

VŠB – Technická univerzita Ostrava

Fakulta strojní

Katedra automatizační techniky a řízení

Vazby mezi PLC a frekvenčními měniči v řídicích úlohách

**Links between PLCs and Frequency Inverters in Control
Tasks**

Student:

Bc. Marek Stančák

Vedoucí diplomové práce:

doc. Ing. Jaromír Škuta, Ph.D.

Ostrava 2019

Zadání diplomové práce

Student: **Bc. Marek Stančák**
Studijní program: N2301 Strojní inženýrství
Studijní obor: 3902T004 Automatické řízení a inženýrská informatika
Téma: **Vazby mezi PLC a frekvenčními měniči v řídicích úlohách**
Links between PLCs and Frequency Inverters in Control Tasks
Jazyk vypracování: čeština

Zásady pro vypracování:

1. Seznamte se s frekvenčními měniči dostupnými na katedře, s možnostmi jejich konfigurace, popište komunikační rozhraní a další.
2. Seznamte se s řídicími systémy firmy ABB, rozšiřujícími moduly dostupnými v laboratořích katedry a s vývojovým prostředím pro jejich programování.
3. S využitím těchto prostředků navrhnete a realizujte úlohu řízení pro střídavé motory s možností konfigurace ze SCADA/MMI prostředí systému Control Web.
4. Zhodnoťte dosažené výsledky a navrhnete směr dalšího řešení.

Seznam doporučené odborné literatury:


- ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace*. 1. díl, Základní pojmy, úvod do programování. Praha: BEN - technická literatura, 1999. ISBN 80-86056-58-9.
- ŠMEJKAL, Ladislav. *PLC a automatizace*. 2. díl, Sekvenční logické systémy a základy fuzzy logiky. Praha: BEN - technická literatura, 2005. ISBN 80-7300-087-3.
- TRZYNADLOWSKI, Andrzej Maria. *Control of induction motors* [online]. Amsterdam: Elsevier, 2007 [cit. 2017-11-16]. ISBN 978-0-12-701510-1.
- WHITT, Michael D. *Successful instrumentation and control systems design* [CD-ROM]. Research Triangle Park: ISA, c2004. ISBN 1-55617-844-1.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí diplomové práce: **doc. Ing. Jaromír Škuta, Ph.D.**

Datum zadání: 21.12.2018

Datum odevzdání: 20.05.2019


doc. Ing. Renata Wagnerová, Ph.D.
vedoucí katedry




prof. Ing. Ivo Hlavatý, Ph.D.
děkan fakulty

Místopřísežné prohlášení studenta

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě dne 20. 5. 2019



.....

Podpis

Prohlašuji, že

- jsem si vědom, že na tuto moji závěrečnou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. Zákon o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (dále jen Autorský zákon), zejména § 35 (Užití díla v rámci občanských či náboženských obřadů nebo v rámci úředních akcí pořádaných orgány veřejné správy, v rámci školních představení a užití díla školního) a § 60 (Školní dílo),
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo užít tuto závěrečnou diplomovou práci nekomerčně ke své vnitřní potřebě (§ 35 odst. 3 Autorského zákona),
- bude-li požadováno, jeden výtisk této diplomové práce bude uložen u vedoucího práce,
- s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 Autorského zákona,
- užít toto své dílo, nebo poskytnout licenci k jejímu využití, mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše),
- beru na vědomí, že – podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů – že tato diplomová práce bude před obhajobou zveřejněna na pracovišti vedoucího práce, a v elektronické podobě uložena a po obhajobě zveřejněna v Ústřední knihovně VŠB-TUO, a to bez ohledu na výsledek její obhajoby.

V Ostravě dne 20. 5. 2019



.....
Podpis

Bc. Marek Stančák
Dolní Jasenka 744
75501 Vsetín

Anotace diplomové práce

STANČÁK, M. *Vazby mezi PLC a frekvenčními měniči v řídicích úlohách: diplomová práce*. Ostrava: VŠB – Technická univerzita Ostrava, Fakulta strojní, Katedra automatizační techniky a řízení, 2019, 66 s. Vedoucí práce: Škuta, J.

Tato práce pojednává o využití PLC pro řízení dvou asynchronních elektromotorů, které jsou připojeny ke dvěma frekvenčním měničům různých výrobců – frekvenční měnič řady ACS355 od společnosti ABB a UniDrive M200 od společnosti EMERSON, které v řídicím řetězci představují akční členy. Laboratorní úloha s měničem ACS355 bude představovat regulační úlohu, jelikož řízený elektromotor má společnou hřídel s tzv. zátěžovým elektromotorem, který je navíc vybaven tachodynamem pro snímání aktuálních otáček. Úloha s měničem M200 bude představovat pouze úlohu ovládní, jelikož není k dispozici reálná zpětná vazba pro snímání otáček elektromotoru. Řízení motorů, sběr procesních dat, řízení komunikace a regulaci na požadovanou hodnotu zajišťuje logický automat typové řady AC500 (ABB). Pro vizualizaci úlohy bude využito prostředí Control Web 8.

Annotation of Master Thesis

STANČÁK, M. *Links between PLCs and Frequency Inverters in Control Tasks: Master Thesis*. Ostrava: VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Control Systems and Instrumentation, 2019, 66 p. Thesis head: Škuta, J.

This thesis deals with the use of PLC for control of two asynchronous electric motors, which are connected to two frequency converters of different manufacturers - the ACS355 frequency inverter from ABB and the UniDrive M200 from EMERSON, which are actuators in the control string. The laboratory task with the ACS355 will be used for regulation, where the controlled electric motor has a common shaft with a so-called load electric motor, which is also equipped with a tachometer for measurement of the current speed. The task with the M200 will be used only for controlling, as there is no real feedback for measurement of the electric motor speed. The AC500 (ABB) logic controller provides motor control, process data collection, communication control, and setpoint control. The Control Web 8 environment will be used for the task visualization.

Obsah

Seznam použitých symbolů a značek.....	8
Seznam použitých veličin.....	9
Úvod.....	10
1 Frekvenční měniče	11
1.1 Způsoby řízení frekvenčního měniče.....	12
1.2 Frekvenční měnič ACS355.....	14
1.2.1 Lokální řízení frekvenčního měniče ACS355	16
1.2.1.1 Manuální ovládání frekvenčního měniče z ovládacího panelu	18
1.2.1.2 Lokální ovládání frekvenčního měniče z PC.....	19
1.2.2 Vzdálené řízení frekvenčního měniče ACS355	22
1.2.2.1 Vzdálené řízení s využitím adaptéru FMBA-01	23
1.2.2.2 Vzdálené řízení s využitím RS232	24
1.3 Frekvenční měnič UniDrive M200	25
1.3.1 Ruční ovládání z panelu frekvenčního měniče.....	27
1.3.2 Ovládání frekvenčního měniče M200 z PC.....	28
1.3.3 Řízení frekvenčního měniče M200 z nadřazené úrovně	31
2 Řídicí systémy společnosti ABB.....	32
2.1 Řídicí systém Codesys	33
2.2 Logický automat AC500-eCo	39
3 Komunikační protokol MODBUS	40
4 Laboratorní úloha řízení elektromotorů	43
4.1 Prvky laboratorní úlohy	44
4.2 Komunikační vazby	47
4.2.1 Tvorba OPC protokolu	48
4.2.2 Komunikace frekvenčních měničů s PLC	50
4.2.3 Konfigurace PLC pro sériovou komunikaci	51
4.3 Řídicí algoritmus PLC	52

4.3.1	Program pro regulaci otáček na požadovanou hodnotu	55
4.3.2	Identifikace systému a syntéza regulátoru.....	57
4.3.3	Regulace otáček na požadovanou hodnotu.....	58
4.4	Uživatelská aplikace v prostředí Control Web 8	59
5	Závěr	62
	Seznam použité literatury	65

Seznam použitých symbolů a značek

CAN	Komunikační sběrnice (<i>Controller Area Network</i>)
CW	Řídicí slovo (<i>Control Word</i>)
DWL	PC aplikace (<i>Drive Window Light</i>)
EMC	Elektromagnetická kompatibilita
FBD	Funkční blokový diagram
GND	Signální uzemnění
I/O	Vstupně/výstupní rozhraní
LOC	Lokální režim ovládání (<i>Local</i>)
Master/Slave	Model počítačové komunikace (<i>Client/Server</i>)
OPC	Otevřená komunikační platforma
PLC	Programovatelný logický automat
REM	Vzdálený režim ovládání (<i>Remote</i>)
Rx	Přijátá data (<i>Received</i>)
ST	Strukturovaný text
TCP/IP	Přenosový protokol/protokol síťové vrstvy
Tx	Odeslaná data (<i>Transmitted</i>)

Seznam použitých veličin

Veličina	Jednotka	Význam
Bd	[bit/s]	Jednotka přenosové rychlosti
f	[Hz]	Frekvence
f_r	[Hz]	Frekvence rotoru
f_{sN}	[Hz]	Jmenovitá frekvence statoru
I	[A]	Elektrický proud
n_s	[ot/min]	Synchronní otáčky statoru
p	[-]	Počet pólových dvojic elektromotoru
R_r	[Ω]	Odpor rotoru
R_s	[Ω]	Odpor statoru
U	[V]	Elektrické napětí
$(U_{sN})_{ef}$	[V]	Jmenovité efektivní napětí statoru
Φ_m	[rad/s]	Mechanická úhlová rychlost

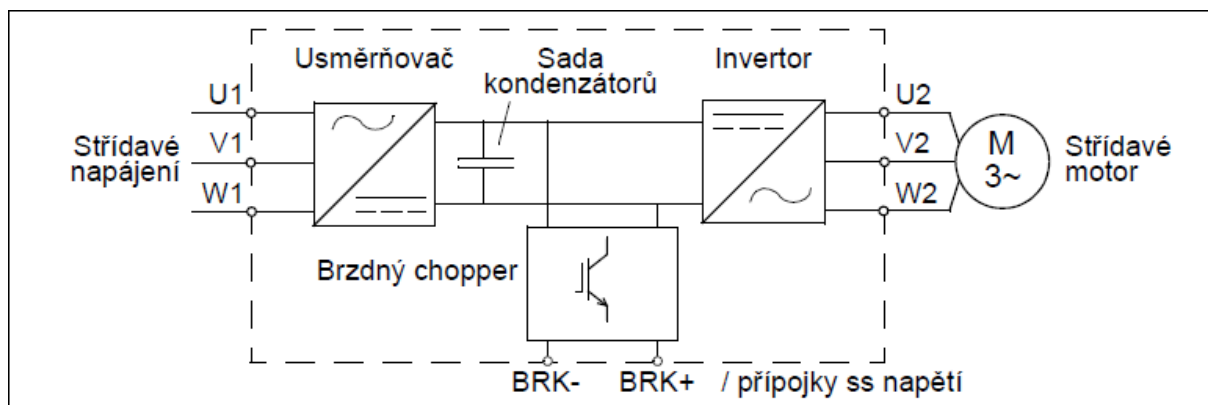
Úvod

V dnešní době je používání výpočetní techniky pro řízení průmyslových úloh nezbytnou součástí vývoje lidské civilizace. Počítače, resp. jejich průmyslová odnož, jsou extrémně robustní zařízení vyvinuté pro nepřetržitý chod v často pro člověka nepříznivých podmínkách. Jejich zpracování logických úloh je z časového hlediska nesrovnatelné s rozhodováním lidského mozku a zvládají obrovského množství výpočetních úloh v krátkém časovém intervalu. Průmyslový logický automat (PLC) je typickým představitelem průmyslového počítače, používaného na procesní úrovni distribuovaného systému řízení. Jeho hlavní odlišností je cyklické zpracování programů a uzpůsobení komunikačních jednotek pro přímé připojení k technologickým procesům. Automatizovaná výroba s využitím PLC velkou mírou přispívá ke zvýšení produkce a kvality výrobků průmyslových firem v oblasti například papírenského, chemického, automobilového či hutního průmyslu.

Frekvenční měnič je zařízení určené k řízení střídavých elektrických pohonů v nejrůznějších sekcích průmyslové výroby. Nejčastěji jsou frekvenční měniče určeny pro řízení asynchronních indukčních motorů nebo synchronních motorů s permanentními magnety, jejichž otáčky závisí na velikosti magnetického toku ve statoru. Frekvenční měnič řídí rychlost otáčení hřídele motoru změnou napájecího napětí a jeho frekvence vstupující do vinutí statoru připojeného motoru. Toho se využívá v průmyslové výrobě například u řízení čerpadel nebo rychlosti dopravníkových pásů, přičemž se ve většině případů jedná o současné řízení více motorů v jednom technologickém procesu. Pro řízení daného technologického procesu je použit jeden nebo více programovatelných logických automatů, které na základě měření procesních veličin upravují požadavky pro jednotlivé frekvenční měniče. Kromě toho se PLC využívá také pro sběr procesních dat, které cyklicky odesílá do operátorských stanovišť, čímž umožňuje v reálném čase sledovat průběh celého technologického procesu. Díky tomu mohou operátoři velmi rychle reagovat na vzniklé chyby procesů a předejít tak zastavení výroby a s tím spojeným finančním ztrátám.

1 Frekvenční měniče

Frekvenční měniče slouží k řízení třífázových elektromotorů. Na vstupu je frekvenční měnič napájen jedno- nebo tří-fázovým napětím, které je uvnitř měniče usměrněno na stejnosměrné napětí, stabilizováno sadou kondenzátorů dle aktuálních požadavků a následně je na výstupu měniče převedeno pomocí invertoru zpět na střídavé napětí (vždy třífázové). Princip činnosti spočívá v řízení otáček elektromotoru, které jsou přímo úměrné vstupující frekvenci napájecího napětí elektromotoru. Změnou frekvence napájení elektromotoru proto dojde k úměrné změně otáček, což vede ke značnému snížení spotřeby elektrické energie, která je přizpůsobena aktuálním požadavkům na výkon elektromotoru. Ekonomická úspora je hlavní předností způsobu řízení elektrických pohonů za pomoci frekvenčních měničů. (ABB, 2010)



Obr. 1 – Zjednodušené schéma hlavního obvodu frekvenčního motoru. (ABB, 2010)

Frekvenční měnič bude použit pro řízení asynchronního motoru. Asynchronní motor je konstrukčně jednoduchý elektrický stroj s bezporuchovým chodem a dlouhou životností. Vysokou účinností elektromotoru s řízením pomocí frekvenčního měniče lze docílit téměř ideální spotřebně elektrické energie vzhledem k aktuálním požadavkům na výkon elektrického pohonu provádějícího pracovní úkon. Otáčky točivého pole statoru, tzn. synchronní otáčky n_s :

$$n_s = \frac{60 \cdot f}{p} \quad [min^{-1}] \quad (1.1)$$

Výsledné otáčky rotoru jsou sníženy o tzv. skluz ($n = n_s \cdot (1 - s)$), jelikož asynchronní motor generuje na výstupní hřídeli kroutící moment pouze při rozdílných otáčkách točivého pole statoru vzhledem ke skutečným otáčkám rotoru.

1.1 Způsoby řízení frekvenčního měniče

Pro řízení frekvenčních měničů, které jsou použity v této práci je využito skalárního řízení. Základní rozdělení způsobů řízení frekvenčního měniče, resp. řízení jeho výstupních veličin, pro asynchronní pohony je skalární řízení, vektorové řízení a přímé řízení momentu asynchronního motoru.

• Skalární řízení:

Jedná se původní způsob řízení asynchronních motorů a dělí se v zásadě na dvě skupiny, napěťově-frekvenční regulace a proudově-frekvenční regulace. Nejčastěji se využívá současného řízení napájecího napětí a frekvence, kdy je při stálém momentu na hřídeli ($M = konst.$) nutné udržet konstantní magnetický tok ($\Phi_m = konst.$) což vede ke konstantnímu poměru mezi napájecím napětím a frekvencí ($U/f = konst.$). (NEBORÁK, 2011)

- Pro frekvenci napájecího napětí v rozsahu do jmenovité frekvence motoru ($0 \div f_N$) můžeme regulovat napájecí napětí v rozsahu $0 \div U_N$, pro udržení poměru $U/f = konst.$ Tímto docílíme konstantního momentu na hřídeli rotoru, který může odpovídat až maximálnímu momentu M_{max} .
- Pro napájecí frekvenci $f > f_N$ již nelze zvyšovat napájecí napětí U_N , které dosáhlo jmenovité hodnoty, čímž dochází ke snižování momentu na hřídeli ($M < M_{max}$).
- Pro napájecí frekvenci $f \rightarrow 0$ by se mělo také napájecí napětí blížit nule, ovšem to způsobí, že se při požadavku na velmi nízké otáčky ($f \rightarrow 0$) motor nerozběhne, jelikož nízké napětí nedovolí průchod dostatečně velkého proudu statorem. Z tohoto důvodu je u požadavku na nízké otáčky ($f \rightarrow 0$) nutné k napájecímu napětí připočíst ΔU , které napájecí napětí zvýší a zajistí dostatečně velký proud procházející statorem pro rozběh motoru. Hodnota ΔU je určena:

$$\Delta U = \frac{(U_{sN})_{ef} \cdot \sqrt{2} \cdot R_s}{R_r \cdot f_{sN}} \cdot f_r \quad (1.2)$$

• Vektorové řízení:

Vektorové řízení oproti skalárnímu neřídí pouze velikost magnetického toku statoru, ale navíc řídí ještě polohu natočení. V podstatě se u vektorového řízení ruší vazby proudu motoru na magnetický tok pomocí výpočtu charakteristiky motoru z matematického modelu s aktivní zpětnou vazbou polohy rotoru nebo bez zpětné vazby. (NEBORÁK, 2011)

Vektorové řízení se v základu dělí na dvě části podle způsobu získání informace o vektoru magnetického toku na přímé a nepřímé.

- Přímé: vektor magnetického toku je vypočten z hodnot napětí a proudu procházejících státorem. Pokud je znám matematický model pro výpočet skluzu, není třeba aktivní zpětná vazba polohy rotoru.
- Nepřímé: nutnost aktivní zpětné vazby pro měření polohy rotoru a znalost matematického modelu pro určení skluzu.

Vektorové řízení se používá, pokud je nutná rychlá reakce na změnu otáček či momentu.

• Přímé řízení momentu asynchronního motoru:

Pro přímé řízení momentu je nutná znalost matematického modelu asynchronního motoru, který je získán při identifikaci připojeného motoru k frekvenčnímu měniči. Většina frekvenčních měničů, kromě nejlevnějších typů, disponuje automatickou detekcí připojeného motoru - tzv. funkcí autotuning, kdy měnič rozběhne motor a provádí sérii měření jejichž výsledkem je získání matematického modelu připojeného indukčního motoru. (NEBORÁK, 2011)

Přímé řízení momentu je založeno na vytvoření točivého magnetického pole statoru, přičemž můžeme řídit rychlost otáčení tohoto pole a tím v každém časovém okamžiku také řídit výsledný moment motoru.

Rychlost otáčení točivého magnetického pole statoru n_s viz vztah (1.1).

1.2 Frekvenční měnič ACS355

Jedná se o frekvenční měnič vyrobený firmou ABB, sloužící k řízení střídavých (asynchronních i synchronních) elektromotorů. Frekvenční měnič ACS355 je dodáván v několika verzích, lišících se v napájecím napětí (200 – 240 V nebo 380 – 480 V), fázi napájecího napětí (jedno- nebo tří-fázové) nebo ochranném stupni krytí IP66/IP67. Rozsah výkonu elektromotorů pro řízení měničem ACS355 je 0,37 kW až 22 kW. (ABB, 2010)

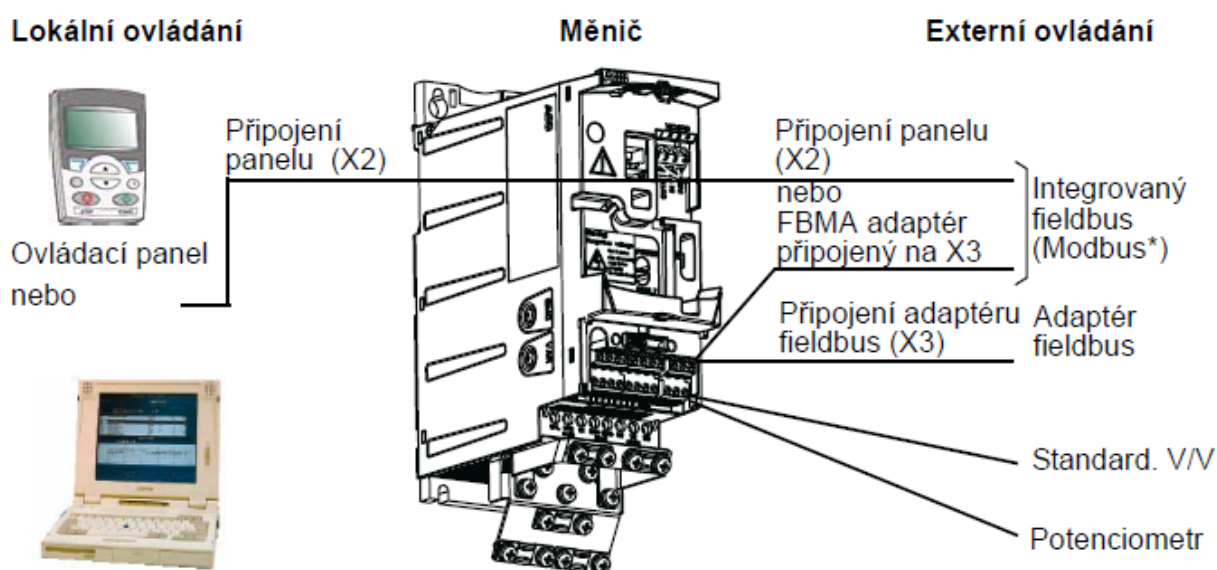
Komunikaci s ostatními účastníky automatizačních systémů zajišťuje procesní aplikační sběrnice (module fieldbus), která podporuje následující komunikační protokoly:

- PROFIBUS DP,
- DeviceNet,
- CANopen,
- Modbus RTU,
- Ethernet IP / Modbus TCP/IP,
- EtherCat.



Obr. 2 – Frekvenční měnič ACS355. (ABB, 2010)

Frekvenční měnič disponuje několika způsoby ovládání. Může přijímat povely pro start, stop, reverzaci směru nebo změnu referenčních hodnot z ovládacího panelu nebo přes digitální/analogové vstupy z nadřazené úrovně (PLC). Další možností je využití sběrnice fieldbus (integrováný nebo s nadstandardním adaptérem) pro připojení k nadřazené úrovni s PC nebo logickým automatem. V této práci je využito integrovaného fieldbus s připojením X2, viz obr. 3, které slouží buď pro připojení ovládacího panelu nebo propojení měniče s PC. Na zvoleném PC musí být nainstalována aplikace DriveWindow Light 2, která byla vyvinuta pro konfiguraci frekvenčních měničů společnosti ABB. (ABB, 2010)

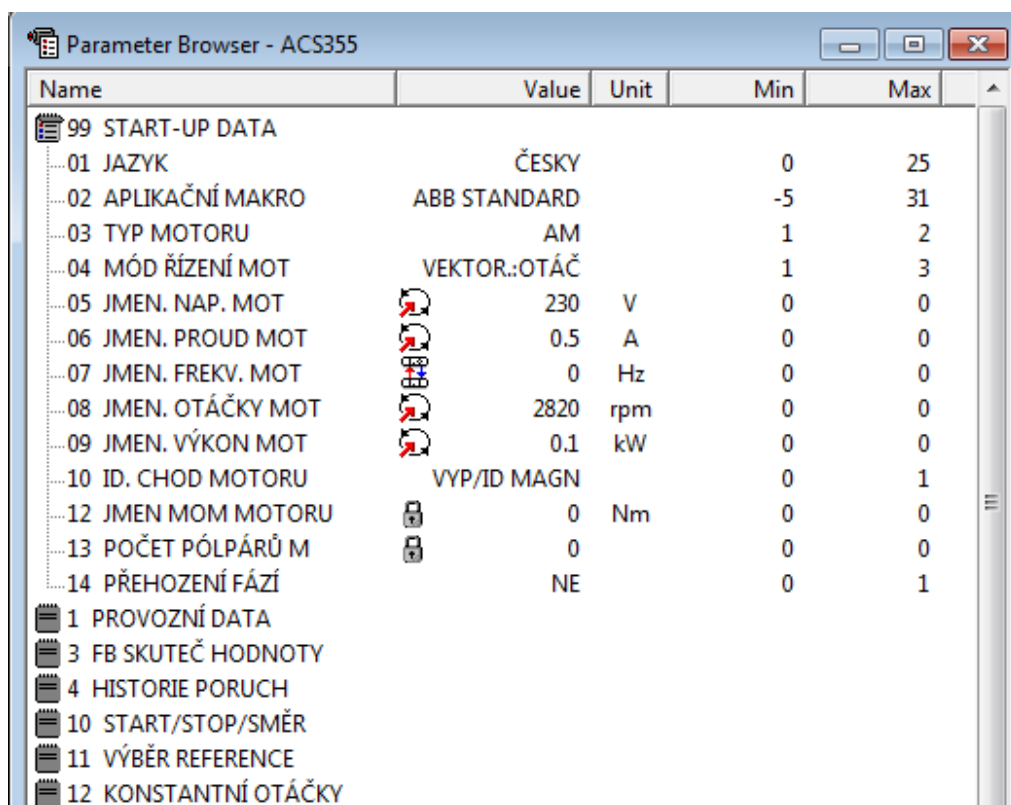


Obr. 3 – Možnosti ovládání frekvenčního měniče ACS355. (ABB, 2010)

Integrovaný fieldbus frekvenčního měniče ACS355 podporuje protokol Modbus RTU. Jedná se o sériový, asynchronní protokol, jehož transakce jsou v polovičním duplexu (v jeden časový okamžik probíhá přenos dat pouze jedním směrem). Připojení k integrovanému fieldbus je přes sériovou linku RS-232, která je u ACS355 koncipována pro aplikace typu point-to-point (jeden master řídí jeden slave). Komunikace s využitím protokolu MODBUS funguje na principu předávání datových zpráv mezi klientem a serverem – komunikace typu Master/Slave. Modbus je protokol typu žádost/odpověď (request/reply), tzn. klient odešle požadavek serveru, který jej vykoná a následně odešle odpověď zpět klientu. (ABB, 2010)

1.2.1 Lokální řízení frekvenčního měniče ACS355

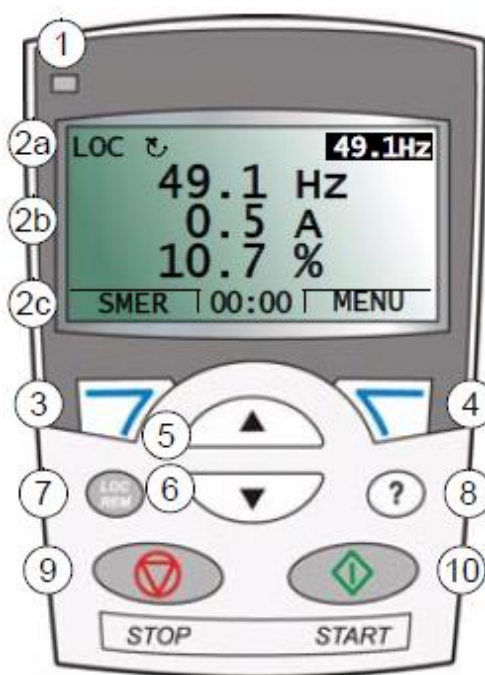
Nejjednodušší možností konfigurace frekvenčního měniče je skrze lokální ovládání přímo z ovládacího panelu (viz obr. 5) Při prvním spuštění měniče je nutné nastavit základní parametry. Parametr je číselné označení, které nese informaci o stavu (např. 99.03 – typ motoru) či aktuální hodnotě sledované veličiny (např. 01.03 – aktuální frekvence). Parametry jsou rozděleny do několika skupin, které charakterizuje první dvojčíslí (např. 99 – Start-up data). Každá skupina je dále rozdělena na dílčí části, přičemž každou dílčí část reprezentuje druhé dvojčíslí (např. 99.01 – výběr jazyka ve skupině Start-up data). K nastavení základních parametrů je vhodné použít START-UP asistenta, který je předdefinovaný a provede uživatele základní konfigurací. Na obr. 4 je zobrazena část parametrů v aplikaci DriveWindow Light 2.



Name	Value	Unit	Min	Max
99 START-UP DATA				
01 JAZYK	ČESKY		0	25
02 APLIKAČNÍ MAKRO	ABB STANDARD		-5	31
03 TYP MOTORU	AM		1	2
04 MÓD ŘÍZENÍ MOT	VEKTOR.:OTÁČ		1	3
05 JMEN. NAP. MOT	230	V	0	0
06 JMEN. PROUD MOT	0.5	A	0	0
07 JMEN. FREKV. MOT	0	Hz	0	0
08 JMEN. OTÁČKY MOT	2820	rpm	0	0
09 JMEN. VÝKON MOT	0.1	kW	0	0
10 ID. CHOD MOTORU	VYP/ID MAGN		0	1
12 JMEN MOM MOTORU	0	Nm	0	0
13 POČET PÓLPÁRŮ M	0		0	0
14 PŘEHOZENÍ FÁZÍ	NE		0	1
1 PROVOZNÍ DATA				
3 FB SKUTEČ HODNOTY				
4 HISTORIE PORUCH				
10 START/STOP/SMĚR				
11 VÝBĚR REFERENCE				
12 KONSTANTNÍ OTÁČKY				

Obr. 4 – Přehled vybraných parametrů v aplikaci DriveWindow Light 2.

Po nastavení všech parametrů, s předpokladem správného hardwarového propojení řízeného motoru s frekvenčním měničem, je uživatel schopen ovládat daný motor z ovládacího panelu. K dispozici je START/STOP, reverzace směru chodu motoru a možnost zadávání referenčních hodnot (otáčky, frekvence, moment). Součástí frekvenčního měniče je také PID regulátor.



Obr. 5 – Vzhled asistenčního ovládací panelu. (ABB, 2010)

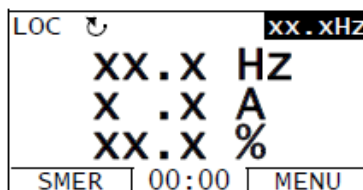
Práce s ovládacím panelem je velmi intuitivní a přehledná. Displej obsahuje potvrzující a návratové tlačítko a pro pohyb mezi jednotlivými řádky slouží šipky nahoru/dolů. Popis prvků ovládacího panelu je znázorněn v tab. 1.

Tab. 1 – Popis funkčních tlačítek asistenčního ovládacího panelu








Odkaz č.	Popis funkce
1	Stavová signalizační LED.
2	LCD displej – funkčně rozdělen do tří částí.
	2a – stavový řádek (základní stavové informace frekvenčního měniče).
	2b – střed (variabilní zobrazení signálů/parametrů, menu, poruch, alarmů).
	2c – dolní řádek (zobrazení aktuální funkce soft tlačítek, zobrazení času).
3	Soft tlačítko 1 – text v dolním levém rohu LCD displeje zobrazuje funkci.
4	Soft tlačítko 2 – text v dolním pravém rohu LCD displeje zobrazuje funkci.
5	Nahoru – listování v menu, zvyšování hodnoty parametru.
6	Dolů – listování v menu, snižování hodnoty parametru.
7	LOC/REM – přepínání mezi lokálním (LOC) a vzdáleným (REM) ovládním.
8	Nápověda – související informace zvýrazněné položky ve středí části displeje.
9	STOP – zastavení frekvenčního měniče v režimu lokálního ovládní.
10	START - spouštění frekvenčního měniče v režimu lokálního ovládní.

1.2.1.1 Manuální ovládání frekvenčního měniče z ovládacího panelu

- Připojte napájecí napětí (ovládací panel se zapne ve výstupním režimu)

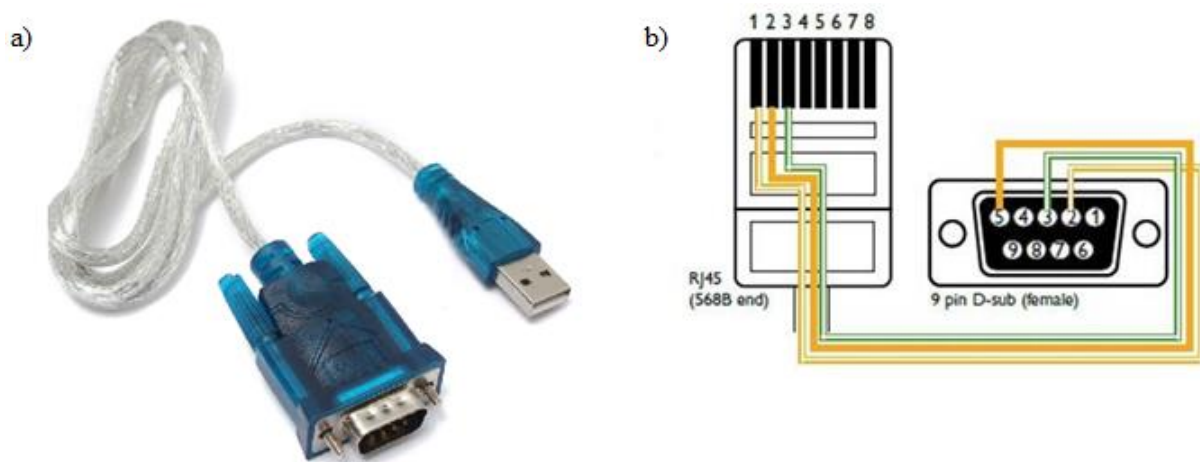


Obr. 6 – Display ovládacího panelu ve výchozím režimu. (ABB, 2010)

- Asistenční ovládací panel se dotáže: "Chcete použít start-up asistenta?".
- Stiskněte  pro použití start-up asistenta (doporučeno).
- Stiskněte  pro ruční nastavení parametrů (nutná znalost parametrů).
- Při výběru start-up asistenta následuje postupné nastavení důležitých parametrů.
- Volba jazyka (parametr 99.01): ACS355 podporuje český jazyk.
- Aplikační makro (par. 99.02): ABB STANDART postačí ve většině případů.
- Typ motoru (par. 99.03): AM (asynchronní motor), PMSM (motor s permanentními magnety).
- Mód řízení motoru (par. 99.04): např. skalární nebo vektorové řízení.
- Data motoru (par. 99.05 – 99.14): např. jmenovité otáčky, jmenovitý proud.
- Dále pokračuje nastavení dalších parametrů, důležitých pro správnou funkci frekvenčního měniče. V případě, že uživatel není schopen rozhodnout o nastavení specifického parametru, ponechá výchozí nastavení, které vyhovuje ve většině případů. Přímou na ovládacím panelu je k dispozici tlačítko  (Help), viz obr. 5 – odkaz č. 8.
- Po úspěšném nastavení parametrů bude Start-up asistent ukončen a ovládací panel se přepne do výchozího režimu, který je znázorněn na obr. 6.
- Pokud je ovládací panel přepnut do výchozího režimu, je možno manuálně ovládat připojený motor. Tlačítkem  je spuštěn chod motoru, tlačítkem  je motor pozastaven. Tlačítka   slouží ke zvýšení/snížení výstupní frekvence měniče a tím ke změně otáček motoru. (ABB, 2010)

1.2.1.2 Lokální ovládání frekvenčního měniče z PC

Frekvenční měnič ACS355 nabízí také možnost lokálního ovládání z PC. V podstatě je ovládací panel nahrazen počítačem, na kterém je nainstalována softwarová aplikace DriveWindow Light 2. Ovládací panel je připojen k frekvenčnímu měniči s využitím integrovaného fieldbus přes konektor RJ-45, viz obr. 3 (přípojka X2). Jedná se Modbus komunikaci s využitím sériového rozhraní RS-232. Na straně frekvenčního měniče je použit konektor RJ-45 a na straně PC je použito rozhraní USB. Je tedy nutné použít převodník RS232/USB a jelikož RS232 standardně používá konektor Cannon 9 je nutné vytvořit také redukci RJ-45/Cannon 9.



Obr. 7 – a) Převodník RS232/USB, b) Zapojení kabelu RJ-45/Cannon 9.

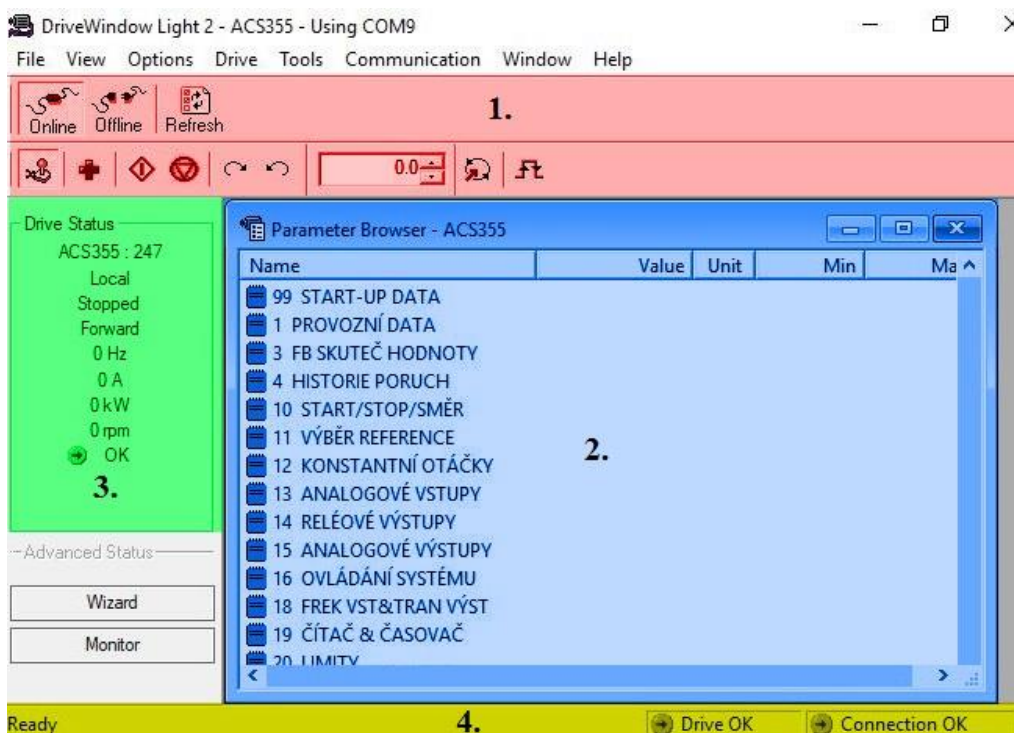
Tab. 2 – Popis pinů u kabelu RJ-45/Cannon 9

RJ-45	Cannon 9
Pin 1 (bílo-oranžová)	Pin 2 (Vstup, příjem dat – Rx)
Pin 2 (oranžová)	Pin 5 (uzemnění – GND)
Pin 3 (bílo-zelená)	Pin 3 (Výstup, přenos dat – Tx)

Pro zajištění správné komunikace mezi počítačem, na kterém běží aplikace DWL a frekvenčním měničem je nutné nastavit parametry frekvenčního měniče do stejného stavu jako při ovládání z panelu měniče, viz kapitola 1.2.1.1. Frekvenční měnič musí být také přepnut do lokálního režimu, a nikoliv do režimu vzdáleného řízení (i přesto, že je měnič ovládán ze vzdáleného místa). Připojení k PC s využitím aplikace DWL totiž funguje stejně jako při použití ovládacího panelu, pouze je panel nahrazen počítačem.

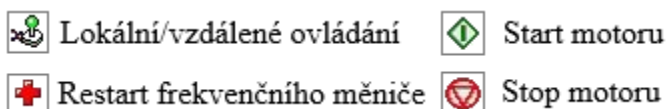
• Drive Window Light 2

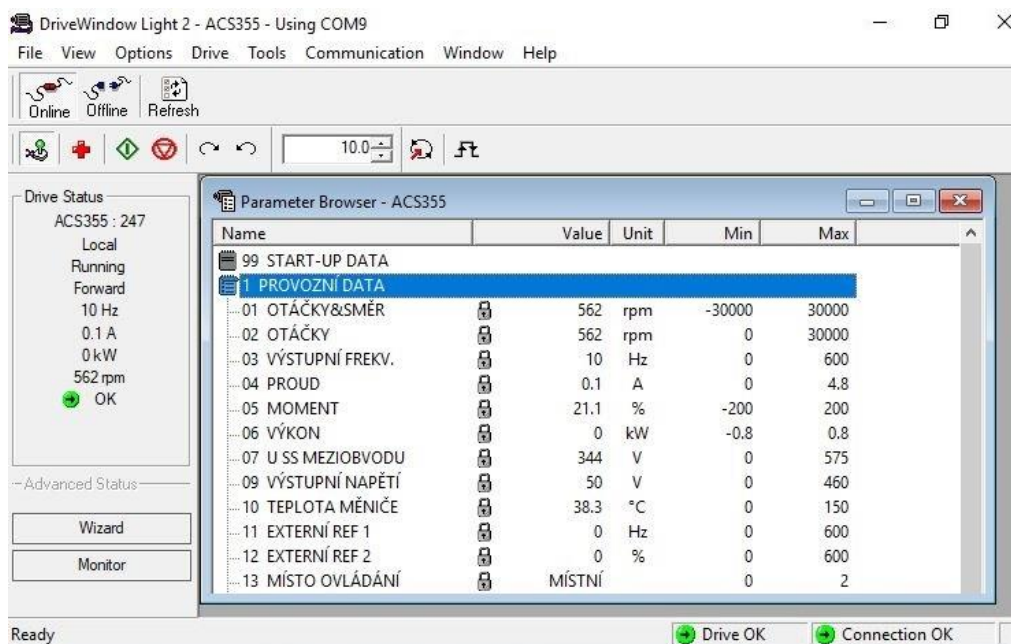
Aplikace společnosti ABB s. r. o. vyvinutá pro konfiguraci frekvenčních měničů. Jedná se o snadno použitelný softwarový nástroj pro spouštění, monitorování a ladění výkonu frekvenčních měničů. Po spuštění aplikace započne DriveWindow Light (dále jen DWL) automatický skenovací cyklus. Pokud skenovací cyklus z nějakého důvodu selže (není připojeno zařízení nebo není možné připojené zařízení identifikovat) naskytne se možnost ručního nastavení a připojení nového zařízení. (ABB, 2014)



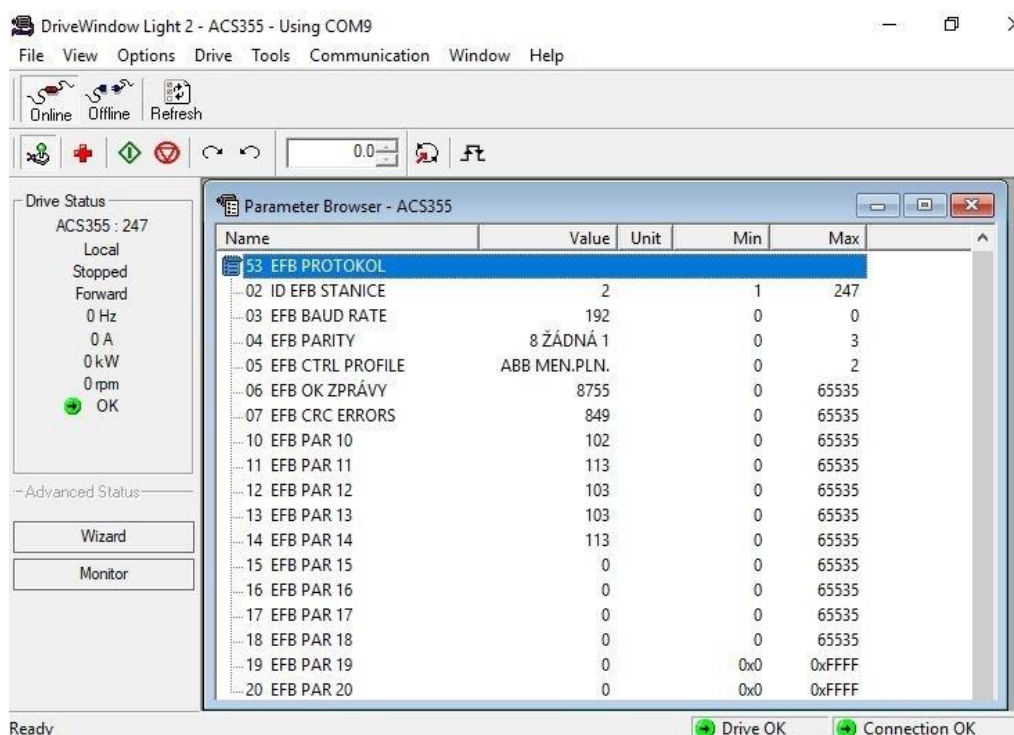
Obr. 8 – Úvodní obrazovka aplikace DWL.

- 1 – Panel nástrojů (sloužící pro ovládání frekvenčního měniče).
 - 2 – Prohlížeč parametrů (slouží pro konfiguraci parametrů frekvenčního měniče).
 - 3 – Stavové okno (informuje o aktuálním stavu připojeného měniče).
 - 4 – Informační lišta (informuje o stavu komunikace a připojeného zařízení).
- Tlačítko "Wizard" představuje průvodce aplikací DWL.
 - Tlačítko "Monitor" slouží pro monitorování chodu frekvenčního měniče.
 - Ovládací tlačítka panelu nástrojů:





Obr. 9 – Aplikace DWL – sledování provozních dat.

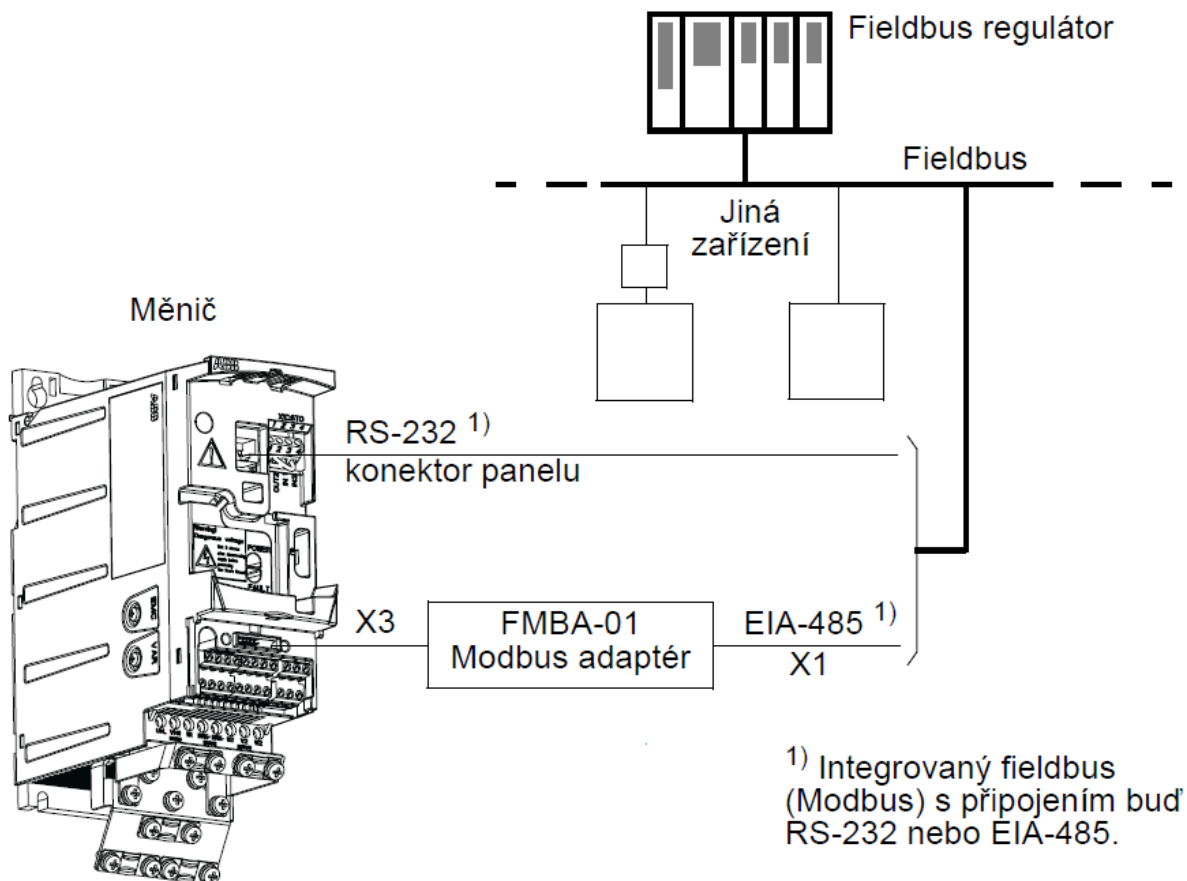


Obr. 10 – Aplikace DWL – nastavení komunikačního protokolu.

Parametr 53.02–53.04 slouží k nastavení komunikace. Parametry 53.10–53.17 jsou určeny ke čtení po sériové lince a vkládá se do nich tzv. odkaz na parametry – tedy číslo parametru, jehož hodnotu chceme znát (např. 1.02 – aktuální otáčky). 53.18 nastavuje časové zpoždění linky před odesláním Modbus odpovědi. 53.19 obsahuje kopii řídicího slova (CW). 53.20 obsahuje informaci o zvoleném profilu řízení frekvenčního měniče

1.2.2 Vzdálené řízení frekvenčního měniče ACS355

Vzdálené řízení frekvenčního měniče bude zprostředkováno programovatelným logickým automatem. Frekvenční měnič může být připojen ke vzdálenému řídicímu systému (PLC) prostřednictvím průmyslové sběrnice fieldbus, u které je možno využít dvojího zapojení, a to buď s využitím standardu RS232 nebo komunikačního adaptéru. (ABB, 2010)



Obr. 11 – Způsob připojení k integrovanému fieldbus. (ABB, 2010)

Frekvenční měnič může komunikovat s řídicím systémem přes fieldbus adaptér, který podporuje několik sériových komunikačních protokolů:

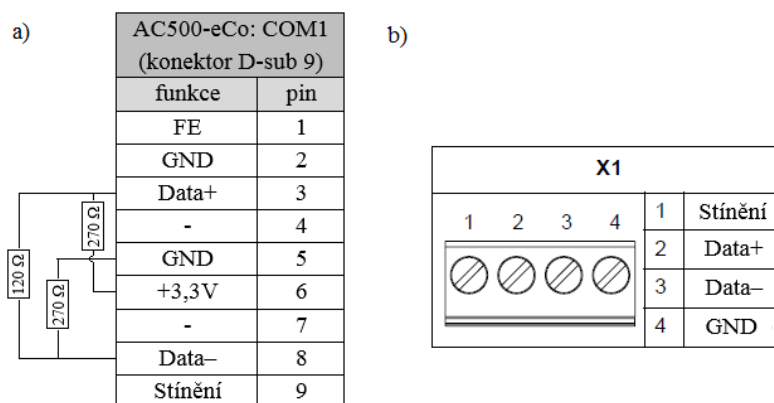
- PROFIBUS-DP (adaptér FPBA-01),
- Ethernet (adaptér FENA-01),
- CANopen (adaptér FCAN-01),
- ModbusRTU (adaptér FMBA-01).
- DeviceNet™ (adaptér FDNA-01),

Použitým řídicím systémem bude PLC společnosti ABB – **AC500-eCo**. Jelikož dostupné PLC v laboratoři je ve verzi eCo, což je úspornější varianta systému AC500, lze pro komunikaci využít pouze sériovou linku RS485. Ethernet je použit pro připojení PLC k počítači.

1.2.2.1 Vzdálené řízení s využitím adaptéru FMBA-01

Zapojení s komunikačním adaptérem se standardně využívá v průmyslové automatizaci a v této práci je využito pro seznámení se s principem a možnostmi sériové komunikace. Adaptér FMBA-01 pro komunikaci využívá integrovaný fieldbus měniče s podporou komunikačního protokolu Modbus RTU.

Pro zajištění bezchybné výměny informací je třeba vytvořit datový kabel pro sériovou komunikaci po lince EIA485. Ve verzi AC500eCo není rozhraní COM opticky izolované, proto se doporučuje připojit uzemnění (třívodičové zapojení *Data+*, *Data-* a *GND*, namísto klasického dvou vodičového zapojení *Data+* a *Data-*). Na straně PLC musí být pro zamezení odrazu signálu umístěn zakončovací odpor 120 Ω. Adaptér FMBA-01 tento zakončovací odpor již obsahuje, a proto nemusí být součástí datového kabelu na straně frekvenčního měniče. Pro udržení definované logické hodnoty na vodiči v klidovém režimu (nedochází k výměně dat nebo je odpojená linka) je nutné připojit na datový kabel ještě tzv. Pull Up/Down rezistor, který má v případě verze eCo hodnotu 270 Ω. (ABB, 2005)



Obr. 12 – a) Zapojení pinů COM1 (PLC AC500eCo), b) Zapojení pinů adaptéru FMBA-01.



Obr. 13 – Adaptér FMBA-01. (ABB, 2005)

Nastavení frekvenčního měniče bylo popsáno v kapitole 1.2.1. Pro vzdálené ovládání se bude nastavení parametrů mírně lišit, jelikož příkazy pro start, stop či reverzaci nepřichází z ovládacího panelu či PC, ale přichází po komunikační lince z nadřazené úrovně. Frekvenční měnič musí být přepnut z lokálního režimu do vzdáleného (Remote).

Tab. 3 – Nastavení parametrů ACS355 pro řízení s adaptérem FMBA-01

Parametr	Význam	Nastavení
Parametry sériové komunikace		
98.02	Výběr komunikačního protokolu	STD MODBUS
53.02	EFB ID Stanice	1 až 247
53.03	EFB Baud Rate	19,2
53.04	EFB Parita	8ŽÁDNÁ1 (8NONE1)
53.05	EFB CTRL Profil komunikace	ABB MEN. PLN.
Parametry řízení frekvenčního měniče		
10.01	EXT1 příkazy	KOM
10.03	Směr otáčení	ŽÁDOST
11.03	Výběr REF1	KOMUNIKACE
16.01	Umožnění chodu	COMM
16.04	Výběr resetu poruchy	KOM
16.08	Umožnění startu 1	COMM
30.18	Funkce porucha komunikace	Porucha
40.01	Zesílení (PID regulátor 1)	0,1 až 100
40.02	Integrační čas (PID regulátor 1)	0 = nevybráno, jinak 0,1 až 3600 [s]
40.03	Derivační čas (PID regulátor 1)	0 = nevybráno, jinak 0,01 až 10,0 [s]
40.10	Výběr žádané hodnoty	KOMUNIKACE

1.2.2.2 Vzdálené řízení s využitím RS232

Poslední možností, která bude také využita pro finální úlohu s oběma měniči, je vzdálené ovládání frekvenčního měniče přes integrovaný fieldbus s přípojkou RS232 (viz obr. 3 – přípojka X2). Tento způsob byl využit pro připojení měniče k PC s tím rozdílem, že nyní budou příkazy pro start, stop, referenční hodnoty atd. přijímány z nadřazené úrovně po sériové lince. Pro připojení k logickému automatu bude využit stejný komunikační kabel jako v případě připojení k počítači (viz kapitola 1.2.1.2).

Nyní bude měnič přepnut do vzdáleného režimu (remote), bude mu přiřazena jedinečná adresa a veškeré příkazy bude měnič očekávat ze sériové linky, a tudíž nebude možné ovládat jej z ovládacího panelu. Nastavení parametrů bude totožné jako při vzdáleném ovládání s využitím adaptéru FMBA-01, pouze se bude lišit parametr pro výběr komunikačního protokolu. Základní sada parametrů a jejich nastavení je znázorněno v tab. 3, přičemž parametr 98.02 musí být nastaven na "RS232".

1.3 Frekvenční měnič UniDrive M200

Jedná se o frekvenční měnič společnosti EMERSON, typové řady UniDrive, použitelný k řízení třífázových elektrických pohonů. Jedná se o modulární frekvenční měnič, který lze dle potřeb rozšířit o komunikační či I/O jednotky. Měniče řady UniDrive jsou rozděleny do čtyř typových velikostí a dále do pěti tříd jmenovitého vstupního napětí (100, 200, 400, 575, 690 V). Rozsah výkonu elektromotorů řízených frekvenčním měničem M200 je 0,37 kW až 22 kW. (CONTROL TECHNIQUES BRNO, 2002)



Obr. 14 – UniDrive M200 typové velikosti 2. (CONTROL TECHNIQUES BRNO, 2002)

Frekvenční měnič je možno ovládat několika způsoby. Základní možností je ruční ovládání přímo z ovládacího panelu frekvenčního měniče, ale tato možnost není příliš pohodlná vzhledem k velikosti displeje a velkému množství parametrů. Druhou možností je připojit měnič k PC, na kterém je nainstalovaná aplikace UniDrive M Connect.

Poslední možností je využití komunikačního adaptéru AI-485. Tento způsob se využívá pro komunikaci s nadřazenou úrovní v průmyslové automatizaci. Adaptér AI-485 podporuje na fyzické vrstvě standard RS485 a pro přenos dat využívá komunikační protokol Modbus RTU.

Veškeré informace o stavu frekvenčního měniče, možnostech jeho ovládání či základních datech o připojeném motoru jsou uloženy do tzv. parametrů se kterými měnič dále pracuje. Některé parametry jsou určeny pouze pro čtení, do jiných je možno zapisovat a také je číst.

Základní tezí u frekvenčních měničů UniDrive řady M je rozdělení všech parametrů do jednotlivých podskupin – tzv. Menu. Měnič M200 obsahuje několik skupin Menu, přičemž se každá skupina zaměřuje na určitou specifickou část pracovního cyklu nebo hardware zařízení měniče (poruchy, řízení, monitoring, digitální/analogové I/O jednotky atd.). Navíc je vytvořeno tzv. Menu 0, které představuje základní výběr parametrů ze všech ostatních skupin Menu, jejichž nastavení ve většině případů stačí pro jednoduché operace.

Tab. 4 – Přehled jednotlivých skupin Menu (CONTROL TECHNIQUES BRNO, 2002)

Menu	Popis	Menu	Popis
0	Uživatelské menu.	10	Stavy a poruchy.
1	Zadávání otáček.	11	Sériová komunikace, obecné nastav.
2	Rampy.	12	Programovatelné komparátory, řízení externí brzdy.
3	Řízení otáček.		
4	Regulace momentu a proudu.	14	Uživatelský PID regulátor.
5	Řízení motoru.	15	Menu volitelného modulu slotu 1.
6	Režimy.	18	Aplikační menu 1.
7	Analogové vstupy a výstupy.	20	Aplikační menu 2.
8	Digitální vstupy a výstupy.	21	Parametry druhého motoru.
9	Programovatelná logika, časovače.	22	Nastavení Menu 0.

Velká část parametrů je určena ke čtení i zápisu a zbytek pouze ke čtení. Každý parametr disponuje unikátním číslem registru, pro zpřístupnění parametrů sériové komunikaci. Nadřazený systém má možnost získat informaci o hodnotě daného parametru vyčtením odpovídajícího registru frekvenčního měniče. K tomuto účelu se používá Modbus funkce 3 (Read Holding Registers – Čti uchovávací registry) – viz kapitola 3.

• **Výpočet odpovídajícího registru jednoho parametru:**

$$R = m \cdot 100 + p - 1 \quad (1.3)$$

kde R ... je číslo registru, m ... je číslo menu, p ... je číslo parametru.

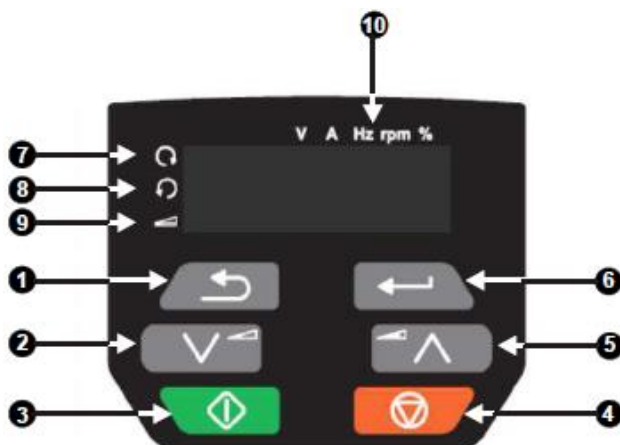
Například pro přečtení jmenovitého napětí motoru, které je uloženo v parametru 05.009.

$$R = 5 \cdot 100 + 9 - 1 = 508$$

V uchovávacím registru č. 508 je uložena aktuální hodnota jmenovitého napětí motoru.

1.3.1 Ruční ovládání z panelu frekvenčního měniče

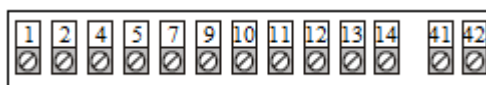
Manuální ovládání frekvenčního měniče přímo z ovládacího panelu se využívá spíše při prvotním připojení motoru k měniči s následným testováním funkčnosti. Množství parametrů s širokými možnostmi jejich nastavení je při použití tlačítek Vstoupit/Opustit a Nahoru/Dolů zdlouhavé a nepohodlné.



Obr. 15 – Ovládací panel měniče M200. (CONTROL TECHNIQUES BRNO, 2002)

- | | |
|-----------------------------|------------------------------------|
| 1- Tlačítko "Escape" (zpět) | 6- Tlačítko "Enter" (vstoupit) |
| 2- Tlačítko "dolů" | 7- Indikátor "Provoz vpřed" |
| 3- Tlačítko "Start" | 8- Indikátor "Provoz vzad" |
| 4- Tlačítko "Stop/Reset" | 9- Indikátor zadávání z klávesnice |
| 5- Tlačítko "nahoru" | 10- Indikátor jednotek |

• **Svorkovnice řízení:**



Obr. 16 – Svorkovnice řízení.

Tab. 5 – Přehled popisu svorkovnice řízení

Svorka	Význam	Svorka	Význam
1	Společná zem (GND).	10	Digitální vstup/výstup 1 (DI/O 1).
2	Analogový vstup 1 (AI1).	11	Vstup "Blokování" (Enable).
4	Zdroj +10 V.	12	Vstup "Provoz vpřed".
5	Analogový vstup 2 (AI2).	13	Vstup "Provoz vzad".
7	Analogový výstup 1 (AO1).	14	Volba AI1 nebo AI2
9	Zdroj +24 V.	41,42	Spínací kontakty relé.

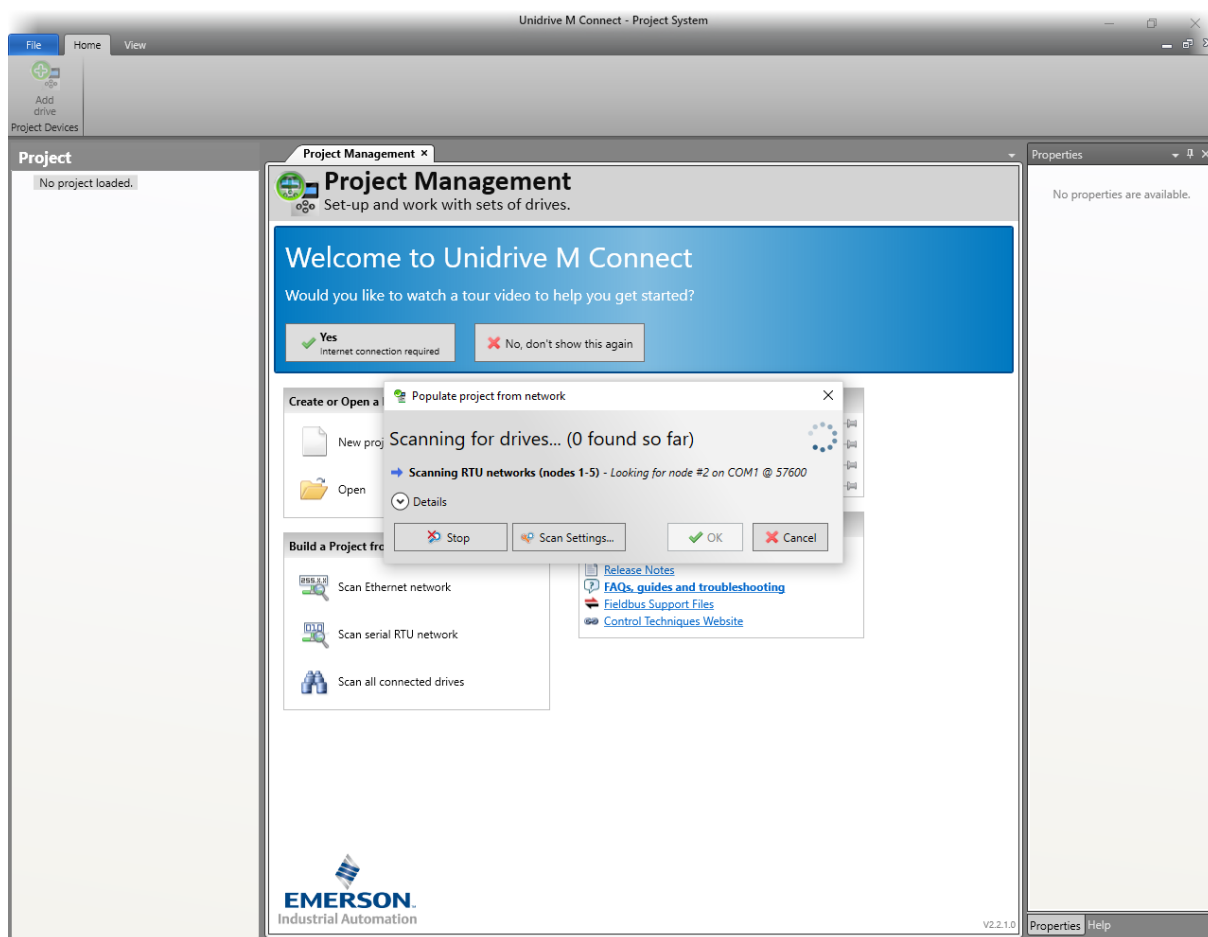
• Postup manuálního ovládání:

- Frekvenční měnič připojíme do elektrické sítě, čímž dojde k jeho zapnutí.
- Na displeji se nejprve na okamžik objeví status "OPEn.LP" (Open Loop – otevřená smyčka (parametr 11.031)) což značí nastavenou kategorii měniče. Poté se na displeji objeví status "inh", který signalizuje blokování měniče (nelze zapnout motor).
- Přivedením log 1 na svorku 11 odblokujeme měnič – na displeji se zobrazí status "rdy". Log 1 na svorku blokování přivedeme tak, že propojíme svorku 9 (24V) se svorkou 11.
- Tlačítkem "Start" je motor uveden do chodu (vnutím protéká proud, ale rotor má nulové otáčky).
- Držením tlačítka "nahoru" dojde k postupnému zvyšování frekvence napájení motoru, čímž dojde ke zvýšení otáček (do hodnoty omezení maximální frekvence (pr 01.006)).
- Držením tlačítka "dolů" dojde k postupnému snižování frekvence napájení motoru, dokud frekvence nedosáhne omezení minimální frekvence (parametr 01.007).
- Tlačítkem "Stop" dojde k zastavení motoru po přednastavené rampě (parametr 02.021). (CONTROL TECHNIQUES BRNO, 2002)

1.3.2 Ovládání frekvenčního měniče M200 z PC

Frekvenční měnič UniDrive M200 může být pro účely ovládání připojen k počítači, který je vybaven aplikací UniDrive M Connect. K fyzickému propojení s počítačem slouží výrobcem dodávaný CT USB komunikační kabel. Frekvenční měnič musí být vybaven adaptérem AI-485, který podporuje standard EIA 485 (dříve RS 485) – viz kapitola 1.3.3, obr. 19. USB CT komunikační kabel disponuje na straně měniče konektorem RJ45 a na straně PC je vybaven USB konektorem.

Velkou výhodou aplikace UniDrive M Connect je automatický skenovací cyklus, který kromě nalezení připojeného portu počítače, definuje ID připojeného zařízení a jeho přenosovou rychlost. Uživatel tedy nemusí nastavovat žádné komunikační parametry, pouze propojí příloženým kabelem počítač s měničem a následně spustí skenovací cyklus. Aplikace nejprve nalezne připojený port počítače (tzv. COM) a poté postupným zvyšováním přenosové rychlosti nalezne použitou přenosovou rychlost.



Obr. 17 – Úvodní obrazovka aplikace UniDrive M Connect s aktivním skenovacím cyklem.

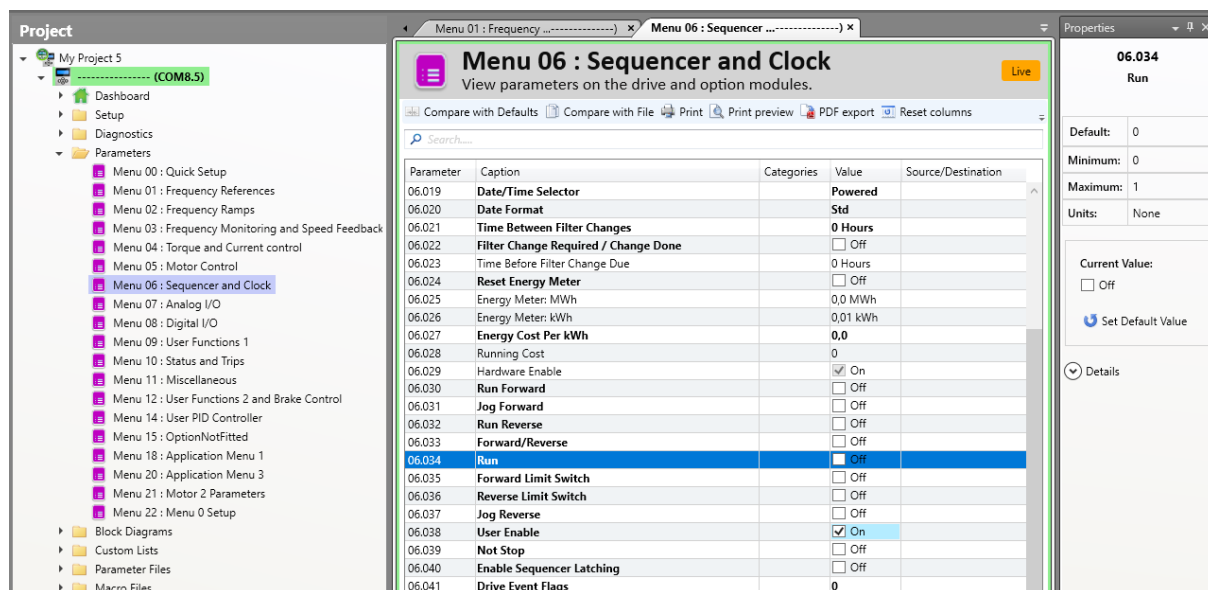
UniDrive M Connect kromě správy parametrů umožňuje také diagnostiku připojeného motoru a dále umožňuje tvorbu logických schémat pro řízení. Aplikace neslouží pouze pro ruční ovládání měniče z počítače, ale je možné ji použít i pro jednodušší logické úlohy. Ovšem u logických úloh je nutné si uvědomit, že osobní počítače nedisponují takovou robustností systému jako průmyslové počítače a je přítomno větší riziko selhání systému.

Možnost ručního ovládání z počítače je velmi pohodlná v případě oživování systému po jeho prvotním připojení. Technik po zapojení hardwarových částí připojí k měniči přenosný počítač a může otestovat jeho funkčnost nebo ověřit správnost jmenovitých hodnot motoru.

Pro ruční ovládání motoru je nutné znát několik základních parametrů, které jsou umístěny v různých menu měniče. Samotná aplikace nedisponuje start/stop tlačítkem jako v případě aplikace DWL u měniče ACS355 a je možné ji ovládat pouze přes parametry. Seznam základních parametrů nutných pro ovládání připojeného motoru je znázorněn v tab. 6.

Tab. 6 – Přehled základních parametrů pro start/stop motoru a změnu otáček

Parametr	Význam	Stav
06.015	Blokování (na svorkovnici řízení – svorka 11)	Log. 0 = off
06.030	Konfigurační bit: provoz vpřed	Log. 1 = on
06.032	Konfigurační bit: provoz vzad	Log. 1 = on
06.033	Konfigurační bit: reverzace	Log. 0 = off
06.034	Konfigurační bit: provoz	Log. 1 = on
01.015	Volba přednastavených otáček	1 až 8
01.021	Přednastavené otáčky 1	[Hz]
⋮	⋮	⋮
01.028	Přednastavené otáčky 8	[Hz]



Obr. 18 – Screen aplikace UniDrive M Connect pro spuštění motoru.

• Funkce Autotune

Tato interní funkce frekvenčního měniče slouží pro bližší charakteristiku připojeného motoru. Funkce Autotune může být spuštěna **bez otáčení motoru** nebo **s otáčením motoru**. Pokud připojený motor umožňuje jeho spuštění (za předpokladu, že není připojen do výrobního provozu) je vhodné použít funkci Autotune s otáčením motoru s následujícím postupem:

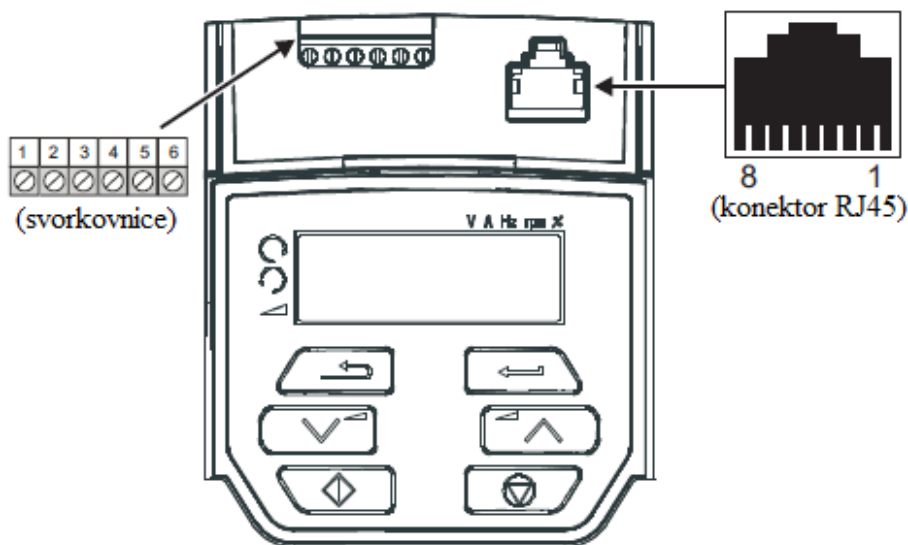
- nejprve se provede test bez otáčení motoru, ve kterém se měří odpor statoru a rozptylová indukčnost – hodnoty se využívají pro definici výchozích hodnot parametrů regulátoru,
- poté započne test s otáčením motoru, kdy se postupně zvyšují otáčky až na hodnotu 2/3 jmenovitého kmitočtu motoru – na tomto kmitočtu motor zůstane 40 sekund,
- měří se indukčnost statoru, body zlomu magnetizační charakteristiky motoru a účinník. (CONTROL TECHNIQUES BRNO, 2002)

1.3.3 Řízení frekvenčního měniče M200 z nadřazené úrovně

Pro připojení frekvenčního měniče k PLC slouží standard EIA485 s podporou protokolu ModbusRTU, který je dostupný ve výbavě měniče jakožto volitelný adaptér. Adaptér AI-485 je umístěn nad ovládacím panelem a disponuje dvěma možnostmi připojení. První možností je konektor RJ45, druhou možností je použití svorkovnice – viz tab. 7. K ovládání frekvenčního měniče z logického automatu se využívá stejných parametrů jako v případě ovládání z PC (viz kapitola 1.3.2). (CONTROL TECHNIQUES BRNO, 2002)

Tab. 7 – Pinout adaptéru AI-485

Pin	Konektor RJ45	Svorkovnice
1	Ukončovací odpor 120 Ω	GND
2	Rx Tx (Data+)	Rx\ Tx\ (Data-)
3	GND	Rx Tx (Data+)
4	+24 V (100mA)	Ukončovací odpor 120 Ω
5	-	Povolení Tx (enable)
6	Povolení Tx (enable)	+24 V (100mA)
7	Rx\ Tx\ (Data-)	-
8	Rx\ Tx\ (při použití ukončovacího odporu)	-



Obr. 19 – Adaptér AI-485 s výstupem RS485. (CONTROL TECHNIQUES BRNO, 2002)

Tab. 8 – Nastavení sériové komunikace

Parametr	Význam	Nastavení
11.023	Sériová adresa	1 až 247
11.024	Režim sériové linky	např. 8.2NP (8 data bit, 2 stop bit, NoneParite)
11.025	Přenosová rychlost	115200 [Bd]

2 Řídicí systémy společnosti ABB

V současné době využívá společnost ABB distribuovaný řídicí systém **800xA**, ve spojení s nejnovějším programovatelným logickým automatem **AC 800M Controller** (existují možnosti propojení i s jinými logickými automaty, ale přednostně je systém 800xA určen pro AC 800M). Vývojové prostředí automatizačních procesů je v současné době **Control Builder**, který slouží k vytvoření logického kódu pro řízení logického automatu. Pro vývoj kompletní aplikace, zajišťující řízení dílčích procesů, ale i tvorba operátorských stanovišť a jejich vizualizace (HMI or MMI – human (man) machine interface) slouží platforma **Engineering Workplace**, která spolupracuje s výše zmíněným softwarem Control Builder.

Navazujícím a v současné době vyvíjeným softwarem pro tvorbu průmyslových automatizačních systémů je platforma **EBase**, která zahrnuje kompletní sestavu vývoje automatizovaných systémů (výkresová dokumentace, vizualizace operátorských stanovišť a vývoj řídicích algoritmů). Software EBase by měl ulehčit tvorbu vyvíjených automatizovaných systémů, propojením dílčích platforem a měl by být schopen sám vytvořit část řídicího algoritmu z výkresové dokumentace, čímž dojde k časové úspoře (automatická tvorba tzv. maker, opakujících se úloh v řídicím algoritmu) a celkovému snížení nákladů při vývoji automatizovaných systémů v průmyslu.

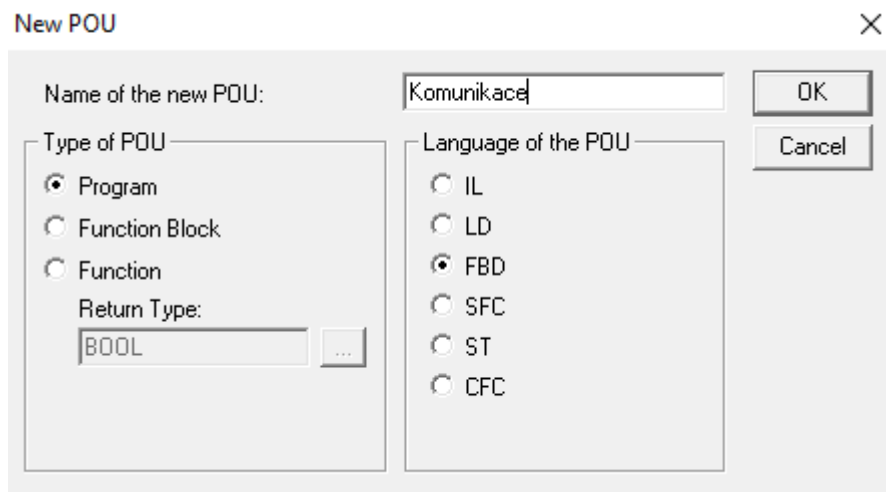
V této práci je ovšem použit programovatelný logický automat řady **AC500**, který je určen pro řízení náročných technologických procesů. Vývojovým prostředím pro AC500 je **Control Builder Plus**, v novějších verzích pak **Automation Builder**. AC500 je modulární logický automat, s možností připojení nespočtu řídicích jednotek, procesorových jednotek, komunikačních nebo I/O modulů atd. K jedné procesorové jednotce (CPU) lze připojit až deset I/O modulů, přičemž jeden I/O modul obsahuje až 32 I/O kanálů. Komunikační možnosti AC500 jsou přizpůsobeny různorodým požadavkům a v základní sestavě obsahují dva sériové porty (ASCII, Modbus, CS31), rozhraní pro Ethernet a FieldBusPlug pro průmyslové sběrnice Profibus s komunikačním standardem CANopen nebo DeviceNET. Samozřejmostí řady AC500 je možnost vytvoření redundantního PLC s využitím standartních komponent pro nasazení do bezpečnostních systémů (tzv. Safety systems). Vývojovým prostředím pro tvorbu řídicích algoritmů je software **CoDeSys**, který je implementován do Control Builder Plus. Předností tohoto prostředí je možnost kombinace různých jazyků pro tvorbu programů (instrukční list (IL), strukturovaný text (ST), kontaktní schéma (LD), funkční blokový diagram (FBD), vývojový diagram (CFC)).

2.1 Řídicí systém Codesys

Softwarová aplikace CoDeSys je programovací prostředí vyvinuté firmou *3S-Smart Software Solutions GmbH*, pro průmyslové PLC v souladu s mezinárodním standardem IEC 61131-3. Pro tvorbu řídicích algoritmů jsou k dispozici tyto programovací jazyky:

- posloupnost instrukcí (IL – Instruction List),
- strukturovaný text (ST – Structured Text),
- kontaktní schéma (LD – Ladder Diagram),
- jazyk pro sekvenční programování (SCF – Sequential Function Chart),
- jazyk funkčních bloků (FBD – Functional Block Diagram),
- jazyk vývojových diagramů (CFC – Continuous Function Chart).

Velkou výhodou prostředí CoDeSys je možná kombinace výše uvedených jazyků, což vede ke znatelnému ulehčení tvorby složitých řídicích programů. Další výhodou tohoto vývojového prostředí je možnost simulace vytvořených programů bez nutnosti připojení reálného PLC. Při tvorbě nového projektu je vyžadován jedinečný název projektu a dále je nutno definovat, zda se jedná o program, funkční blok nebo funkci. (KOVÁŘ, 2010)



Obr. 20 – Založení nového projektu v prostředí CoDeSys.

- **Program** – hlavní odlišností programu je, že se dá časovat pomocí tzv. tasku.
- **Funkční blok** – obsahuje libovolný počet vstupů a libovolný počet výstupů. Časování je stejné jako časování programu, ve kterém je daný funkční blok použit.
- **Funkce** – obsahuje libovolný počet vstupů, ale pouze jeden výstup (např. AND, OR).

Programovací jazyky se v základu rozdělují podle způsobu zadávání jednotlivých příkazů, a to na grafické a textové jazyky. Grafické jazyky ke své činnosti využívají kombinaci prvků a bloků, které jsou uspořádány tak, aby jejich konečná struktura zajistila požadovaný úkon, či sérii úkonů (obecně výstup). Oproti tomu se textové jazyky tvoří čistě sérií textových instrukcí (tzv. syntaxí), které po provedení či cyklickém opakování, zajistí požadovaný výstup. Grafické jazyky jsou obecně jednodušší, jelikož obrazová komunikace mezi člověkem a počítačem je snáze pochopitelná a představitelná. Textové jazyky jsou složitější a ke své správné činnosti potřebují vyšší informovanost a zkušenost programátora, ale dovolují vytvořit i velmi specifické požadavky konečného algoritmu.

• **Posloupnost instrukcí – IL:**

Zápis řídicího programu posloupností instrukcí je nejjednodušší způsob zpracování algoritmů. V programu jsou využity nejjednodušší operace (matematické operace, přesun dat mezi registry, volání proměnných atd.) a operandy odkazujícími na jednotlivé části paměti či registry. Zápis s využitím základních příkazů umožňuje vytvořit paměťově úsporný a rychlý program bez zbytečných operací, přesunů dat či bezvýznamného volání proměnných. Ovšem zápis s využitím základních operací dovoluje zpracovat pouze jednoduché a krátké algoritmy, jelikož se zvyšující se složitostí programu také roste nepřehlednost. (KOVÁŘ, 2010)

Výhody:

- přesná definice chování programu,
- tvorba paměťově a rychlostně úsporných programů.

Nevýhody:

- špatná orientace u delších programů,
- znalost příkazů a registrů.

• **Strukturovaný text (ST):**

Jedná se o programovací jazyk vyšší úrovně, ve němž se vyskytují předdefinované funkce a instrukce, který je velice oblíbený a rozšířený mezi většinou programátorů. Umožňuje zápis složitých algoritmů, jejichž syntaxe je velice podobná programovacímu jazyku Pascal či C (C++). Strukturovaný text je ideální pro práci s daty a řetězci dat, zpracování analogových signálů, digitálních filtrů, linearizaci signálů či složitých výpočetních algoritmů, ovšem je nutná znalost syntaxe kódu a příkazů, což není ideální kombinace pro začátečníky. (KOVÁŘ, 2010)

Výhody:

- rychlé a snadné programování složitých výpočetních operací,
- velice vhodný pro zpracování objemných dat a databází,
- vhodný pro práci s textovými řetězci,
- vhodný pro realizaci a zpracování datové komunikace.

Nevýhody:

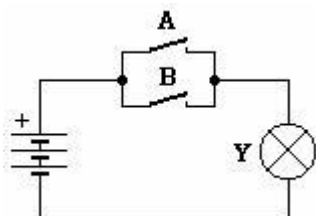
- vyšší časová náročnost pro znalost syntaxe a příkazů,
- méně vhodný pro zpracování velkého množství logických signálů.

0001	IF A AND B THEN	A = TRUE	B = FALSE
0002	AND_OUT := TRUE ;	AND_OUT = FALSE	
0003	END_IF		
0004			
0005	IF A OR B THEN	A = TRUE	B = FALSE
0006	OR_OUT := TRUE ;	OR_OUT = TRUE	
0007	END_IF		

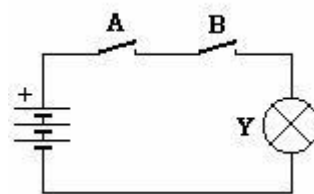
Obr. 21 – Příklad logické funkce OR/AND ve strukturovaném textu.

• Kontaktní schéma (LD):

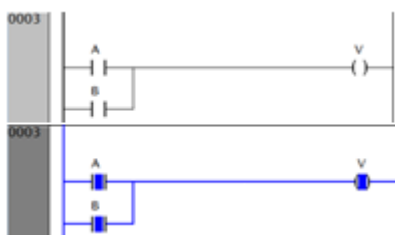
Kontaktní schéma je grafický jazyk obsahující softwarové ekvivalenty spínacích/rozpínacích kontaktů, paralelního či sériového řazení, obvody paměti, čítače a časovače (a další), jejichž kombinací docílíme požadovaného výstupu. Kontaktní schéma je více méně softwarová obměna soustav vzájemně propojených relátek s jejichž pomocí bylo možné sestavit logické funkce AND a OR (a jejich negace) v době, kdy ještě nebyly k dispozici mikrokontroléry, logické automaty či integrované obvody. Svou strukturou je kontaktní schéma ideální pro zpracování digitálních vstupů a výstupů (čítání, časování, cyklické opakování (např. spínání diod) atd.) či jednoduchého vyhodnocení analogových signálů. Předností tohoto způsobu zápisu řídicího algoritmu je jasná posloupnost programu, což lze využít při tvorbě dílčích podprogramů, které jsou implementovány do algoritmu složeného z více programovacích jazyků. (KOVÁŘ, 2010)



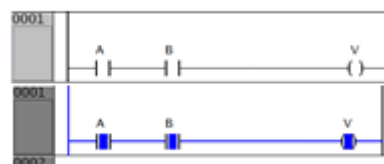
Obr. 22 – Schéma součtového členu (OR) s využitím relé.



Obr. 23 – Schéma součinnového členu (AND) s využitím relé.



Obr. 24 – Realizace logické funkce OR v prostředí CoDeSys (ladder diagram).



Obr. 25 - Realizace logické funkce AND v prostředí CoDeSys (ladder diagram).

Výhody:

- jasně definovaná posloupnost zápisu, kterou není možné porušit,
- rychlé programování logických operací (sčítání, časování signálů, ...).

Nevýhody:

- Nevhodný pro složité algoritmy s aritmetickými operacemi a práci s daty.

• Jazyk pro sekvenční programování (SFC):

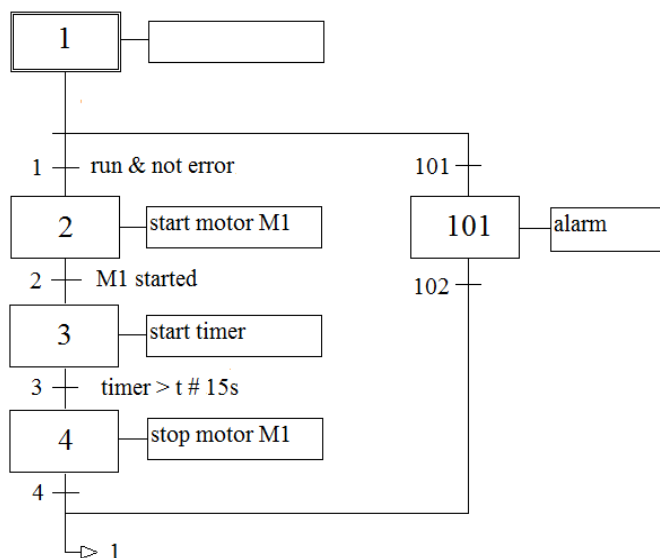
Sekvenční programování patří mezi grafické programovací jazyky, které je principiálně podobné vývojovému diagramu používanému ke grafickému znázornění jednotlivých kroků algoritmu, pracovního postupu či jiného obecného procesu. Algoritmus tvořený sekvenčním programováním je jasně definovaný a jeho průběh je závislý na splnění rozhodujících podmínek, čímž pokračuje postupné větvení programu a zpracovávání následujících rozhodujících podmínek. Tento způsob je vhodný pro definici posloupnosti volání jednotlivých podprogramů, vytvořených jiným programovacím jazykem (např. strukturovaný text), které jsou závislé na hodnotách stavových proměnných získaných z technologického procesu (např. pokud je aktivní alarm/error, který signalizuje poruchu v procesu (např. plná nádrž), nebude spuštěn motor (např. zajišťující běh čerpadla)). (KOVÁŘ, 2010)

Výhody:

- snadná definice a ošetření různých stavů programu,
- ideální pro tvorbu sekvenční logiky a tvorbu hlavní větve programu s voláním podprogramů.

Nevýhody:

- obtížné zpracování analogových či velkého počtu číslicových signálů,
- není vhodný pro složité algoritmy.



Obr. 26 – Příklad SFC programovacího jazyku.

• Jazyk funkčních bloků (FBD):

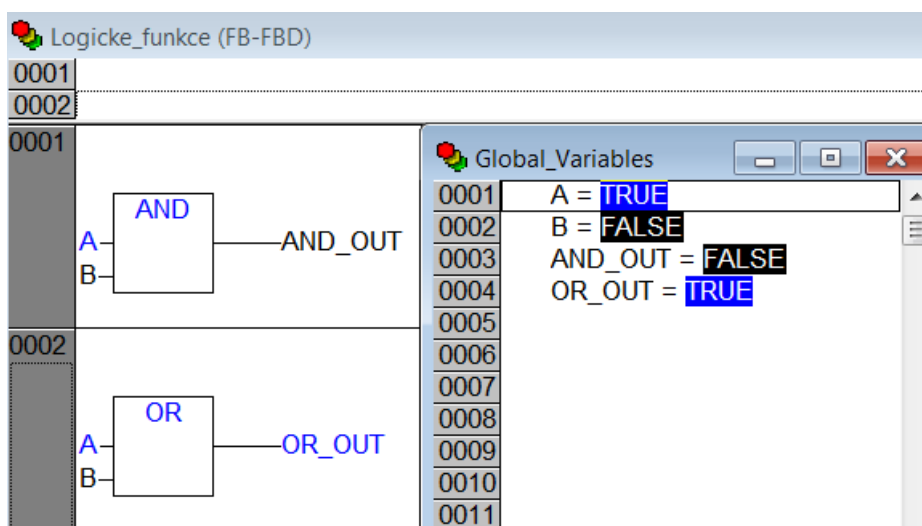
Programování s využitím tzv. funkčních bloků patří mezi grafické programovací jazyky, kde je posloupnost algoritmu definována návazným propojením bloků. Funkční blok je softwarový prvek, který s určitým souborem vstupních dat vytvoří jeden či více výstupů (výsledků) a disponuje interním ukládáním dat (např. PID, čítače, časovače, logické funkce atd.). Jazyk funkčních bloků se vytváří postupným propojením funkčních bloků a funkcí (např. $\sin()$, $\cos()$ atd.), které dohromady vytvářejí samostatný blok, který vykazuje specifické vlastnosti a jako takový může být vstupem do jiného funkčního bloku vykonávajícího jiný výpočetní úkon. (KOVÁŘ, 2010)

Výhody:

- přehledný zápis i pro složitější grafické programy (proti jazyku LD),
- snadné zpracování velkého počtu I/O signálů.

Nevýhody:

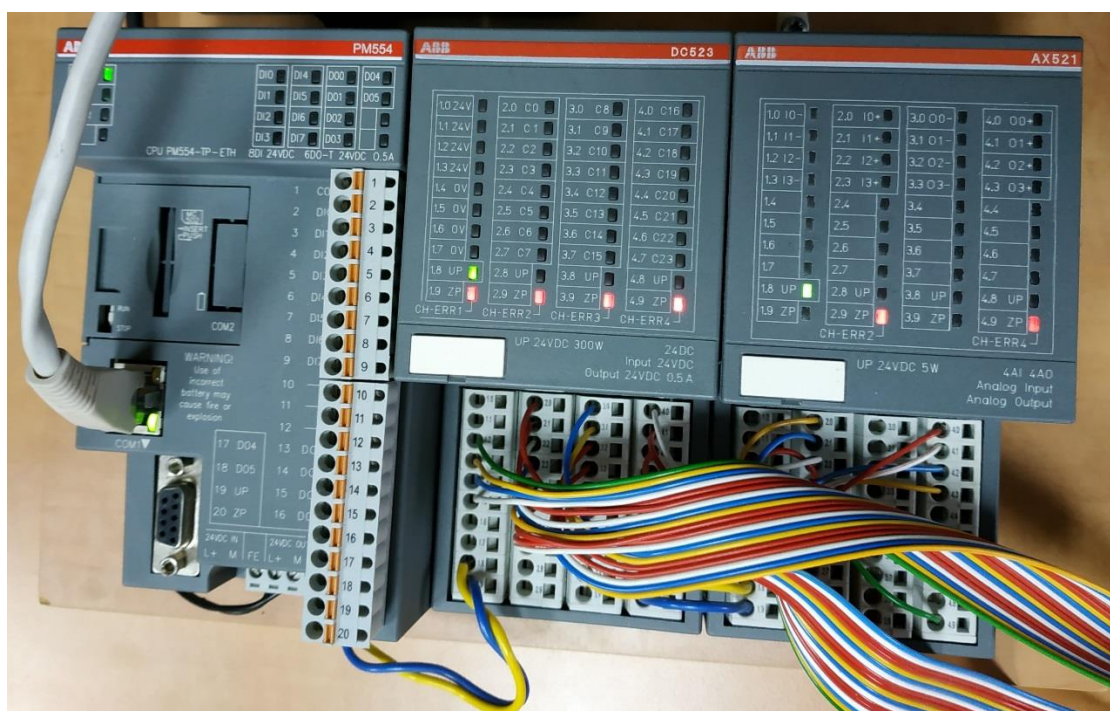
- nevhodný pro práci s velkým počtem analogových signálů,
- u složitých algoritmu se stává nepřehledným.



Obr. 27 – Příklad logické funkce OR/AND ve funkčním blokovém diagramu.

2.2 Logický automat AC500-eCo

V současné době využívá společnost ABB pro své automatizační systémy logický automat AC800, což je zatím poslední vyvinutá řada logických automatů firmy ABB. Ve vybavení laboratoře katedry automatického řízení je ovšem k dispozici starší řada s označením AC500, která je pro školní účely ve verzi eCo. Logický automat AC500 sice oproti AC800 obecně disponuje nižším výkonem a nemá přímou podporu nejnovějších komunikačních protokolů ovšem je to stále plnohodnotný a časem ověřený logický automat, který je navíc stále podporován ze strany výrobců (v roce 2019 vyšla nejnovější aktualizace verze 3.0).



Obr. 28 – Logický automat AC500-eCo.

Na obr. 28 je znázorněno PLC AC500-eCo použité v této práci. Jedná se o modulární PLC, kdy vlevo je modul procesorové jednotky PM554. Dále jsou k vidění dva I/O moduly – prostřední modul s označením DC523 slouží jako digitální karta vstupů/výstupů a vpravo je umístěn analogová vstupně/výstupní karta s označením AX521.

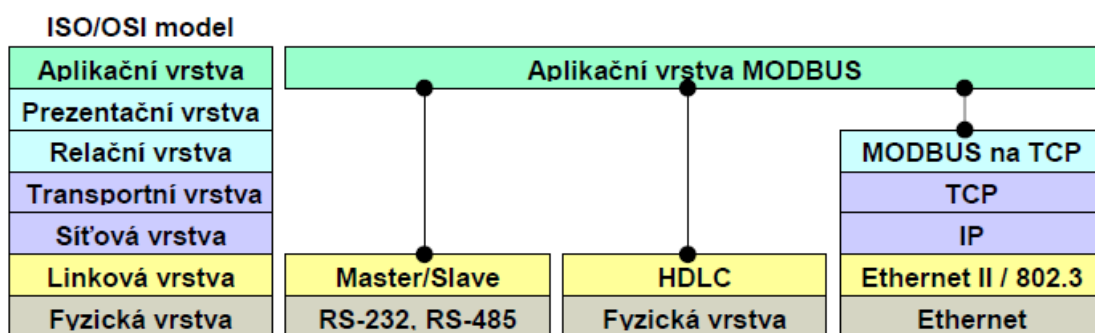
V této práci je využíván pouze modul procesorové jednotky PM554, který je vybaven jedním ethernetovým konektorem pro připojení k nadřazené úrovni řízení (PC) a dále jedním komunikačním portem COM1, který podporuje standard RS485. Komunikační port COM1 slouží pro připojení podřízených zařízení (Slave jednotky), popis pinů tohoto portu je znázorněn na obr. 12a.

3 Komunikační protokol MODBUS

Jelikož v této práci využívají oba dva frekvenční měniče komunikační protokol Modbus RTU, bez ohledu na použitou fyzickou vrstvu (RS232 či RS485), budu se v této kapitole věnovat bližšímu popisu komunikačního protokolu Modbus.

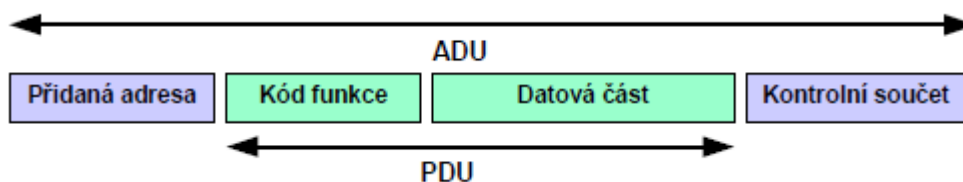
Definice komunikačního protokolu Modbus je dle (RONEŠOVÁ, 2005) stanovena takto:

„Modbus je komunikační protokol na úrovni aplikační vrstvy ISO/OSI modelu, umožňující komunikaci typu klient-server (master-slave) mezi zařízeními na různých typech sítí a sběrnic. Vytvořen v roce 1979 firmou MODICON. V současné době je podporována celá řada komunikačních médií např. sériové linky typu RS-232, RS-422 a RS-485, optické a rádiové sítě nebo síť Ethernet s využitím protokolu TCP/IP. Komunikace probíhá metodou požadavek-odpověď a požadovaná funkce je specifikována pomocí kódu funkce jež je součástí požadavku.“



Obr. 29 – Příklady implementace protokolu Modbus. (RONEŠOVÁ, 2005)

Modbus definuje strukturu odeslané zprávy na úrovni protokolu nezávisle na typu použité komunikační vrstvy – tzv. PDU (Protocol Data Unit) část zprávy. Zpráva PDU je dále rozšířena o další segmenty podle toho, na jakém typu sítě je protokol použit. Výsledkem je souhrnná zpráva PDU s dílčími segmenty na aplikační úrovni – tzv. ADU (Application Data Unit).



Obr. 30 – Základní tvar Modbus zprávy. (RONEŠOVÁ, 2005)

Kód funkce definuje serveru (slave) jaký druh operace má být vykonán – čtení, zápis, diagnostika. Rozsah kódu funkce je 1 až 255, přičemž kód funkce 128 až 255 je vyhrazen pro chybové odpovědi serveru (slave). **Datová část** je vyhrazena pro realizaci operace definované kódem funkce (např. čtení hodnoty uložené v uchovávacím registru). (RONEŠOVÁ, 2005)

• **Datový model:**

Datový model protokolu Modbus je rozdělen do čtyř základních částí – tabulek. Každá tabulka obsahuje 9999 hodnot. Dvě tabulky jsou určeny pro čtení i zápis a dvě jsou určeny pouze pro čtení. Počet dostupných registrů je 2^{16} (65536).

Tab. 9 – Rozdělení registrů protokolu Modbus

Číslo registru	Typ	Název	Zkratka
1-9999	Čtení/zápis	Cívky (diskrétní výstupy)	DO
10001-19999	Čtení	Diskrétní vstupy	DI
30001-39999	Čtení	Vstupní registry	AI
40001-49999	Čtení/zápis	Uchovávací registry	AO

Přístup k jednotlivým tabulkám je definovaný kódem funkce. Rozdělení a význam základních Modbus funkcí je znázorněn v tabulce pod textem.

Tab. 10 – Základní funkce protokolu Modbus

Kód funkce	Význam funkce		Typ hodnoty	Přístup
01 (0x01 _H)	Čtení DO	Čtení stavu cívky	Diskrétní	Čtení
02 (0x02 _H)	Čtení DI	Čtení stavu vstupů	Diskrétní	Čtení
03 (0x03 _H)	Čtení AO	Čtení holding registrů	16bitová	Čtení
04 (0x04 _H)	Čtení AI	Čtení vstupních registrů	16bitová	Čtení
05 (0x05 _H)	Zápis jednoho DO	Nastavení stavu jedné cívky	Diskrétní	Zápis
06 (0x06 _H)	Zápis jednoho AO	Nastavení jednoho registru	16bitová	Zápis
15 (0x0F _H)	Zápis více DO	Nastavení stavu více cívek	Diskrétní	Zápis
16 (0x10 _H)	Zápis více AO	Nastavení více registrů	16bitová	Zápis

Pzn.: (žlutě zvýrazněné funkce jsou funkce použité v této diplomové práci).

• **Kód funkce 03:**

Tato funkce slouží ke čtení obsahu souvislého bloku až 125 uchovávacích registrů. V požadavku klienta (master) je vždy definována adresa prvního registru a počet registrů určených ke čtení.

Adresa registru vždy začíná od 0 – např. pro práci s uchovávacími registry je tedy nutné vždy od cíleného registru odečíst hodnotu 40001 (začátek sady uchovávacích registrů).

Příklad: požadavek pro vyčtení sady 10 uchovávacích registrů, kdy první registr začíná na adrese 40101.

Kód funkce	Adresa registru	Počet registrů
03	100	10

Výsledkem bude přečtení registrů 40101 až 40110 a vrácení jejich hodnoty klientu.

• **Kód funkce 06:**

Tato funkce slouží pro zápis jednoho uchovávacího registru. V požadavku klienta (master) je vždy definována adresa registru, který se má zapsat a hodnota, která má být zapsána. Správnou odpovědí je v kopie požadavku poté, co proběhl zápis registru.

Adresa registru opět vždy začíná od 0 – např. pro zápis jednoho holding registru je nutné od tohoto registru odečíst 40001.

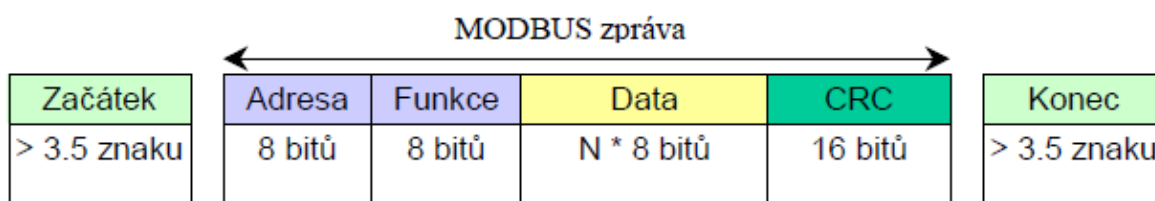
Příklad: požadavek pro zápis jednoho uchovávacího registru (např. řídicího slova do slave zařízení), přičemž adresa registru řídicího slova je 40001.

Kód funkce	Adresa registru	Počet registrů	Hodnota pro zápis
06	0	1	Typicky číslo v DEC tvaru.

Výsledkem bude zápis určité hodnoty do registru 40001, který je určen pro řídicí slovo daného zařízení. Po zápisu server (slave) jednotka vrátí klientu (master) kopii zapsané hodnoty.

• **Struktura ModbusRTU zprávy:**

Zpráva v rámci režimu Modbus RTU je složena z adresy Slave jednotky, použité Modbus funkce, datových bitů a CRC kontrolního součtu. V této práci je u obou frekvenčních měničů zvolen 16bitový profil čili data jsou přenášena v podobě 16bitového slova.



Obr. 31 – Skladba zprávy v režimu Modbus RTU. (RONEŠOVÁ, 2005)

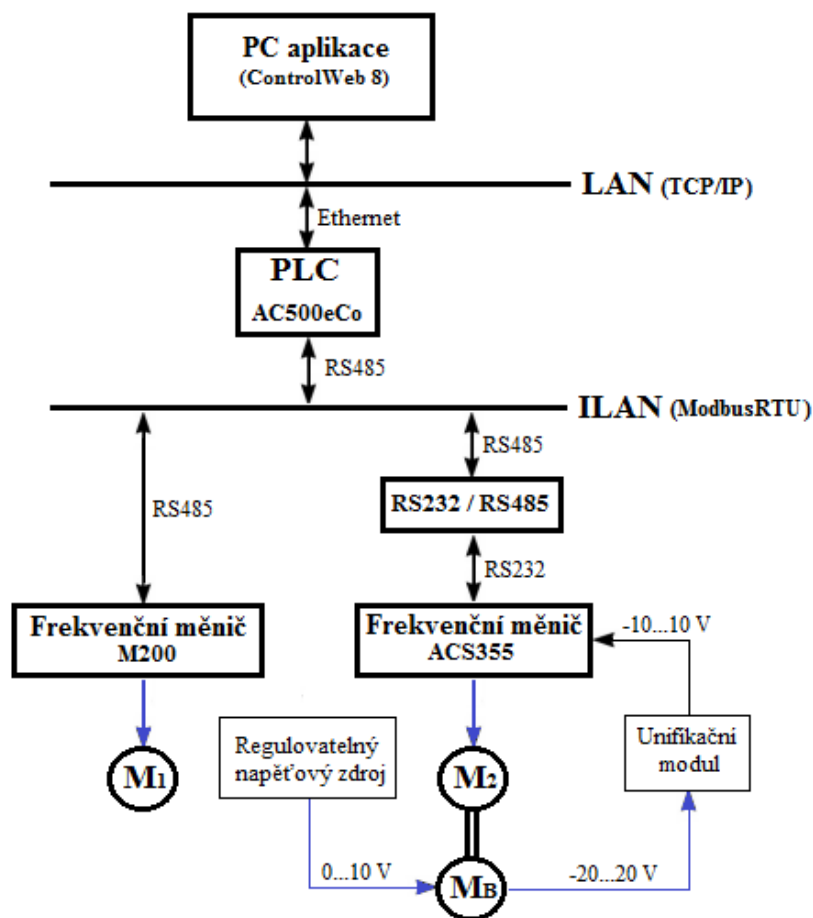
Formát bytu (11 bitů): každý byte v režimu Modbus RTU obsahuje:

- 1 start bit,
- 8 datových bitů,
- 1 parita bit,
- 1 stop bit.

Pokud není použita parita, je nahrazena druhým stop bitem.

4 Laboratorní úloha řízení elektromotorů

Na obr. 32 je znázorněno schéma SCADA/MMI úlohy pro řízení indukčních elektromotorů z uživatelské aplikace běžící na PC. Jako řídicí systém je využit logický automat společnosti ABB AC500 eCo, dostupný v laboratořích katedry automatického řízení. PLC se v reálném čase stará o sběr procesních dat, následné vyhodnocení a také zajišťuje interakci mezi uživatelem (resp. uživatelskou aplikací) a řízenými elektromotory. Komunikaci mezi nejvyšší úrovní řízení (PC) a procesní úrovní řízení (PLC) zajišťuje OPC komunikační protokol s použitím Ethernetu na fyzické úrovni.



Obr. 32 – Distribuovaný systém řízení laboratorní úlohy.

Logický automat je spolu s frekvenčními měniči připojen na společnou sériovou sběrnici na jejíž fyzické úrovni je použit standard RS485 a komunikační protokol Modbus RTU. Každý frekvenční měnič představuje samostatnou podúlohu, lišící způsobem řízení a hardwarovým vybavením.

4.1 Prvky laboratorní úlohy

První podúloha představuje úlohu ovládání. Sestává z frekvenčního měniče UniDrive M200 a k němu připojeného třífázového asynchronního elektromotoru, který není vybaven snímačem pro měření aktuálních otáček. Z nadřazené úrovně řízení je proto tento motor možno pouze ručně ovládat.



Obr. 33 – Frekvenční měnič M200 s připojeným elektromotorem.

- Třífázový asynchronní elektromotor FT4C52G:

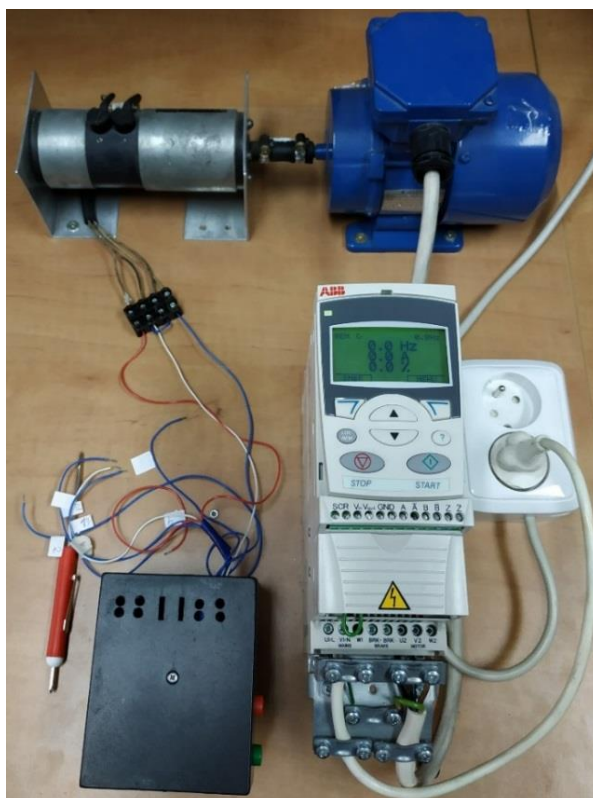


Jmenovité hodnoty:

Motor = 3~
U = 230 V
P = 500 W
f = 400 Hz
I = 3 A
n = 23700 ot/min

Obr. 34 – Třífázový asynchronní elektromotor FT4C52G.

Druhá podúloha představuje regulační úlohu, která sestává z frekvenčního měniče ACS355 s připojeným třífázovým asynchronním elektromotorem, který má navíc společnou hřídel s druhým, tzv. zátěžovým motorem. Tento zátěžový (brzdňý) motor je vybavený tachodynamem, které generuje stejnosměrné napětí přímo úměrné aktuálním otáčkám společné hřídele. Kromě realizace zpětné vazby slouží zátěžový motor pro simulaci zatížení, přičemž je zátěžový elektromotor ovládán z laboratorního napět'ového zdroje (0 až 10 V, max 2 A).



Obr. 35 – Frekvenční měnič ACS355 s připojeným elektromotorem.

- Zátěžový stejnosměrný elektromotor PIVT 6-25/3:



Obr. 36 – Zátěžový elektromotor model PIVT 6-25/3.

Jmenovité napětí motoru je 30 VDC a maximální proud 4 A. Jelikož motor disponuje velkým výkonem je pro účely simulace zatížení využito napětí 0 až 10 V a proudu max. 2 A.

- Asynchronní (též indukční) elektromotor Elektrim Sf56-2A:



Jmenovité hodnoty:

Motor = 3~

$U = 220 \text{ V}$

$f = 50 \text{ Hz}$

$P = 0,09 \text{ kW}$

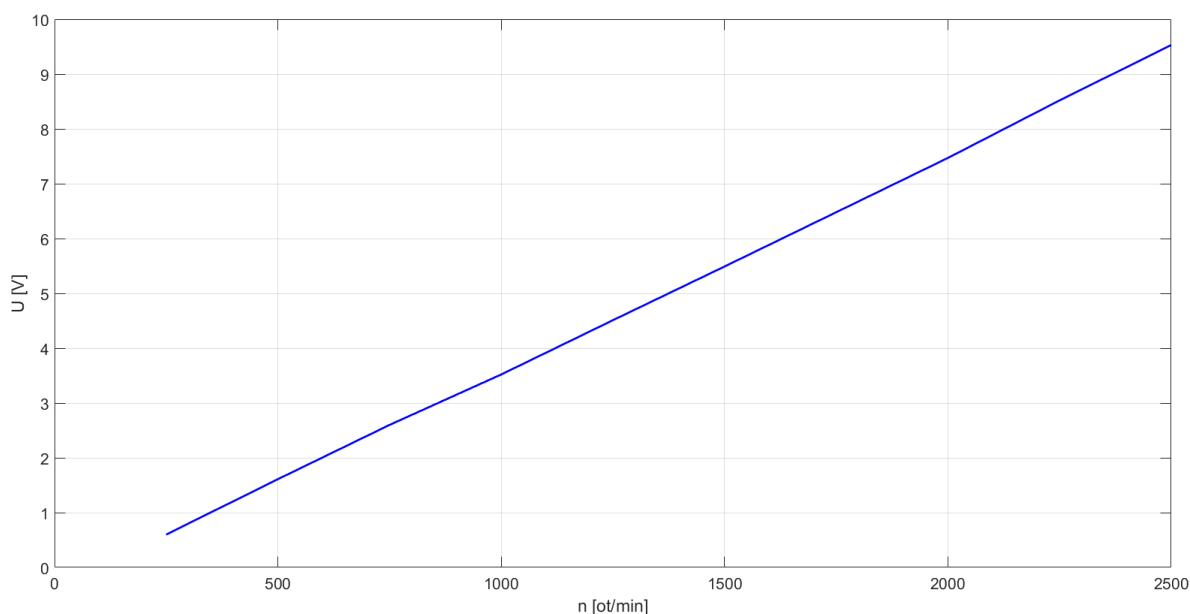
$I = 0,5/0,29 \text{ A}$

$n = 2820 \text{ ot/min}$

$\cos\varphi = 0,73$

Obr. 37 – Indukční elektromotor Elektrim Sf56-2A.

Jelikož pro snímání aktuálních otáček bylo použito tachodynamo, které na svém výstupu generuje stejnosměrné napětí přímo úměrné otáčkám hřídele, bylo nutné vytvořit statickou charakteristiku závislosti unifikovaného výstupního napětí na aktuálních otáčkách hřídele.



Obr. 38 – Závislost unifikovaného napětí tachodynamu na otáčkách motoru.

Ze statické charakteristiky, která vykazuje lineární závislost generovaného napětí na otáčkách hřídele elektromotoru byl zjištěn následující přepočtení vzorec, kde n je:

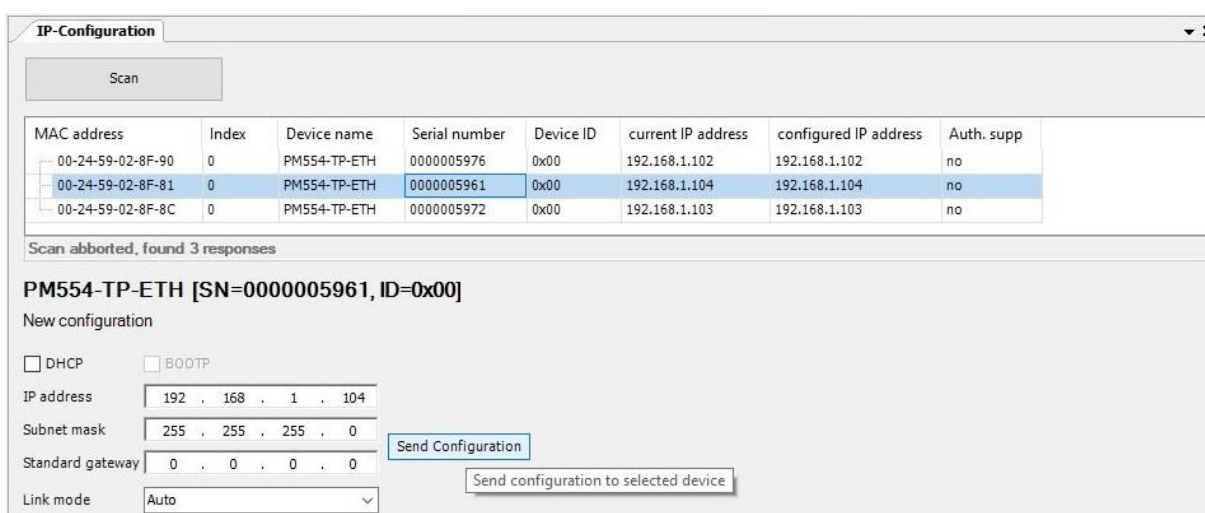
$$n = \frac{U + 0,3987}{0,004} \quad [\text{ot} \cdot \text{min}^{-1}] \quad (4.1)$$

Modul pro unifikaci signálu již byl součástí úlohy s ACS355, viz (HÝL, 2013).

4.2 Komunikační vazby

Komunikace počítače s programovatelným logickým automatem je vyřešena připojením, do již vytvořené sítě v laboratoři s použitím protokolu TCP/IP. Pro připojení počítače a PLC (fyzická vrstva ISO/OSI modelu) je použit ethernetový kabel. PLC má přidělenou jedinečnou IP adresu, pomocí které můžeme k danému PLC přistupovat z kteréhokoliv počítače připojeného do laboratorní sítě. IP adresa je použita ve formátu IPv4, prezentují ji tedy čtyři oddělené byty – tzv. oktety (1 okten = 8 bitů). Každý byte je oddělen tečkou, přičemž první dva byty jsou vždy stejné a zařízení se odlišují různou hodnotou třetího a čtvrtého bytu.

Příklad přiřazené IP adresy: 192.168.1.104.



Obr. 39 – Control Builder Plus – hledání připojených zařízení v laboratorní síti.

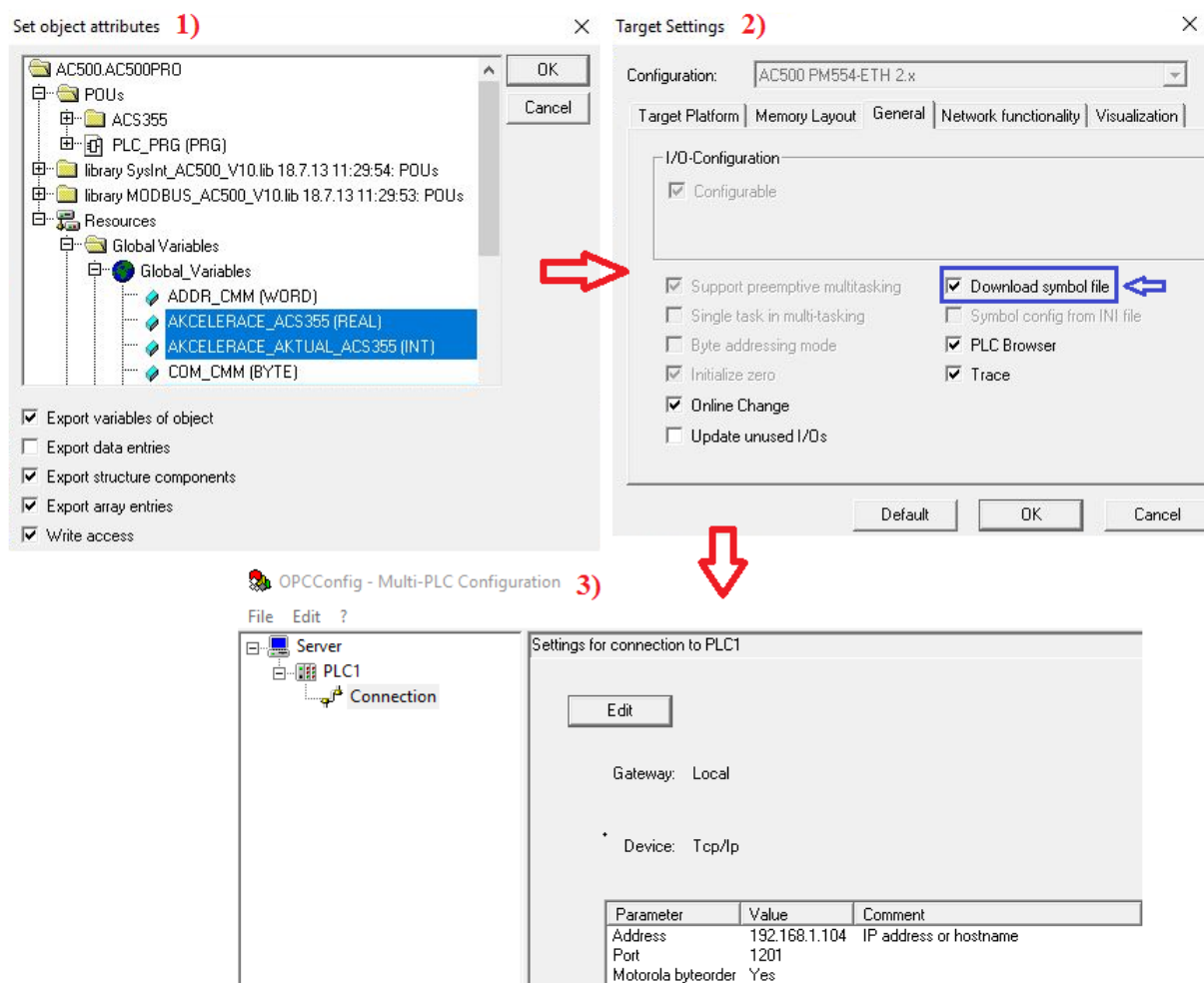
Záložka "IP-Configuration" programu Control Builder Plus slouží pro připojení se k logickému automatu. Počítač, ve kterém byl vytvořen řídicí program, který má být nahrán do PLC je připojen buď do společné internetové sítě spolu s PLC (laboratoř, firemní síť) nebo je fyzicky spojen s daným PLC přímo. V obou případech je pro připojení využito ethernetového kabelu. Po fyzickém spojení jsou identifikována všechna zařízení na lince a tlačítkem "Send Configuration" je programu Control Builder Plus přiřazeno dané PLC. Poté stačí pouze přejít do online režimu – pro správu stavu PLC v Control Builder Plus, pro nahrávání nového programu do PLC v programu CoDeSys.

4.2.1 Tvorba OPC protokolu

Pro úspěšnou komunikaci mezi uživatelskou aplikací vytvořenou v prostředí ControlWeb 8 a logickým automatem je nutné vytvořit OPC protokol jak na straně ControlWebu, tak na straně CoDeSysu, resp. řídicího algoritmu.

• CoDeSys OPC Server:

- nejprve je nutné vybrat globální proměnné určené pro výměnu dat,
- poté povolit stáhnutí konfiguračního souboru do paměti PLC,
- následně je nutné celý program nahrát do PLC s příslušnou IP adresou,
- nakonec vytvořit OPC server s využitím konfigurátoru *OPCConfig*.

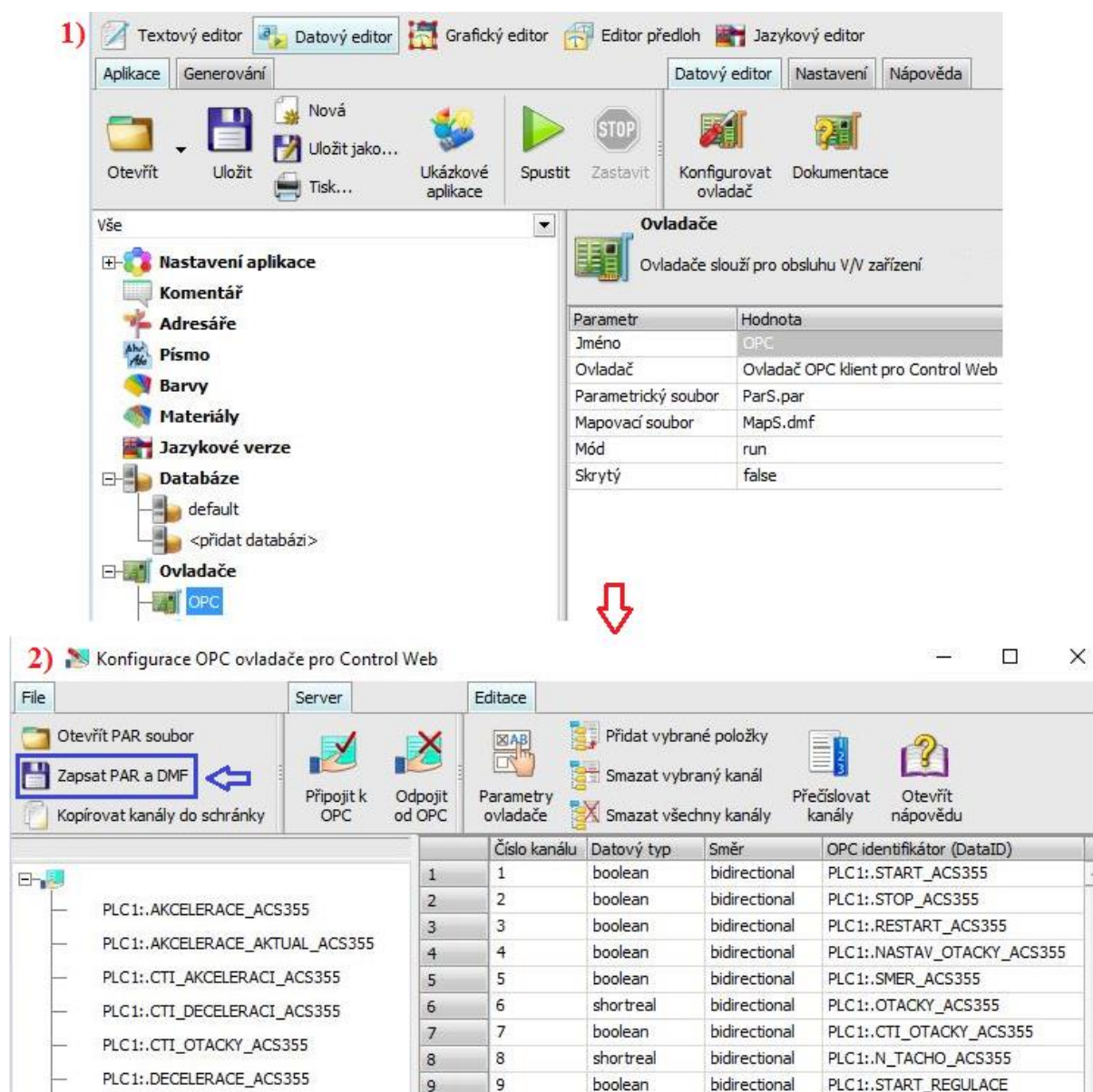


Obr. 40 – Příklad tvorby OPC Serveru na straně PLC.

V této práci byl využit *OPC Server for CoDeSys V2.0*.

• **ControlWeb OPC Server:**

- v prostředí ControlWeb 8, v záložce *Datový editor* vytvořit OPC ovladač,
- při tvorbě OPC ovladače použít "Ovladač OPC klient pro ControlWeb",
- s pomocí OPC konfigurátoru namapovat kanály pro všechny komunikační proměnné,
- v konfigurátoru vytvořit mapovací a parametrický soubor,
- vytvořené mapovací a parametrický soubor přiložit k použitému ovladači.



Obr. 41 – Příklad tvorby OPC Serveru na straně uživatelské aplikace.

V konfigurátoru OPC ovladače pro ControlWeb jsou v levé části obsaženy komunikační proměnné načtené z PLC. V pravé části se poté těmto proměnným přidává číslo kanálu pro použití v uživatelské aplikaci

4.2.2 Komunikace frekvenčních měničů s PLC

Jak bylo uvedeno frekvenční měniče jsou připojeny na společné sériové lince, přičemž je využito standardu EIA485 (dříve RS485). Frekvenční měnič UniDrive M200 má výstup sériové komunikace v podobě standardu EIA485, je tedy k PLC připojen přímo. U měniče ACS355 je ovšem využito připojení namísto ovládacího panelu, které využívá standard RS232, je proto nutné umístit do řídicího řetězce mezi PLC a ACS355 převodník fyzické úrovně RS232/EIA485. Logický automat AC500eCo podporuje standard EIA485 a pro tyto účely slouží COM1 představený 9pinovým D-Sub konektorem (samice/female). Zapojení konektoru je znázorněno na obr. 12a na straně 23.

Jelikož frekvenční měnič UniDrive M200 podporuje pouze komunikační protokol ModbusRTU, je tento protokol využit pro komunikaci frekvenčních měničů s PLC po sériové lince. Frekvenční měniče představují v řídicím řetězci "Slave" zařízení, každému je proto přiřazena jedinečná adresa a je nastavena společná přenosová rychlost. PLC představuje v řídicím řetězci "Master" zařízení a jeho výstup COM1 musí být vhodně nastaven.

Tab. 11 – Společné nastavení komunikačních parametrů frekvenčních měničů

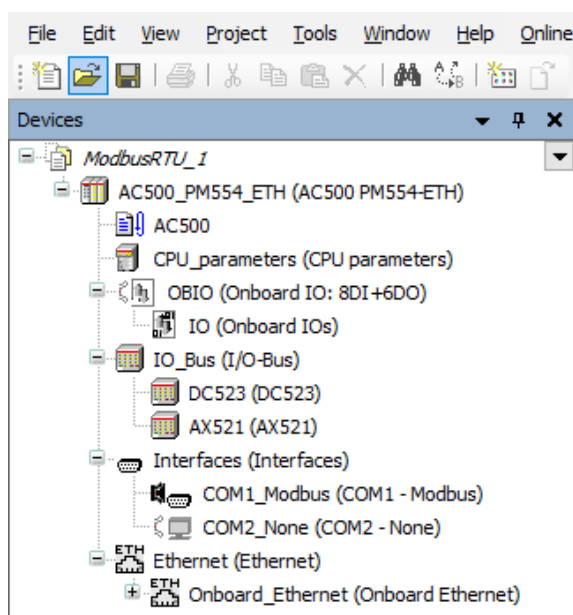
Parametr	Význam	Nastavení
Frekvenční měnič ACS355		
53.02	Sériová adresa	2
53.03	Přenosová rychlost sériové linky	115.2 [kbit/s]
53.04	Režim sériové linky	8NONE1
53.05	CTRL profil měniče	ABB MEN.PLN
Frekvenční měnič UniDrive M200		
11.023	Sériová adresa	5
11.024	Režim sériové linky	8.1NP
11.025	Přenosová rychlost sériové linky	115200 [bit/s]
11.026	Zpoždění linky před odesláním odpovědi	3 [ms]

Nastavení přenosové rychlosti a režim sériové linky musí být u všech zařízení na jedné sériové lince totožné. Přenosová rychlost byla zvolena 115200 bit/s a režim sériové linky nastaven na 8 datových bitů, 2 stop bity a není použit kontrolní bit (parita).

4.2.3 Konfigurace PLC pro sériovou komunikaci

Programovatelný logický automat AC500eCo představuje v řídicím řetězci Master zařízení a připojené frekvenční měniče Slave zařízení. V předešlé kapitole byly nastaveny parametry sériové komunikace obou měničů a zbývá nastavit PLC. K tomuto účelu slouží vývojové prostředí Control Builder Plus, v novějších verzích se jedná o nástroj Automation Builder. V této práci je využita starší verze softwaru ve verzi Control Builder Plus 2.3, jelikož použité PLC AC500eCo disponuje starším firmwarem, který již není v novějším Automation Builderu podporován.

Prvotní fázi nastavení komunikace v PLC představuje virtuální vytvoření použitého PLC v laboratorní úloze, je proto nutné zjistit typ procesorové jednotky a dalších připojených modulů. Na základě získaných informací je v prostředí Control Builder Plus vytvořen strom připojených zařízení pod procesorovou jednotkou. Při mapování jednotlivých modulů je nutné dodržet takové pořadí modulů, jaké odpovídá realitě.



Obr. 42 – Virtuální PLC v prostředí Control Builder Plus.

Po vytvoření PLC je nutné nastavit komunikační port COM1 (viz. obr. 43), který zprostředkovává komunikaci se slave zařízeními. Rozhraní COM1 bude nastaveno jako Modbus, jelikož frekvenční měniče podporují komunikační protokol ModbusRTU. Použitému PLC je přiřazena adresa 0, která značí, že se jedná o Master zařízení (novější terminologie Client).

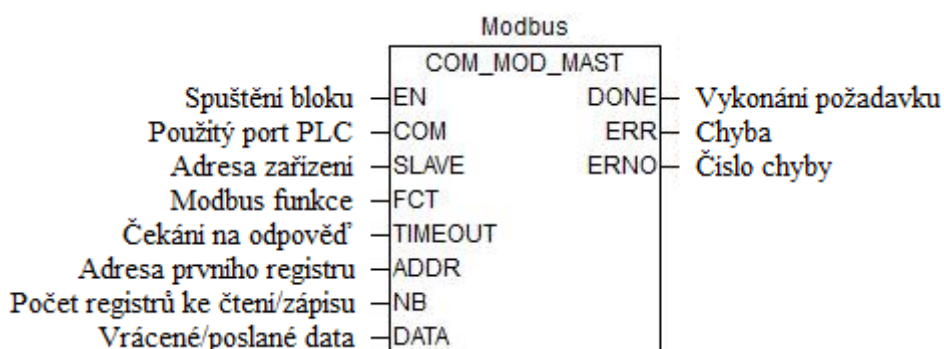
Parameter	Type	Value	Default Value	Unit	Description
Enable login	Enumeration of BYTE	Disabled	Disabled		Check for CoDeSys login
RTS control	Enumeration of BYTE	Telegram	None		RTS control must be set to 'telegram' for RS485 !
Telegram ending value	WORD(0..65535)	3	3		Set the telegram ending value in ms or characters
Baudrate	Enumeration of DWORD	19200	19200	bits/s	Set the baudrate in bits per seconds
Parity	Enumeration of BYTE	None	even		Set the parity bit type
Data bits	Enumeration of BYTE	8	8	bits/character	Set the character size
Stop bits	Enumeration of BYTE	1	1		Set the number of stop bits per character 2 means 1,5 when character
Run on config fault	Enumeration of BYTE	No	No		Start PLC program even on configuration fault
Operation mode	Enumeration of BYTE	Client	None		Set the operating mode
Address	BYTE(0..255)	0	0		Set the address of the device. (Note: Client requires address 0)

Obr. 43 – Nastavení Modbus komunikace rozhraní COM1.

4.3 Řídicí algoritmus PLC

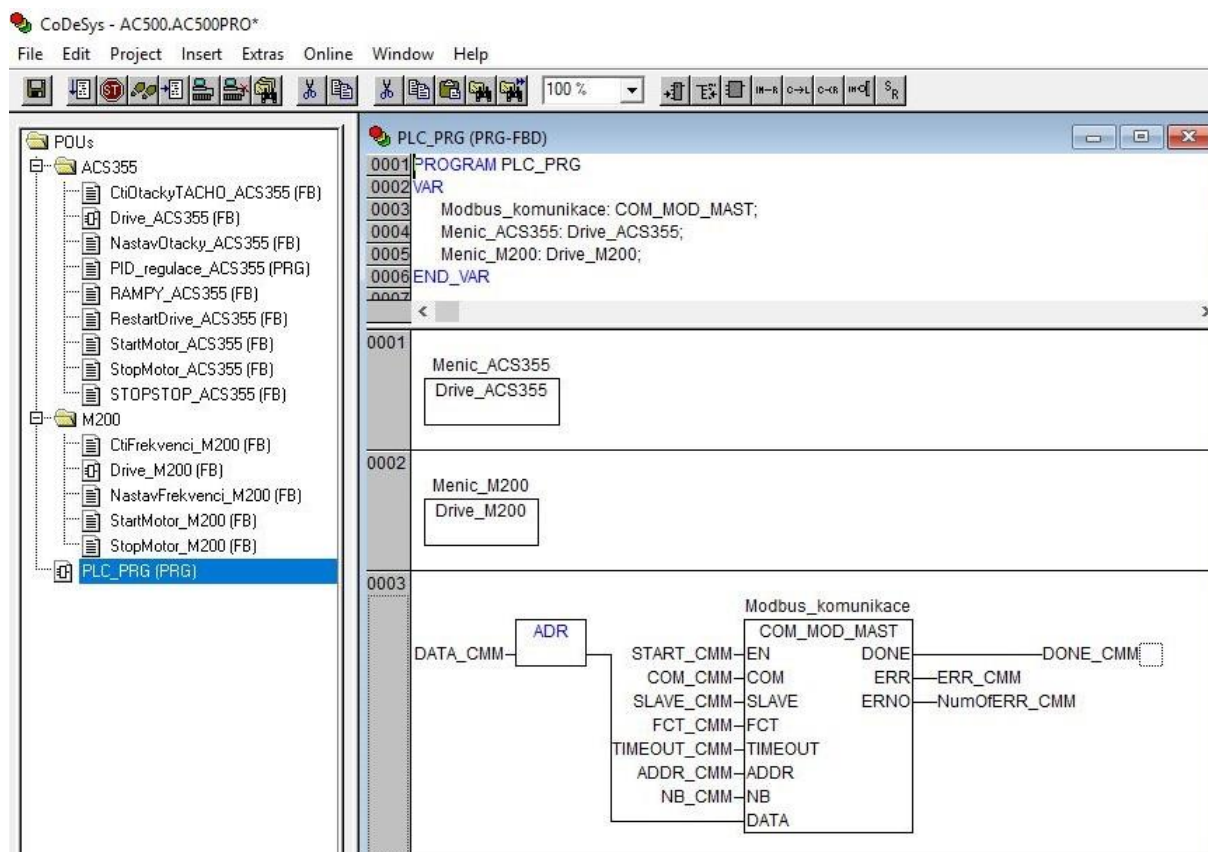
Po nastavení komunikačních vazeb je PLC připraveno pro tvorbu řídicího programu. K vývoji řídicího programu slouží nástroj CoDeSys, který je součástí balíčku Control Builder Plus. Pro správnou funkci musí být vždy vytvořena nejprve konfigurace použitého PLC v Control Builderu a poté vytvořen řídicí program v CoDeSysu, který je následně poslán do paměti PLC a dále vykonáván dle požadavků.

Řídicí program je založen na principu zapisování/čtení do/z odpovídajících registrů měničů podle aktuální potřeby. Každému parametru je přiřazena jedinečná adresa registru, jehož číslo odpovídá tabulce tzv. uchovávacích registrů – viz kapitola 3, tab. 9. Pro rozlišení vykonávané operace se používá tzv. kód funkce. Pro řízení elektrických pohonů postačuje využití funkce pro zápis a čtení registrů – viz tab. 10. Pro tyto účely je v knihovnách ABB dostupný funkční blok COM_MOD_MAST, který slouží pro obecné navázání Modbus komunikace.



Obr. 44 – Funkční blok COM_MOD_MAST.

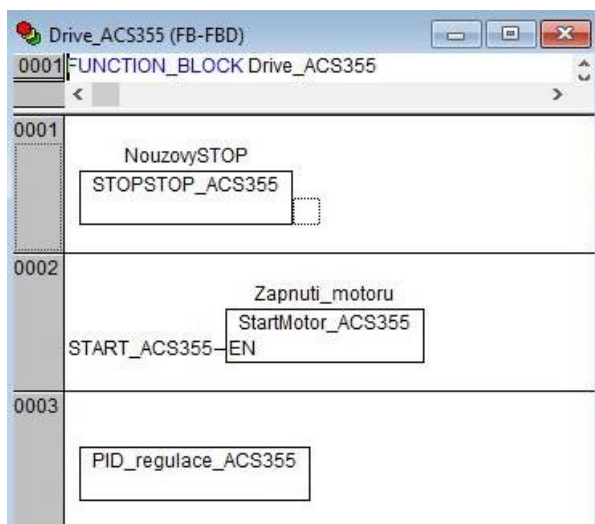
Hlavní program jsem se rozhodl vytvořit v grafickém jazyce FBD (funkční blokový diagram). K tomuto účelu jsem si vytvořil několik funkčních bloků, kdy každý jeden funkční blok vykonává určitou operaci (např. start motoru, čtení aktuálních otáček, PID regulace). Jednotlivé funkční bloky jsou psané v jazyce ST (strukturovaný text) a jsou volány v hlavním programu.



Obr. 45 – Programové prostředí CoDeSys s hlavním programem řídicího algoritmu.

Na obr. 45 je znázorněn vzhled vývojového prostředí CoDeSys určeného pro tvorbu algoritmů pro logické automaty AC500. V levé části je vidět skladba řídicího algoritmu, který obsahuje několik programů nebo funkčních bloků. Hlavní program "PLC_PRG" je vykonáván cyklicky s časovou periodou 10 ms (tzv. task).

Pravé okno je určeno pro tvorbu algoritmu. V 1. a 2. příčce hlavního programu jsou kvůli přehlednosti umístěny funkční bloky odpovídající jednotlivým frekvenčním měničům. V poslední příčce je volán funkční blok COM_MOD_MAST, který je součástí knihoven ABB a je určen pro obecnou Modbus komunikaci s podřízenými zařízeními.



Obr. 46 – Funkční blok pro ACS355.

Ve funkčním bloku "Drive_ACS355", který je volán v hlavním programu, jsou postupně volány další funkční bloky realizující určitou operaci. Na obr. 46 je znázorněn částečný obsah funkčního bloku "Drive_ACS355" – k vidění je volání bloku realizující nouzový stop, zapnutí motoru či program pro PLC regulaci.

```

StartMotor_ACS355 (FB-ST)
0001 FUNCTION_BLOCK StartMotor_ACS355
0002 VAR_INPUT
0003 EN: BOOL;
0004 END_VAR
0005 VAR_OUTPUT
0006 END_VAR
0007 VAR
0008 KROK: BYTE := 0;
0009 END_VAR
0010
0011 IF EN AND CTI_OTACKY_ACS355=FALSE THEN
0012   KROK := KROK+1;
0013
0014   CASE KROK OF
0015     1:
0016       START_CMM := FALSE;
0017       SLAVE_CMM := 2;
0018       FCT_CMM := 6;
0019       ADDR_CMM := 0;
0020       NB_CMM := 1;
0021       DATA_CMM := 1150; (* Rezim PROVOZ *)
0022     2:
0023       START_CMM := TRUE;
0024     3:
0025       START_CMM := FALSE;
0026       SLAVE_CMM := 2;
0027       FCT_CMM := 6;
0028       ADDR_CMM := 0;
0029       NB_CMM := 1;
0030       DATA_CMM := 1151; (* Rezim RUN *)
0031     4:
0032       START_CMM := TRUE;
0033       IF CTI_OTACKY_ACS355=FALSE THEN
0034         START_ACS355 := FALSE;
0035       END_IF
0036       KROK := 0;
0037     END_CASE
0038   END_IF
0039 END_IF

```

Obr. 47 – Funkční blok pro start motoru.

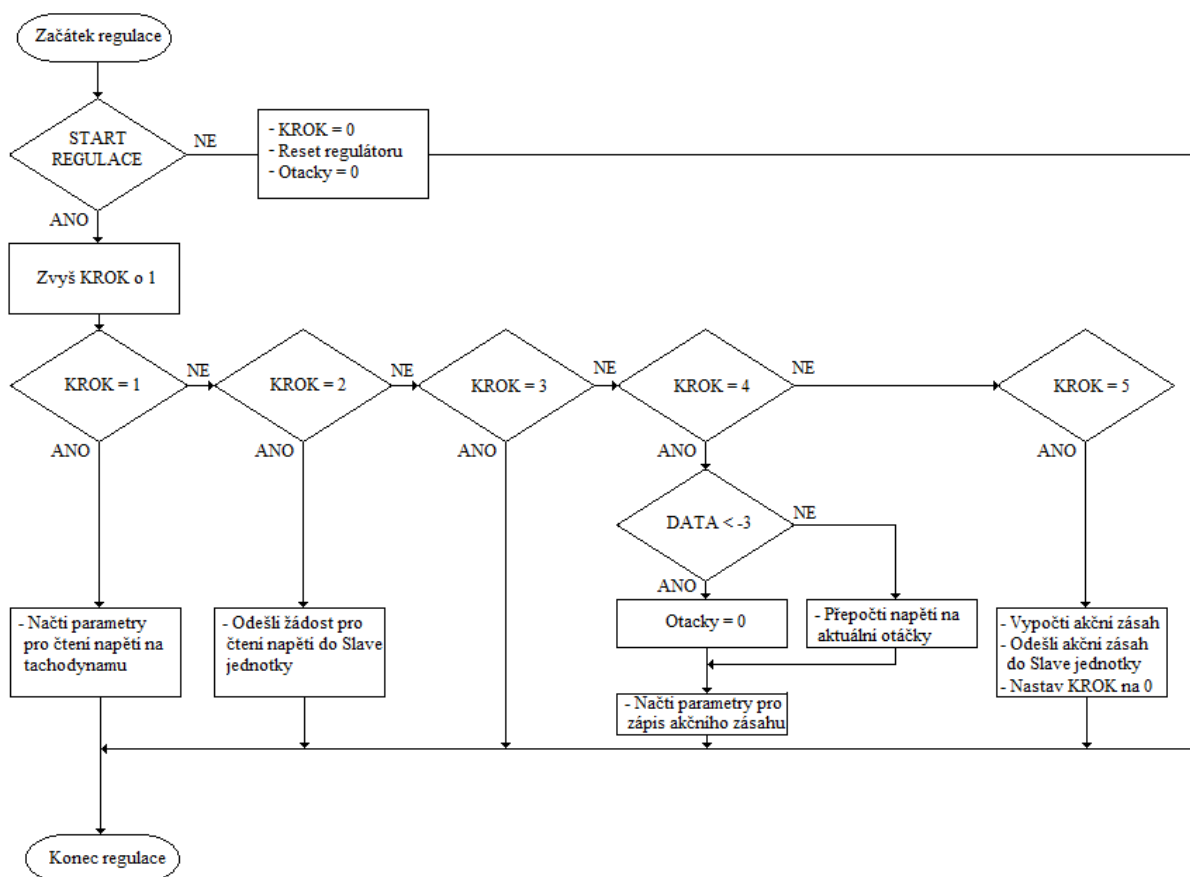
Ukázka funkčního bloku realizujícího start motoru. Frekvenční měnič přijímá povely pro základní operace (start, stop, reset, ...) v podobě řídicího slova zapisovaného do registru 40001.

Z bezpečnostních důvodů není možné připojený motor zapnout pouze jedním řídicím slovem, ale musí být nejprve uveden do provozu a až poté může být provoz spuštěn. Funkční blok "StartMotor_ACS355" nejprve odešle 16bitové řídicí slovo, které uvede měnič do provozu a poté odešle druhé řídicí slovo aktivující motor.

Hodnota řídicího slova je představena dekadickou hodnotou (1150 pro PROVOZ a 1151 pro RUN), která je přepočtena ze skladby 16bitového řídicího slova.

4.3.1 Program pro regulaci otáček na požadovanou hodnotu

Regulaci na požadovanou hodnotu otáček jsem se rozhodl realizovat na úrovni PLC. K tomuto účelu byl využit funkční blok "PID", který je dostupný v knihovně ABB. Snímání aktuálních otáček hřídele bylo realizováno tak, že je unifikovaný signál z tachodynamu přiveden na vstupní analogovou svorku s číslem registru 40121. Aktuální hodnota napětí na svorce AI frekvenčního měniče je cyklicky čtena přes Modbus a vrácena do PLC, kde je tato hodnota po přepočtu na otáčky namapována jako zpětná vazba PID regulátoru.



Obr. 48 – Vývojový diagram pro regulaci otáček na požadovanou hodnotu.

Komunikační standard RS485 použitý na fyzické úrovni sériové komunikace je v polovičním duplexu čili v jeden časový okamžik je možné buď číst z nebo zapisovat do Slave jednotky. Pro správnou regulaci otáček je nutné oddělit od sebe cyklus čtení otáček z frekvenčního měniče a cyklus zápisu akčního zásahu do frekvenčního měniče. Tento problém jsem vyřešil krokováním algoritmu PID regulace s využitím příkazu CASE.

Při průchodu tělem algoritmu (za předpokladu aktivní PID regulace), je vždy KROK zvýšen o 1. Podle aktuální hodnoty kroku je vykonána požadovaná procedura (čtení napětí generovaného tachodynamem, zápis akčního zásahu do frekvenčního měniče). Rychlost krokování je definována cyklem (task) chodu PLC, který jsem zvolil 10 ms.

```

PID_regulace_ACS355 (PRG-ST)
PROGRAM PID_regulace_ACS355
0001 IF START_REGULACE THEN
0002   KROK := KROK+1;
0003   CASE KROK OF
0004     1:
0005       START_CMM := FALSE;
0006       SLAVE_CMM := 2;
0007       FCT_CMM := 3; (* Modbus funkce WRITE ONE WORD *)
0008       ADDR_CMM := 120;
0009       NB_CMM := 1;
0010     2: (* odeslani pozadavku cteni rychlosti *)
0011       START_CMM := TRUE;
0012     4: (* Nacteni vracene hodnoty - prevod na otacky - nacteni komun. parametru *)
0013       IF DATA_CMM >= -3 THEN
0014         OTACKY_PID := 0;
0015       ELSE
0016         Tacho_pom := INT_TO_REAL(DATA_CMM); (* nacteni vracenyh dat *)
0017         TACHO_PID := Tacho_pom/100; (* vysledkem je 0-10 V *)
0018         OTACKY_PID := (TACHO_PID-0.3987)/(-0.004);
0019       END_IF
0020
0021       SLAVE_CMM := 2;
0022       FCT_CMM := 6;
0023       ADDR_CMM := 1;
0024       NB_CMM := 1;
0025       START_CMM := FALSE;
0026     5: (* Volani regulatoru *)
0027       Regulator (ACTUAL:=OTACKY_PID, SET_POINT:=W_PID, KP:=KP_PID, TN:=TN_PID, TV:=TV_PI
0028         Y_PID:=Regulator.Y;
0029         LIMITS_PID:=Regulator.LIMITS_ACTIVE;
0030         OVERFLOW_PID:=Regulator.OVERFLOW;
0031       Y_POM := REAL_TO_INT(Y_PID); (* urceni akcniho zasahu *)
0032       DATA_CMM := Y_POM;
0033       START_CMM := TRUE;
0034       RESET_PID := FALSE;
0035       KROK := 0;
0036     END_CASE
0037   ELSE
0038     KROK := 0;
0039     RESET_PID := TRUE;
0040     OTACKY_PID := 0;
0041   END_IF

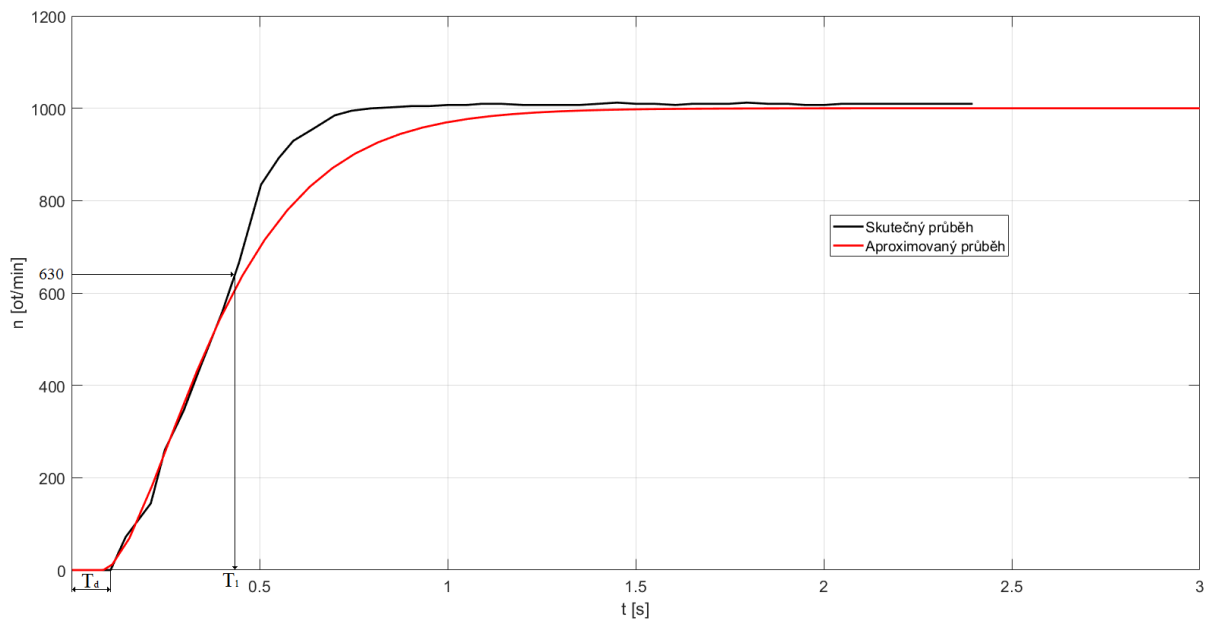
```

Obr. 49 – Program pro regulaci na požadovanou hodnotu otáček.

Krok s číslem 3 je vynechán pro to, aby mezi žádostí o čtení napětí na tachodynamu a přepočtem vrácené hodnoty na aktuální otáčky byl zajištěn dostatek času pro přenos dat po sériové lince. Změřená prodleva frekvenčního měniče mezi přijetím žádosti o čtení napětí a odesláním této hodnoty je přibližně 6 ms. K tomuto času je nutno připočíst ještě dobu přenosu dat po sériové lince, která je cca 1,15 ms.

4.3.2 Identifikace systému a syntéza regulátoru

Přechodovou charakteristiku jsem se rozhodl aproximovat proporcionální soustavou se setrvačností 1. řádu. Z naměřených dat jsem odečetl dopravní zpoždění cca $T_d = 100$ ms, které představuje dopravní zpoždění soustavy *PC – PLC – frekvenční měnič – motor*, a nikoliv samotného motoru, protože je ovlivněno přenosem informace po sériové lince, dobou zpracování požadavku logickým automatem a frekvenčním měničem a samotnou odezvou elektromotoru. Pracovní bod pro regulaci byl zvolen 1000 ot/min.



Obr. 50 – Aproximace naměřené přechodové charakteristiky.

- Aproximace proporcionální soustavou se setrvačností 1. řádu:

$$k_1 = \frac{y(\infty)}{u(\infty)} = \frac{1000}{1000} = 1 \qquad T_1 = 0,63 \cdot y(\infty) \cong 0,345 \text{ s}$$

$$G_s(s) = \frac{k_1}{T_1 s + 1} \cdot e^{-T_d s} = \frac{1}{0,345 s + 1} \cdot e^{-0,1 s}$$

- Syntéza analogového regulátoru metodou požadovaného modelu:

Požadavek – seřízení bez překmitu v nejkratším možném čase.

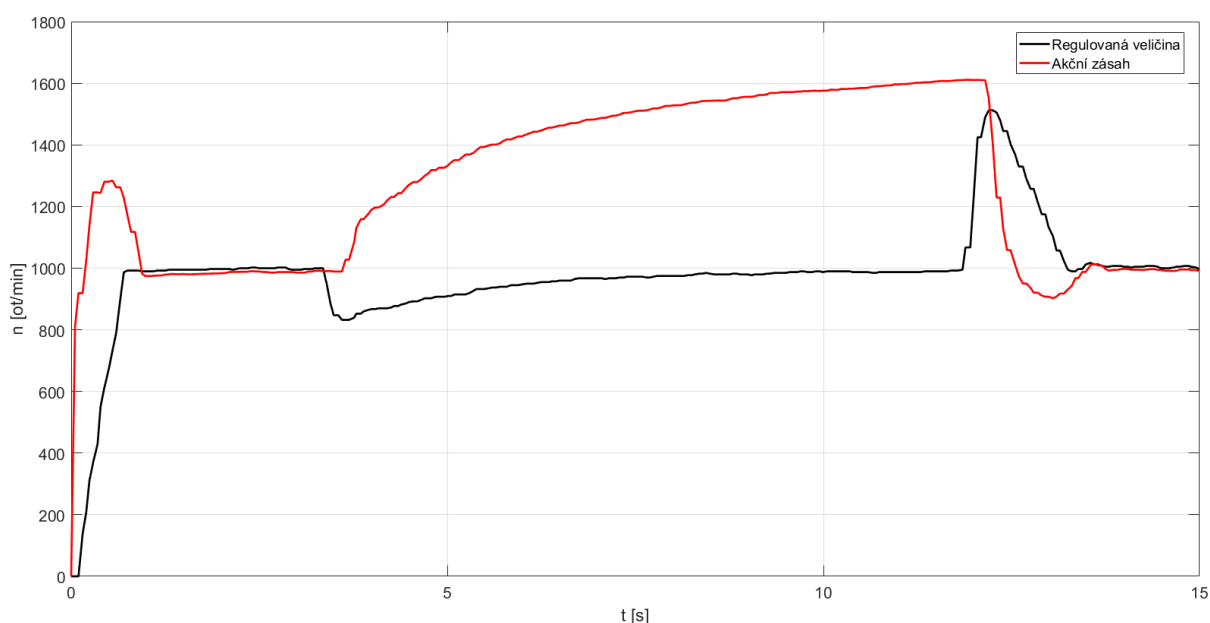
$$T_I^* = T_1 = 0,345 \qquad k_P^* = \frac{T_I^*}{k_1 \cdot \beta \cdot T_d} = \frac{0,345}{1 \cdot 2,718 \cdot 0,1} = 1,27$$

4.3.3 Regulace otáček na požadovanou hodnotu

Po návrhu regulátoru bylo nutné ověřit správnost výpočtu stavitelných parametrů. Při testování funkčnosti navrženého regulátoru se projevila nepřesnost aproximovaného přenosu vůči přechodové charakteristice soustavy, která zapříčinila zejména delší dobu regulace. Zvýšení hodnoty zesílení a integrační časové konstanty regulátoru způsobilo kratší dobu regulace, což byl pozitivní výsledek vzhledem k požadavkům na regulaci. Výsledné hodnoty stavitelných parametrů pro žádanou hodnotu $w = 1000$ ot/min jsou:

$$k_p = 5,$$

$$T_I = 0,45.$$



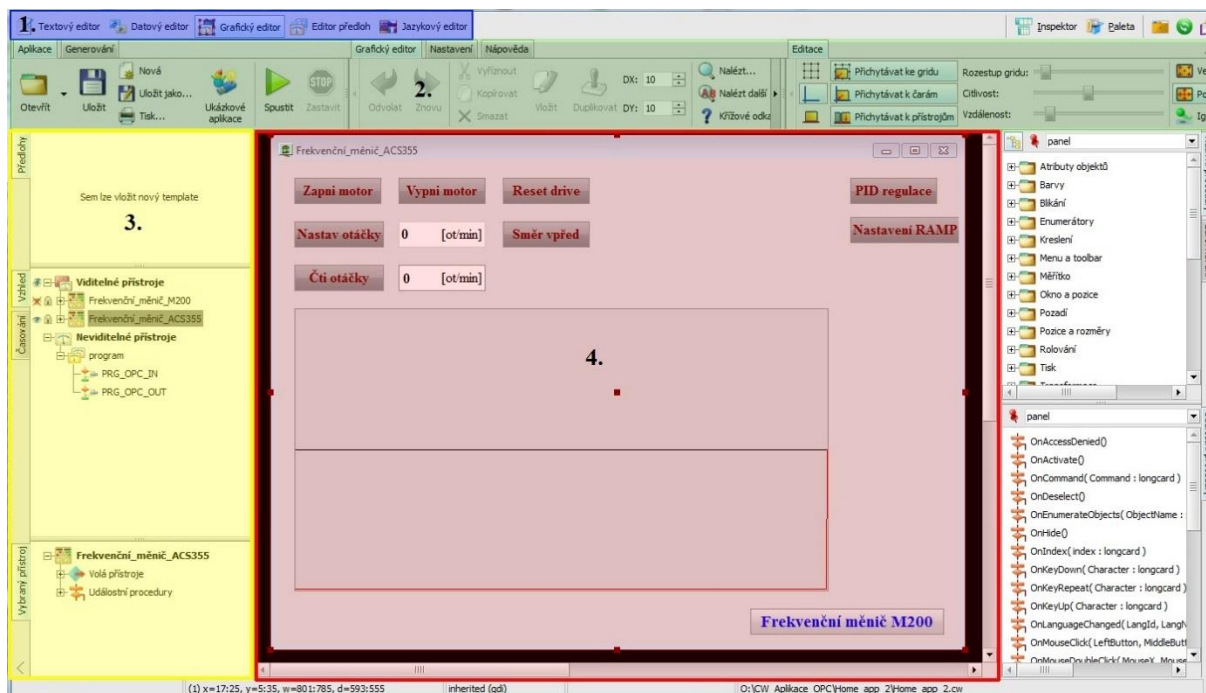
Obr. 51 – Regulace otáček na požadovanou hodnotu se simulovanou zátěží.

Na obr. 51 je znázorněna regulace otáček na požadovanou hodnotu 1000 ot/min. V průběhu regulace byl postupně aktivován a poté deaktivován zátěžový elektromotor, který je napájen z laboratorního napěťového zdroje. Pro zatížení je zátěžový elektromotor napájen napětím opačné polaritě v rozsahu 0 až 10 V s maximálním proudem 2 A. V případě aktivace zdroje s nulovým napětím dojde k pouhému zkratu výstupního statorového napětí zátěžového elektromotoru, postupným zvyšováním napětí opačné polaritě dojde k nárůstu brzdné síly.

Z průběhu regulace je možné odečíst časový interval mezi snížením/zvýšením otáček aktivací/deaktivací zátěžového motoru a zvýšením/snížením akčního zásahu regulátoru. Reakční doba regulátoru byla v případě aktivace a deaktivace brzdy přibližně 250 ms.

4.4 Uživatelská aplikace v prostředí Control Web 8

Uživatelská aplikace umožňuje současné řízení obou připojených motorů, resp. frekvenčních měničů. Tvorbou aplikace v prostředí Control Web 8 je intuitivní a velkou výhodou je česká a velmi podrobná nápověda. Na druhou stranu Control Web způsobuje problémy při časování dynamických vazeb mezi statickými členy aplikace. I malé množství časovaných položek může způsobit pád aplikace, často z nejasných důvodů. Po optimalizaci časování ovšem aplikace v prostředí Control Web 8 pracuje velmi dobře.



Obr. 52 – Popis prostředí aplikace Control Web 8.

1. Editor.
2. Panel nástrojů.
3. Strom použitých prvků.
4. Prostor pro tvorbu vizualizace.

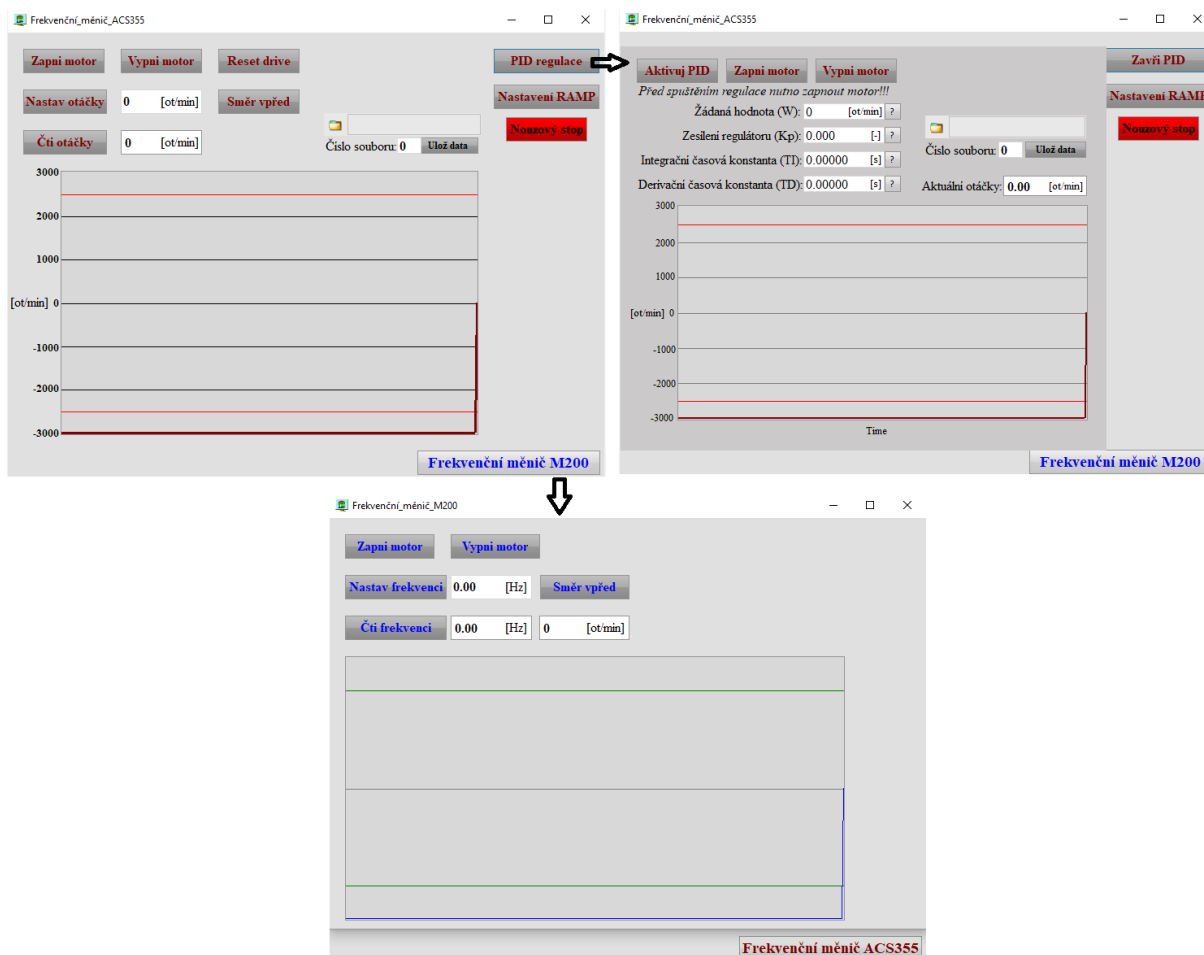
• **Editor:** slouží pro tvorbu grafické, datové či textové části vizualizace. Jednotlivé editory jsou mezi sebou propojené. Grafický editor slouží pro tvorbu vizuální stránky aplikace, datový editor slouží pro tvorbu proměnných a textový editor sdružuje vazby mezi prvky a proměnnými.

• **Panel nástrojů:** je určen pro práci s vytvářenou aplikací (ukládání, spouštění).

• **Strom použitých prvků:** sdružuje všechny prvky aplikace. Je rozdělen na část viditelných přístrojů (tlačítka, grafy, panely) a neviditelných prvků (programy, časovače).

• **Prostor pro tvorbu vizualizace:** je určen pro vlastní návrh aplikace a dynamických vazeb.

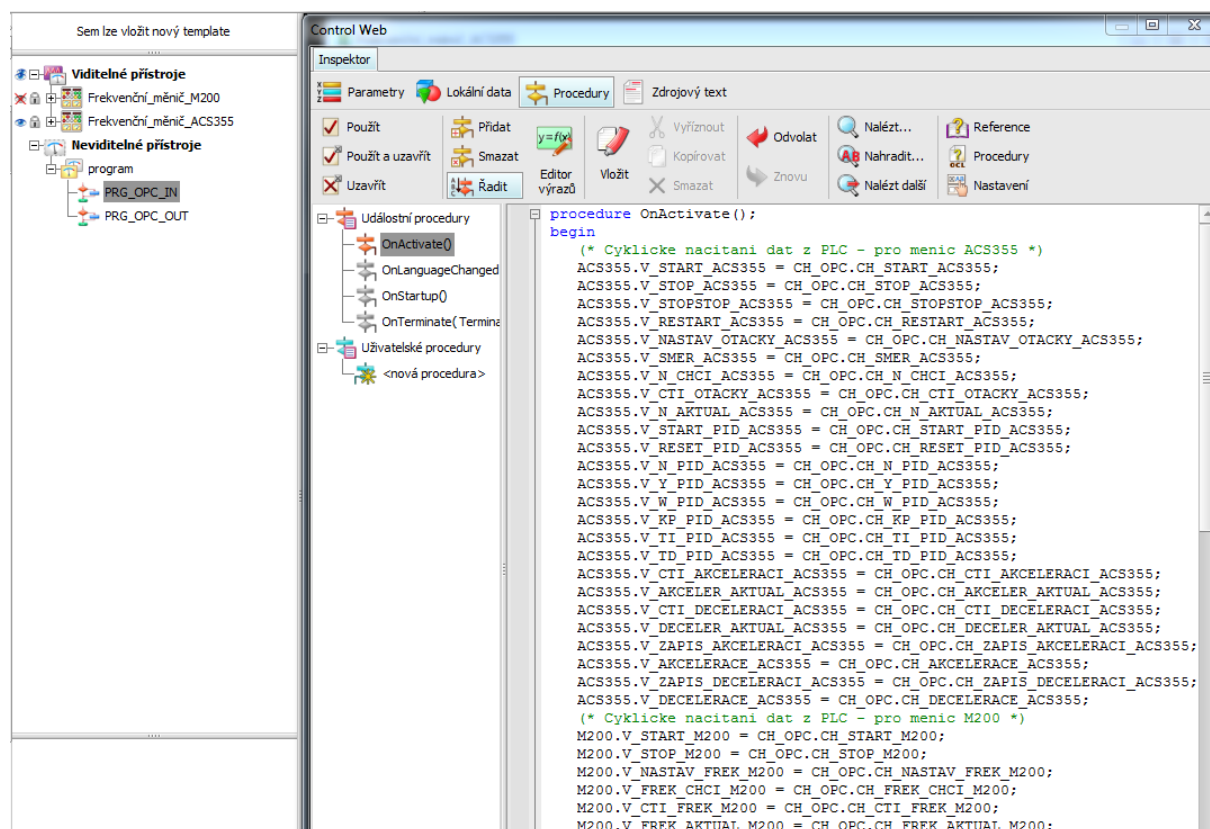
Z uživatelské aplikace jsou zadávány pokyny pro ovládání jednotlivých motorů (start, stop, nastavení otáček motoru) či v případě regulační úlohy pro zápis požadované hodnoty a parametrů regulátoru. Jelikož z důvodu nedostatečné paměti není možné ukládání dat na úrovni PLC, je aplikace využita také pro ukládání měřených hodnot. K tomuto účelu je využit přístroj *Archiver*, obsažený v paletě nástrojů, který umožňuje archivaci měřených proměnných v podobě tabulky hodnot ve formátu *DBF*.



Obr. 53 – Uživatelská aplikace.

Aplikace obsahuje několik oken, které jsou v zásadě rozděleny do dvou částí. Při prvním spuštění je aktivována aplikace s výchozím panelem, který slouží pro ovládání úlohy s frekvenčním měničem ACS355 (obr. 53 – nahoře vlevo). Z výchozího okna je možno například aktivovat panel pro regulaci otáček u měniče ACS355 (obr. 53 – nahoře vpravo) nebo se přepnout do okna určeného pro ovládání úlohy s frekvenčním měničem M200 (obr. 53 – dole), jehož panel je rozložením totožný s výchozím panelem ovšem obsahuje pouze základní funkce potřebné pro ovládání připojeného motoru.

Pro účely komunikace byly vytvořeny dva programy PLC_PRG_IN a PLC_PRG_OUT. První program PLC_PRG_IN zajišťuje příjem dat z logického automatu AC500 do aplikace, přičemž musí být pro zajištění cyklického načítání dat časován s pevně danou časovou periodou. Druhý program PLC_PRG_OUT slouží pro odesílání dat do logického automatu, přičemž k aktivace programu nedochází cyklicky, ale je spuštěn v okamžik aktivace některého z přístrojů (stisknutí tlačítka, změna požadované hodnoty).



Obr. 54 – Ukázka programu pro cyklické načítání procesních dat do aplikace.

V kapitole 4.2.1 bylo ukázáno, jak vytvořit OPC komunikační protokol. Kromě nastavení parametrů OPC bylo nutné definovat proměnné určené k výměně dat mezi aplikací a logickým automatem. Každé proměnné OPC protokolu byl přiřazen jedinečný kanál s číselným označením, přes který je možný přístup k hodnotám jednotlivých proměnných v aplikaci, viz obr. 41. Tyto kanálové proměnné ovšem není vhodné připojit přímo k jednotlivým přístrojům aplikace (tlačítka, grafy), jelikož přímá vazba OPC kanálových proměnných může zapříčinit nečekaný pád Control Webu. Doporučeným postupem je vytvoření pomocných proměnných v aplikaci, které svým datovým typem odpovídají jednotlivým kanálům OPC. Tyto pomocné proměnné již mohou být namapovány k jednotlivým přístrojům aplikace. Na obr. 54 je znázorněn program PLC_PRG_IN aktualizující pomocné proměnné (označení V_ (Variable)) na základě hodnot kanálových proměnných (označení CH_ (Channel)).

5 Závěr

Výsledkem této práce je vytvoření distribuovaného systému řízení dvou asynchronních elektromotorů ovládaných z frekvenčních měničů připojených na společnou sériovou sběrnici s možnostmi konfigurace z uživatelské aplikace vytvořené v prostředí Control Web 8. Úloha řízení je rozdělena do dvou samostatných úloh, přičemž úloha s měničem M200 představuje úlohu ovládání, protože není k dispozici snímač pro měření otáček. Úloha s měničem ACS355 představuje regulační úlohu, jelikož má řízený motor společnou hřídel s tzv. zátěžovým elektromotorem určeným pro simulaci zatížení. Brzdňý motor je navíc vybaven tachodynamem pro snímání aktuálních otáček. Interakci mezi uživatelskou aplikací a frekvenčními měniči zprostředkovává programovatelný logický automat AC500 eCo, které se stará o sběr procesních dat, řízení komunikace a v případě regulační úlohy plní funkci regulátoru.

Prvním krokem bylo seznámení se s frekvenčními měniči použitými v řídicí úloze. Oba měniče disponují třemi možnostmi ovládání – manuální ovládání přímo z ovládací panelu, připojení měniče k počítači s využitím odpovídající aplikace daného výrobce a poslední možností bylo připojení měniče k nadřazené úrovni řízení (PLC), což je způsob použitý v této práci. Po otestování manuálního ovládání a připojení měničů k PC přišlo na řadu seznámení se s vývojovým prostředím CoDeSys pro logický automat AC500, kde bylo vytvořeno několik úvodních projektů a následně otestována jejich funkčnost. Po seznámení se s jednotlivými prvky použitými v laboratorní úloze bylo nutné definovat parametry komunikace – společným komunikačním protokolem byl zvolen Modbus RTU. Pro řízení Modbus komunikace na úrovni PLC slouží funkční blok COM_MOD_MAST, s jehož použitím jsem do té doby neměl zkušenosti, a proto jsem se rozhodl první komunikaci otestovat mezi dvěma logickými automaty. Jedno PLC jsem nastavil jako Master zařízení a druhé jako Slave jednotku, přičemž jsem nejprve použil Modbus funkci 3 pro čtení ze Slave jednotky a poté Modbus funkci 6 pro zápis do Slave jednotky.

Po fyzickém propojení frekvenčních měničů a PLC následoval postupný návrh řídicího algoritmu. Hlavní program (main) jsem vytvořil v jazyce funkčních blokových diagramů (FBD) a jsou v něm postupně volány jednotlivé funkční bloky realizující jednu operaci (například zapnutí motoru, nastavení otáček atd). Všechny vytvořené funkční bloky jsou napsány v jazyce strukturovaného textu (ST), jelikož FBD je z důvodu přehlednosti vhodný spíše pro tvorbu hlavního programu. Samotné ovládání měničů je založeno na zápisu do nebo čtení z odpovídajících registrů, čemuž odpovídají vytvořené funkční bloky, které se liší

v nastavených hodnotách komunikačního bloku COM_MOD_MAST (Slave ID, Modbus funkce čtení nebo zápis, číslo registru atd).

Součástí regulační úlohy s frekvenčním měničem ACS355 byla možnost simulace zatížení pomocí brzděného motoru, který je pro vytvoření zátěžného momentu buďto zkratován (0 V) nebo pro zvýšení zatížení napájen napětím s opačnou polaritou (1 až 10 V). Pro ovládání zátěžového elektromotoru je použit laboratorní regulovatelný zdroj, jelikož je využíván pro napájení komunikačního převodníku RS485/RS232 použitého u regulační úlohy s měničem ACS355. Laboratorní zdroj poskytuje napětí 0 až 30 V a proud 0 až 4 A. Jmenovité napětí zátěžového motoru je 30 V a maximální dovolený proud jsou 4 A, nehrozí proto nechtěné zničení zátěžového elektromotoru neuváženě velkým proudem či napětím.

Posledním krokem byla tvorba uživatelské aplikace v prostředí ControlWeb 8 umožňující pohodlné ovládání obou připojených motorů z PC. Pro komunikaci mezi PLC a PC s uživatelskou aplikací byl zvolen OPC protokol, jelikož PLC je standartně připojené do laboratorní sítě stejně jako PC a stačilo tak pouze definovat ovladače pro OPC komunikaci na straně PLC a na straně ControlWebu.

Zlepšením stávajícího řešení by mohla být realizace PID regulace ve frekvenčním měniči namísto regulace na úrovni PLC. Použitý komunikační protokol Modbus RTU umožňuje maximální přenosovou rychlost 115200 bit/s. Jelikož je Modbus koncipován tak, že Slave jednotka nesmí nikdy vysílat data bez pověření Master zařízením, je výsledná přenosová rychlost snížena na polovinu, navíc je k ní nutno připočítat dobu nutnou pro zpracování požadavku frekvenčním měničem. Stávající řešení regulace spočívá v cyklickém opakování těchto kroků – čtení aktuálního napětí generovaného tachodynamem – převedení tohoto napětí na aktuální otáčky – výpočet akčního zásahu vzhledem k aktuální regulační odchylce – odeslání vypočteného akčního zásahu do měniče ACS355. Regulační pochod je spolu s omezením maximální přenosové rychlosti časově náročný, a použití frekvenčního měniče jakožto regulátoru by značně zjednodušilo, a především zrychlilo celý proces regulace. Z logického automatu by byla zadávána pouze požadovaná hodnota otáček a cyklicky čteny aktuální otáčky, případně hodnota regulační odchylky, akčního zásahu atd.

Tato diplomová práce vznikla za podpory projektu Aplikační výzkum v oblasti řízení strojů a procesů – SP2019/51 financovaného Ministerstvem školství, mládeže a tělovýchovy České republiky.

Seznam použité literatury

- ABB. *User's manual: Modbus Adapter Module FMBA-01* [online]. B. m., 2005 [cit. 2018-11-08]. Dostupné z: https://library.e.abb.com/public/f7a43c83974ccdf5c125708800349535/EN_FMB_A_UN_REV_A.pdf
- ABB. *User's manual: Startup and maintenance DriveWindow Light* [online]. B. m., 2014 [cit. 2019-5-14]. Dostupné z: <https://new.abb.com/drives/software-tools/drivewindow-light>
- ABB. *Uživatelská příručka: Frekvenční měniče ACS355* [online]. B. m.; 2010 [cit. 2018-09-22]. Dostupné z: <https://search-ext.abb.com/library/Download.aspx?DocumentID=3AUA0000066143CZ&LanguageCode=cs&DocumentPartId=&Action=Launch>
- BALÁTĚ, Jaroslav. *Automatické řízení*. Praha: BEN – technická literatura, 2003. ISBN 80-730-0020-2.
- CONTROL TECHNIQUES BRNO. *Uživatelská příručka: Unidrive M200/M201* [online]. B. m., 2002 [cit. 2018-10-03]. Dostupné z: https://www.pohony-menice.cz/Data/ke-stazeni/M200_Uzivatelaska_prirucka_1-6_3N.pdf
- HÝL, Radim. *Využití PLC pro řízení laboratorního modelu s možností vizualizace*. Ostrava, 2013. Diplomová práce. VŠB – Technická univerzita Ostrava, Fakulta strojní.
- KOVÁŘ, Josef, Zuzana PROKOPOVÁ a Ladislav ŠMEJKAL. *Programování dle normy IEC 61131* [online]. Kolín, 2010 [cit. 2019-2-10]. Dostupné také z: https://www.spszl.cz/soubory/plc/programovani_dle_normy_iec61131.pdf
- KRTIČKA, Pavel. *Třířákový asynchronní motor FT4C* [online]. Náchod, 2010 [cit. 2018-12-21]. Dostupné z: <https://www.atas.cz/files/FT4C.pdf>
- LOS, Jaroslav. *Konfigurace monitorování testovacího stavu vysokootáčkového rotačního stroje*. Ostrava, 2009. Diplomová práce. Vysoká škola Báňská – Technická univerzita Ostrava.
- NEBORÁK, Ivo. *Skalární řízení asynchronních motorů* [online]. B. m., 2011 [cit. 2019-4-9]. Dostupné z: http://homen.vsb.cz/~hav278/Elektricke_regulovane_pohony_I/Studijni_opory/Skalarni_rizeni_AM.pdf.
- NOSKIEVIČ, P. *Modelování a identifikace systémů*. 1. vyd. Ostrava: Montex, 1999. 276 s. ISBN 80-7225-030-2

- RONEŠOVÁ, Andrea. *Přehled protokolu MODBUS* [online]. Plzeň, 2005 [cit. 2018-11-27]. Dostupné z: <http://home.zcu.cz/~ronesova/bastl/files/modbus.pdf>
- ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace*. 1. díl, Základní pojmy, úvod do programování. Praha: BEN – technická literatura, 1999. ISBN 80-86056-58-9.
- ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace*. 2. díl, Sekvenční logické systémy a základy fuzzy logiky. Praha: BEN – technická literatura, 2005. ISBN 80-7300-087-3.
- TRZYNADLOWSKI, Andrzej Maria. *Control of induction motors* [online]. Amsterdam: Elsevier, 2007. ISBN 978-0-12-701510-1.
- VLACH, Jaroslav. *Řízení a vizualizace technologických procesů*. Praha: BEN – technická literatura, 1999. ISBN 80-860-5666-X.
- WHITT, Michael D. *Successful instrumentation and control systems design* [CD-ROM]. Research Triangle Park: ISA, 2004. ISBN 1-55617-844-1.
- ZEISEK, Jakub. *Regulace otáček asynchronního motoru* [online]. Plzeň, 2017 [cit. 2019-04-24]. Bakalářská práce. Západočeská univerzita v Plzni. Dostupné z: <https://otik.zcu.cz/bitstream/11025/27590/1/Regulace%20otacek%20asynchronniho%20motoru.pdf>.