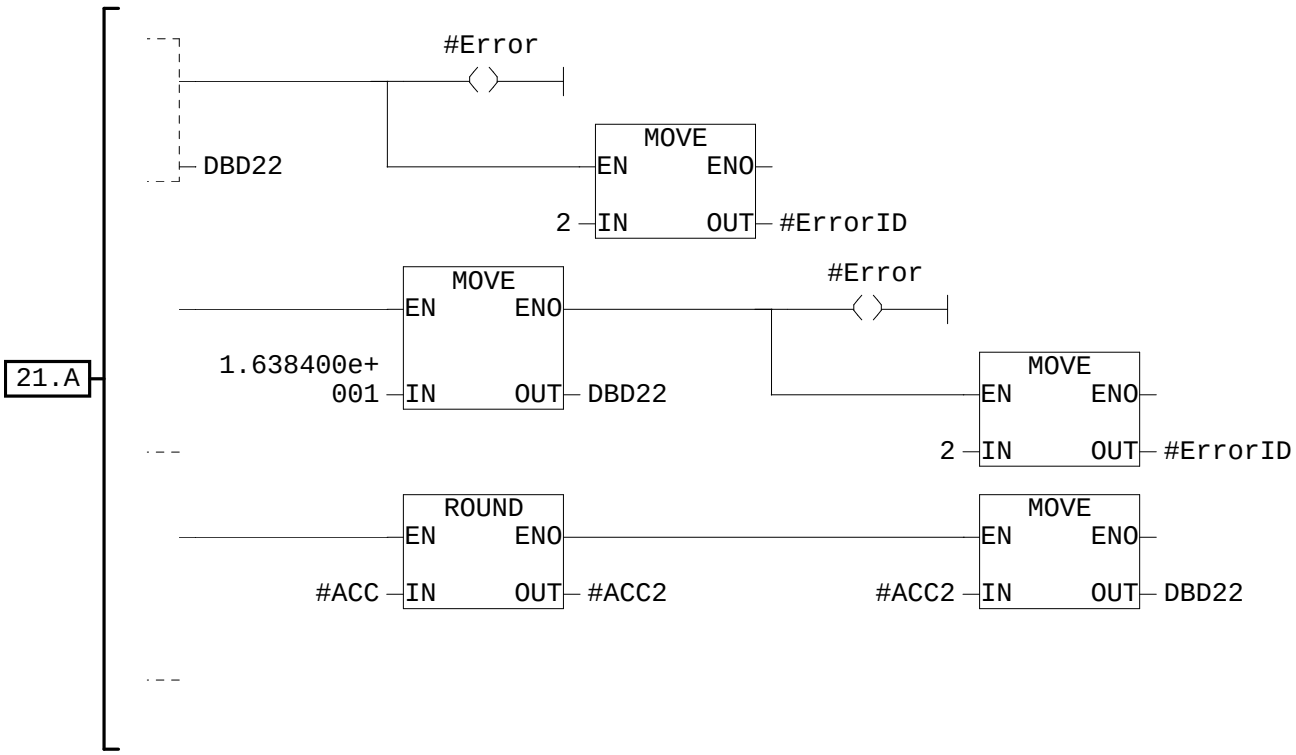
Nastaveni Override pro rychlost - 100%



Network: 21 Acceleration

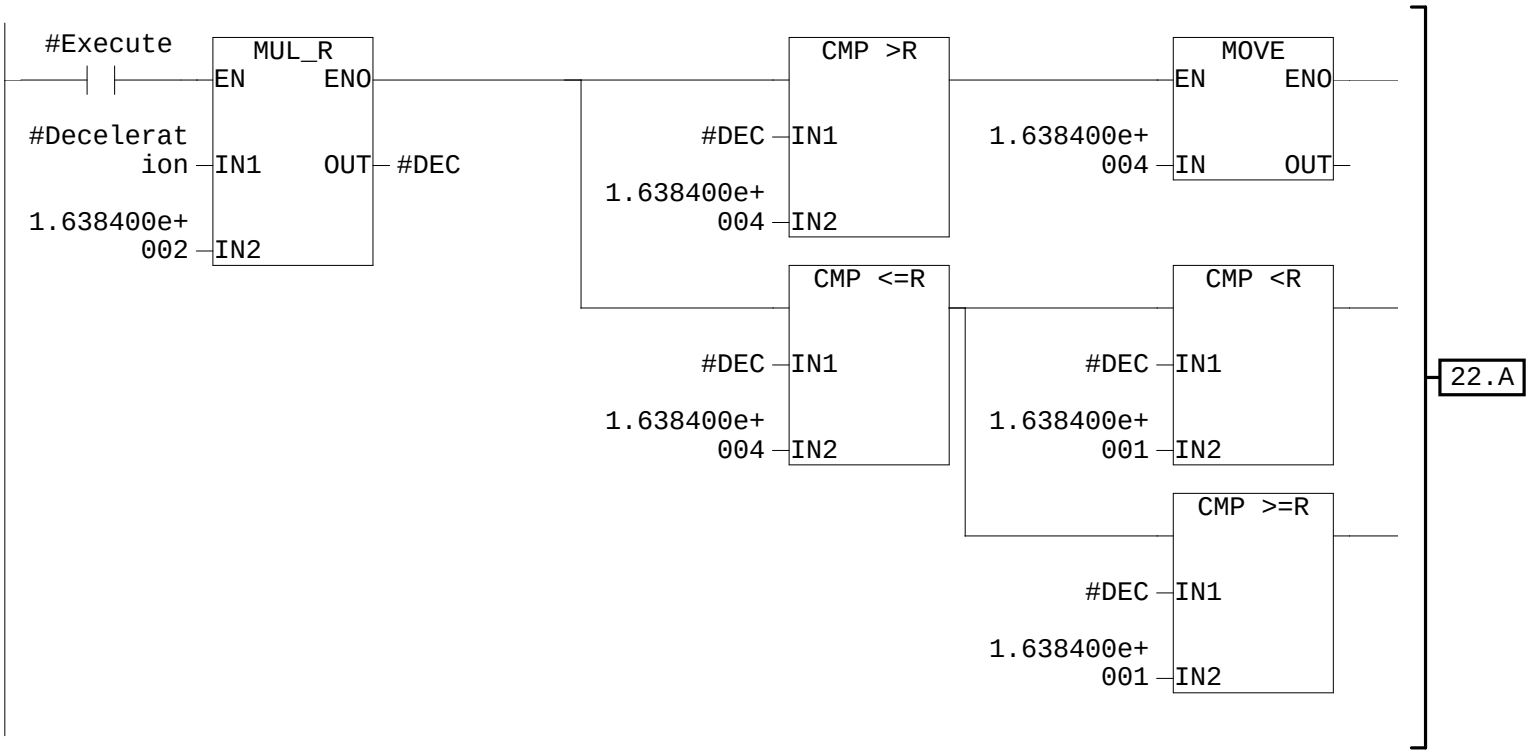
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

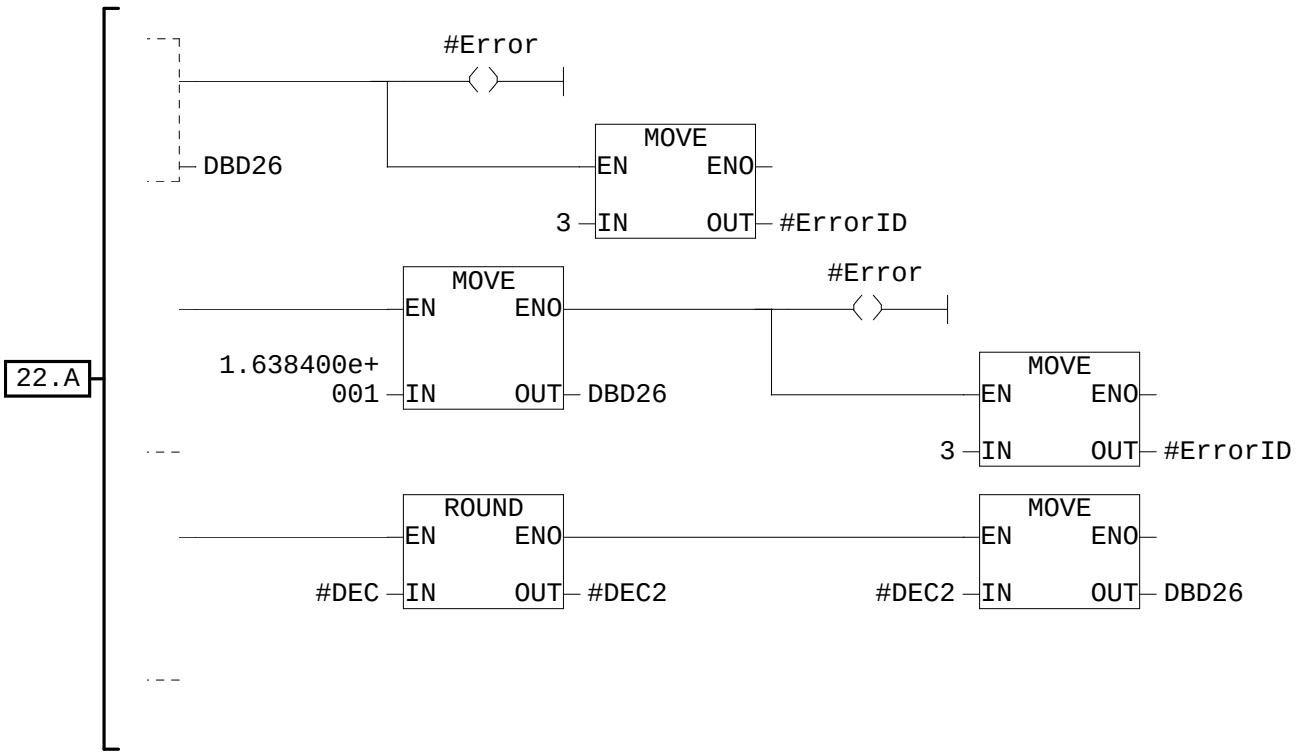




Network: 22 Deceleration

Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

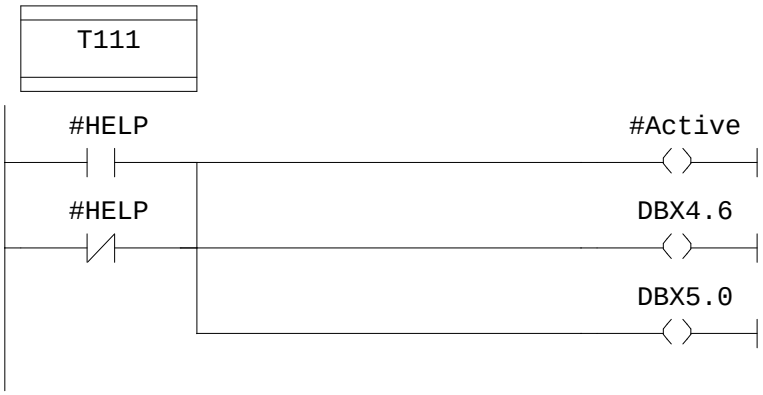




Network: 23 Konec

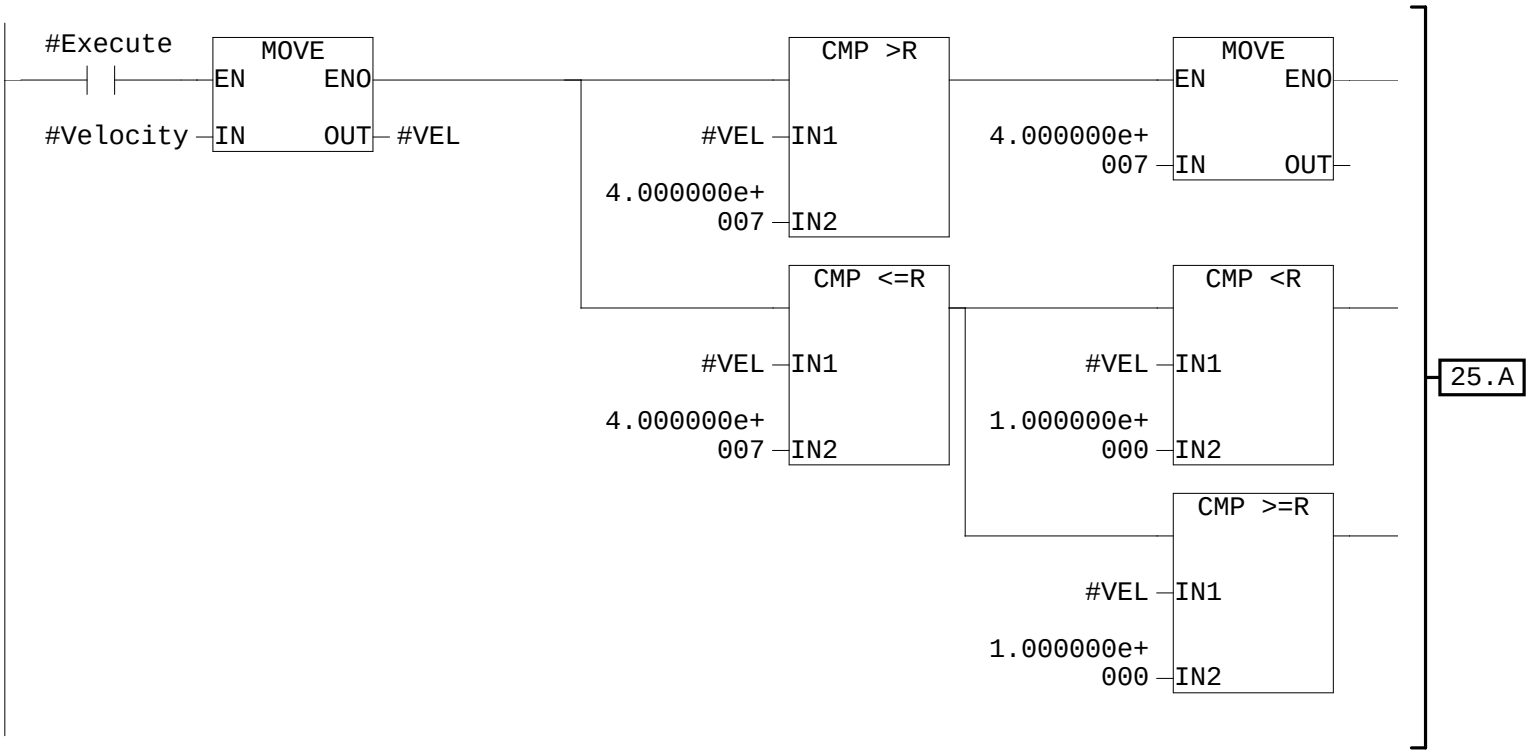
Network: 24 Telegram 111

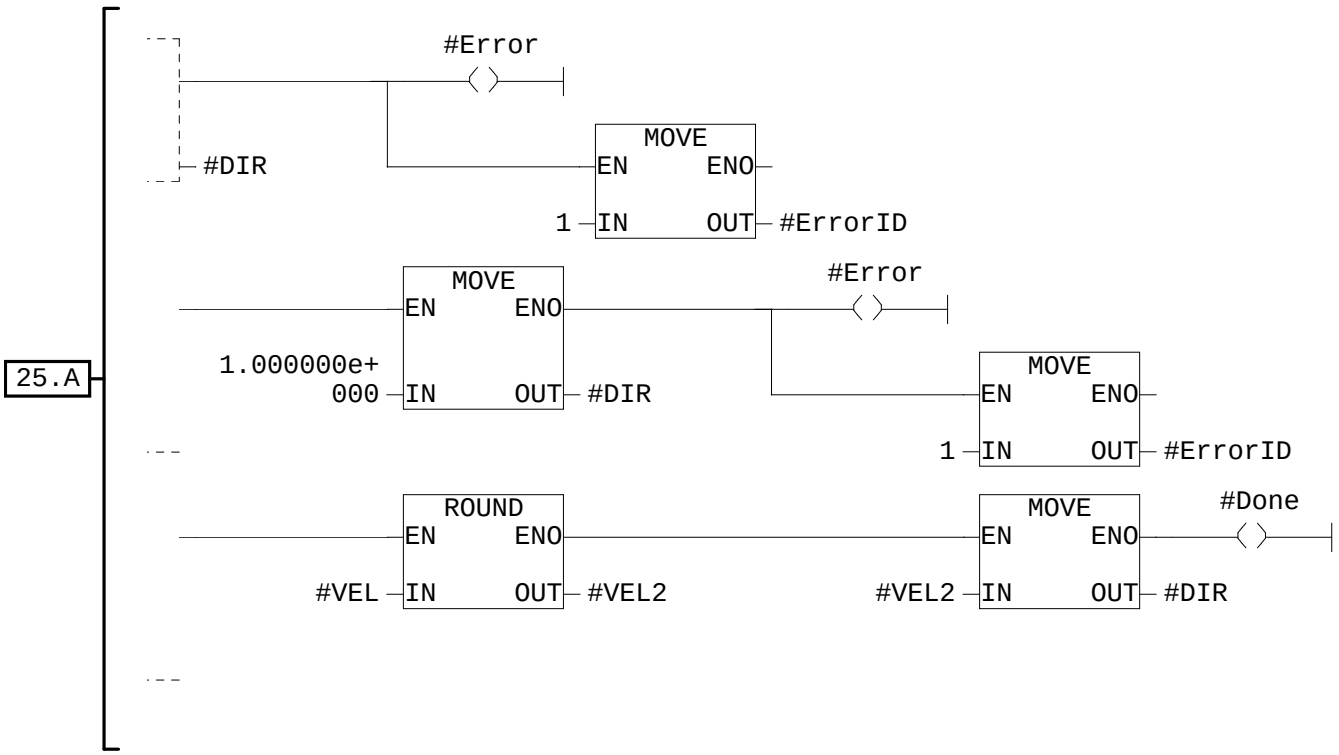
Bity EPOS_activate a EPOS_jog_1 jsou nastaveny na hodnotu true, coz je nezbytne pro povoleni funkce jog



Network: 25 Velocity

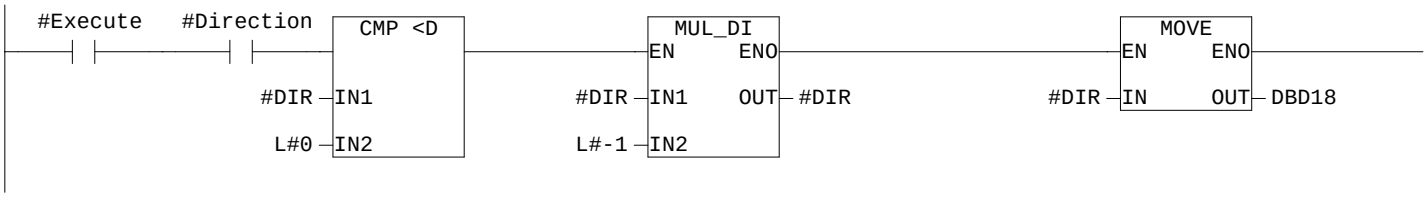
Regulovatelna v rozmezi 1 - 40000000. Co je ale maximalni rychlost, urcuje nastavitelny parametr p2571!
Normalizace: 1 hex = 1000 LU/min





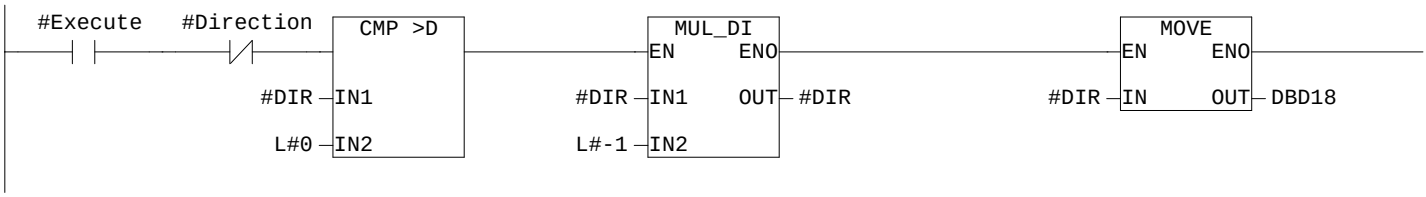
Network: 26 Direction

Smer otaceni nastaven na Forward
Direction = True



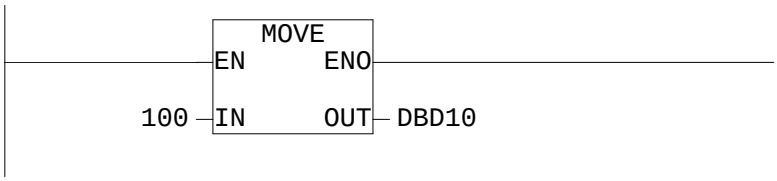
Network: 27 Direction

Smer otaceni nastaven na Backward
Direction = False



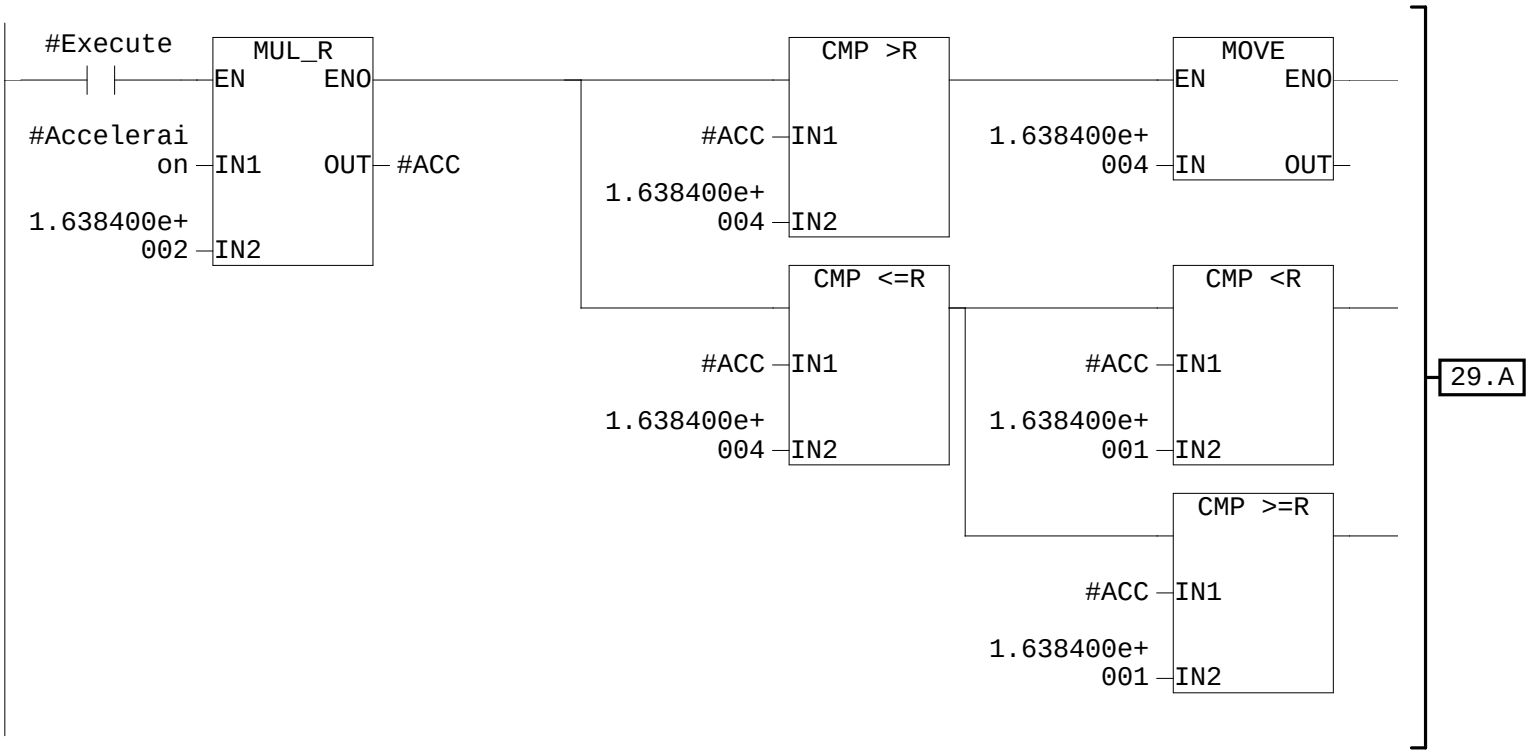
Network: 28 Override

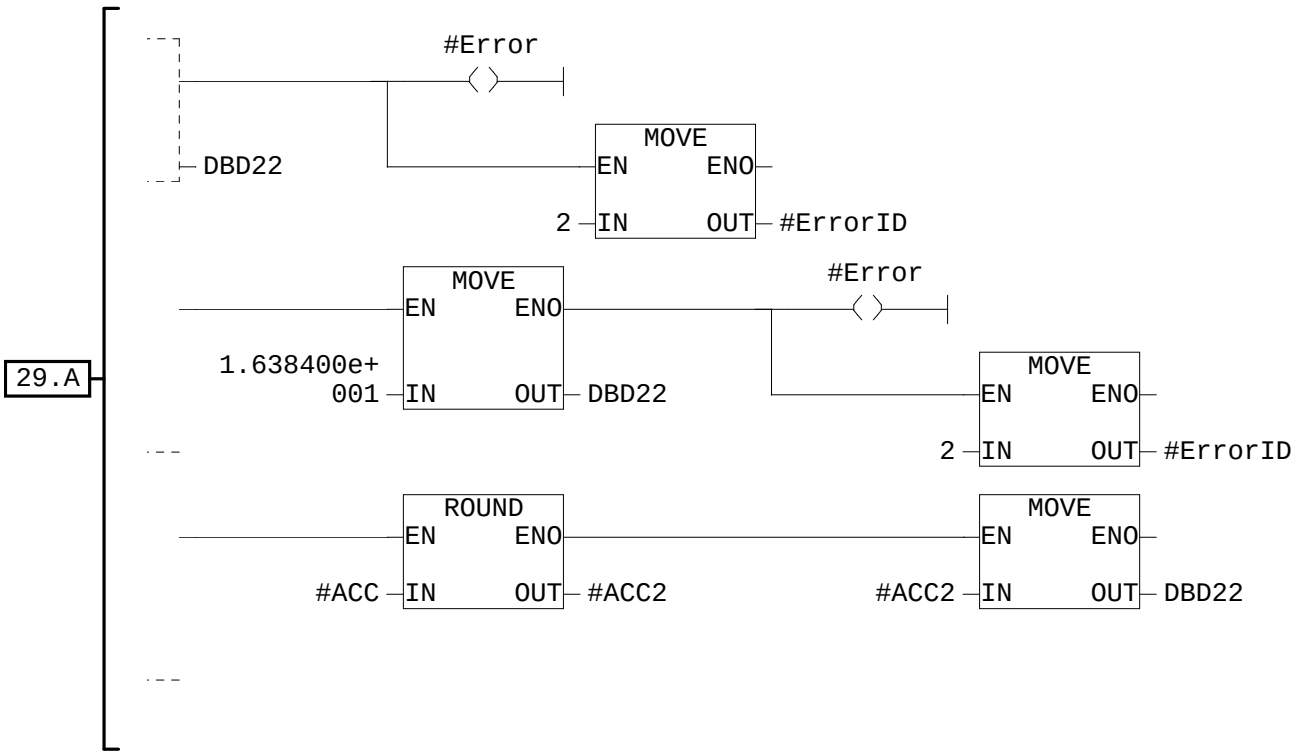
Nastaveni Override pro rychlost - 100%



Network: 29 Acceleration

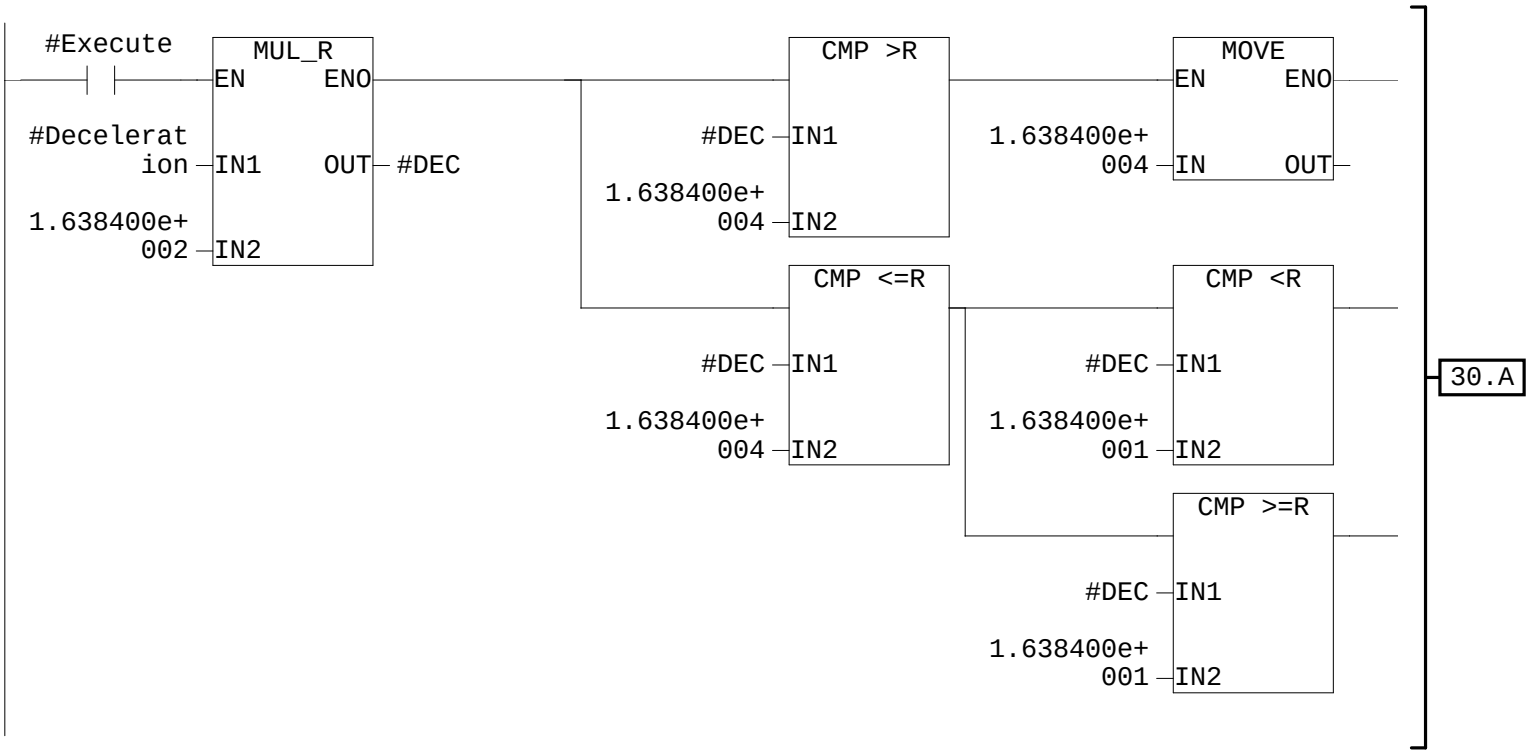
Nastaveni zrychleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer

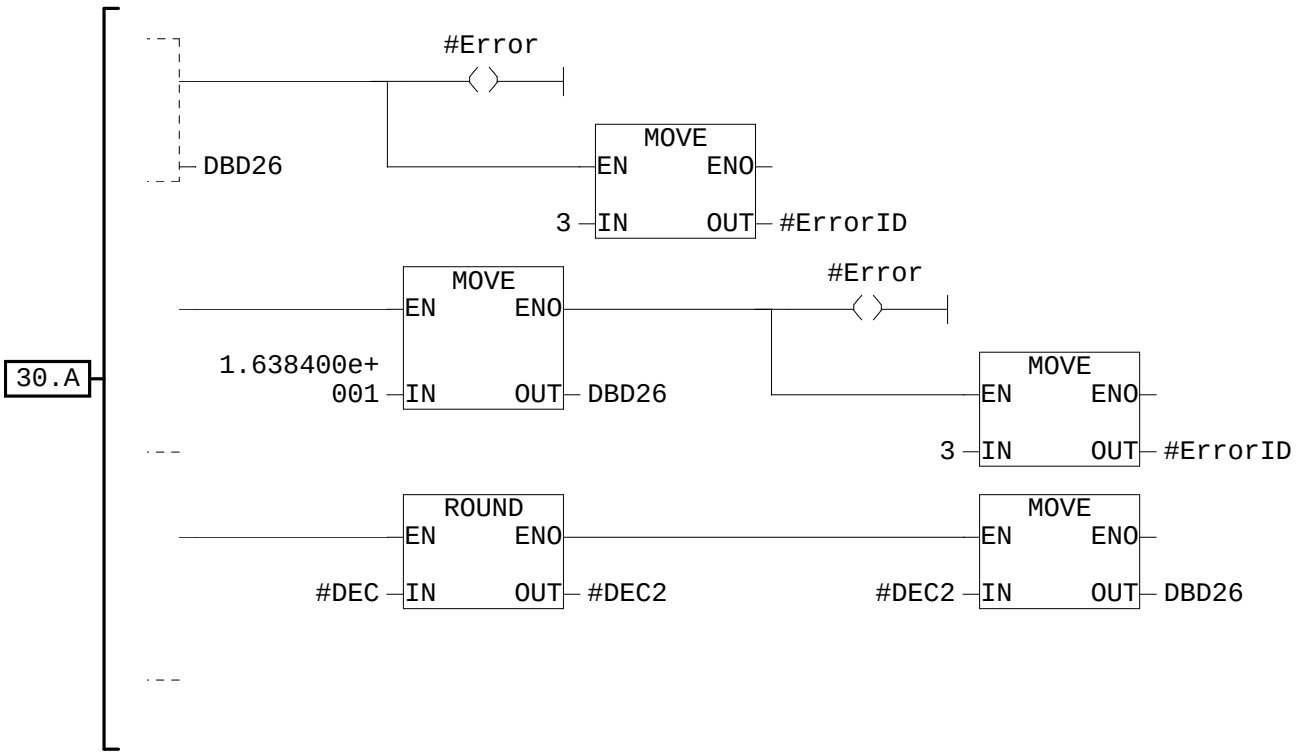




Network: 30 Deceleration

Nastaveni zpomaleni v procentech (0-100)
Standardni normalizace v menicich je 4000 hex = 100%
Provadi se omezeni na hodnoty double integer





Network: 31 Konec

Network: 32 Konec

Network: 33