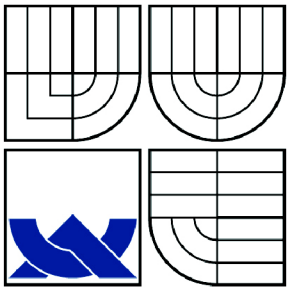
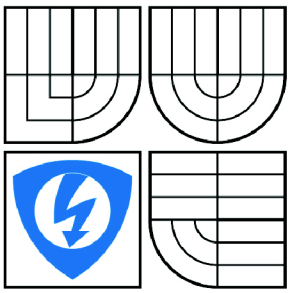


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

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 POLOHOVÝCH APLIKACÍCH

FUNCTION BLOCK FOR SIMATIC, SINAMICS DEVICES APPLIED IN POSITION APPLICATIONS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

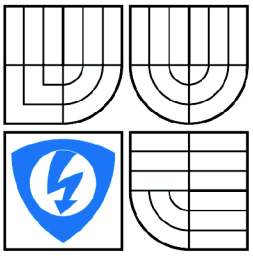
AUTOR PRÁCE — VOJTĚCH HLOŽEK

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:	Vojtěch Hložek	***ID:***	98125
Ročník:	3	***Akademický rok:***	2008/2009

NÁZEV TÉMATU:

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

POKYNY PRO VYPRACOVÁNÍ:

1. Naprogramujte ve vývojovém prostředí Simatic Step7 sadu funkčních bloků pro PLC Simatic S7, které utvoří knihovnu využitelnou programátorem pro polohové 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

Cílem této bakalářské práce bylo vytvořit sadu funkčních bloků a funkcí, pomocí nichž by komunikoval programovatelný automat (PLC) Siemens S7-300 s frekvenčním měničem Siemens Sinamics S120. Komunikace mezi těmito dvěma zařízeními je zajištěna pomocí sběrnice Profinet nebo Profibus a datagramů.

Vytvořené funkční bloky slouží pro nastavení parametrů datagramů, pomocí kterých je možno ovládat na frekvenční měnič připojené motory. Funkce byly vytvořeny pro rychlé a snadné ovládání základních vlastností motoru (spuštění, zastavení atd.) .

Tato práce obsahuje popis použitého programového vybavení, technických prostředků použitých při testování a vlastních algoritmů funkčních bloků a funkcí.

Klíčová slova

Datagram, frekvenční měnič, funkce, funkční blok, PLC, Profibus, Profinet, Simatic, Sinamics S120, STEP7, S7-300.

Abstract

The objective of this Bachelor's thesis was to make a series of function blocks and functions which would establish a communication between the programmable logic controller (PLC) Siemens S7-300 and the drive system Siemens Sinamics S120 through Profinet or Profibus and datagrams.

The function blocks adjust the datagram parameters through which we can control motors connected to a drive system. The functions were made for easy and fast control of the basic motor operations (startup, shutdown, etc.) .

The thesis includes a definition of a used software, hardware and the algorithms of the function blocks and functions.

Keywords

Datagram, drive system, function, function block, PLC, Profibus, Profinet, Simatic, Sinamics S120, STEP7, S7-300.

Bibliografická citace

HLOŽEK, V. *Funkční bloky pro Simatic S7-300 s Sinamics S120 v polohových aplikacích.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 59 s. Vedoucí bakalářské práce Ing. Radek Štohl, Ph.D.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Funkční bloky pro Simatic S7-300 s Sinamics S120 v polohových aplikacích jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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. června 2009** ……
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Radku Štohlovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce a sekci Automatizace a pohony společnosti Siemens s.r.o. za poskytnuté technologie a odborné rady.

V Brně dne: **1. června 2009** ……
podpis autora

OBSAH

1. ÚVOD

Cílem této bakalářské práce bylo seznámit se s programovatelným automatem (PLC) S7-300 společnosti Siemens, frekvenčním měničem Sinamics S120 a jejich vzájemným propojením pomocí průmyslové sběrnice. Na základě spolupráce se sekcí Automatizace a pohony společnosti Siemens s.r.o. (sídlem na ulici Technická v Brně), byly konkretizovány cíle této práce a díky poskytnutí potřebného technického vybavení zde mohlo dojít k otestování funkčnosti vytvořeného programu.

Hlavním cílem bylo vytvoření funkčních bloků, pomocí nichž by komunikoval programovatelný automat S7-300 s frekvenčním měničem Sinamics S120 a které by byly programátorem využitelné v polohových aplikacích. Jako komunikační sběrnice byl vybrán Profinet, přes který si měl PLC s řídicí jednotkou frekvenčního měniče předávat data pomocí datagramu č. 110 a 111 v režimu RT (Real Time).

Dodatečnou částí práce bylo vytvoření tří funkcí podle funkčního manuálu společnosti Siemens. Ty by měly obsluze určitého technologického zařízení umožňovat snadné a rychlé ovládání základních vlastností frekvenčního měniče. Jednalo se o funkce Power, Stop a Reset.

Protože ale CPU schopné komunikovat přes sběrnici Profinet bylo k dispozici až v pozdější době, využil jsem náhradní možnosti a propojil jsem PLC s frekvenčním měničem přes průmyslovou sběrnici Profibus. Navíc nebylo vždy možné použít stejný testovací box a proto se v této práci vyskytnou dvě různé sestavy použitých PLC a frekvenčních měničů.

2. PROGRAMOVATELNÝ AUTOMAT (PLC)

2.1 HISTORIE

Za období vzniku programovatelných automatů (PLC – Programmable Logic Controller) se považuje konec 60. let minulého století[17]. Hlavním úkolem bylo především nahradit efektivnějším způsobem reléovou a později bezkontaktní logiku. Proto byla jejich architektura navržena na zpracování binárních informací a jádro tvořil bitový procesor.

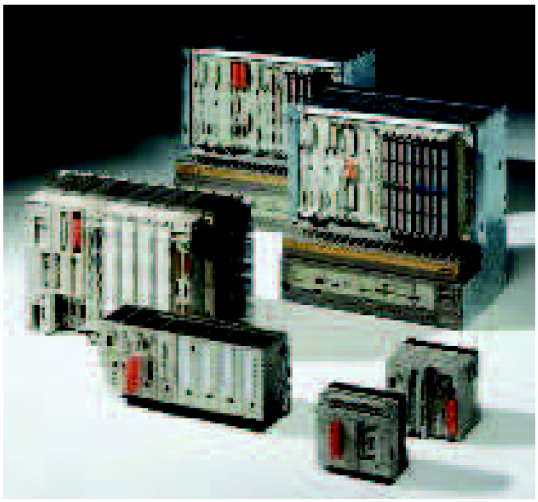
To se v 70. letech, kdy byly standardem pomalé procesory s 8 a 16bitovým slovem, jevilo jako velmi rychlé řešení. Z tohoto důvodu se na architekturu PLC kladly tyto nároky:

- bitově orientovaná CPU a paměť dat
- slovně orientovaná paměť programu
- interface pro programovací přístroj
- jednoduchý instrukční soubor a soubor speciálních funkcí (čítače, časovače atd.)

V dnešní době se však s takovýmto PLC (viz 0brázek 1) již nesetkáme a to proto, že rychlost a především cena výkonných mikroprocesorů v dnešní době umožňuje použít slovně orientovaný mikroprocesor i u nejmenších programovatelných automatů.

2.2 ZÁKLADNÍ VLASTNOSTI

V průběhu 80. let se podařilo programovatelným automatům díky rozmachu mikroelektroniky vytlačit centralizované řídicí systémy (řídicí počítače a minipočítače) a malou automatizaci (průmyslové regulátory, reléová logika) a nahradit je distribuovanou řídicí technikou[17]. Jedním z hlavních požadavků projektantů a inženýrů se tak stal požadavek na jednoduchý programovací jazyk podobný jazyku logických schémat, reléovým schématům a assembleru, na které byli vývojáři zvyklí.



Obrázek 1 Siemens Simatic S5 (1979) [13]

I když by se mohlo zdát, že použití PLC má jen samé výhody, není tomu tak. Uvedu zde proto srovnání jeho hlavních výhod a nevýhod[17].

1. Výhody použití PLC

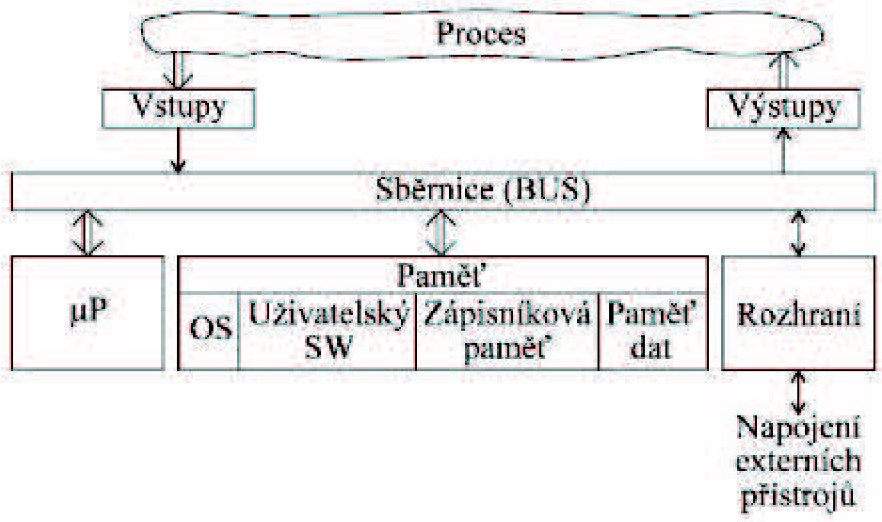
- rychlá úprava programu
- malý počet náhradních dílů
- značná flexibilita (vytvoření PLC zákazníkovi na míru)
- modularita (jednoduché rozšíření původního PLC)
- nízké náklady (velmi malé a malé automaty)
- vnitřní diagnostický okruh a možnost vytvoření vnějšího diagnostického okruhu
- jednoduché programování
- u moderních automatů možnost použití vyšších programovacích jazyků
- jednoduchý a spolehlivý operační systém reálného času
- široké spektrum výrobců a výrobků

2. Nevýhody použití PLC

- nižší komfort programování než u minipočítačů a IPC (Industrial PC)
- při ekvivalentním výkonu jakou u IPC vychází PLC dražší
- pro propojení automatů do sítí se mnohdy využívá nedostatečně standardizovaných komunikačních sběrnic
- nezbytnost hierarchické architektury při propojování do velkých celků

2.3 HARDWARE

Jak je vidět z obrázku 2, základ celého PLC tvoří sběrnice (16 nebo 32bitová), kolem které je programovatelný automat modulárně vytvořen. U prvních PLC byla paměť programu oddělena od paměti dat, od čehož se ale v průběhu vývoje ustoupilo a dnešní PLC mají jednu operační paměť. V ní je prostor rozdělen na vstupní a výstupní data, vnitřní proměnné a vlastní uživatelský program. Dále jsou v paměti uloženy i uživatelské a systémové funkční bloky a funkce.

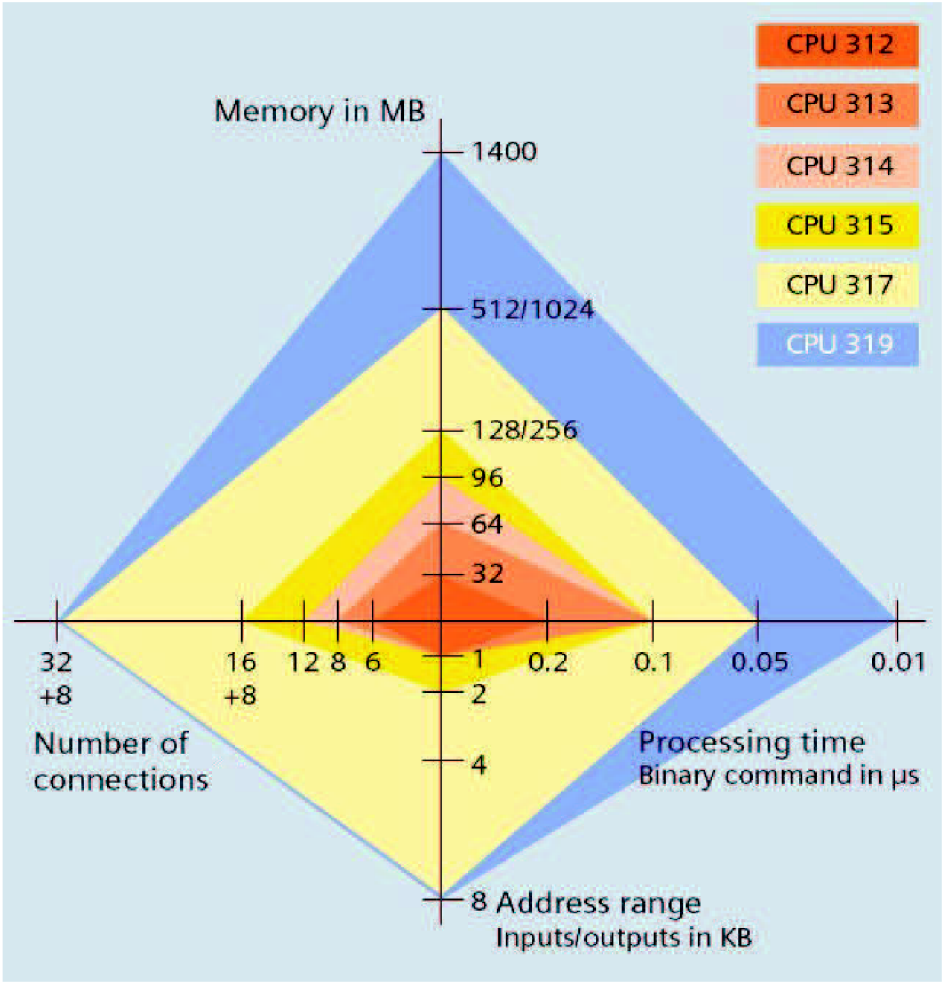


Obrázek 2 Blokové chéma modulárního PLC [16]

2.3.1 CPU

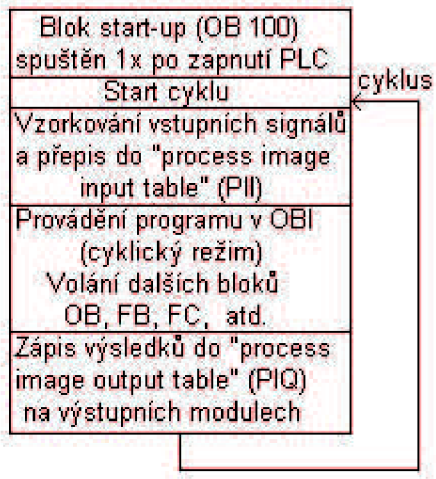
Centrální procesorová jednotka poskytuje programovatelnému automatu inteligenci, je tedy jeho mozkem, pokud bychom použili přirovnání k člověku. Realizuje soubor instrukcí a systémových služeb, zajišťuje základní komunikační funkce s vlastními i odloučenými moduly, s nadřazeným systémem i programovacím přístrojem[15].

Vyrábí se v různých provedeních v závislosti na výkonu procesoru, velikosti pracovní paměti, rozsahu vstupních/výstupních adres a počtu připojitelných modulů. Přibližný přehled verzí CPU od společnosti Siemens je zobrazen na obrázku níže.



Obrázek 3 Rozdělení CPU podle hlavních parametrů [2]

CPU se týká ještě jedna vlastnost, charakterizující PLC, a to způsob práce, který zůstal od vzniku PLC nezměněn – tzv. cyklický režim (viz Obrázek 4). Ten je v dnešní době rozšířen i o režim přerušení (od časovače či od procesu), čímž může dojít při výskytu kritické situace k obsloužení přerušení mimo cyklus. Dá se tedy mluvit o jisté formě multitaskingu.



Obrázek 4 Cyklický režim PLC [17]

Čas trvání cyklu bývá definován jako doba, kterou potřebuje PLC k načtení vstupních dat, zpracování 1000 instrukcí a odeslání výstupních dat. V dnešní době se tato doba pohybuje u moderních PLC v řádu 1-5 ms i když výkonné CPU se mohou dostat i hluboce pod 1 ms (např. Mitsubishi) [17].

Základní architektura CPU[17]**:**

- procesor s 16 nebo 32bitovým slovem
- možnost doplnění o bitový procesor (u některých středních a velkých PLC)
- bitové registry (tzv. flagy)
- paměť typu EPROM pro operační systém
- paměť RAM pro program a pro vstupně/výstupní data
- programovací a sériové rozhraní (pro komunikaci s nižší a vyšší úrovní)

2.3.2 Vstupy a výstupy PLC

Jedním z mnoha důvodů, proč se stal PLC tak oblíbeným automatizačním prostředkem je bezesporu propracovaný systém vstupních a výstupních jednotek. V dnešní době lze při výběru programovatelného automatu volit z pevně nastavené konfigurace vstupů a výstupů (tzv. kompaktní PLC) či dle potřeby přikoupit k CPU potřebný počet modulů s odpovídajícím počtem vstupů a výstupů (v/v).

Podle druhu zpracovávané informace se dělí v/v jednotky (moduly) na digitální (číslicové) a analogové[14, 16].

1. **Digitální vstupy** – bývají v klasickém či univerzálním provedení (pro střídavé i stejnosměrné napětí) a vynikají hned několika ochranami. Například galvanickým oddělením pomocí optosoučástek, RC filtrem pro odstranění poruchových signálů či diodami zabraňujícími nechtěnému přepólování a chránícím vstup před napěťovými špičkami. Využití digitálních vstupů je velmi rozmanité (informace a stavu snímače, přítomnosti objektu, stisku tlačítka). Moduly většinou obsahují 8, 16 či 32 vstupů.
2. **Digitální výstupy** – slouží především pro ovládání relátek, stykačů, indikace stavů (zapnuto, rozběh) a varování atd. Dělají se v provedení s tranzistorem (nevýkonové provedení do stovek mA) nebo s tyristorem (výkonové provedení). I digitální vstupy bývají galvanicky odděleny z důvodu zamezení průniku rušivých signálu do systému.
3. **Analogové vstupy** – jejich využití tvoří například snímání teploty odporových teploměrů (Pt100, Ni200 atd.), měření napětí, proudu a odporu. Dělí se na napěťové (pro stejnosměrné napětí, různá přesnost A/D převodu, většinou 12 bitů) a proudové (0-20 mA, 4-20 mA). Lze se setkat i s tzv. univerzálními analogovými moduly, na které lze připojit široké spektrum snímačů a měřit jak napěťové, tak proudové signály v širokém rozsahu hodnot (± 50 mV, 256 mV, 1 V, 10 V, 1 mA, 5 mA, 20 mA).
4. **Analogové výstupy** – jsou často realizovány pomocí šířkově modulovaných impulzů konstantní délky a dělí se na napěťové a proudové.

2.3.3 Napájecí zdroj

Napájecí zdroj tvoří nedílnou součást každého PLC a bývá umístěn hned vedle CPU. Mění síťové napětí (120/230 V AC) na napětí, na které je PLC stavěn (24 V DC). Výstupní proud se volí podle počtu a typů modulů, které bude zdroj napájet. Obvyklými hodnotami jsou 2, 4, 5, nebo 10 A (u velkých sestav a velkých PLC i 20 A).

2.3.4 Speciální moduly

Při použití modulárních PLC se můžeme v mnoha případech setkat s širokou nabídkou speciálních modulů, které mohou velmi významně rozšířit působnost programovatelného automatu. Dají se rozdělit do několika skupin[15]:

- **Polohovací moduly** – slouží k měření polohy a pohybu pomocí absolutních nebo inkrementálních čidel, pro nastavování a řízení polohy. Uplatnění nacházejí také při řešení úloh geometrického charakteru (měření úhlu, rychlosti, zrychlení a řízení po požadované dráze). PLC tak může v jednodušších případech nahradit CNC systém.
- **Moduly regulátorů a fuzzy logiky –** jde například o moduly schopné realizovat až 8 regulačních smyček PID regulátorů s volitelnými parametry nebo moduly využívající fuzzy logiku a fuzzy regulaci.
- **Moduly pro řízení hydraulických a pneumatických systémů**
- **Moduly pro diagnostiku**
- **Moduly pro měření a řízení teploty**
- **Komunikační moduly** – spadají sem moduly pro připojení PLC na průmyslovou sběrnici, pro vzájemné propojení programovatelných automatů, různé adaptéry mezi sběrnicemi či modemy pro bezdrátový přenos.

2.4 SOFTWARE

Jako příklad softwarového vybavení PLC uvedu to, které využívá PLC Simatic od společnosti Siemens. O programu vytvořeném uživatelem lze hovořit jako o modulárním. Využívá tři druhy programových bloků – uživatelské, systémové a standardní[4, 16]. Blok je funkcí, strukturou nebo použitím ohraničená část uživatelského programu.

1. **Uživatelské bloky** – lze je vytvářet (vyjma OB) a volat. Dělí se na:

- Organizační bloky (OB) - představují rozhraní mezi systémem a uživatelským programem. Systémový program CPU volá organizační bloky při určitých událostech (např. při přerušeních z procesu nebo od hodin). Hlavní program je obsažen v organizačním bloku OB 1, který je v normálním cyklickém režimu volán v každém cyklu. Ostatní bloky mají přidělena čísla vzhledem ke spouštěcím událostem (např. OB 100 – je volán při kompletním restartu).
- Funkční bloky (FB) - jsou parametrizovatelné. Mají paměť proměnných uloženou v některém datovém bloku. Tento datový blok je tomuto funkčnímu bloku pevně přiřazen (mluvíme pak o tzv. datovém bloku instance a kombinaci volání funkčního bloku s tímto datovým blokem nazýváme instancí).
- Funkce (FC) - slouží pro programování často se vyskytujících úloh. Jsou parametrizovatelné a vrací zpět volajícímu bloku hodnotu funkce (někdy i další výstupní parametry). Funkce neukládají žádné informace protože nemají žádné pevně přiřazené datové bloky.
- Datové bloky (DB) - obsahují data uživatelského programu. Programováním datových bloků se určuje, v jaké formě budou data ukládána (v kterém datovém bloku, v jakém pořadí a jakém typu dat).

2. **Systémové bloky –** na rozdíl od uživatelských bloků je lze pouze volat. Dělení:

- Systémové funkční bloky (SFB) – jsou to funkční bloky integrované v CPU, který mohou být volány uživatelským programem. Musí být přiřazeny k datovým blokům, které musí být uloženy v CPU jako součást uživatelského programu.
- Systémové funkce (SFC) – jedná se o předprogramované funkce integrované v CPU sloužící například pro nastavování parametrů modulů, podporu datové komunikace nebo kopírování funkcí.
- Systémové datové bloky (SDB) – jde o datové bloky obsahující nastavení systému a modulů. Vytvářejí se a mění při konfiguraci hardwaru.

3. **Standardní bloky** - vedle funkcí a funkčních bloků, které programátor vytváří, existují již hotové bloky (standardní bloky), které lze získat na některém datovém médiu nebo jsou dodávány přímo se systémem (např. matematické a aritmetické funkce, funkce pro komunikaci, PID regulátory apod.).

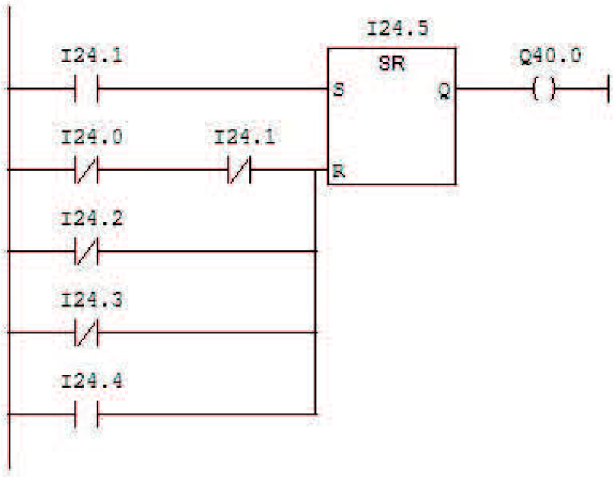
2.5 VYTVÁŘENÍ PROGRAMU

Spolu s vývojem programovatelných automatů se pro vytvoření uživatelského programu vyvinulo i několik odlišných programovacích jazyků. Ty stále ve velké míře odpovídají svému původnímu poslání – umožnit projektantům zvyklým na reléovou či bezkontaktní logiku přejít na řešení Booleovských rovnic pomocí programu.

Jazyky systémů různých výrobců jsou si podobné, ale ne stejné. Přímá přenositelnost mezi PLC různých výrobců většinou není možná. Mezinárodní norma IEC 1131-3 sjednocuje programovací jazyky PLC a určuje čtyři typy jazyků[15], které lze rozdělit do dvou hlavních skupin – na grafické a algebraické.

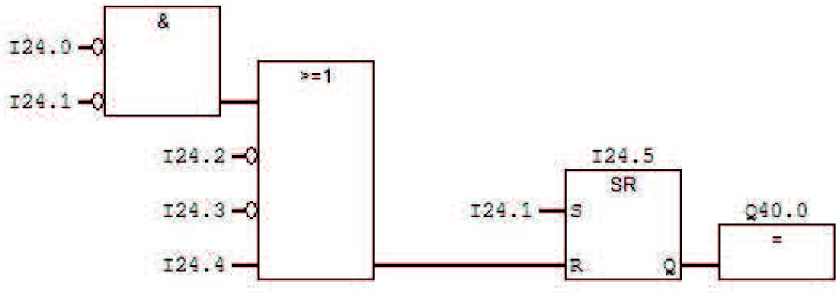
2.5.1 Grafické jazyky

- Jazyk kontaktních (reléových) schémat – neboli také Ladder Diagram (LD). Základní logické operace jsou zde zobrazeny ve formě obvyklé pro kreslení reléových schémat. Spínací kontakt je tvořen dvojicí svislých čárek, rozpínací kontakt je doplněn o lomítko, cívka je označena dvojicí závorek (Obrázek 5).



Obrázek 5 Program v jazyce LD

- Jazyk logických schémat – neboli jazyk funkčních bloků (Function Block Diagram, FBD). Základní logické operace, čítače a časovače, ale i např. aritmetické operace jsou popsány obdélníkovými značkami (Obrázek 6). Tím se tento jazyk hodí pro uživatele zvyklé na práci se schématy integrovaných obvodů.



Obrázek 6 Program v jazyce FBD

2.5.2 Algebraické jazyky

- Jazyk mnemokódů – Instruction List (IL) je obdobou assembleru u počítačů a je také strojově orientován. Princip assembleru se zřejmý například ze symbolického pojmenování návěští pro skoky, symbolických pojmenování vstupních, výstupních a vnitřních proměnných. Příklad programu v IL je na obrázku 7.

```
A     I     24.1
S     I     24.5
A(
AN    I     24.0
AN    I     24.1
ON    I     24.2
ON    I     24.3
O     I     24.4
)
R     I     24.5
A     I     24.5
=     Q     40.0
```

Obrázek 7 Program v jazyce IL

- Jazyk strukturovaného textu – Structured Text (ST) je podobný vyšším programovacím jazykům pro PC, jako je např. Pascal nebo C. Umožňuje názorný a úsporný zápis algoritmů.

Jakousi nadstavbu nad těmito čtyřmi jazyky tvoří jazyk pro sekvenční programování (SFC, GRAFCET). Ten je založen na principu přechodového grafu konečných automatů a určité třídy Petriho sítí. Využívá tedy značky stavů, přechodů a větvení. Podmínky přechodů nebo akce vykonávané v určitých stavech jsou obvykle popsány nějakým z výše uvedených jazyků.

2.6 ROZDĚLENÍ PLC A JEJICH VÝROBCI

Zvolení správné konfigurace programovatelného automatu je jedním ze základních a velmi důležitých úkonů, které musí projektant při návrhu řešené úlohy vyřešit. Na jedné straně je mnohdy požadavek na co nejnižší cenu a na druhé na maximální možný výkon zařízení a bezpečnost. Do toho se ještě začleňují vlastnosti řízeného objektu (typy a množství zpracovávaných signálů, typy ovládaných zařízení atd.) a geografické (rozlehlost řízeného celku) nebo komunikační podmínky (nutnost komunikovat s nadřazeným systémem). Proto lze typy PLC rozdělit na několik skupin[5]:

1. **Podle velikosti** – velikost není jen dána rozměry, ale hlavně výkonem celého zařízení. Tím je myšlena rychlost zpracování informací, velikost vnitřní paměti, počet čítačů a časovačů nebo maximálním možným počtem vstupů a výstupů. Obecně používané rozdělení dle velikosti je na mikro, malé, střední a velké PLC (viz Obrázek 8, 9 a 10).

Obrázek 8 Mikro PLC Siemens LOGO! [9]

Obrázek 9 Malé PLC Siemens S7-300 [9]

Obrázek 10 Střední až velké PLC Siemens S7-400 [9]

2. **Podle modularity** – dělí se na kompaktní a modulární. Kompaktní PLC jsou ta, která jsou tvořena CPU a pevně daným počtem vstupů a výstupů. Jsou určena pro řízení strojů či malých technologických zařízení. Dělají se obvykle v provedení mikro a malé PLC. Modulární PLC poskytují nesrovnatelně větší volnost při konfiguraci systému. Ten tvoří CPU a potřebné množství různých typů modulů.

3. **Podle komunikačních rozhraní** – každé PLC obsahuje základní rozhraní pomocí nějž komunikuje s programovacím přístojem. Ve velké části případů však potřebuje systém komunikovat jak s nadřazeným, tak podřazeným systémem nebo navzájem mezi několika PLC. V tomto případě se do konfigurace zahrne CPU s integrovaným potřebným rozhraním nebo je možno sestavu PLC rozšířit o komunikační modul (např. pro DeviceNet, CAN, Profibus, Profinet, AS-I).

4. **Podle hodin reálného času –** v některých aplikacích nejsou nutné, jinde být musí.

5. **Podle integrovaného displeje** – mezi hlavní výhody integrovaného displeje patří jednoduché nastavení požadovaného kroku či parametru, což znamená pro obsluhu usnadnění a zrychlení práce. Displeje mohou být s tlačítky či dotykové, grafické nebo textové a černobílé nebo barevné.

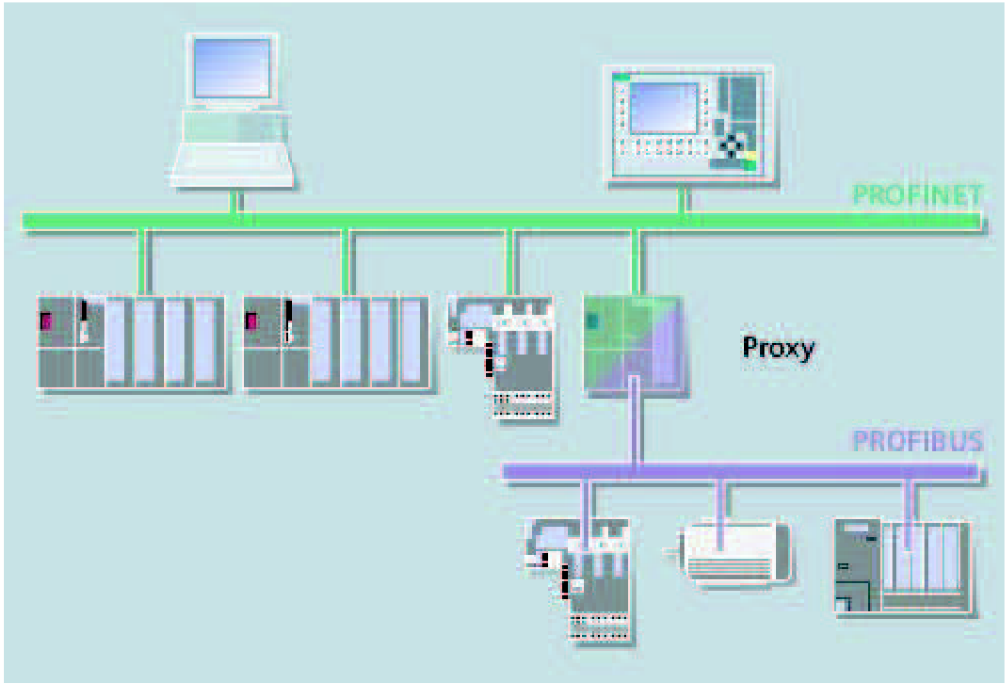
Mezi přední výrobce programovatelných automatů dnes patří tyto společnosti: ABB, Allen-Bradley, B&R, Mitsubishi, Omron, Schneider Electric, Siemens.

3. KOMUNIKAČNÍ SBĚRNICE

Jak již bylo napsáno v úvodu, v rámci této bakalářské práce jsem k propojení programovatelného automatu Siemens Simatic S7-300 a frekvenčního měniče (přesněji řečeno jeho řídicí jednotky) využil nejen moderní průmyslové sběrnice Profinet, ale i neméně známé a roky prověřené sběrnice Profibus.

3.1 PROFINET

Profinet (IEC 61158) pracuje na bázi Ethernetu (IEEE 802.3), vyhovuje ale daleko vyšším nárokům na průmyslovou automatizaci, jako jsou např. schopnost práce v reálném čase, integrace distribuované přístrojové techniky, rychlé instalační metody atd. Využívá také osvědčených standardů IT jako např. TCP/IP. Nabízí integraci stávajících řešení na bázi průmyslových sběrnic Profibus (Obrázek 11), Interbus, AS-Interface atd. a tím zajišťuje plnou podporu, kontinuitu a ochranu všech dosavadních řešení[10].



Obrázek 11 Propojení Profinetu a Profibusu pomocí proxy jednotky [10]

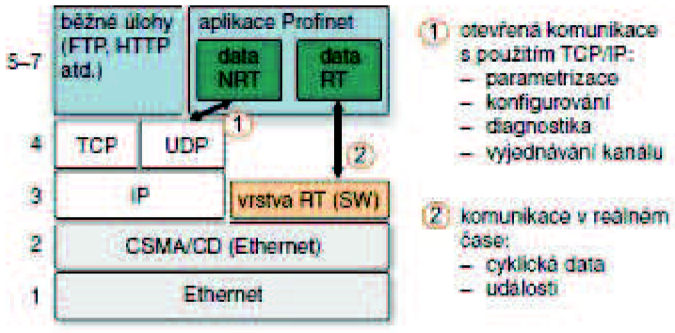
3.1.1 Varianty Profinetu

V základě lze rozlišovat podle typu úloh dvě varianty Profinetu. Tou první je Profinet IO (Input/Output) a druhou Profinet CBA (Component Based Automation).

1. **Profinet IO** – používá se pro přímé připojení distribuovaných periferií a zařízení v technologickém provozu k průmyslovému Ethernetu. Koncepce sítě je typu producent – konzument a v síti se vyskytují tři typy uzlů. Jde o řídicí stanici (Controller), řízenou stanici (Device) a inženýrskou stanici (Supervisor). Profinet IO musí umět zajišťovat tyto funkce[18]:

- cyklickou výměnu dat mezi řízenými a řídicími moduly
- přenos s velkou prioritou a potvrzování varovných hlášení
- přenos acyklických dat v režimu bez reálného času (tzv. režim NRT)
- výměnu dat mezi řízenými stanicemi bez zásahu stanic řídicích
- synchronizaci stanic pracujících v režimu RT
- automatické přidělení adres zařízením

Princip komunikace je vysvětlen na obrázku 12 (čísla nalevo odpovídají vrstvám modelu ISO/OSI). Z něj je zřejmé, že část komunikačních úloh, která nevyžaduje přenos dat v reálném čase (konfigurace, parametrování) využívá kanálu TCP/UDP/IP. Ten je typu NRT (Non Real Time). Naopak úlohy vyžadující režim RT (Real Time) kanál TCP/UDP/IP paralelně obcházejí.

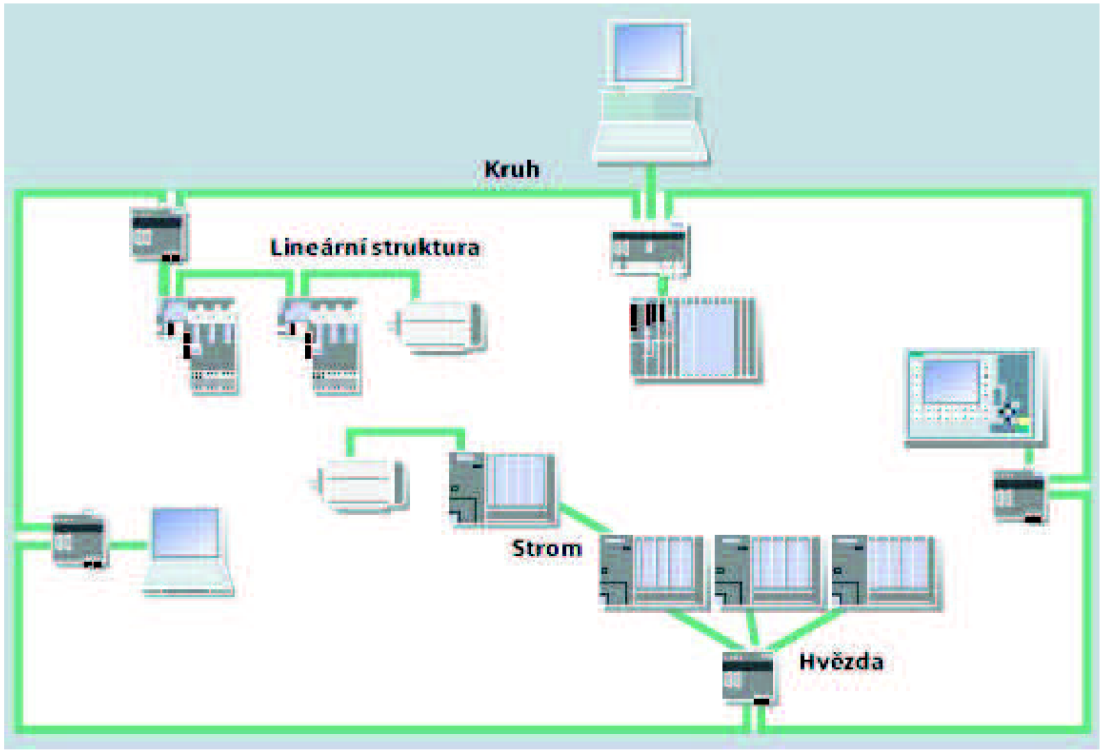


Obrázek 12 Komunikační model Profinetu IO [18]

2. **Profinet CBA** – tato verze Profinetu definuje jeden pohled na funkci automatizačního zařízení. Tím je myšleno to, že automatizační celek lze rozdělit na určitý počet komponent a ty poté pomocí jazyka XML popsat. Vznikne tak tzv. PCD (Profinet Component Description) soubor, v němž jsou obsaženy všechny údaje o komunikujících zařízeních. Komunikační systém pak v této síti není vytvářen psaním komunikačních programů, ale projektováním spojení s využitím databáze PCD. Obdobně jako Profinet IO i Profinet CBA je typu producent-konzument[18].

3.1.2 Topologie sítě a rozšířené služby

Jelikož Profinet vychází ze sítě Ethernet, lze u něj využít stejných topologií. Mezi nejčastější patří především hvězda, lineární struktura, strom a kruh[10]. Samozřejmostí (a v praxi využívanou) vlastností je kombinace více těchto topologií navzájem, jak ukazuje obrázek 13.



Obrázek 13 Kombinace topologií u Profinetu [10]

Pro spojení mezi zařízeními lze použít metalických kabelů (stejných jako u Ethernetu) nebo optických kabelů.

Větší flexibility v rámci automatizačního procesu lze docílit použitím komponent pro bezdrátovou síť (Wireless LAN, WLAN). Ta umožní připojit na Profinet zařízení, která jsou v pohybu nebo ke kterým by bylo složité přivést kabely.

V rámci správy sítě lze provádět jak spravování IP adres, tak vlastní diagnostiku sítě. Díky tomu, že Profinet podporuje technologie HTTP, XML, HTML či skriptování, může být přístupný i webovým klientům[10].

3.2 PROFIBUS

Komunikační sběrnice Profibus (IEC 61158/ EN 50170/ DIN 19245) je zavedeným pojmem v automatizaci více než 10 let a jsou pro něj stále vyvíjeny nové specifikace a rozšíření. Např. PROFIdrive pro řízení pohonů, přenos časových značek a PROFIsafe pro komunikaci vyhovující většině známých bezpečnostních požadavků. Obecně je sběrnicový systém Profibus určen pro nižší až střední rozsah komunikační výkonnosti[7]. Přenosovým médiem je stíněná kroucený pár (standard RS 485) nebo optický kabel (skleněná nebo plastová vlákna). Přenosová rychlost je až 12 Mbit/s.

3.2.1 Varianty Profibusu

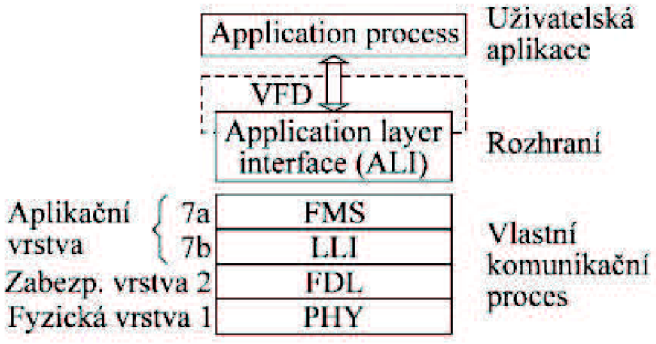
1. **Profibus DP** – jedná se o nejrozšířenější a nejjednodušší variantu, určenou pro rychlou komunikaci typu Master-Slave. Je vhodný především pro rychlý přenos signálů z procesu pomocí decentralizovaných periferií a odloučených v/v jednotek. Zkratka DP znamená Decentralized Periphery[3].

2. **Profibus PA-** u této varianty je zohledňován především požadavek na bezpečnost a ne na maximální rychlost, protože je určen pro řízení procesů v místě nebezpečí výbuchu (podporuje standard IEC 1158-2). Zařízení připojená na tuto sběrnici jsou napájena přímo ze sběrnice. Zkratka PA znamená Process Automation[3].

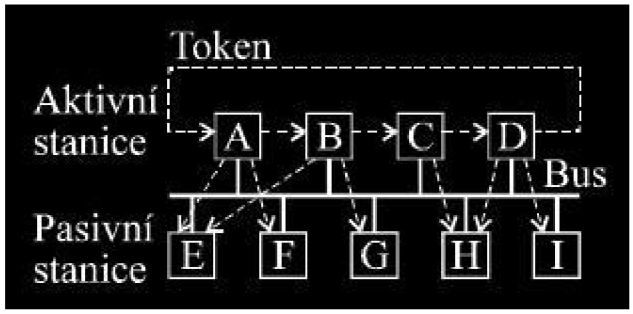
3. **Profibus FMS** - používá se pro heterogenní komunikaci na vyšší úrovni. Nabízí širokou nabídku služeb pro práci s daty, programy a alarmy, umožňuje vzájemnou komunikaci automatizačních členů různých výrobců. Jedna se o nejméně rozšířenou variantu Profibusu. Zkratka FMS znamená Fieldbus Message Specification[3].

3.2.2 Komunikační model a přístup k síti

Jako mnoho sítí typu fieldbus i Profibus implementuje pouze několik vrstev modelu ISO/OSI. Jedná se o první (fyzickou) vrstvu, druhou (zabezpečovací) vrstvu a sedmou (aplikační) vrstvu, jak je zřejmé z obrázku 14. Nad sedmou vrstvou je tzv. komunikační rozhraní ALI (Aplication Layer Interface) zajišťující komunikujícím zařízením přístup k tomuto modelu. Druhá vrstva určuje také metodu přístupu k síti. Ta je Token Bus s přístupem Master – Slave[17] (viz Obrázek 15).



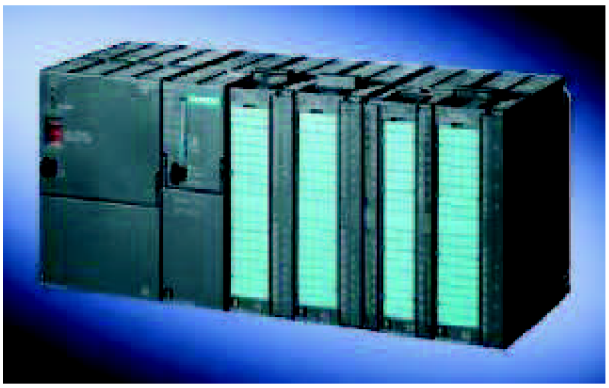
Obrázek 14 Komunikační model Profibusu [16]



Obrázek 15 Přístup k síti Profibus [16]

4. PLC SIEMENS SIMATIC S7-300

Programovatelný automat Siemens Simatic S7-300, jehož možná konfigurace je zobrazena na obrázku 16, je jedním z nejrozšířenějších řídících systémů z nabídky společnosti Siemens. Jak vyplývá ze zadání mé bakalářské práce, tvoří toto PLC hlavní technologické zařízení, se kterým jsem se zabýval a proto zde uvedu jeho krátkou charakteristiku.



Obrázek 16 Možné provedení Simatic S7-300 [9]

4.1 TYPICKÉ APLIKACE

Poskytuje univerzální automatizační platformu pro systémová řešení s hlavním důrazem na výrobní technologii (automobilový průmysl, výroba strojů a zařízení, balicí technika, potravinářský průmysl, výroba a rozvody energie atd.)[6].

4.2 ROZDĚLENÍ CPU

Jádrem programovatelného automatu řady S7-300 je jednotka CPU, která zpracovává uživatelský program, doplněná o napájecí zdroj a sestavu přídavných karet (modulů). CPU lze rozdělit na několik typů podle různých požadavků aplikace[6]:

- **Standardní** - s komunikačním prostředím MPI a Profibus (resp. Profinet) – například 317-2DP.
- **Kompaktní** (označené písmenem C) - doplněné digitálními a analogovými v/v a nejčastěji vyžadovanými základními technologickými funkcemi (rychlé čítání, měření frekvence, polohování, PID regulace). Příkladem je 314C-2DP.
- **Bezpečnostní** (označené písmenem F) **-** používají se všude tam, kde je třeba zajistit co nejvyšší stupeň bezpečnosti obsluhy, výrobního zařízení či okolního prostředí.
- **Technologické** (označené písmenem T) – v tomto CPU jsou přímo začleněny výkonné technologické funkce a funkce pro řízení polohy a pohybu. Je navrženo pro dynamické řízení pohybu v několika osách současně.

4.3 ROZŠIŘUJÍCÍ MODULY

Všechny moduly lze jednoduše instalovat na profilovou lištu. K dispozici je široká paleta různých modulů, které umožňují projektantovi vytvořit potřebnou konfiguraci sestavy a optimálně se tak přizpůsobit parametrům řízeného procesu či technologie. Jedná se o[6]:

- **Signálové moduly** (SM) - DI/DO, AI/AO pro všechny typy běžných procesních signálů (včetně verze „Ex“ – provedení pro výbušná prostředí).
- **Moduly rozhraní** (IM) - pro víceřadá uspořádání
- **Funkční moduly** (FM) - pro zpracování komplexních nebo časově kritických procesů nezávisle na CPU, např. rychlé čítání, měření frekvence, polohování, PID algoritmy.
- **Komunikační procesory** (CP) - zprostředkovávají propojení různých sběrnicových systémů nebo spojení s dalšími přístroji sériovou linkou

5. SOFTWAROVÉ VYBAVENÍ

Jak již vyplývá z názvu programovatelný automat, jedná se o zařízení, které je pro uvedení do chodu nutno naprogramovat. A k tomu je potřeba mít nainstalován odpovídající software. V rámci mé bakalářské práce se to týkalo především dvou programů (STARTER a STEP7) a jedné doplňkové sady (Drive ES) od společnosti Siemens.

Program STARTER si lze z webových stránek společnosti Siemens volně stáhnout a ke spuštění nepotřebuje žádnou licenci. STEP7 spolu s Drive ES byl dodán na DVD vedoucím sekce Automatizace a pohony Siemens s.r.o., Ing. Dočkalem. A to spolu s licenčním klíčem pro STEP7.

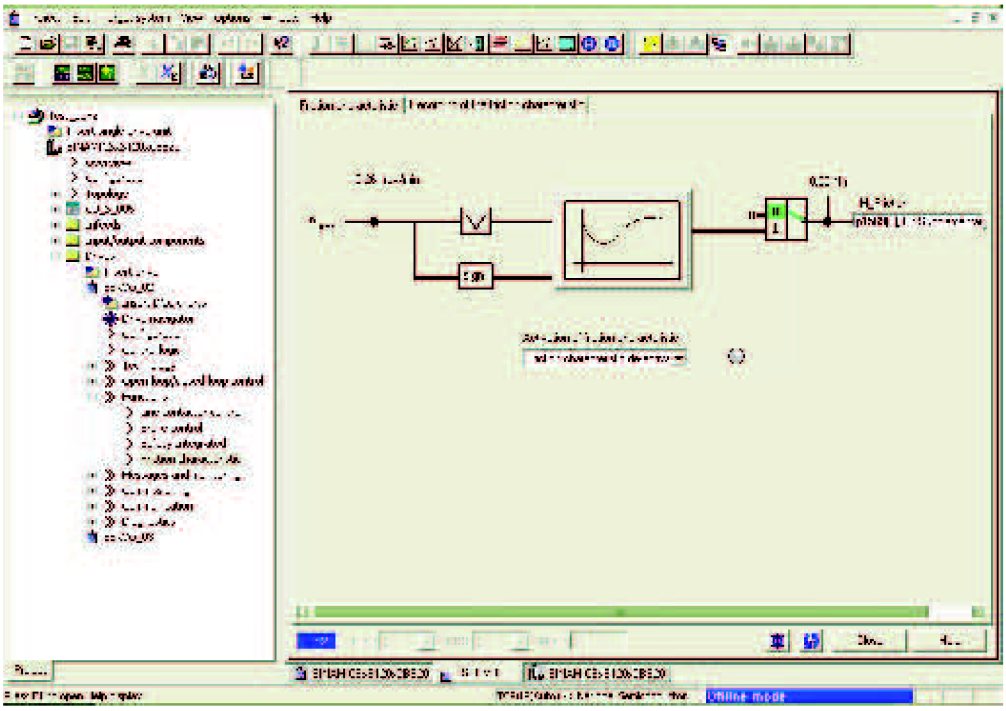
Teprve po nainstalování všech těchto komponent mohlo dojít k praktické realizaci bakalářské práce. V průběhu tohoto semestru jsem navíc ještě provedl update programu STARTER na nejnovější verzi, aby byl schopen pracovat s datagramem č.111.

5.1 STARTER

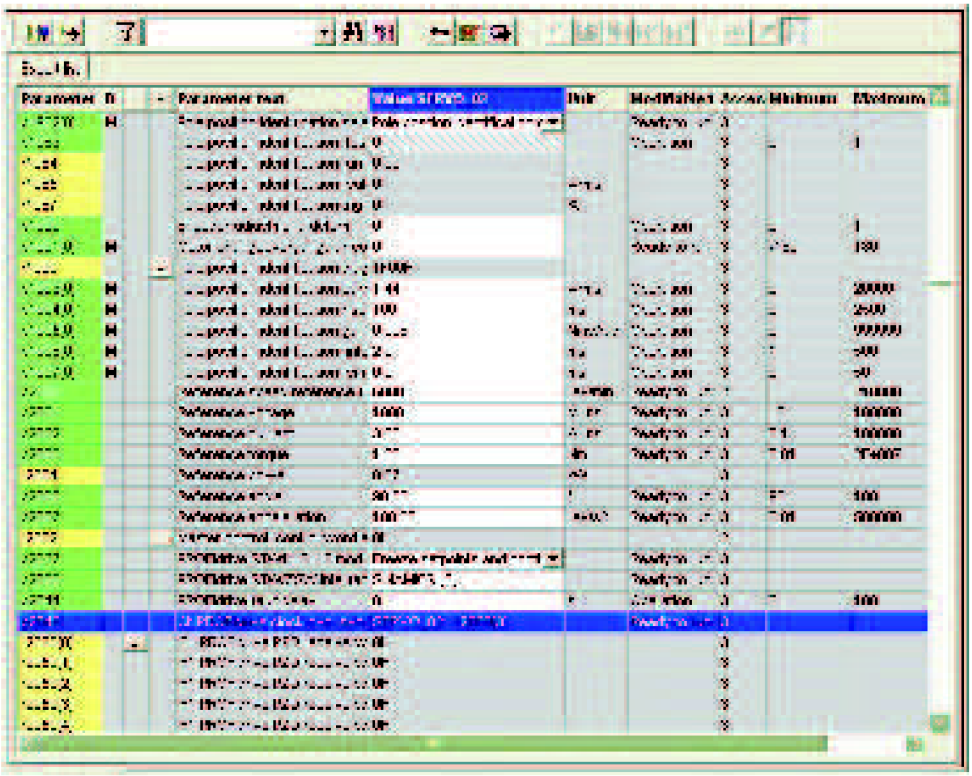
Tento program poskytuje podporu pro nastavení parametrů frekvenčních měničů typu Sinamics a Micromaster. Obsahuje mnoho integrovaných testovacích funkcí, které podporují optimální nastavení pohonů pro různé pracovní nasazení. Dále jsou také implementovány grafická znázornění vlastností měniče (Obrázek 17), které umožňují jednoznačnou diagnostiku pro rychlou orientaci uživatele a kompletní monitoring (např. funkce osciloskopu)[11].

Veškeré parametry měniče se nastavují v tzv. Expert listu (Obrázek 18), což je jakýsi detailní soupis parametrů, funkcí frekvenčního měniče a jejich vzájemných vazeb. Nachází se zde také údaje o limitních hodnotách veličin.

Při pouhém nastavování parametrů měniče lze STARTER používat jako samostatnou aplikaci pod MS Windows, přičemž rozhraní pro komunikaci s měničem je Profibus DP. Při práci v programovacím prostředí pro PLC Simatic – v systému STEP 7, je software STARTER začleněn do prostředí Simatic Manager.



Obrázek 17 Sledování brzdné charakteristiky v programu STARTER



Obrázek 18 Expert list

5.2 STEP 7

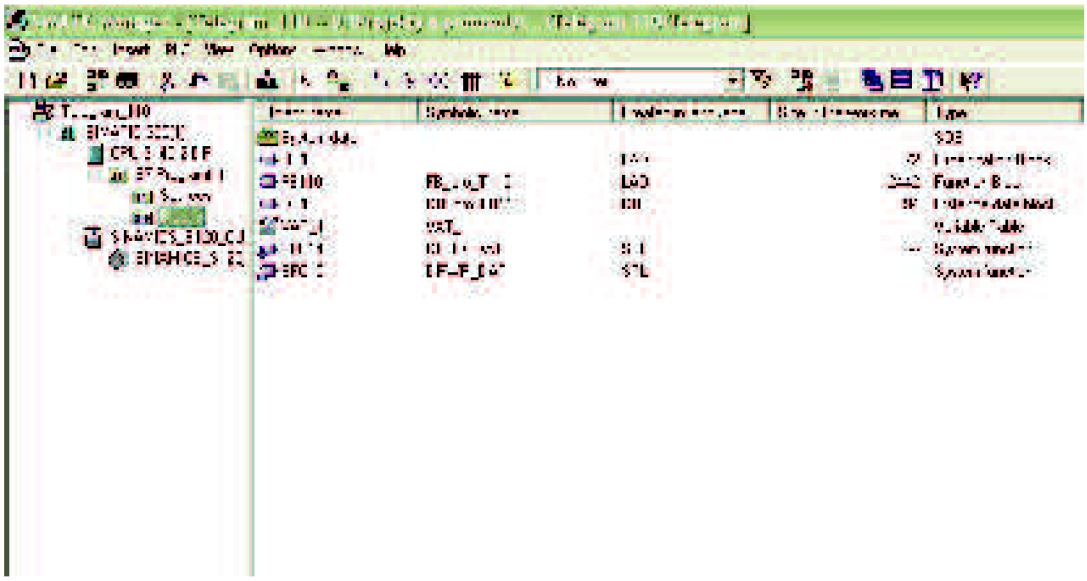
Tvoří základní software pro konfiguraci a programování PLC Simatic. Obsahuje výkonné nástroje a funkce pro řadu úloh spojených s automatizačními projekty. Nabízí uživatelsky příjemný způsob práce ve všech fázích vývoje projektu jakými jsou obvykle konfigurace a parametrizování hardwaru, definování komunikace, programování, testování a oživování projektu, servis, správa dokumentace a archivování, provozní a diagnostické funkce[8].

Program STEP7 obsahuje především prostředí Simatic Manager, programový editor, editor symboliky pro správu globálních proměnných, nástroj pro konfiguraci hardwaru (tvz. HW Config), diagnostiku hardwaru a program NetPro pro nastavení datových spojení přes komunikační rozhraní. Mezi mnou nejčastěji využívané součásti STEP7 patřil Simatic Manager a programový editor.

5.2.1 Simatic Manager

Slouží k integrované správě všech nástrojů a dat daného projektu. Jde o jakousi základnu (viz Obrázek 19), odkud programátor přistupuje ke všem potřebným nástrojům (např. STARTER, programový editor, NetPro, HW Config a mnohé další) a kde vzniká každý automatizační projekt.

Odtud je do paměti PLC nahrán řídicí program skládající se z uživatelských, systémových a resp. standardních bloků a funkcí[12].



Obrázek 19 Simatic Manager

5.2.2 Programový editor

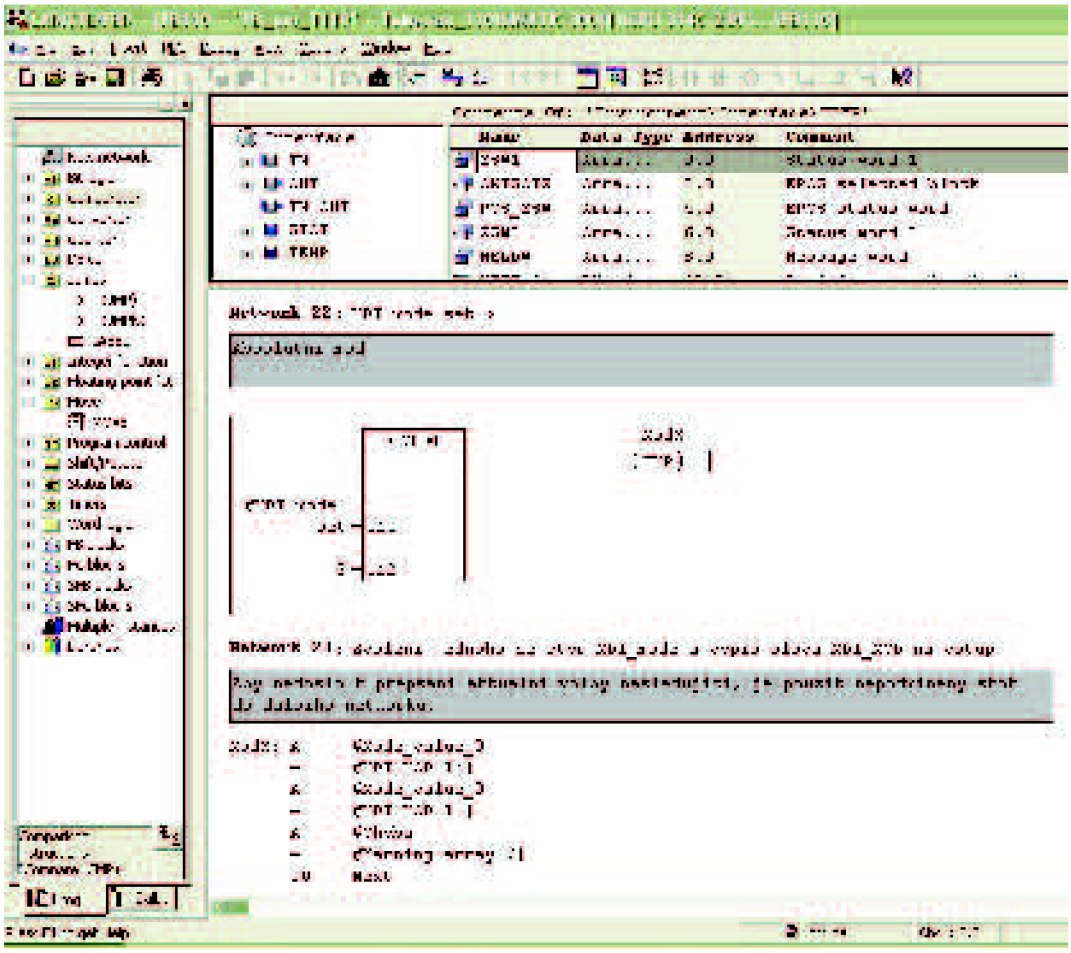
Slouží k vlastnímu vytváření uživatelských funkčních bloků (FB) a funkcí (FC), které tvoří základní prvky uživatelského programu a umožňují rozdělit komplexní program do dílčích, vzájemně propojených částí.

Programátor zde vytváří řídicí logiku (Obrázek 20), pomocí které programovatelný automat řídí připojené vstupy a výstupy, měniče, motory apod. Zápis logiky je možný ve třech jazycích (všechny dle normy IEC 1131-3):

- **LAD** (Ladder diagram, LD) - čili jazyk reléových logických schémat. Ten vychází z kdysi v automatizaci používané reléové logiky.
- **STL** (Statement list, obdoba IL) - něm se programuje pomocí instrukcí (mnemokódů).
- **FBD** (Function Block Diagram) - neboli programování pomocí funkčních bloků. Ten je v jisté míře podobný jazyku LAD.

Každá část programu zapsaná v LAD/FBD je převeditelná do STL. Naopak to platí také, ale pouze v případě, kdy je dodržována správná syntaxe psaní instrukcí (např. zápis časového členu v STL musí obsahovat nejprve instrukci načtení a pak instrukci nastavení časové funkce).

Výjimku tvoří případy, kdy se v jazyce LAD nebo FBD nevyskytuje ekvivalentní grafický popis (ikona) příkazu jazyka STL. V takovémto případě zůstává neekvivalentní část programu v STL a zbytek programu je převeden do požadovaného jazyka.



Obrázek 20 Vytváření programu v Programovém editoru

6. DALŠÍ TECHNICKÉ VYBAVENÍ

Dalším důležitým technologickým prvkem této bakalářské práce je kromě PLC Simatic S7-300 (viz kapitola 4) také frekvenční měnič Siemens Sinamics S120. Ten bývá spolu s ovládanými servomotory umístěn v tzv. testovacím boxu.

6.1 TESTOVACÍ BOX

Jedná se o snadno přepravitelný hliníkový box, pomocí kterého se testují programy pro řízení frekvenčních měničů Sinamics S120 a tedy velkého množství pohonových aplikací. Obsahuje již zmíněný měnič a dvojici servomotorů. Pro manuální ovládání je k dispozici ovladač s možností ručního nastavení hodnot analogových i digitálních hodnot vstupů.

6.1.1 Frekvenční měnič Sinamics S120

Jeho koncepce je navržena pro řízení náročných aplikací především v oblasti pohonů a výrobních strojů a robotů. Zvládne dynamické polohování a synchronizaci pohybů pro více os. Může pracovat samostatně nebo mu může být nadřazeno PLC z řady Simatic[11].

Frekvenční měnič (viz Obrázek 21) se skládá ze dvou hlavních částí. Jsou jimi řídicí jednotka (Closed-loop control) CU320, která je schopná řídit až čtyři výkonové členy, a dvojitý výkonový člen (Power unit) řídicí dva servomotory. Řídicí jednotka provádí výpočty pro řízení výkonového členu a současně poskytuje rozhraní pro komunikaci s nadřazeným systémem. Výkonový člen (též nazývaný motorový modul) napájí připojené motory.

Typickým příkladem použití tohoto frekvenčního měniče jsou stroje a linky na balení materiálu a výrobků, stroje pro sklářský a dřevařský průmysl, lisy, manipulační a zdvihací zařízení, montážní a testovací linky[11].

Obrázek 21 Sinamics S120 [1]

6.1.2 Servomotory

V testovacím boxu byly k názorné simulaci pohonů použity dva servomotory Siemens 1FK7. Ty nachází uplatnění v mnoha průmyslových aplikacích s vysokými nároky na přesnost polohy při vysoké dynamice[11]. Pro představu, jak takový servomotor vypadá, je zobrazen na obrázku 22.

Obrázek 22 Servomotor 1FK7 [1]

7. SESTAVY POUŽITÝCH ZAŘÍZENÍ

7.1 PRO KOMUNIKACI PŘES PROFIBUS

V tomto případě komunikovalo PLC s frekvenčním měničem pomocí sběrnice Profibus DP rychlostí 1,5 Mbit/sec po stíněném krouceném páru.

Konfigurace PLC Simatic S7-300:

- Napájecí zdroj PS (kat.č. 6ES7307-1KA01-0AA0)
- CPU 314C-2 DP (kat.č. 6ES7314-6CF02-0AB0, firmware V2.0), který disponuje pracovní pamětí 64kB, zaváděcí pamětí až 8MB pomocí karty MMC, rychlostí zpracování 0,1 ms na 1000 instrukcí, 24DI/16DO, 5AI/2AO, čtyřmi výstupními kanály pulzně-šířkové modulace (2,5 kHz), čtyřmi kanály pro čítací a měřicí funkce, integrovanou polohovací funkcí, komunikačním rozhraním MPI a Profibus, možnost připojení až 31 slave modulů.
- Signálový modul SM 323 - 8DI 24V + 8DO 24V/0.5A (kat.č. 6ES7323-1BH00-0AA0)

Konfigurace frekvenčního měniče Sinamics S120:

- Řídicí jednotka CU320 (kat.č. 6SL3040-0MA00-0A00)
- Terminálová karta TB30 (kat.č. 6SL3055-0AA00-2TA0) pro připojení více vstupů/výstupů na řídicí jednotku (2AI/2AO, 4DI/4DO)
- Dvojitý výkonový člen (kat.č. 6SL3120-2TE13-0AA3) o výkonu 2x1,6 kW
- Dvojice servomotorů 1FK7 (oba kat.č. 1FK7022-5AK71-1LG0) z nichž každý dosahoval maximálně 6000 ot/min a krouticího momentu 0,6 Nm.

7.2 PRO KOMUNIKACI PŘES PROFINET

Médiem, přes které komunikace pomocí sběrnice Profinet IO probíhala, byl osmižilový UTP kabel zakončený koncovkami RJ45. Maximální rychlost přitom byla 100 Mbit/sec. Komunikace probíhala v RT režimu.

Konfigurace PLC Simatic S7-300 (jednotka ET200S):

- Integrovaný napájecí zdroj PM-E (kat.č. 6ES7138-4CA01-0AA0)
- CPU IM151-8 PN/DP (kat.č. 6ES7151-8AB00-0AB0, firmware V2.7), který disponuje pracovní pamětí 128kB a rychlostí zpracování 0,3 ms na 1000 instrukcí. Je schopen připojení na Profinet IO i CBA a disponuje třemi Profinet porty. Pro verzi IO se může jednat o řídicí stanici (Controller) a pro verzi CBA i o proxy-jednotku. Je možno využít i RT (Real Time) režim.
- Integrované vstupy a výstupy: 8DI 24V (kat.č. 6ES7131-4BF00-0AA0) a 8DO 24V/0.5A (kat.č. 6ES7132-4BF00-0AA0)

Konfigurace frekvenčního měniče Sinamics S120:

- Řídicí jednotka CU320 (kat.č. 6SL3040-0MA00-0A00)
- Komunikační karta CBE20 (kat.č. 6SL3055-0AA00-2EB0) se čtyřmi porty pro připojení na Profinet IO
- Dvojitý výkonový člen (kat.č. 6SL3120-2TE13-0AA3) o výkonu 2x1,6 kW
- Dvojice servomotorů 1FK7 (kat.č. 1FK7022-5AK71-1LG0 a 1FK7022-5AK71-1AG0) z nichž každý dosahoval maximálně 6000 ot/min a krouticího momentu 0,6 Nm.

8. ROZBOR ZADÁNÍ

8.1 DATAGRAM

Jak již bylo uvedeno v úvodu, komunikace mezi PLC a frekvenčním měničem bude probíhat pomocí tzv. datagramů. Pojmem datagram je myšlen datový rámec konstantní délky, pomocí kterého si řídicí a řízený objekt navzájem vyměňují požadované informace. Každý datagram přitom pracuje s určitým počtem vstupních a výstupních datových slov (tzv. wordů). Vstupním slovem je myšleno to, které jde z řídicího do řízeného objektu, výstupním poté to v opačném směru.

Každý word je tvořen 16 bity (v případě doubleword - dvojitého slova tedy 32 bity) a má svůj specifický význam a pořadí, které nelze zaměnit. Krátký přehled některých datagramů společnosti Siemens a jejich datových slov (tzv. PZD) je uveden na obrázku 23. Jen pro upřesnění uvedu, že společnost Siemens používá ve svých programech i manuálech místo pojmu datagram pojem telegram (Telegramm).

Telegramm:	102		103		105		106		110		111		116	
Appl. - Class	1, 4		1, 4		4 DSC		4 DSC		3		3		4 DSC	
PZD 1	STW1	ZSW1	STW1	ZSW1	STW1	ZSW1	STW1	ZSW1	STW1	ZSW1	STW1	ZSW1	STW1	ZSW1
PZD 2	NSOLL_B	NIST_B	NSOLL_B	NIST_B	NSOLL_B	NIST_B	NSOLL_B	NIST_B	SATZANW	AKTSATZ	POS_STW1	POS_ZSW1	NSOLL_B	NIST_B
PZD 3									POS_STW	POS_ZSW	POS_STW2	POS_ZSW2		
PZD 4	STW2	ZSW2	STW2	ZSW2	STW2	ZSW2	STW2	ZSW2	STW2	ZSW2	STW2	ZSW2	STW2	ZSW2
PZD 5	MOMRED	MELDW	MOMRED	MELDW	MOMRED	MELDW	MOMRED	MELDW	OVERRIDE	MELDW	OVERRIDE	MELDW	MOMRED	MELDW
PZD 6	G1_STW	G1_ZSW	G1_STW	G1_ZSW	G1_STW	G1_ZSW	G1_STW	G1_ZSW	MDI_TAR POS	KÜST_A	MDI_TAR POS	KÜST_A	G1_STW	G1_ZSW
PZD 7		G1_KÜST1	G2_STW	G1_KÜST1	KÜRH	G1_KÜST1	G2_STW	G1_KÜST1	MDI_VELO CITY		MDI_VELO CITY	NIST_B	G2_STW	G1_KÜST1
PZD 8							KERR						KERR	
PZD 9		G1_KÜST2		G1_KÜST2	KPC	G1_KÜST2		G1_KÜST2	MDI_ACC		MDI_ACC	FAULT_CODE	KPC	G1_KÜST2
PZD 10							KPC		MDI_DEC		MDI_DEC	WARN_CODE		
PZD 11				G2_ZSW				G2_ZSW	MDI_MODE		<3>	<3>		G2_ZSW
PZD 12				G2_KÜST1				G2_KÜST1						G2_KÜST1
PZD 13														
PZD 14				G2_KÜST2				G2_KÜST2						G2_KÜST2
PZD 15														
PZD 16														ÄNST_GLÄTT
PZD 17														NSOLL_GLÄTT
PZD 18														NIST_GLÄTT
PZD 19														STÄT_GLÄTT
PZD 20														

Obrázek 23 Příklad komunikačních datagramů [4]

8.2 FUNKČNÍ BLOK

Aby mohla komunikace mezi zařízeními pomocí datagramů cyklicky probíhat, je nutné navrhnout funkční blok (FB), kterému bude přiřazen jeho instanční datový blok (DB). Následně se vytvoří v organizačním bloku OB1 instance tohoto FB, pomocí které se budou moci nastavovat parametry ovládající frekvenční měnič.

Z pohledu uživatele je FB takový „black box“ s mnoha vstupy a výstupy. Co je uvnitř uživatele většinou nezajímá. Jeho požadavkem je stoprocentní funkčnost a bezchybnost při komunikaci.

8.2.1 Datová slova

Prvním krokem při vymýšlení koncepce FB bylo zjištění, jaká slova a v jakém pořadí datagram obsahuje. Tuto informaci lze nejlépe zjistit v programu STARTER v záložce Communication – Profibus u daného servomotoru.

Datová slova lze z hlediska významu bitů rozdělit na ta, která jsou tvořena 16-ti bity, z nichž každý má svůj unikátní význam (povoluje operaci, definuje styl polohování atd.) a na ta, u kterých tvoří kombinace bitů (ať 16 nebo 32) číselnou hodnotu, obvykle vyjádřenou hexadecimálně. Ty pak určují hodnotu rychlosti, zrychlení, zpomalení a také třeba vzdálenost, která se má ujet.

8.2.2 Jednotka vzdálenosti

Jelikož mým cílem bylo vytvořit funkční blok pro polohové aplikace, musel jsem si zvyknout na skutečnost, že se zde vzdálenost neudává v metrech nebo milimetrech a rychlost v metrech za sekundu, nýbrž v tzv. LU (Length Unit) a LU/min. LU je jednotka míry nastavitelná u každého servomotoru v záložce Technology – Position Control – Mechanics, která nám umožňuje převádět otáčku motoru na konkrétní vzdálenost v závislosti na nastavitelném rozlišení (LU per load revolution v téže záložce). Pokud tedy nechám nastaveno standardních 10000 LU/ot. a vím, že na hřídeli motoru je umístěn válec o průměru 1 m, poháněj ící např. pás, pak při jedné otáčce motoru tento pás ujede π*1 m. Pak je tedy 1 LU rovna 314 μm .

8.2.3 Interface

Každý funkční blok má tzv. Interface tvořený vstupy IN, výstupy OUT, vstupně-výstupními proměnnými IN_OUT, statickými proměnnými STAT a dočasnými proměnnými TEMP. Při každém cyklu PLC je sejmuta hodnota vstupů, nahrána do TEMP, a až teprve poté je s touto hodnotou v programu pracováno. V případě výstupů je princip opačný. Výsledek logické operace je nejdříve nahrán do TEMP a až poté je odeslán na fyzický výstup. Statické proměnné jsou v úvodu programu nastaveny na své hodnoty a ty se již při běhu programu nemění.

8.3 FUNKCE

Všechny funkce jsem měl navrhnout tak, aby uživateli (především obsluze zařízení) umožnily rychlé a jednoduché ovládání vybraných vlastností frekvenčního měniče Sinamics (a na něj napojeného motoru). A to v případě, že PLC komunikuje s frekvenčním měničem přes datagram č. 1, 2, 9, 110 nebo 111.

Princip práce bude spočívat v nastavování a čtení vybraných parametrů datového bloku, který tvoří instanční datový blok funkčního bloku. Tento funkční blok přitom bude sloužit pro komunikaci přes jeden z výše uvedených pěti datagramů.

Každá funkce bude povinně obsahovat dva vstupy – název datového bloku se kterým bude pracovat a vstup pro vykonání funkce. Mezi povinné výstupy bude patřit stav ve kterém se funkce nachází (funkce je prováděna nebo je dokončena) a také informace o chybě a jejím původu.

Jelikož funkce je uživatelský blok, který nemá vlastní DB, je ochuzen o možnost definování statických proměnných. Má ale navíc návratovou hodnotu RETURN. V ostatních vlastnostech je Interface funkce shodný s tím z FB.

9. NÁVRH VLASTNÍCH PROGRAMŮ

Při programování funkčních bloků a funkcí jsem jako primární programovací jazyk použil LAD. V některých částech funkčních bloků je navíc doplněn o jazyk STL a to z důvodu přesunu hodnot konkrétních bitů některých datových slov. Tato operace by v LAD zabrala příliš mnoho místa a program by nebyl tak přehledný.

Názvy vstupů FB (ovládajících měnič) a výstupů FB (které zobrazují hodnoty a stavy paramterů měniče) a jejich datový typ jsem zvolil podle hodnot, kterých mohou nabývat a sekundárně také podle toho, aby byly pro uživatele co nejvíce logické. U funkcí to bylo poté podle funkčního manuálu Siemens a požadavků společnosti Siemens s.r.o.[14].

V následujícím textu někdy použiji pojem network. Tím je myšlena jakási sekce programu mající svůj specifický účel. Každý network má své číslo, může mít název a lze k němu ještě připsat komentář. Slouží k rozdělení programu na menší celky a tím k jeho zpřehlednění.

9.1 FB110

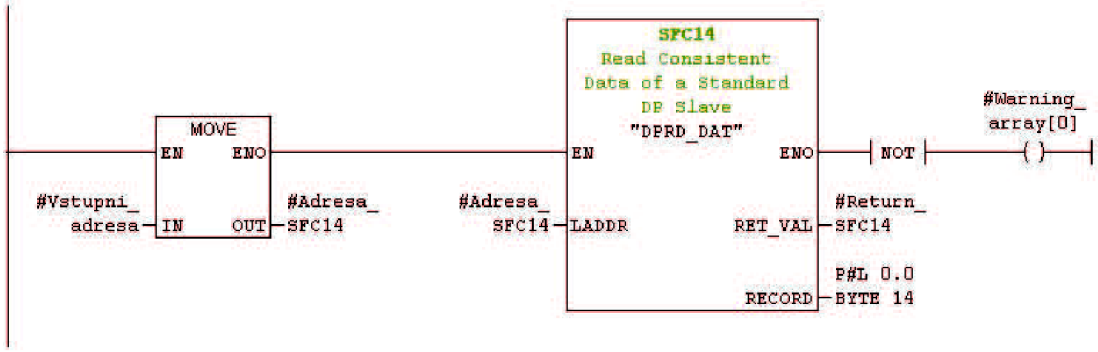
Tento funkční blok byl vytvořen pro komunikaci mezi PLC a frekvenčním měničem pomocí datagramu č.110. Ten je tvořen dvanácti výstupními slovy (jdoucích z PLC do FM) a sedmi vstupními slovy (z FM do PLC). Pro správnou funkci tohoto FB je nutné nahrát do paměti PLC systémové funkce SFC14 a SFC15.

9.1.1 Čtení, přesun a zápis dat

V úvodu programu (network 1 - 3) dojde k vynulování místa pro dočasné proměnné (vstupní a výstupní slova). Tím jsem předešel tomu, že by při jejich zápisu nebo čtení mohlo dojít v důsledku existence nějaké předešlé uložené hodnoty ke stavu, který by systém vyhodnotil jako chybový.

Pro správnou komunikaci s frekvenčním měničem je nutné nejprve načíst jeho data do TEMP. K tomu slouží systémová funkce SFC14 použitá v networku 4 (viz Obrázek 24). Jejím vstupem je adresa měniče (Adresa_SFC14), ze které budou

data čtena a výstupy pak návratová hodnota (Return_SFC14) a adresa v lokální paměti spolu s počtem čtených bajtů, kam budou data zapisována (adresa 0.0 a 14 bytů). V TEMP pak budou výstupní slova měniče uložena do datových slov se stejným názvem.



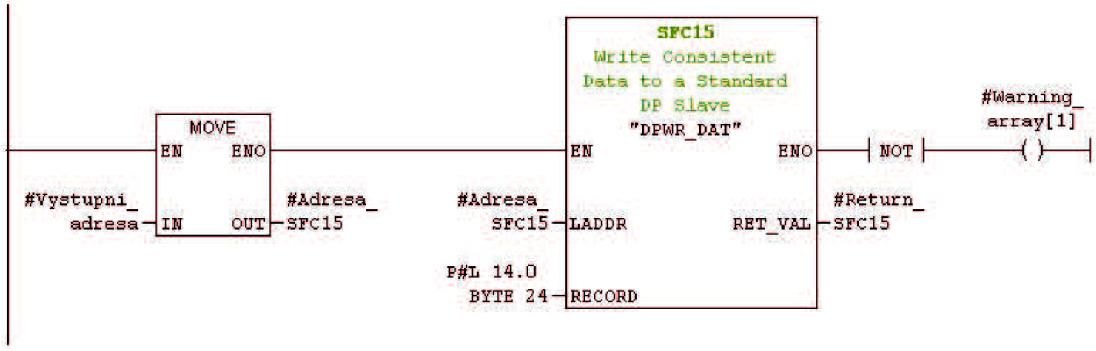
Obrázek 24 Načtení dat z měniče

Přesun konkrétních binárních hodnot stavových, řídicích a ostatních slov typu bitové pole na vstup (network 6 – 9) a výstup FB (network 23 - 26) jsem napsal v STL, a to z důvodů uvedených na začátku kapitoly 9. Musel jsem si ale dát pozor na jedno úskalí zápisu do bitových polí a čtení z bitových polí. Tím myslím to, že v tomto případě není systém kódování uložení a následného čtení dat ve FM a PLC identický a dochází ke kolizi little endian a big endian. Proto je osmý (ZSW1[9]) bit čteného slova zapsán jako nultý (Ready_to_power_up), devátý bit (ZSW1[10]) jako první (Ready) atd. Příklad tohoto zápisu je na obrázku uvedeném níže.

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

A     #ZSW1[9]
=     #Ready

A     #ZSW1[10]
=     #Operation_enabled

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

Obrázek 25 Přesun bitů

K zápisu dat do frekvenčního měniče slouží v networku 27 se nacházící blok SFC15 (Obrázek 26), který plní přesně opačnou funkci jako SFC14. Ze zadaného místa v lokální paměti čte příslušný počet bytů a posílá je na adresu, která je mu zadána na vstupu. Výstupem je jako u SFC14 návratová hodnota, jejíž velikost je v případě bezchybného odeslání dat rovna nule. Pokud tomu tak není, vyjadřuje hexadecimální číslo kód chyby, jejíž popis lze nalézt v Helpu.

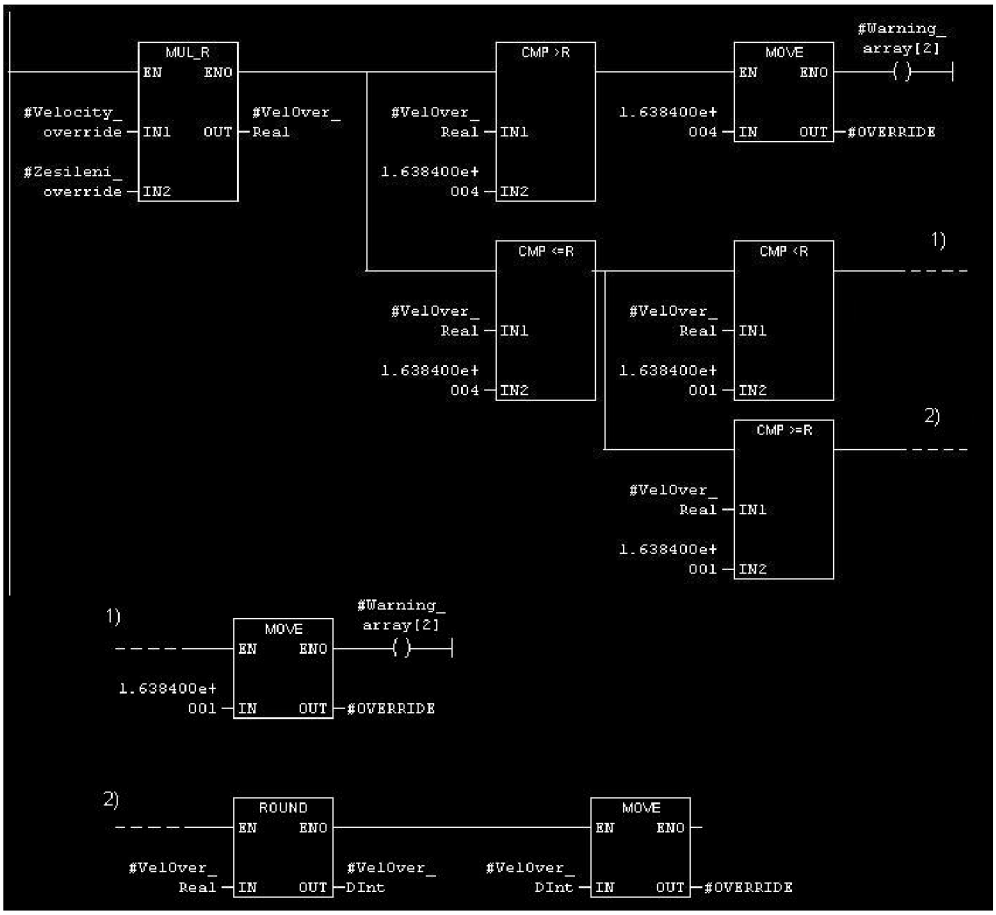


Obrázek 26 Zápis dat do měniče

9.1.2 Nastavení override, rychlosti a vzdálenosti k ujetí

Při návrhu schémat pro nastavování override (zjemnění) rychlosti, zrychlení a zpomalení (slova MDI_VELOC, MDI_ACC a MDI_DEC), které lze zadat v rozmezí 0,1 – 100%, jsem se snažil předejít chybě, které se může obsluha dopustit při zadávání hodnoty. Při nastavení hodnoty větší než maximální, dojde k jejímu nastavení na 100%. Obdobně je tomu u minimální hodnoty, kde dojde k nastavení spodní hranice 0,1%. Ještě před touto porovnávací sekvencí dochází k vynásobení hodnoty zadávané v procentech určitou konstantou a to kvůli normalizaci a následnému převodu na hexadecimální číslo, se kterým měnič pracuje (Obrázek 27).

Blok ROUND slouží k zaokrouhlení reálného čísla (Real) na double integer (DInt), který se pak převede na word. Blok MOVE umí přesouvat byty, wordy a double wordy a současně také vzájemně konvertovat typy real, integer, double integer, word a double word. Reálné číslo v tomto případě ale musí mít vždy desetinná místa nulová.



Obrázek 27 Nastavení hodnoty Velocity_override s ošetřením chyb

U nastavení vzdálenosti v LU, kterou má motor ujet – MDI_TARPOS (network 13) předcházím chybovým stavům stejně, jako v případě override, jen limitními hodnotami jsou rozsahy datového typu double integer.

Na stejném principu jako nastavení override funguje i následující nastavení rychlosti v tisících LU/min (MDI_VELOC). Zde jsou limitní hodnoty nastaveny dle Expert listu, reálně ale platí hodnota uvedená v parametru p2571, která lze nastavit ve STARTERu (v sekci nastavení servomotoru – záložka Technology – Basic positioner – Limit – okénko Max. velocity).

9.1.3 Volba polohovacího módu

V posledním slově ze vstupních – MDI_MOD, lze pomocí vzájemné binární kombinace dvou bitů navolit, zda bude polohovací režim probíhat v jednom ze čtyř režimů (absolutní, relativní, absolutní pozitivní a absolutní negativní).

Aby obsluha nemusela složitě tuto binární kombinaci zadávat, stačí jen na vstup MDI_mode_set zadat hodnotu 0 – 3, podle toho, který režim chceme použít. Přičemž hodnota zadaná zde dekadicky se rovná binární hodnotě dvou bitů slova MDI_MOD. V networku 17 – 22 dojde pomocí bloku CMP k porovnání čísla zadaného režimu (ve formátu integer) s relevantními i chybnými možnostmi. Je-li vybrán možný režim (Network 18 – 21), provede se pomocí instrukce podmíněného skoku (JMP) skok na příslušné návěští do networku 23 (ukázka dvou návěští je na obrázku 28), kde dojde k nastavení požadovaných bitů slova. Poté je proveden skoku ven na následující network.

```
ModX: A     #Mode_value_0
      =     #MDI_MOD[12]
      A     #Mode_value_0
      =     #MDI_MOD[13]
      A     #Chyba
      =     #Warning_array[7]
      JU    Next

Mod0: A     #Mode_value_0
      =     #MDI_MOD[12]
      A     #Mode_value_0
      =     #MDI_MOD[13]
      JU    Next
```

Obrázek 28 Nastavení MDI_MOD

Pokud by došlo k zadání hodnoty MDI_mode_set různé od hodnoty 0 – 3, dojde ke skoku na návěstí ModX, provede se nastaven bitů tak, aby se jednalo o absolutní režim a navíc je do bitového pole Warning_array na příslušné místo zapsána log. 1, aby byla obsluha informována o své chybě.

9.1.4 Hlášení o chybách

Jak jsem již výše uvedl, pro lepší informovanost obsluhy, kde udělala při zadávání parametrů chybu, jsem v TEMP vytvořil bitové pole Warning_array. Nulové hodnoty znamenají bezchybný stav. Logická 1 v příslušném bitu znamená chybu, která je okomentována v úvodních informacích na začátku funkčního bloku nad samotným programem.

I když by ale k nějaké takové chybě došlo, je díky ošetřujícím podmínkám hodnota posílaná měniči ve správném rozsahu. Ze 16 bitů bitového pole Warning_array je jich využito jen 8. Zbytek zůstává v záloze pro další možné využití.

9.2 FB111

FB111 byl vytvořen pro komunikaci přes datagram č.111. Ten tvoří dvanáct vstupních i výstupních datových slov. Pro správnou funkci tohoto FB je nutné nahrát do paměti PLC systémové funkce SFC14 a SFC15.

Dá se říci, datagram č.111 je upravenou a rozšířenou verzí datagramu č.110 (viz Obrázek 23). Z toho také vyplívá podobný princip výměny dat. Nebudu se zde proto zabývat popisem vlastního programu(viz Příloha 1). Spíše se zaměřím na vyjmenování některých zásadních rozdílů mezi FB111 a FB110.

Podstatné změny FB111 oproti FB110:

- Pracuje navíc s pěti vstupními slovy – dvojitým slovem NIST_B udávajícím rychlost motoru (v programu je hexadecimální číslo převedeno na rychlost v ot/min), slovy typu bitové pole FAULT_CODE a WARN_CODE a mnou definovaným slovem FREE2 (poznámka <3> v obrázku 23 znamená volné slovo)

- Posledním výstupním slovem není MDI_MOD pro výběr polohovacího módu, ale jedná se o slovo FREE1, které má stejný význam jako slovo FREE2 (oba jsem nastavil datového typu bitové pole).

- Došlo k nahrazení slov SATZANW, PosSTW, AKTSATZ a PosZSW slovy PosSTW1, PosSTW2, PosZSW1 a PosZSW2. Datový typ zůstal stejný.

- Ve stavových slovech ZSW1 a ZSW2 došlo k využití některých nevyužitých (reserved) bitů a ke změně logiky u některých bitů. Tím je myšleno to, že nyní už bit s hodnotou logická 1 neznamená kladný výraz (např. Ano, aktivní apod.) a bit s hodnotou logická 0 záporný výraz, ale je tomu naopak.

- V poli Warning_array je využito pouze sedm bitů ze šestnácti. To z důvodu nahrazení slova MDI_MOD slovem FREE1.

9.3 FC1 – MC_POWER

Funkce FC1 vznikla za účelem zapnutí a vypnutí polohovacího zařízení. Zapnutím není myšleno např. roztočení motoru, ale pouze odblokování všech potřebných podmínek pro start a uvedení generátoru pulzů do stavu připraven. Za názvy vstupů a výstupů je v závorce uveden použitý datový typ. Pro správný chod této funkce je nutné nahrát do paměti PLC systémovou funkci SFC24.

Vstupy funkce:

- Axis_DB (Block_DB) – název datového bloku, který bude funkce ovládat
- Enable (Bool) – přivedením logické 1 dojde k zapnutí zařízení, přivedením logické 0 dojde k vypnutí zařízení a to podle módu, který je vybrán na vstupu StopMode
- StopMode (Bool) – definuje dvě možnosti zastavení – okamžité (nastavena logická 0) a s brzděním po rampě (nastavena logická 1)

Výstupy funkce:

- Status (Bool) – funkce je používána
- Busy (Bool) – funkce právě vykonává příkaz
- Active (Bool) – funkce vykonala příkaz
- Error (Bool) – došlo k výskytu chyby
- ErrorID (DWord) – popis příčiny chyby (ta je uvedena v komentáři k programu)

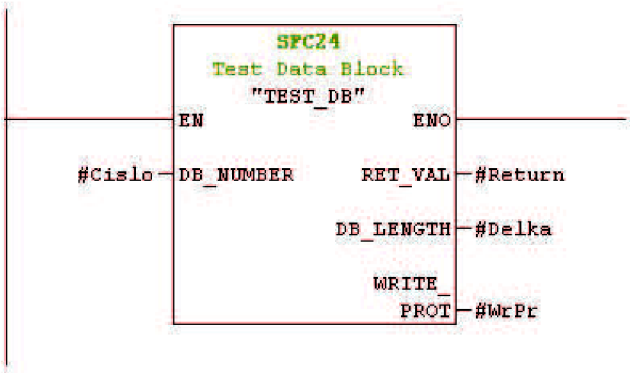
9.3.1 Rozeznání vybraných datagramů

Protože všechny funkce měly být vytvořen tak, aby bylo pomocí nich možno ovládat zařízení komunikující pomocí datagramů č. 1, 2, 9, 110 a 111, musí se vždy na začátku programu zjistit o jaký datagram jde. Toho je dosaženo následujícím postupem.

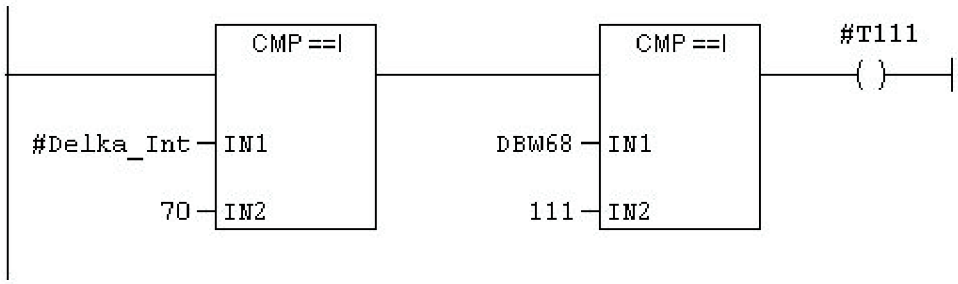
Funkce si pomocí příkazu pro otevření (OPN) otevře datový blok, se kterým bude pracovat. Zjistí se číslo datového bloku, které se v hexadecimální formě nahraje do akumulátoru PLC (příkaz L DBNO) a pomocí příkazu pro přenos z akumulátoru se nahraje do proměnné Cislo (T #Cislo). Tím dochází i k jeho automatické konverzi na datový typ Word.

Následuje použití SFC24 (viz Obrázek 29), která dokáže spočítat délku libovolného datového bloku. Na její vstup DB_NUMBER je přivedena proměnná Cislo. Na výstupu DB_LENGTH poté obdržíme délku DB v bytech, která je zapsána do proměnné Delka (datový typ Word). SFC24 navíc umožňuje výpis návratové hodnoty funkce na výstupu RET_VAL (při hodnotě 0 vše proběhlo v pořádku) a zjištění, zda není DB chráněn proti přepsání (výstup WRITE_PROT).

Hodnota proměnná Delka je zkopírována do proměnné Delka_Int, která je ve formátu integer. Použitý datagram je rozeznán podle dvou informací. Na základě porovnání Delka_Int se známou hodnotou délky každého z datagramů a také pomocí shody identifikačního čísla (které má každý z pěti datagramů na konci svého DB) s definovaným číslem (Obrázek 30). Pak je proveden skok na požadované návěští. V případě neznámého datagramu dojde k výpisu chyby.



Obrázek 29 Použití systémové funkce SFC24



Obrázek 30 Princip rozeznání použitého datagramu (zde č.111)

9.3.2 Zapnutí a vypnutí zařízení

Zde se rozlišují dvě verze zapnutí a vypnutí a to v závislosti na použitých datagramech. Pro datagram č. 1 a 2 se využívá prvních sedm bitů řídicího slova STW1. U datagramů č. 9, 110 a 111 pak jen prvních čtyř. Princip ovládání je ale stejný.

Zapnutí:

Na začátku programu dojde k nastavení všech používaných bitů s výjimkou bitu ON/OFF1na hodnotu logická 1. Program čeká (stav Busy), až ovládané zařízení nastaví bit Ready_for_switch_on (první bit stavového slova ZSW1) na hodnotu logická 1 a pak teprve dojde k aktivaci bitu ON/OFF1. Stav Active je signalizován poté, co zařízení oznámí, že je možné s ním provádět operace (je nastaven bit Operation_enabled nebo Run - v závislosti na použitém datagramu).

Vypnutí:

První možností vypnutí zařízení je okamžité zastavení (StopMode = 0) s využitím maximálního brzdného potenciálu. Toho bude docíleno tak, že motor nebude brzdit po rampě (grafické nastavení brzdné charakteristiky), ale bude mu ihned vypnuto napájení. Dalo by se říci, že se motor zasekne. Programově se toho dosáhne tak, že budou všechny ovládané bity najednou nastaveny na hodnotu logická 0. Této možnosti by mělo být využíváno spíše v krizových situacích.

Druhou možností, jak zařízení vypnout je brzdit po rampě (StopMode = 1) a až se motor úplně zastaví, odpojit mu napájení z měniče. V programu je to řešeno tím způsobem, že je nejprve bit OFF3 (aktivace brždění po rampě při hodnotě 0) nastaven na 0 a poté co je bit Operation_enabled (Run) také v 0, dojde k vynulování zbývajících ovládaných bitů. U této možnosti je navíc pomocí časovače nastavena podmínka, že když brzděním po rampě nedojde k zastavení do 3 sekund, aktivuje se okamžité zastavení jako při StopMode = 0.

9.4 FC2 – MC_STOP

Funkce FC2 slouží k nastavení hodnoty brždění polohovacího zařízení. Za názvy vstupů a výstupů je v závorce uveden použitý datový typ. Pro správný chod této funkce je nutné nahrát do paměti PLC systémovou funkci SFC24.

Vstupy funkce:

- Axis_DB (Block_DB) – název datového bloku, který bude funkce ovládat
- Execute (Bool) – přivedením logické 1 dojde k nastavení hodnoty zpomalení v ovládaném DB
- Deceleration (Real) – udává hodnotu override pro zpomalení v % (v popisu programu jsou poté uvedeny čtyři možnosti nastavení hodnot tohoto vstupu)

Výstupy funkce:

- Done (Bool) – zastavení proběhlo a funkce byla ukončena
- Busy (Bool) – funkce právě vykonává příkaz
- Active (Bool) – funkce je používána
- Error (Bool) – došlo k výskytu chyby
- ErrorID (DWord) – popis příčiny chyby (ta je uvedena v komentáři k programu)

Na začátku programu dojde k rozeznání datagramu stejným způsobem jako u funkce FC1. Pouze s tím rozdílem, že je umožněno pracovat jen s datagramem č.9, 110 a 111. Ty totiž disponují datovým slovem MDI_DEC určeným pro nastavení override pro zpomalení.

Nastavení hodnoty vstupu Deceleration do MDI_DEC, ke kterému dojde po aktivaci vstupu Execute, je možné provést dvěma způsoby a lze se dopustit dvou chyb. Proto jsem v programu vytvořil několik rozhodovacích podmínek, jejichž výstupem je poté skok na příslušné návěští.

Nejprve je zadaná hodnota testována na interval –nekonečno až -1.0 (včetně). V tomto případě se jako hodnota zpomalení použije hodnota přednastavená systémem v měniči. V rozsahu -1.0 až 0.1 je hodnota vyhodnocena jako chyba a dojde k nastavení výstupu Error a ErrorID. Relevantním intervalem je 0.1 (včetně) až 100.0 (včetně), kdy dojde k nahrání zadané hodnoty do MDI_DEC. Pokud by obsluha zadala větší číslo než je 100.0, nahraje systém do MDI_DEC hodnotu 100.0.

Až funkce zjistí, že se již motor netočí (bit s názvem „|n_act| < speed threshold value 3“ neboli Drive_at_standstill = 1), tak sama vynuluje vstup Execute a tím se ukončí (stav Done).

9.5 FC3 – MC_RESET

Funkce FC3 slouží k nastavení bitu Ack_fault polohovacího zařízení. Za názvy vstupů a výstupů je v závorce uveden použitý datový typ. Pro správný chod této funkce je nutné nahrát do paměti PLC systémovou funkci SFC24.

Vstupy funkce:

- Axis_DB (Block_DB) – název datového bloku, který bude funkce ovládat
- Execute (Bool) – přivedením logické 1 dojde k nastavení hodnoty Ack_fault

Výstupy funkce:

- Done (Bool) – reset proběhl
- Busy (Bool) – funkce právě vykonává reset
- Error (Bool) – došlo k výskytu chyby
- ErrorID (DWord) – popis příčiny chyby (ta je uvedena v komentáři k programu)

Nastavením vstupu Execute na hodnotu logická 1 dojde k nastavení osmého bitu řídicího slova STW1 (Ack_fault - Acknowledge faults). To má za výsledek, že v případě výskytu nějakého varování či nezávažné chyby dojde ke jejímu vymazání. Obsluha tím dala najevo že o chybě ví.

Stejně jako v kapitole 9.3.1. je i zde použito rozeznávání pěti datagramů.

10. ZÁVĚR

Vypracováním této bakalářské práce jsem se naučil základům programování programovatelných automatů Siemens a poznal, jak velké možnosti řízení poskytuje práce s frekvenčním měničem. Zajímavé pro mě bylo i setkání s moderními průmyslovými sběrnicemi Profibus DP a Profinet IO.

Hlavním cílem práce bylo vytvořit sadu funkčních bloků pro komunikaci mezi programovatelným automatem Siemens Simatic S7-300 a frekvenčním měničem Siemens Sinamics S120. Tohoto cíle se mi podařilo dosáhnout díky velké podpoře zástupců sekce Automatizace a pohony společnosti Siemens s.r.o. Poskytnuli mi jak technologické zázemí, tak spoustu odborných rad a díky nim jsem měl tu možnost získat mnoho zkušeností týkajících se praxe v tomto oboru.

Oba funkční bloky byly vyzkoušeny na testovacích sestavách zařízení a mohou se tak po dohodě se zástupci společnosti Siemens s.r.o. zařadit mezi sadu standardních funkčních bloků, které tato společnost používá pro své zákazníky. Funkce, které jsem dodatečně navrhl a které by měly sloužit pro ovládání základních vlastností frekvenčního měniče budou teprve testovány.

11. SEZNAM LITERATURY

[1] *Catalog PM 21 - 2008 English.* Siemens AG. 2008. 884 s.

[2] KOSEK, Rostislav. *Simatic Innovation tour 2008 - Simatic PLC.* 2008. 51 s.

[3] KUČERA, Pavel. *Přednáška – Profibus.* Brno, 2008, 43 s.

[4] *List Manual - SINAMICS S120/S150.* Siemens AG. 10/2008. 1980 s.

[5] PÁSEK, Jan. *Programovatelné automaty v řízení technologických procesů.* Brno, 2007. 128 s. Skriptum k předmětu BPGA.

[6] PÁTÍK, Martin. *PLC Simatic S7-300* [online]. c2008, [cit. 2009-5-20]. <http://www1.siemens.cz/ad/current/index.php?ctxnh=ee5ad951ae&ctxp=home>

[7] PÁTÍK, Martin. *Profibus* [online]. c2008, [cit 2009-5-18]. <http://stest1.etnetera.cz/ad/current/index.php?ctxnh=cf23b3a6ff&ctxp=home>

[8] PÁTÍK, Martin. *STEP7* [online]. c2008, [cit. 2009-5-20]. <http://www1.siemens.cz/ad/current/index.php?ctxnh=3a01fb9720&ctxp=home>

[9] *Products for Totally Integrated Automation and Micro Automation - Catalog ST 70 2007.* Siemens AG. 2006. 804 s.

[10] *Prospekt Profinet 04/2005 CZ* [online]. Siemens s.r.o., 2005, [cit. 2009-5-18]. 16 s. <http://stest1.etnetera.cz/ad/current/index.php?ctxnh=bae95e75f4&ctxp=doc_prospekty>

[11] *Přehledový prospekt Elektrické pohony.* Brno: Siemens s.r.o., září 2006. 20 s.

[12] *SIMATIC S7 Programování 1+2 (SIMATIC S7-300/400).* Siemens s.r.o., 2003. 672 s.

[13] *Simatic79.* Siemens s.r.o., <http://www.siemens.sk/download/visions/02-2008/_assets/simatic79.jpg>

[14] *SIMOTION PLCopen Blocks Function Manual.* Siemens AG, 07/2008. 118s.

[15] ŠMEJKAL, Ladislav – MARTINÁSKOVÁ, Marie. *PLC a automatizace.* Grafický návrh Libor Kubica. 1. vyd. Praha: BEN, 2007. 224 s. ISBN 978-80-86056-58-6.

[16] ZEZULKA, František. *Prostředky průmyslové automatizace.* Brno, 2004. Skriptum k předmětu BPPA.

[17] ZEZULKA, František. *Prostředky průmyslové automatizace.* Návrh obálky Pavel Luffer. 1. vyd. Brno: VUTIUM, 2004. 176 s. ISBN 80-214-2610-1.

[18] ZEZULKA, František – HYNČICA, Ondřej. Průmyslový Ethernet VIII: Ethernet Powerlink, Profinet. *AUTOMA* [online]. 2008, č. 5 [cit. 2009-5-18]. <http://www.odbornecasopisy.cz/index.php?id_document=37288>

Seznam obrázků

Seznam zkratek

A/D	Analog/Digital
AI/AO	Analog Input/Analog Output
ALI	Aplication Layer Interface
CNC	Computer Numeric Control
CPU	Central Processing Unit
DB	Data Block – datový blok
DI/DO	Digital Input/Digital Output
DIN	Deutsche Industrie Norm – Německá průmyslová norma
EN	European Norm – evropská norma
FB	Function Block – funkční blok
FBD	Function Block Diagram – program. jazyk funkčních bloků
FC	Function - funkce
FM	Frekvenční měnič
IEC	International Electrotechnical Commission – Mezinárodní elektrotechnická komise
IEEE	Institute of Electrical and Electronics Engineers - Společnost pracovníků v elektrotechnice a elektronice v USA
IP	Internet Protocol
IPC	Industrial PC
ISO	lnternational Organization for Standardization - Mezinárodní organizace pro normalizaci
IT	Information Technology
LAD,LD	Ladder Diagram – program. jazyk reléových logick. schémat
LU	Length Unit
MPI	Multi-Point Interface – systémové komunikační rozhraní
NRT	Non Real Time
OB	Object Block – objektový blok
OSI	Open Systems Interconnection
PLC	Programmable Logic Controller – programovatelný automat
RT	Real Time – režim reálného času

SDB	System Data Block – systémový datový blok
SFB	System Function Block – systémový funkční blok
SFC	System Function – systémová funkce
SFC14	System Function 14
SFC15	System Function 15
SFC24	System Function 24
STL	Statement List – programovací jazyk mnemokódů
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

Seznam příloh

Příloha 1	Přiložené CD, které obsahuje projekt s vytvořenými funkčními bloky a funkcemi, kompletní výpis programu funkčních bloků a funkcí ve formátu PDF a PDF verzi bakalářské práce.

Poznámka

Verze programů použité při vytváření funkčních bloků a funkcí:

1) STARTER V4.1.2.4, release number V4.1.2.4
2) Drive ES Basic V5.4 + SP2, release number V5.4.2.0
3) STEP7 V5.4 + SP4, release number K5.4.4.0