

Operátorské panely a jejich komunikace s programovatelnými automaty

Vít Kvíčala

Bakalářská práce
2022



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Vít Kvičala
Osobní číslo: A19584
Studijní program: B3902 Inženýrská informatika
Studijní obor: Inteligentní systémy s roboty
Forma studia: Kombinovaná
Téma práce: Operátorské panely a jejich komunikace s programovatelnými automaty
Téma práce anglicky: Operator Panels and Their Communication with Programmable Logic Controllers

Zásady pro vypracování

1. Vypracujte literární rešerši na téma Průmyslové operátorské panely.
2. Popište síťové protokoly nejčastěji využívané pro komunikaci programovatelnými automaty (PLC) mezi sebou a především PLC s operátorským panelem.
3. Navrhněte a realizujte laboratorní úlohy demonstrující, jak propojit vybraný operátorský panel s PLC, popř. jeden operátorský panel s PLC různých výrobců.
4. Pro laboratorní úlohy vytvořte návody pro studenty, aby byli podle těchto návodů studenti schopni úlohy z bodu 3 pochopit a realizovat.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace, 1.díl. 1. Vyd. Praha: BEN – technická literatura, 1999. ISBN 80-860-5658-9.
2. ŠMEJKAL, Ladislav. PLC a automatizace, 2.díl. 1. Vyd. Praha: BEN – technická literatura, 2005. ISBN 80-7300-087-3.
3. MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty. Vyd. 2. Praha: Vydavatelství ČVUT, 2004. ISBN 80-010-2925-5.
4. KRŮPALA, Martin. Komunikační protokoly programovatelných automatů Tecomat Foxtrot a SAIA. Univerzita Tomáše Bati ve Zlíně, 2019. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně. Vedoucí práce Navrátil Pavel.
5. BOLTON, William. Programmable Logic Controllers. 6th Edition. 2015. ISBN 9780128029299.
6. BERGER, Hans. Automating with SIMATIC S7-1500: Configuring, Programming and Testing with STEP 7 Professional. Germany: Publicis MCD, 2014. ISBN 978-3-89578-919-9.
7. BERGER, Hans. Automating with SIMATIC S7-1200: Configuring, Programming and Testing with STEP 7 Basic Visualization with HMI Basic. 2nd enlarged and revised edition. Germany: Publicis MCD, 2013. ISBN 978-3-89578-901-4.

Vedoucí bakalářské práce: **Ing. Tomáš Sysala, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **15. ledna 2022**
Termín odevzdání bakalářské práce: **20. května 2022**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan



prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 15. ledna 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Kvíčala Vít v. r.
podpis studenta

ABSTRAKT

Cílem bakalářské práce je popis a následné použití operátorského panelu v oblasti programovatelných logických automatů (PLC) a jejich vzájemné komunikace. V první části je teoreticky popsán systém PLC, operátorských panelů (HMI) a komunikační protokoly, které jsou nejčastěji použity u těchto zařízení.

V praktické části jsou jednoduché úlohy, které byly vytvořeny pro ověření vzájemné komunikace mezi zvoleným operátorským panelem značky Weintek a programovatelnými logickými automaty Siemens a Saia, které jsou dostupné ve školních laboratořích UTB. K těmto úlohám je vždy vypracován podrobný manuál, tak aby čtenář po prostudování teoretické části a manuálu, byl schopen dané zadání zpracovat.

Klíčová slova: PLC, HMI, Weintek, Siemens, Saia

ABSTRACT

The aim of this bachelor thesis is to describe and flowingly demonstrate the usage of operator panels (HMI) within the area of programmable logic controllers (PLC) and their mutual communication. In the first part the PLC and HMI systems are described on a theoretical base together with the communication protocols, which are used most abundantly in these devices.

In the practical part are used simple practical sample tasks which were created with the intention to verify the communication between chosen operation panel by Weintek and PLC by Siemens and Saia. All of these systems are available in the laboratories of Tomas Bata University. For each demonstration has been created detailed manual for the reader to ensure the understanding of practical application of the tasks after familiarizing with the theoretical part of this work.

Keywords: PLC, HMI, Weintek, Siemens, Saia

Poděkování

Touto cestou bych rád poděkoval Ing. Tomáši Sysalovi, Ph.D. za vedení práce a za spolupráci při tvorbě a za rady při práci se systéme pro PLC Saia. Jelikož v této práci jsem se s tímto typem PLC setkal poprvé tak, i to mi umožnilo osobní rozvoj zkušeností.

Prohlášení

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 PLC	10
1.1 POJEM PLC	10
1.2 HARDWARE PLC.....	10
1.2.1 CPU	10
1.2.2 Paměť	11
1.2.3 Input/Output [I/O]	11
1.2.4 Komunikační rozhraní.....	12
1.2.5 Zdroj.....	12
1.3 VYKONÁVÁNÍ PROGRAM	12
1.3.1 Otočka cyklu	12
1.4 PROGRAMOVACÍ JAZYKY.....	13
1.4.1 Jazyk mnemokódů.....	13
1.4.2 Ladder diagram	14
1.4.3 Jazyk logických schémat.....	14
1.4.4 Jazyk strukturovaného textu.....	15
2 HMI	16
2.1 HMI.....	16
2.2 TAGY.....	17
2.3 PANEL WEINTEK MT 8051 IP.....	18
3 KOMUNIKAČNÍ PROTOKOLY	23
3.1 ISO/OSI.....	23
3.1.1 Fyzická vrstva	23
3.1.2 Linková vrstva.....	23
3.1.3 Síťová vrstva	24
3.1.4 Transportní vrstva	24
3.1.5 Relační vrstva.....	24
3.1.6 Prezentační vrstva	24
3.1.7 Aplikační vrstva	24
3.2 TCP/IP.....	24
3.2.1 Fyzická vrstva	25
3.2.2 Datová vrstva	25
3.2.3 Síťová vrstva	25
3.2.4 Transportní vrstva	26
3.2.5 Aplikační vrstva	26

3.3	ETHERNET/IP	26
3.4	PROFINET	27
3.5	PROFIBUS	28
3.6	IO-LINK	29
3.7	CONTROLNET	30
3.8	ETHERCAT	30
3.9	MODBUS	30
II	PRAKTICKÁ ČÁST	32
4	PRVNÍ ÚLOHA	34
4.1	ZADÁNÍ	34
4.2	POUŽITÉ SOUČÁSTI	34
4.3	REALIZACE ÚLOHY	36
4.3.1	Vytvoření Hardware konfigurace	36
4.3.2	Vytvoření I/O	38
4.3.3	Program	40
4.3.4	Bitová kombinace	42
4.4	VIZUALIZACE	43
4.5	VIZUALIZACE PRVNÍ ÚLOHY NA PANELU SIEMENS	48
	PŘIDÁNÍ PANELU	49
5	DRUHÁ ÚLOHA	55
5.1	ZADÁNÍ	55
5.2	REALIZACE	55
6	TŘETÍ ÚLOHA	60
6.1	ZADÁNÍ	60
6.2	REALIZACE	60
6.2.1	Program	60
6.2.2	Postup	62
6.2.3	Vizualizace	65
7	SAIA-BURGESS	68
7.1	ZADÁNÍ	68
7.2	REALIZACE PROGRAMU	68
7.3	VIZUALIZACE	72
8	ALLAN BRADLEY	76
8.1	REALIZACE	76
9	ZÁVĚR	83
	SEZNAM POUŽITÉ LITERATURY	84
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	87
	SEZNAM OBRÁZKŮ	88
	SEZNAM TABULEK	91

ÚVOD

Co se týče problematiky a všeobecně PLC a s tím souvisejícím použitím operátorských panelů je v dnešní době už neodmyslitelná část nejen v oblasti automatizace, ale i ve všech možných prostředích se kterými se denně setkáváme. Této oblasti se dostává velkého ohlasu, právě díky tomu že dochází k nahrazení vybrané pracovní činnosti strojem, který je řízen PLC, ať už je to v rámci rozvoje firmy, nebo zavedení nového výrobního procesu. V určitých situacích právě použití procesu, který je řízen PLC, vede ke zvýšení bezpečnosti na pracovišti, a tím předejití možným bezpečnostním rizikům.

Při použití PLC už z osobních zkušeností neodmyslitelně patří použití operátorského panelu, pro zjednodušení práce s ovládaným systémem, k jeho ovládání, kontrole, nebo vizualizaci procesu.

V této práci je použit panel od firmy Weintek, na kterém je znázorněna vizualizace modelových úloh a jejich ovládání. Pro vizualizaci budou použity PLC od Siemens a Saia. Pro jednotlivé úlohy bude vytvořen manuál, jak příslušné PLC propojit s HMI a nakonfigurovat.

Navíc bude panel použit v pracovní sestavě, která bude sloužit k představení funkčnosti panelu a modelových aplikací, které jsou zpracované na PLC Siemens.

I. TEORETICKÁ ČÁST

1 PLC

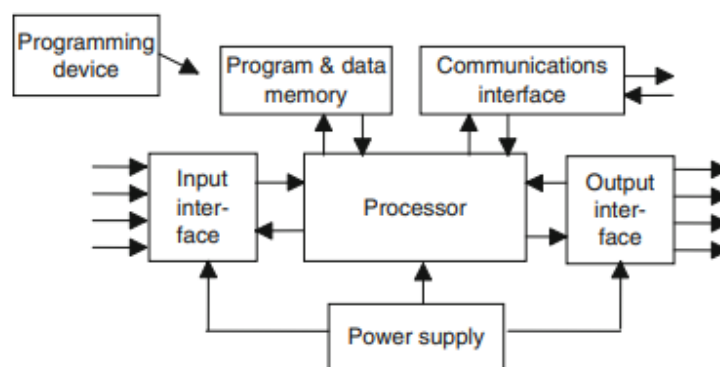
1.1 Pojem PLC

PLC neboli programovatelný logický automat. Je to opakovatelně programovatelný řídicí systém s příslušným hardwarem a softwarem, který je navržen k řízení procesů a ovládání strojů. [1]

Jedná se o specializované číslicové zařízení, které používá paměť k uchování instrukcí a následně jejich implementace jako je logika, časovače, čítače a k řízení a ovládání procesu. Systémy PLC jsou podobné PC, s tím rozdílem že PC jsou optimalizované pro výpočty a řešení úloh, tak PLC jsou optimalizovaná pro řízení úloh a průmyslové prostředí. Díky tomu je jejich konstrukce přizpůsobena podmínkám, které jsou obtížné pro PC, jako například vibrace, vysoké nebo nízké teploty, vlhkost, prach. [2]

To, že tyto systémy jsou velmi odolné svědčí už jen to, že například řízení světelné dopravy na některých křižovatkách je řízené pomocí PLC, které je mnohdy staré i desítky let a jejich program je stále vykonáván, bez výrazných problémů.

1.2 Hardware PLC



Obr. 1 Schématická struktura PLC [2]

1.2.1 CPU

Nedílná součást PLC, obsahuje mikroprocesor, navíc obsahuje i „arithmetic logic unit“ (ALU), která je zodpovědná za manipulaci s daty a provádění aritmetických operací, jako je sčítání, odčítání a logických operací AND, OR, NOT, XOR. [2]

Programovatelnému automatu poskytuje inteligenci, realizuje souborové instrukce a systémové služby. Zároveň zajišťuje komunikaci s vlastními i vzdálenými moduly, s nadřazeným systémem, ale i s programovacím zařízením. U současných programovatelných automatů je obsažen mikroprocesor, mikrořadič, nebo specializovaný řadič, který je zaměřen na rychlé provedení instrukce. [3]

1.2.2 Paměť

Pro správnou funkci PLC je potřeba aby měl přístup ke zpracovaným datům a informacím z digitálních či analogových vstupů. Informace se ukládají do paměti. Použité typy paměti jsou RAM a ROM, popřípadě EPROM. [2]

ROM (*Read only memory*): permanentní úložiště pro operační systém a data pro CPU.

RAM (*Random acces memory*): je použita pro uživatelský program a data. Tento typ paměti je použit pro uložení informací o stavu I/O zařízení.

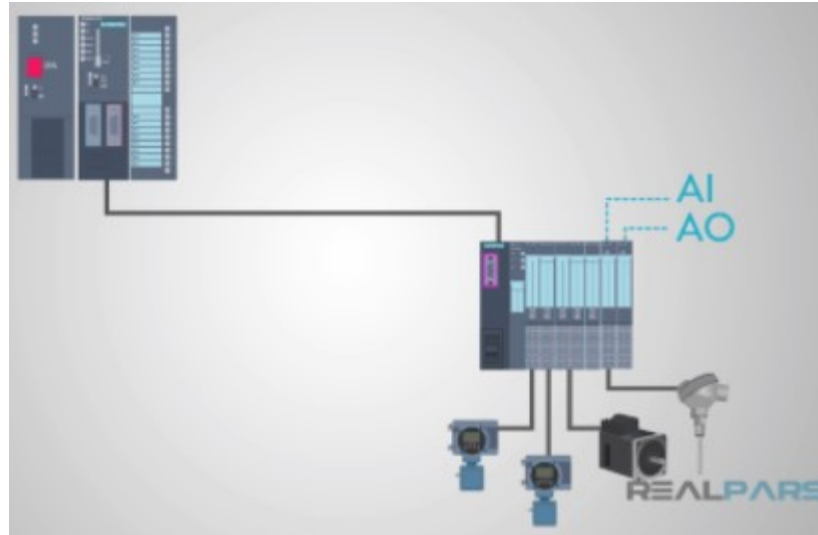
EPROM (*Erasable and programmable read only memory*): je možné použití i tohoto typu paměti k permanentnímu uložení paměti, ne všechny PLC ale mají tento typ paměti. Program je primárně uložen v paměti RAM, takže jakmile dojde k odpojení od zdroje, tak dojde k smazání programu, proto, PLC obsahuje baterii, která uchová program v RAM i v případě odpojení od zdroje, ovšem jen po dobu kapacity baterie. [3]

1.2.3 Input/Output [I/O]

Místo kde procesor obdrží informace z externích zařízení a posílá informace na externí zařízení. Je to takový interface mezi systémem a vnějším světem. Externí zařízení mohou být například indukční, nebo optoelektrické snímače, která nás budou informovat o přítomnosti materiálu. Tyto informace jsou do PLC přivedeny jako 1 nebo 0. Na vstupy mohou jít také informace v podobě analogového signálu, jako například teplota, tlak, průtok, ovšem tyto informace nám nemusí jít napřímo jako daná hodnota, ale jako hodnota v bitovém rozsahu, kterou si následně převedeme pomocí instrukce v program.

Co se týče výstupů, na ty už máme připojená koncová zařízení, jako například frekvenční měniče, které jsou napojené například na motor, žárovku nebo klakson. [2]

Také musíme brát v potaz, že počet fyzických vstupů je omezený. PLC má určitý počet fyzických I/O, tento počet můžeme rozšířit doplňujícími I/O kartami, které jsou kompatibilní pro typ PLC.



Obr. 2 Ukázkové schéma propojení PLC s analogovými vstupy a výstupy [8]

1.2.4 Komunikační rozhraní

Slouží k příjmu nebo přenosu dat ze sítě, nebo dalších PLC připojených na síť. Zabývá se akcemi, jako například ověření zařízení, synchronizace, sběr dat nebo správa připojení. [2]

1.2.5 Zdroj

Je potřebný k převodu AC napětí na nízké DC napětí, které je potřebné pro procesor a pro I/O moduly. [2]

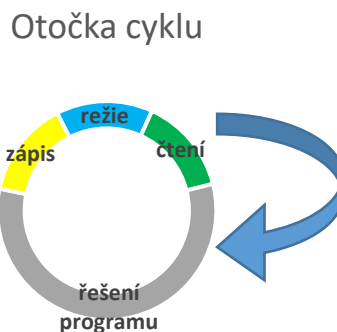
1.3 Vykonávání program

Samotný program vytvořený uživatelem je určitá posloupnost instrukcí a příkazů, které jsou cyklicky vykonávány bez toho, aby programátor musel nastavit zakončení smyčky zpět na začátek, to již zajistí systém. Pokud dojde k zdržení při vykonávání programu, tudíž setrvání ve smyčce, dojde k chybě na překročení doby cyklu. [3]

1.3.1 Otočka cyklu

Program je cyklicky vykonáván v „otočce cyklu“. Nejprve je provedena aktualizace I/O, aktualizace časových údajů pro časovače, systémových registrů a zkontroluje komunikaci.

Během režie jsou na výstupy poslány aktuální hodnoty, poté systém provede režijní operace. [3]



Obr. 3 Otočka cyklu [2]

1.4 Programovací jazyky

K programování PLC jsou čtyři specializované jazyky, které byly navrženy pro snadnou orientaci a realizaci logických funkcí. Přenositelnost programů mezi PLC různých výrobců není možná, je to možné pouze mezi systémy stejného výrobce. Mezinárodní norma IEC 1131-3 sjednocuje programovací jazyky pro PLC na čtyři typy. [3]

1.4.1 Jazyk mnemokódů

Jindy též pod názvem Instruction List (IL). Je to obdoba programovacího jazyku assembleru u PC a je též strojově orientovaný, to znamená že každé instrukci odpovídá příkaz. [3]

Tento jazyk se v dnešní době u systémů PLC už skoro nevyskytuje.

```
ld true
st blinker.run
ld t#1s
st blinker.cycle
cal blinker

ld blinker.q
st trigger.clk
cal trigger

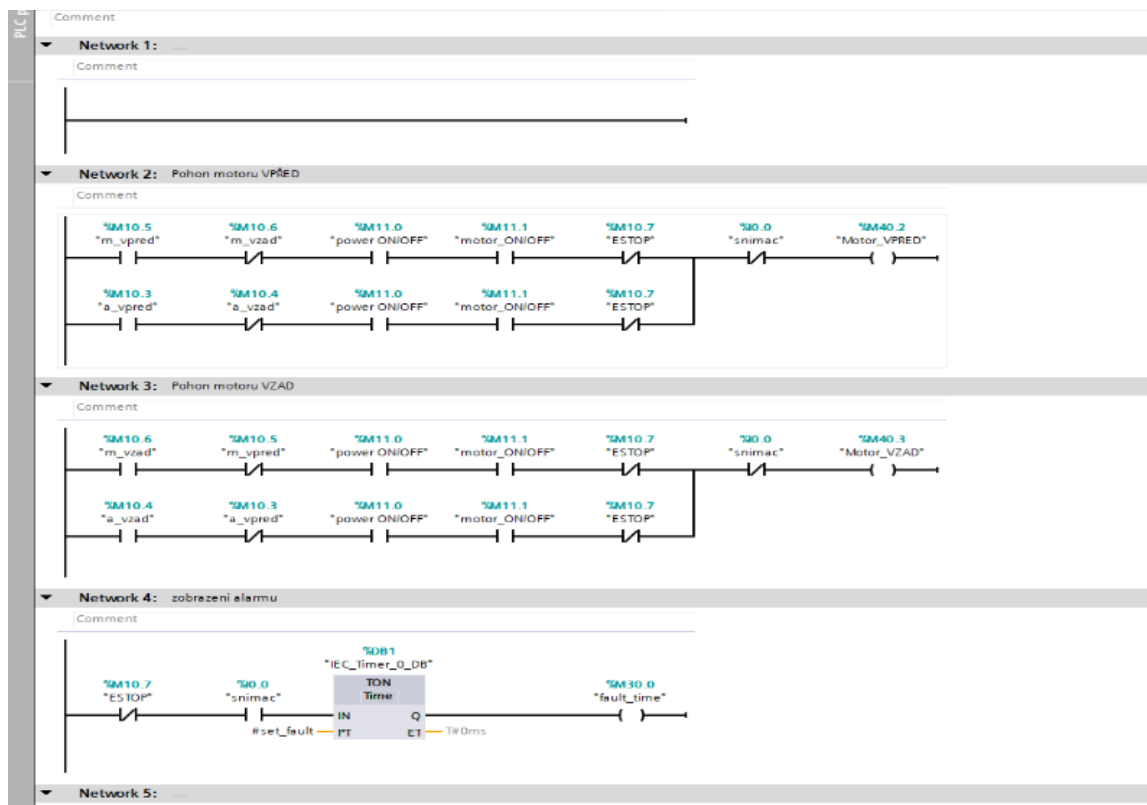
ld trigger.q
jmpnc LBmodulo
ld counter
add 1
st counter
```

Obr. 4 Ukázka IL

1.4.2 Ladder diagram

Jazyk reléových schémat (LAD). Zobrazení logických operací v podobě použitých při kreslení kontaktních schémat při práci s reléovými prvky, jen s úpravami pro různé prvky v grafické podobě. [2]

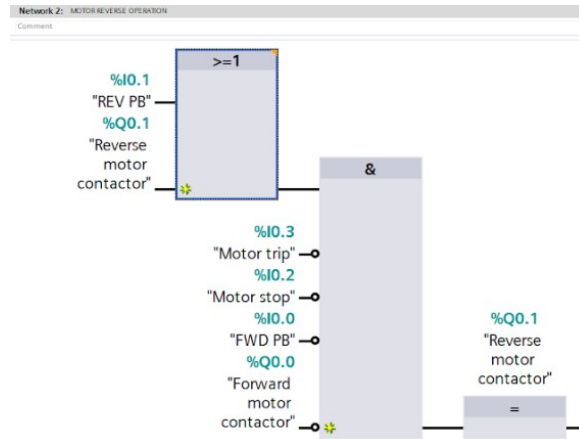
Tento jazyk je výhodný pro programování jednodušších aplikací, kdy pracujeme s menším množstvím I/O, potom je výhodné pracovat právě s LAD, kdy již při programování získává program přehlednost. Tento typ programovacího jazyku, je v oblibě v průmyslu i právě skrz svoji přehlednost, z důvodů servisních úkonů a údržby, kdy zařízení obsluhují, nebo opravují lidé, kteří program nevytvořili. Tudiž v případě že se například něco pokazí na lince, tak servisní mechanik při nahlédnutí do programu může rychleji najít problém.



Obr. 5 Ladder diagram

1.4.3 Jazyk logických schémat

Nebo jinak i jazyk funkčních bloků (FBD). Je to opět grafický jazyk, kdy základní operace mají vizuální podobu obdélníkových značek, které mění svoji velikost a podle počtu vstupů. Tento jazyk je příjemný uživatelům, kteří jsou zvyklí na kreslení logických schémat pro zařízení s integrovanými obvody. [2]



Obr. 6 Jazyk FBD

1.4.4 Jazyk strukturovaného textu

V anglických textech najdeme podle názvu Structure text (ST), je to obdoba vyšších programovacích jazyků, jako je jazyk C, C++, C#. Tento jazyk najdeme u složitějších programů, kdy už použití Ladder diagramu by zabralo mnoho prostoru, co se týká počtu řádků.

```

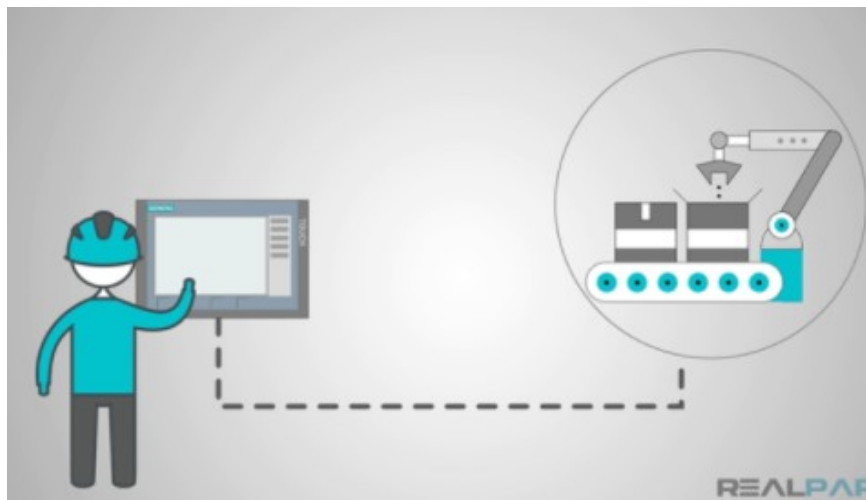
IF... CASE... FOR... WHILE... (*...*) REGION
OF... TO DO... DO...

1
2 // min limitst lead array
3
4 IF #SI_HMI < 1 THEN
5   #SI_HMI := 1;
6 END_IF;
7
8 // max limitst lead array
9
10 IF #SI_HMI > (#CI_MaxRows-9) THEN
11   #SI_HMI := (#CI_MaxRows-9);
12 END_IF;
13
14 // load row of archive
15
16 FOR #TI_Index := 1 TO 10 DO
17   #SA_HMI[#TI_Index] := "DB_Sortiment".Program[#SI_HMI + #TI_Index - 1];
18 END_FOR;
19
20 // copy archive to actual product
21 IF #SI_LoadRow > 0 AND #SI_LoadRow < 11 AND NOT #Automat THEN
22   "DB_Sortiment".Aktualni_program := "DB_Sortiment".Program[#SI_HMI + #SI_LoadRow - 1];
23   "DB_Sortiment".Cislo_programu := #SI_HMI + #SI_LoadRow - 1;
24 END_IF;
25
26 #SI_LoadRow := 0;
27
28 // write actual to archive
29 //
30
31 IF #SBI_Write AND "DB_Sortiment".Cislo_programu > 0 AND "DB_Sortiment".Cislo_programu <= #CI_MaxRows THEN
32   "DB_Sortiment".Program["DB_Sortiment".Cislo_programu] := "DB_Sortiment".Aktualni_program;
33 END_IF;
34
35 #SBI_Write := FALSE;

```

Obr. 7 Jazyk strukturovaný text

2 HMI

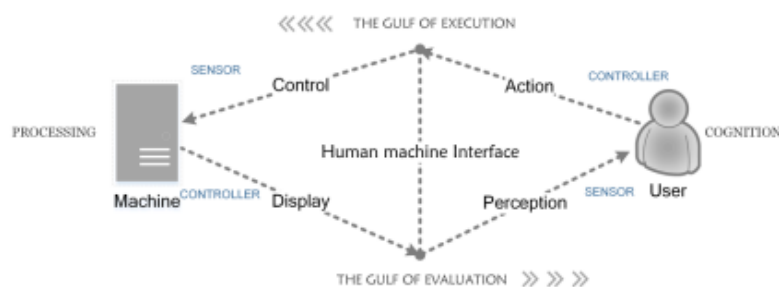


Obr. 8 Použití HMI jako ovládací člen [7]

2.1 HMI

Pod zkratkou HMI se skrývá název „Human machine interface“. Ve spojení s PLC je toto zařízení nástrojem k vizualizaci řízeného prostředí, nebo k možnosti ovládání řídicího programu. Proto může být grafické prostředí na obrazovce uspořádané tak, že napodobuje rozložení řídicího systému, kvůli snazší manipulaci, nebo přehlednosti. Je to zařízení, které spojuje uživatele se strojem, systémem, nebo zařízením. Termín HMI může být technicky aplikován na jakoukoliv obrazovku, která umožňuje vzájemnou interakci člověka se zařízením, přesto je tento termín nejčastěji použit v kontextu průmyslového procesu. [17]

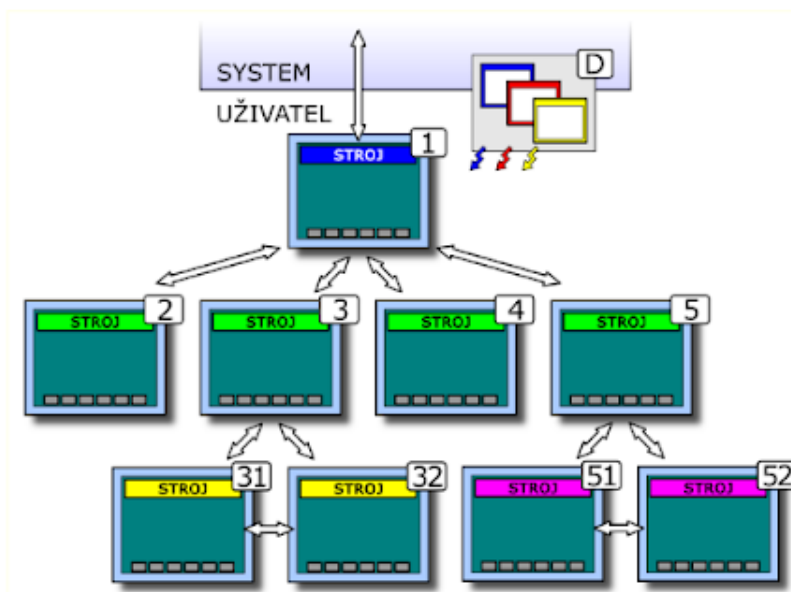
Přestože HMI je jeden z nejpoužívanějších pojmů v této souvislosti, někdy je odkazováno na Man-Machine Interface (MMI), Operator Interface Terminal (OIT), Local Operator Interface (LOI), nebo Operator Terminal (OT). [17]



Obr. 9 Cyklický model interakce uživatel a zařízení [18]

Samotný panel disponuje jednou obrazovkou a ve většině případů jedna obrazovka nestačí. Díky velkému množství informací a datům, které je třeba zobrazit, je výhoda

strukturovanosti HMI obrazovek. Vycházíme ze základní obrazovky a poté rozšiřujeme strukturu podle potřeby.



Obr. 10 Obecná struktura HMI obrazovek [15]

2.2 Tagy

Moderní HMI panely a software který je použit k jejich konfiguraci je výhradně založen na použití tagů, mohlo by to být vyjádřeno i jako datové značky, které jsou spojeny s každým použitým prvkem na obrazovce, a právě tento tag je propojen s PLC. Při základním programování při použití základních I/O, které má PLC, jsou tagy odkazem na tyto I/O. [5]

V podstatě každá část paměti, na kterou se budeme odkazovat v programu, má své pojmenování, které je přiřazeno úseku paměti. Toto pojmenování je potom použito při tvorbě programu.

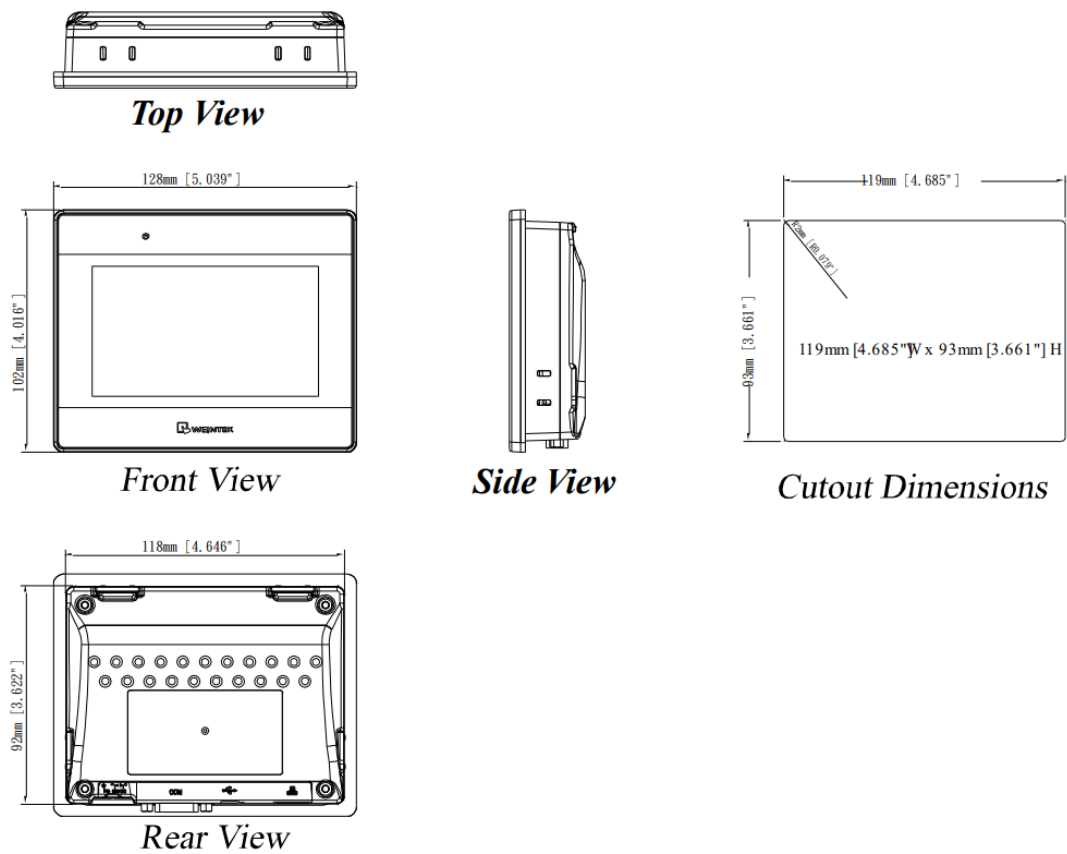
Například máme-li na první vstup napojen snímač, tak v PLC budeme mít tag “%I0.0 snímač” a jeho adresa je I0.0.

14	Clock_0.5Hz	Default tag table	Bool	%M0.7
15	OB_Motor_1	OUTPUTS	Bool	%Q0.1
16	OB_Motor_2	OUTPUTS	Bool	%Q0.2
17	a_vpřed	INPUTS	Bool	%M10.3
18	a_vzad	INPUTS	Bool	%M10.4
19	m_vpřed	INPUTS	Bool	%M10.5
20	m_vzad	INPUTS	Bool	%M10.6
21	ESTOP	INPUTS	Bool	%M10.7
22	power ON/OFF	INPUTS	Bool	%M11.0
23	motor_ON/OFF	INPUTS	Bool	%M11.1
24	svetelko_HMI	OUTPUTS	Bool	%M20.4
25	Budiz_svetlo	INPUTS	Bool	%M11.2
26	snimac	INPUTS	Bool	%I0.0
27	fault_time	OUTPUTS	Bool	%M30.0
28	OB_Motor_0	OUTPUTS	Bool	%Q0.0
29	OB_Motor_3	OUTPUTS	Bool	%Q0.3
30	rychlost_1	INPUTS	Bool	%M20.0
31	rychlost_2	INPUTS	Bool	%M20.1
32	rychlost_3	INPUTS	Bool	%M20.2
33	rychlost_4	INPUTS	Bool	%M20.3
34	rychlost_5	INPUTS	Bool	%M20.5
35	rychlost_6	INPUTS	Bool	%M20.6
36	rychlost_7	INPUTS	Bool	%M20.7
37	rychlost_8	INPUTS	Bool	%M21.3
38	rychlost_9	INPUTS	Bool	%M21.1
39	rychlost_10	INPUTS	Bool	%M21.2
40	rychlost_11	INPUTS	Bool	%M21.3
41	rychlost_12	INPUTS	Bool	%M21.4
42	rychlost_13	INPUTS	Bool	%M21.5
43	rychlost_14	INPUTS	Bool	%M21.6
44	rychlost_15	INPUTS	Bool	%M21.7
45	rychlost_16	INPUTS	Bool	%M22.0
46	IB_Motor_RUN	INPUTS	Bool	%I0.1
47	rychlost_1_ON	INPUTS	Bool	%M40.0
48	rychlost_2_ON	INPUTS	Bool	%M40.1
49	Motor_VPRED	INPUTS	Bool	%M40.2
50	Motor_VZAD	INPUTS	Bool	%M40.3
51	rychlost_3_ON	INPUTS	Bool	%M40.4
52	rychlost_4_ON	INPUTS	Bool	%M40.5
53	rychlost_5_ON	INPUTS	Bool	%M40.6
54	rychlost_6_ON	INPUTS	Bool	%M40.7
55	rychlost_7_ON	INPUTS	Bool	%M41.0

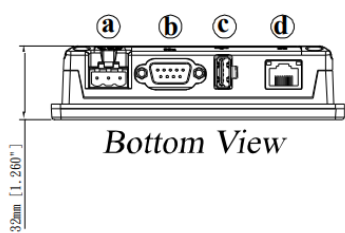
Obr. 11 Ukázka tagů v PLC

2.3 Panel Weintek MT 8051 iP

Modelová řada iP s dotykovou obrazovkou je určena pro vizualizaci a monitorování technologických procesů řízených PLC. Model MT 8051 iP disponuje 4,3“ LCD displejem s podsvícením a hustotou 16,7 milionu barev. Panel obsahuje procesor RISC Cortex-A8, který má takt 600 MHz a k dispozici má 128 MB operační paměti. Komunikační porty, k připojení a komunikaci s PLC, máme k dispozici RS 485, RS 232 nebo Ethernet. Právě zmiňovaný Ethernetový port je využit k propojení panelu s PLC v této práci.



Obr. 12 Rozměry panelu MT8051iP [20]



a.	Power Connector
b.	COM1 RS-232, RS-485 2W/4W, COM3 RS-485 2W
c.	USB Host
d.	Ethernet


Obr. 13 Konektory panelu MT8051iP [20]

Display	Display	4.3" TFT LCD
	Resolution	480 x 272
	Brightness (cd/m ²)	400
	Contrast Ratio	500:1
	Backlight Type	LED
	Backlight Life Time	>30,000 hrs.
	Colors	16.7M
	LCD Viewing Angle (T/B/L/R)	50/70/70/70
	Pixel Pitch (mm)	0.198(H) x 0.198(V)
Touch Panel	Type	4-wire Resistive Type
	Accuracy	Active Area Length(X)±2%, Width(Y)±2%
Memory	Flash	128 MB
	RAM	128 MB
Processor		32-bit RISC 600MHz
I/O Port	USB Host	USB 2.0 x 1
	USB Client	N/A
	Ethernet	10/100 Base-T x 1
	COM Port	COM1 RS-232/RS-485 2W/4W COM3 RS-485 2W
	RS-485 Dual Isolation	N/A
RTC		Built-in (CR2032 3V lithium battery.)
Power	Input Power	10.5~28VDC
	Power Consumption	800mA@12VDC ; 400mA@24VDC
	Power Isolation	Built-in
	Voltage Resistance	500VAC (1 min.)
	Isolation Resistance	Exceed 50MΩ at 500VDC
	Vibration Endurance	10 to 25Hz (X, Y, Z direction 2G 30 minutes)
Specification	PCB Coating	N/A
	Enclosure	Plastic
	Dimensions WxHxD	128 x 102 x 32 mm
	Panel Cutout	119 x 93 mm
	Weight	Approx. 0.25 kg
	Mount	Panel mount
Environment	Protection Structure	NEMA4 / IP65 Compliant Front Panel
	Storage Temperature	-20° ~ 60°C (-4° ~ 140°F)
	Operating Temperature	0° ~ 50°C (32° ~ 122°F)
	Relative Humidity	10~90% (non-condensing)
Certification	CE	CE marked
Software		EasyBuilder Pro EasyAccess 2.0 (Optional)

Obr. 14 Parametry k panelu MT8051 iP [20]

Značka Weintek nabízí, širokou škálu panelů. Z modelové řady iP je model 8051 ten nejmenší nabízený. V porovnání s dalšími panely stejné modelové řady, jsou rozdíly především ve velikosti displeje.

Picture & Dimensions

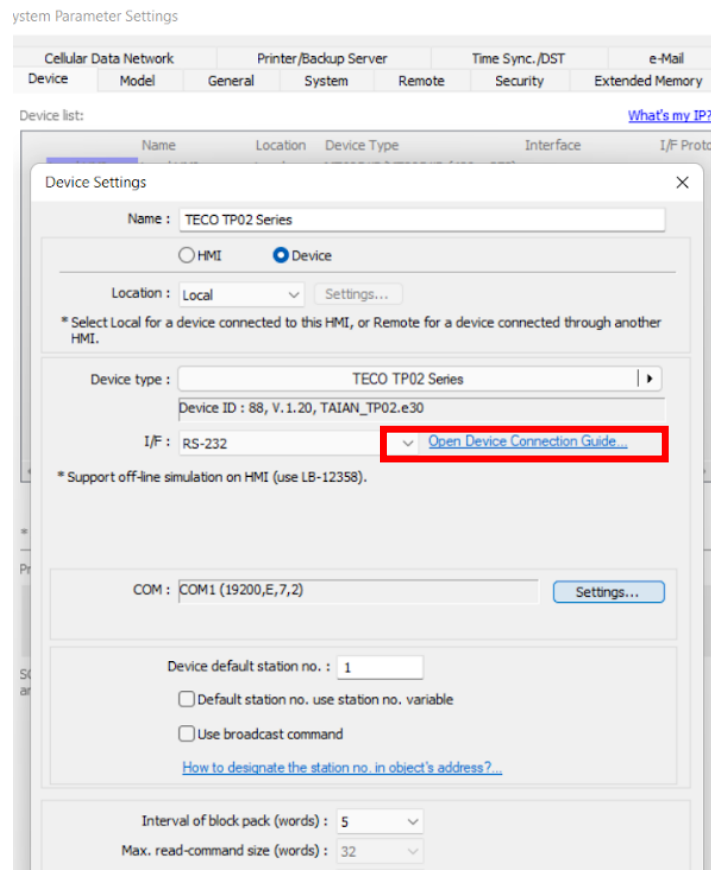


Specification

	4.3" TFT	7" TFT	7" TFT	10.1" TFT
Display	4.3" TFT	7" TFT	7" TFT	10.1" TFT
Resolution	480 x 272	800 x 480	800 x 480	1024 x 600
Brightness (cd/m ²)	400	300	300	350
Contrast Ratio	500:1	500:1	500:1	500:1
Backlight Type	LED	LED	LED	LED
Backlight Life Time	>30,000 hrs.	>30,000 hrs.	>30,000 hrs.	50,000 hrs.
Colors	16.7M	16.7M	16.7M	16.7M
LCD Viewing Angle	50/70/70/70	70/50/70/70	70/50/70/70	70/70/80/80

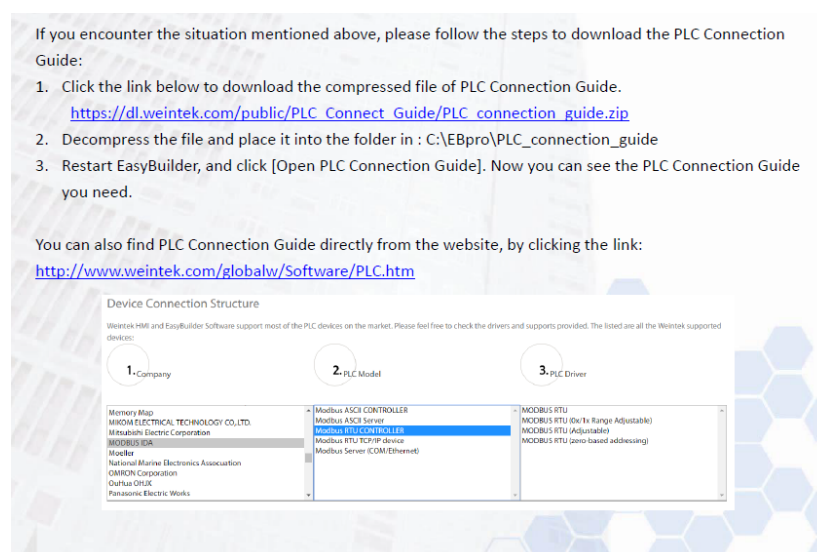
Obr. 15 Modelová řada iP [21]

Při práci s panelem je k dispozici rozšířená základna informací a potřebných souborů z webových stránek výrobce. Jedním z nejdůležitějších využití je pro zjištění předávání dat mezi využitým typem PLC a panelem. Rychlý přístup k těmto údajům získáváme v nastavení zařízení v panelu, po přidání příslušného PLC máme k dispozici možnost stáhnutí dokumentů ze stránek výrobce. Stáhnuté dokumenty podle popisu umístíme do složky v našem PC, nebo je možnost využití webového konfiguratoru k stáhnutí potřebného dokumentu.



Obr. 16 Nastavení zařízení

Po kliknutí na „Open Device Connection Guide“, se otevře dokument se zobrazeným návodem, jak příslušná data stáhnout ze stránek výrobce, nebo pomocí webového odkazu stáhnete příslušný dokument z konfiguratoru.



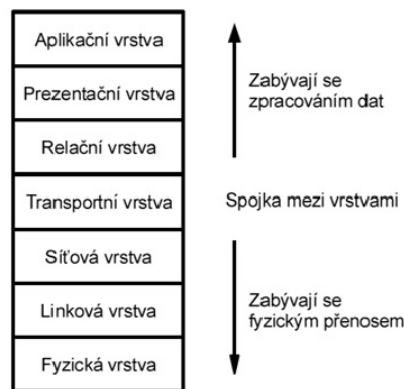
Obr. 17 Náhled na návod pro připojení komunikace

3 KOMUNIKAČNÍ PROTOKOLY

Komunikace je důležitá v systémech PLC, potřebná nejen ke komunikaci se základními I/O, které jsou k dispozici v PLC, ale i I/O kartám, dalším PLC, připojeným zařízením, nebo jiných periférií.

3.1 ISO/OSI

V roce 1979 navrhla Mezinárodní organizace pro normalizaci (ISO – International Organization for Standardization) model, který je použit jako standard pro propojení otevřených systémů (OSI – Open System Interconnection). Komunikace mezi digitálními zařízeními je pomocí protokolu popsána v sedmi vrstvách.[2]



Obr. 18 Referenční model ISO/OSI [12]

Jednotlivé vrstvy jsou přesně definovány určitou funkcí a sousedí s vrstvou nižší/vyšší, výjimky jsou vrstva Aplikační, která sousedí přímo s procesem a Fyzická vrstva, která sdílí rozhraní s fyzickým médiem. [12]

3.1.1 Fyzická vrstva

Zabývá se kódováním a fyzickým přenosem informací. Mezi její funkce patří i synchronizace přenosových dat a přenos bitů mezi systémy. [2]

Řeší technické parametry jako hodnoty signálů 0 a 1, konektory a kabely. Nezajímá se o samotnou informaci v podobě bitů, pouze ji přeposílá. [12]

3.1.2 Linková vrstva

Definuje protokoly pro odesílání a příjem informací mezi systémy, které jsou propojeny. Někdy je označována jako spojovací vrstva, jelikož zde dochází ke spojení bitů do rámců, k

jejich kontrole dochází pomocí kontrolních součtů. Jedna z funkcí je řízení toku, kdy se řídí přenos tak, aby příjemce stíhal zpracovávat informace. [2]

3.1.3 Síťová vrstva

Přepínání cest dat mezi systémy. Data jsou zde členěna do paketů. Pro adresování dat, jsou zde použity IP adresy. [12]

3.1.4 Transportní vrstva

Protokoly zodpovědné za posílání zpráv z jednoho konce sítě na druhý. Kontroluje tok dat. Vyrovnává rozdíly mezi třemi spodními síťově orientovanými vrstvami a horními aplikačně orientovanými vrstvami. Rozpozná data a předává je příslušnému programu. [12]

3.1.5 Relační vrstva

Obstarává operace, které probíhají během komunikace mezi uzly, jako navázání, řízení, rušení, jaký typ komunikace bude probíhat, nebo zda bude použito šifrování. [12]

3.1.6 Prezentační vrstva

Informace z nejvyšší vrstvy jsou zde překládány, tak aby s nimi nižší vrstvy mohli pracovat. Probíhá zde komprese a kódování dat. Je zde důležitější struktura dat než význam. [12]

3.1.7 Aplikační vrstva

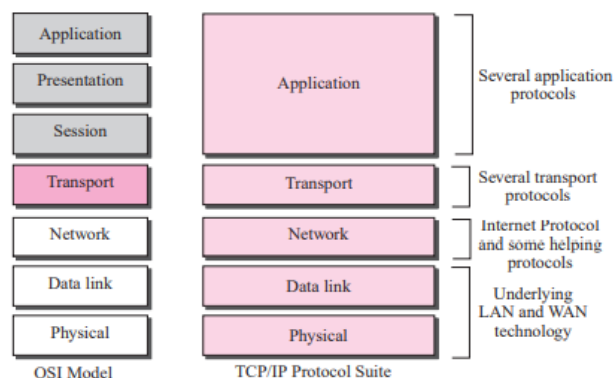
Nejvyšší vrstva poskytuje služby aplikacím. Nachází se zde program na řízení databází, tiskové servery, nebo příkazy operačního systému. Do této jediné vrstvy má uživatel přístup. [12]

3.2 TCP/IP

Sada komunikačních protokolů používaných k propojení síťových zařízení na internetu, používá se i jako komunikační protokol v privátních sítích (intranet). TCP a IP jsou dva hlavní protokoly, které specifikují způsob, jakým bude probíhat výměna dat, komunikace mezi dvěma body, určuje jak budou data rozdělena při přenosu, jak budou adresována, přenášena a přijímána v koncovém bodě. TCP/IP je navrženo tak, aby při selhání zařízení na síti, byla automaticky obnovena spolehlivost. TCP definuje, jak mohou aplikace komunikovat prostřednictvím sítě. Řídí sestavení dat do menších paketů, než jsou přeneseny po síti a poté znovu sestaveny. IP definuje, jak adresovat a jak směřovat každý paket, tak aby bylo

zaručeno dosažení správného cíle. Kontroluje se IP adresa a podle ní je určeno, kam se má zpráva předat. Součástí sady protokolů TCP/IP je i HTTP (Hypertext Transfer Protocol), který se stará o komunikaci mezi web server a web browser a FTP (File Transfer Protocol) přenos mezi PC. [16]

Sada protokolů TCP/IP byli vyvinuty dříve než model OSI, proto se jednotlivé vrstvy neshodují. TCP/IP je hierarchický protokol s interaktivními moduly. Moduly na sobě nemusí být vzájemně závislé. V modelu OSI je určeno, které funkce patří do příslušné vrstvy, naopak TCP/IP obsahuje relativně nezávislé protokoly, které mohou být smíšené, podle potřeb systému. V normálních případech je TCP/IP uváděn se čtyřmi vrstvami, zde uvedeme protokol s pěti vrstvami, s tím že pátá vrstva fyzická, je zařízení. [2]



Obr. 19 Rozdíl mezi OSI modelem a TCP/IP [22]

3.2.1 Fyzická vrstva

TCP/IP nedefinuje žádný specifický protokol pro fyzickou vrstvu. Podporuje všechny standardní protokoly. Na této úrovni probíhá komunikace mezi PC a směrovačem. Jednotkou komunikace je jeden bit. [22]

3.2.2 Datová vrstva

Podporuje standardní protokoly, na této úrovni je také komunikace mezi dvěma segmenty. Komunikační jednotku je zde paket nazvaný „frame“, neboli určitý rámeček dat. V rámci jsou obsažena data přijatá ze síťové vrstvy s přidanou hlavičkou. V hlavičce je obsaženy informace o zdroji a cílové adrese. [22]

3.2.3 Síťová vrstva

Podporuje protokol IP, ten je jako přenosový mechanismus použit u protokolu TCP/IP. IP přenáší data v paketech, nazvaných datagramy, které jsou přepravovány samostatně.

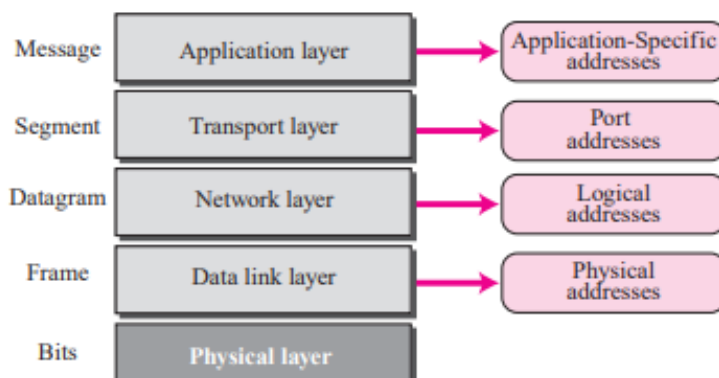
V síťové vrstvě může probíhat kontrola zdroje a cíle paketů, není však povoleno měnit obsah paketu, spíše se určuje vhodnější trasa dat. [22]

3.2.4 Transportní vrstva

Je zodpovědná za doručení celého paketu. Každý paket musí být rozdělen na datagramy a každý datagram musí být dodán do síťové vrstvy pro přenos. Internet definuje rozdílné cesty pro každý datagram, tak datagramy mohou být doručeny v rozdílném pořadí, nebo mohou být ztraceny. Po přijetí jsou všechny datagramy zkontrolovány a opět složeny do paketu. Komunikace mezi transportními vrstvami se děje pomocí výměny bitů. [22]

3.2.5 Aplikační vrstva

U TCP/IP je tato vrstva ekvivalentní kombinaci prezentační a aplikační vrstvě u OSI modelu. Aplikační vrstva umožňuje uživateli přístup k internetu. Mnoho protokolů je definováno touto vrstvou, jako email, nebo přenos souborů. [22]

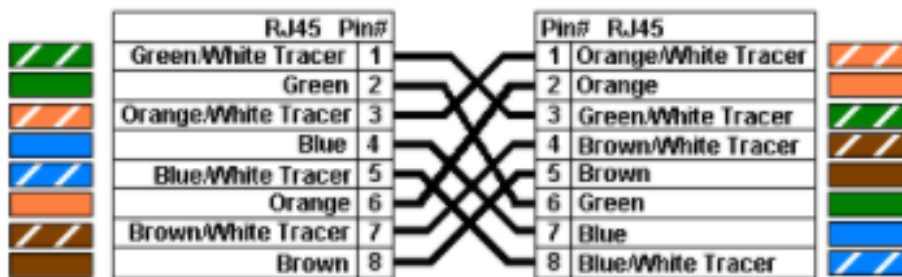


Obr. 20 Adresování u TCP/IP [22]

3.3 Ethernet/IP

Nabízí komplexní sadu zpráv a služeb pro automatizační aplikace. Je to otevřený síťový standart, použitý pro Ethernetové komunikační zařízení, podporující real time I/O komunikaci. Síť Ethernet/IP podporuje i bezpečnostní aplikace, tudíž během standartního přenosu dat, dochází k přenosu bezpečnostních dat na společné síťi. [12]

Každá připojená stanice má stejný status. S ethernetem se dnes setkáváme téměř u všech připojených zařízení, kdy nejčastěji je použita kroucená dvoulinka s konektorem RJ-45.



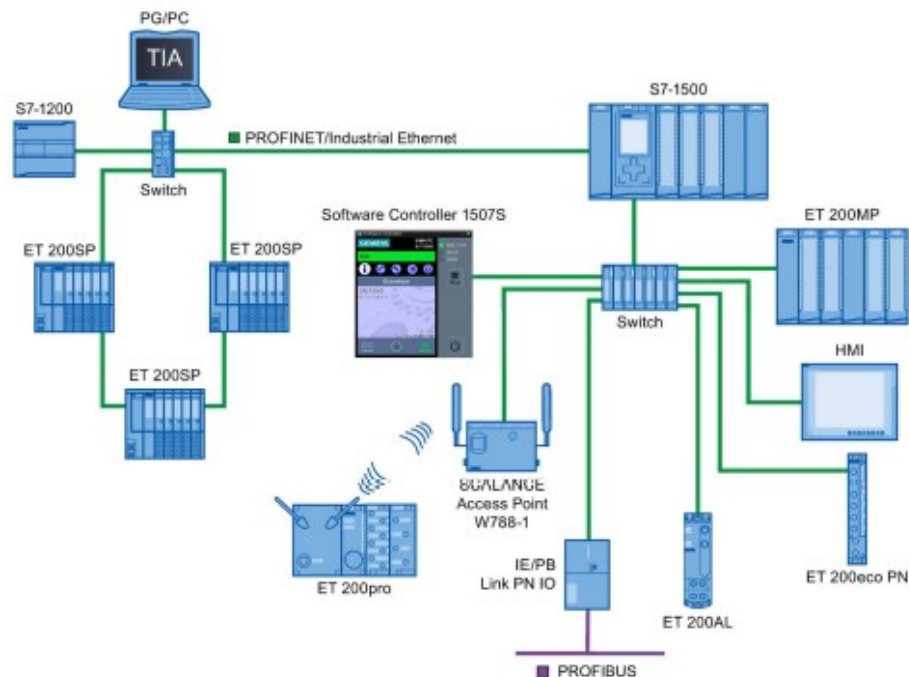
Obr. 21 Křížové propojení u RJ45 [19]

Ethernet má tu výhodu, že jej je možné použít na více zařízení spolu propojených, tudíž může mít teoreticky neomezený počet uzlů, jelikož používá standardní ethernet a switche, díky tomuto připojení má velkou kompatibilitu. [6]

3.4 PROFINET

Profinet vychází z dlouholetých zkušeností s Profibus DP a kombinuje uživatelské operace se současným inovativním konceptem Ethernetové technologie. Tím je zajištěna integrace Profinet DP do světa Profinetu. Profinet je automatizační standart, základem z Ethernetu. Zajišťuje vzájemnou komunikaci mezi dodavateli elektronických zařízení. [9]

Standart pro komunikační aplikace, umožňuje připojení distribuovaných periférií pomocí sítě Ethernet. Síť Ethernetu si můžeme představit jako buňkovou síť, která je v souladu se standardem IEEE 802.3 a je právě určena pro použití v průmyslovém sektoru. Díky tomu máme možnost řešit komplexní síťová řešení s vysokým přenosovým výkonem. V současnosti je Profinet stále mladý sběrníkový systém, tudíž v porovnání s Profibus je mnohem méně modulů, ale stále se vyvíjejí nové. [7]



Obr. 22 Možné zasíťování zařízení pomocí Profinetu [9]

Typy zařízení PROFINET IO:

PROFINET IO Controller: typicky PLC, které po spuštění ustavuje logické spojení s připojenými IO zařízeními a přiřazuje jim parametry.

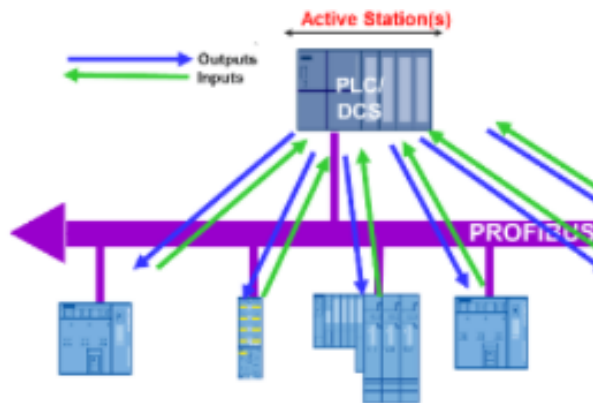
PROFINET IO Device: zařízení které je připojené přes Profinet. Můžeme mít kompaktní, modulární nebo inteligentní.

PROFINET IO Supervisor: zde spadá programovací zařízení (PG), PC, nebo HMI.

Nedílnou součástí sítě Profinet je Switch, díky kterému zvětšujeme použitou síť. [9]

3.5 PROFIBUS

Sběrníkový systém pro buňkové sítě s několika uzly. Při splnění požadavků podle norem, zaručuje PROFIBUS otevřenost pro připojení periferií, od všech výrobců, umožňující tento typ komunikace, které jsou v souladu s normami. Komunikační protokol založený na systému „master-slave“, kdy jedno zařízení (PLC) komunikuje s řadou zařízení. [7]



Obr. 23 Náhled na schéma komunikace [7]

Komunikace probíhá tak, že „*master*“ neustále komunikuje se zařízeními „*slave*“ v určité periodě, nemusí nutně posílat rámce dat. V zařízení má nastavený „*watchdog*“, to znamená že pokud *master* nedostane ze zařízení zpětnou informaci o komunikaci v určitém limitu, nastává poruchové hlášení. [7]

PROFIBUS je standardem v oblasti sběrnicových modulů po delší dobu, tudíž je dostupná široká škála modulů.

PROFINET vs PROFIBUS		
	PROFINET	PROFIBUS
Topologie:	bus, star, ring	bus
Dostupnost modulů:	+	+++
max.přenosová rychlost:	100 Mb/s	12 Mb/s
max.délka kabelů:	100 m	-
max.délka segmentu:	-	1000 m
max.počet segmentů:	-	10
max.počet modulů:	>256	125

Tabulka 1 Profinet x Profibus

3.6 IO-LINK

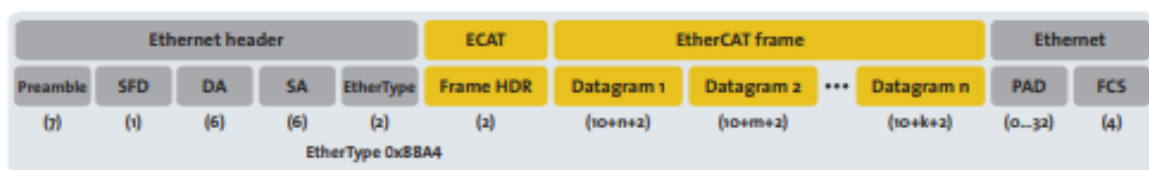
Je vlastně nezávislé řešené rozhraní pro sensory a akční členy s vzájemným použitím Profibus a Profinet. Point to Point rozhraní pro připojení inteligentních zařízení použitých nejen v automatizaci. V dnešní době tyto sítě obsahují větší výpočetní výkon, tudíž je předpoklad že budou potřebovat i rychlejší a výkonnější komunikaci s řídicím zařízením. IO-link je flexibilní a nezávislý na sběrnici. Je navržen pro práci s tradičními zařízeními, bez zbytečných modifikací, tudíž instalace sítě není nijak složitá a dovoluje kdykoliv přidání nového zařízení a úpravu stávající sítě. [10]

3.7 ControlNet

Tato síť je používána u výrobce Allen-Bradley. Použitá data jsou umístěna na síti bez indikátorů, komu je určena. Data mohou být simultánně použita na více stanicích. Díky tomu je snížen počet zpráv na síti a díky tomu je zvýšena rychlost komunikace. To umožňuje PLC sdílet data rovnoměrně mezi více procesorů, místo toho, aby je posílal pouze jednomu. [2]

3.8 EtherCat

Vysoký výkon, nízká cena, jednoduché použití a flexibilní topologie, to je to, čím se prezentuje EtherCat, který výhradně používá firma Beckhoff pro komunikaci u svých zařízení v průmyslové automatizaci. Představena byla někdy v roce 2003 a v dnešní době se její základna rozšiřuje. Klíčový princip spočívá ve zpracování Ethernet frame a to tak, že EtherCat master odešle datagram, který prochází každým uzlem. Každé zařízení přečte data, která jsou mu adresována a zapisuje data na frame během toho co je frame posílán. Samotný frame nesoucí data je tak opožděn pouze hardwarem. Tento způsob vede ke zlepšení komunikace a využití šířky pásma, navíc i k snížení počtu Switch. [11]



Obr. 24 EtherCat frame [11]

3.9 ModBus

Komunikační protokol vyvinut společností Modicon v roce 1979, nyní Schneider Electric. Tradičně je použit spolu s RS232, RS422, nebo RS485. [14]

Protocol Modbus je sériový protokol strukturován jako master – slave system, který komunikuje pomocí „message“ mezi zařízeními. Tyto „message“ obsahují adresu cílového zařízení, příkaz, data CRC (cyclic redundancy check), což je kontrolní součet pro poslaný paket dat, kdy se tímto součtem potvrdí že poslaná data byli přijaty v stejném formátu jako byli odesláni.

Modbus může být rozdělen na 3 části, a to Modbus RTU, ASCII, TCP.

Modbus RTU je nejvíce využíván z nich, je to sériový protokol. Založený na „master – slave“, master může mít na sebe napojených 254 zařízení a posílat na ně data. Každý „slave“

má svoji 8bitovou adresu, zařízení „slave” odpovídá zařízení “master” pouze když je dotázáno a musí odpovědět v určitém čase „watchdog” jinak PLC vyhlásí chybu komunikace. [14]

RTU je komunikace pouze jednoho zařízení „master” a mnoho „slave” zařízení. Master posílá žádost a dostává odpověď pouze od zařízení, na které se adresuje. Speciální případ je, když pošleme žádost na adresu 0, to je vyslán broadcast telegram na všechny „slave” zařízení. [16]

Modbus ASCII je starší verze protokolu, která obsahuje všechny elementy RTU. Není příliš podporován a ani zdaleka tolik využívaný jako RTU. [14]

Modbus TCP, je v podstatě RTU s TCP interface, který funguje na Ethernetu. Není zde už primární adresa zařízení, ale využívá se již IP adresování a došlo ke změně, není zde již master – slave, ale client – server. Velkou změnou je že zde může být několik klientů. Modbus TCP funguje pomocí “peer-to-peer” (PtP) komunikaci na Ethernet IP. [14]

II. PRAKTICKÁ ČÁST

V praktické části je realizace laboratorních úloh, využívajících panel Weintek MT8041 iP a přidružených typů PLC, které jsou dostupné v rámci laboratoří Univerzity Tomáše Baři na Fakultě aplikované informatiky. Konkrétně PLC Siemens a Saia, ty jsou použity i při praktické výuce v předmětu PLC programování. Dále jejich popsání z hlediska funkčnosti a následného vytvoření manuálu, tak aby čtenář, nebo student měl možnost si zmíněnou aplikaci vytvořit, pokud k tomu bude mít dostupné prostředky.

4 PRVNÍ ÚLOHA

4.1 Zadání

Zadání úlohy je vytvoření programu na řízení motoru dopravníku, který dopravuje zpracovaný díl od stroje k obsluze. Motor se bude zastavovat na konci, kdy dojde na indukční snímač, po odebrání dílu se motor opět rozjede, aby nedocházelo ke zpoždění výroby. Obsluha zvolí směr automatického režimu a nadále nemusí nic dělat, popřípadě bude mít možnost s motorem posunout ručně.

Realizujte propojení daného PLC s operátorským panelem a následně vytvořte program v PLC, pomocí kterého budete ovládat připojený motor.

K dispozici budete mít indukční snímač, pomocí kterého budete zastavovat motor.

Veškeré ovládání bude řízené z panelu, kde bude zobrazena přítomnost dílu, popřípadě alarmové hlášení.

Prvky ovládání z panelu: budeme demonstrovat reálné zařízení, tudíž zapínání zdroje na rozvaděči, zapnutí motoru, central stop a následně ovládací funkce které budou na panelu, jako volba směru pohybu.

Během práce nebude použit žádný dopravník, zadání je pouze simulační.

4.2 Použité součásti



Obr. 25 PLC S7-1200 CPU 1212C AC/DC/RLY



Obr. 26 Operátorský panel MT8051iP



Obr. 27 Frekvenční měnič Sinamics V20



Obr. 28 tři fázový motor



Obr. 29 Indukční snímač

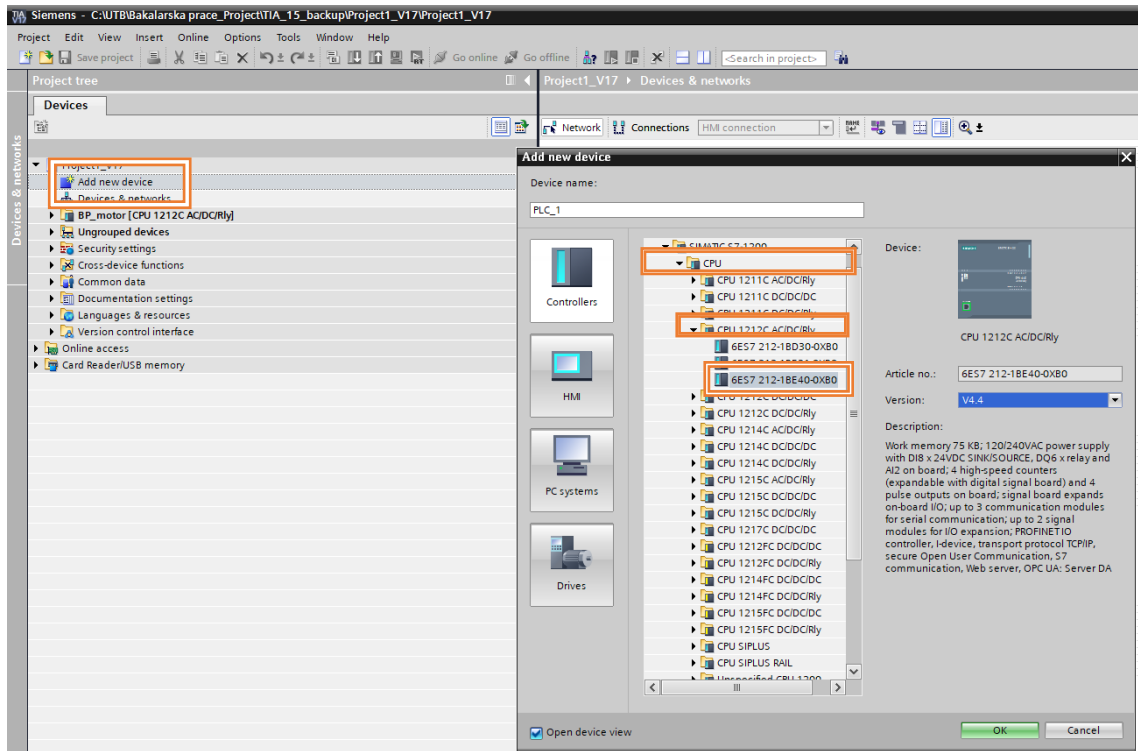
4.3 Realizace úlohy

4.3.1 Vytvoření Hardware konfigurace

Nejdříve si vytvoříme program pro PLC v programu TIA portálu. Po otevření programu si jako první musíme nahrát hardwarovou část, tudíž PLC, se kterým budeme pracovat. Podle označení, které je na PLC si najdeme totožné zařízení v TIA.

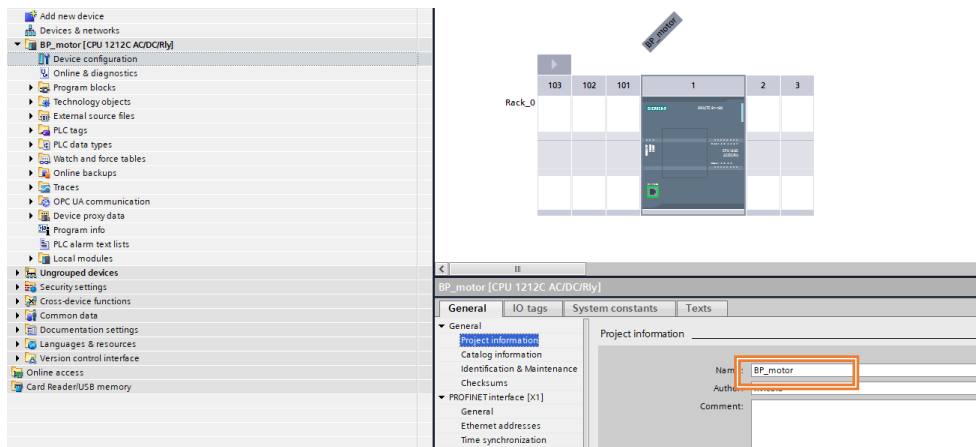


Obr. 30 Značení PLC

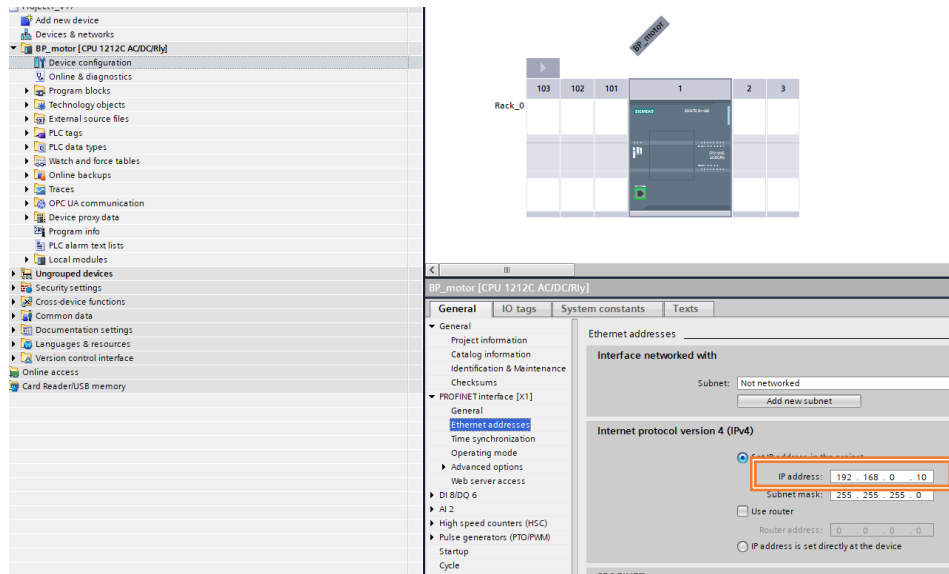


Obr. 31 Naznačené části postupu výběru

Jakmile máme vybrán typ PLC a výběr je potvrzen, automaticky se vytvoří složka v strojové topologii projektu. Poté si otevřeme konfiguraci zařízení, přiřadíme jméno projektu a IP adresu.



Obr. 32 Přiřazení jména PLC



Obr. 33 Přiřazení IP adresy

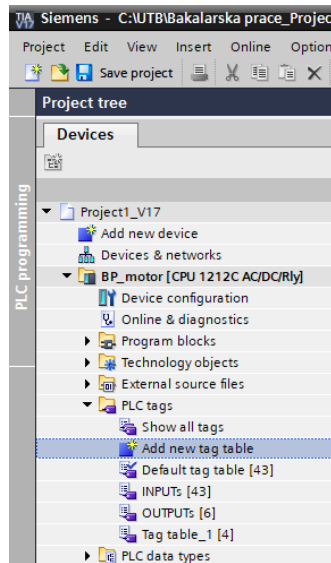
4.3.2 Vytvoření I/O

V normálním případě je programování vždy podle zadání od zákazníka a podle elektro projektu, kde máme zobrazené hardwarovou část a připojené I/O zařízení.

V našem případě nemáme elektro projekt, tudíž volba vstupů je na nás. Tudíž podle zadaných informací si zvolíme potřebné vstupy a výstupy. V tomto případě budeme mít pouze jeden vstupní signál, a to od snímače. Budeme ovládat 4 výstupy, které budou navedeny na frekvenční měnič. Takže budeme mít 1 digital input a 4 digital output, větší množství bude memory bitů, které použijeme na ovládání z panelu.

Pro správné zvolení memory bitů, si musíme představit, jak bude dané zařízení pracovat, tzv.co budeme chtít mít na panelu, jaké tlačítka použijeme, a podobně.

Ve většině případů si memory bity doplňujeme postupně při práci, jelikož nemusí mít každý úplnou představu, jak bude dané zařízení pracovat a co vše bude mít na panelu. Memory bit ve formě bool, je uživatelem definované místo v paměti, kterou si nastavujeme pouze programově na hodnotu 1 nebo 0, naopak na digitálním vstupu dostaneme 1 v návaznosti na reakci I/O zařízení. Nejdříve si ale musíme vytvořit soubor pro umístění tagů, to je naznačeno na obrázku níže.



Obr. 34 Vložení nového souboru pro tagy

Definujeme si vstupy a výstupy, se kterými budeme pracovat. V tomto případě nemáme definované názvosloví I/O podle projektu, tudíž názvy si určíme tak, abychom se v tom sami orientovali.

Tag table_1									
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment	
1	a_vpřed(1)	Bool	%M10.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
2	a_vzad(1)	Bool	%M10.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	m_vpřed(1)	Bool	%M10.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	m_vzad(1)	Bool	%M10.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	ESTOP(1)	Bool	%M10.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
6	power ON/OFF(1)	Bool	%M11.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
7	motor_ON/OFF(1)	Bool	%M11.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
8	snimac(1)	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
9	Motor_VPRED(1)	Bool	%M40.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
10	Motor_VZAD(1)	Bool	%M40.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
11	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

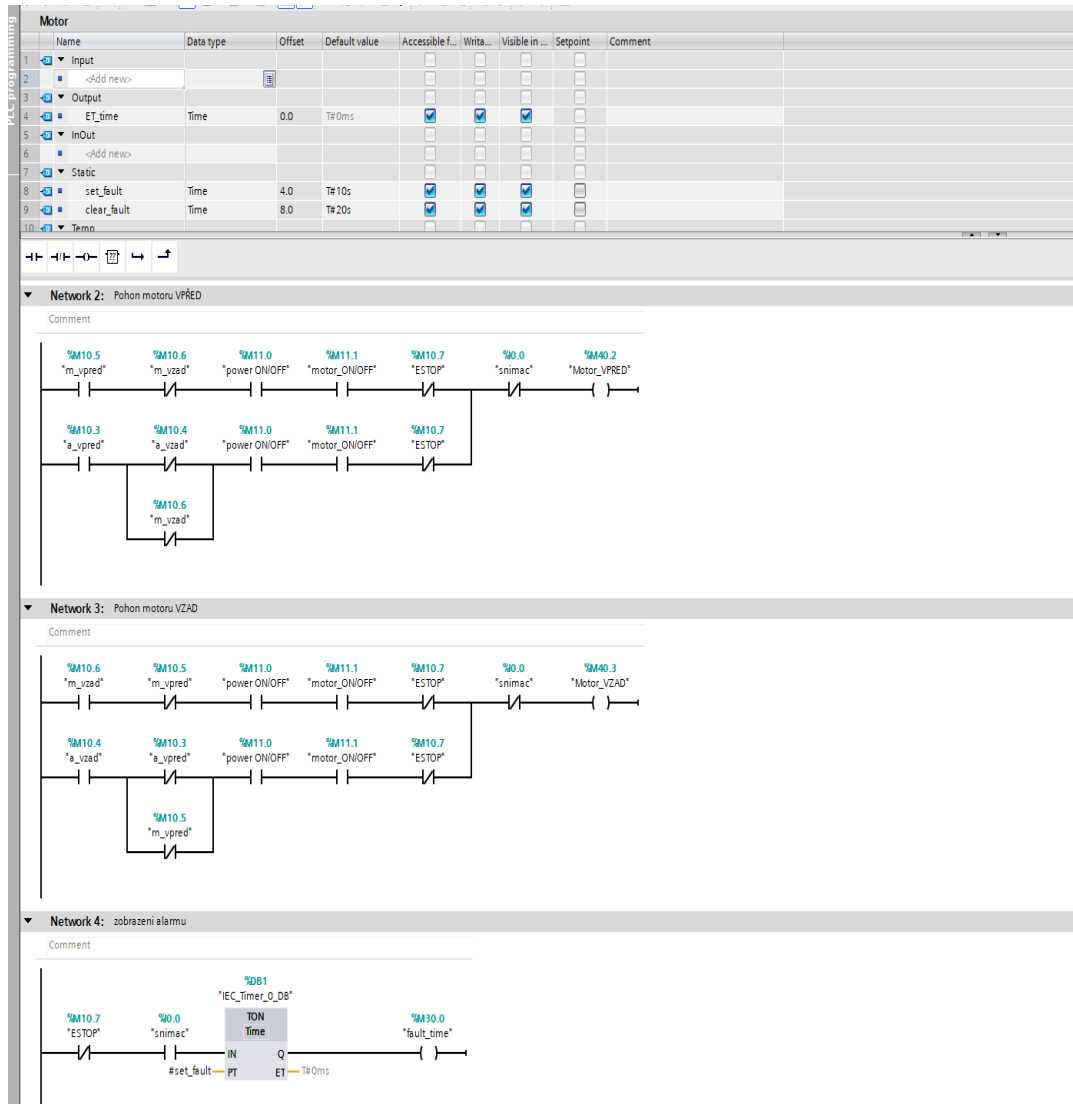
Obr. 35 Seznam použitých vstupů

Tag table_1									
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comme	
1	OB_Motor_1(1)	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
2	OB_Motor_2(1)	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	OB_Motor_0(1)	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
4	OB_Motor_3(1)	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
5	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Obr. 36 Seznam použitých výstupů

4.3.3 Program

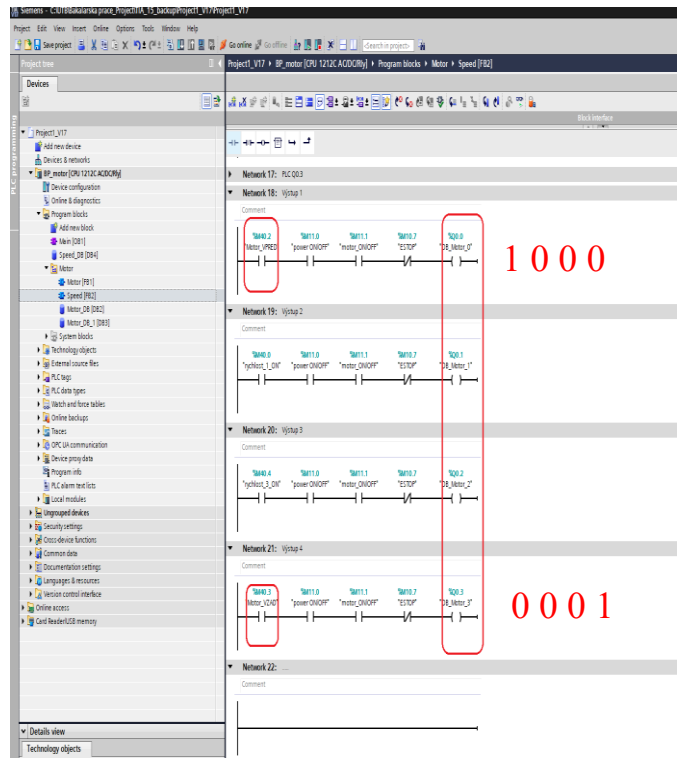
Teď když máme nastavenou konfiguraci a máme vytvořené I/O tagy, tak už můžeme začít s psáním programu. Pro naši aplikaci se nejedná o nic složitého, v podstatě na pár řádků. Pro vytvoření programu jsem zvolil jazyk LAD, ve kterém se jednoduché aplikace dělají nejsnáze a přehledně se vidí probíhající změny. Po složení podmínek, které musí splňovat zařízení pro pohyb jde vidět, že se nejedná o nic složitého. Pokud si představíme zařízení funkčně, tak ani složení podmínek není problém. Program je zpracován jednou z možností. Je více možností, jak mít program vytvořený, tudíž zkušenější může provedení vytvořit jinak, s více možnostmi.



Obr. 37 Schéma programu

Logika programu není nijak složitá, navíc jsem přidal funkci, pokud obsluha neodebere kus po určitém čase, tak nastane chyba, která bude zobrazena na panelu. Teď už jen musíme nastavit ovládání na výstupy. Motor je řízen bitově, tudíž posláním hodnoty 1 na určitý vstup budeme řídit motor vpřed/vzad.

V našem případě budeme motor řídit frekvencí 7 Hz a -7 Hz, a to bitovou kombinací (1000) a (0001), kterou přivedeme na výstupy.



Obr. 38 Propis na výstupy

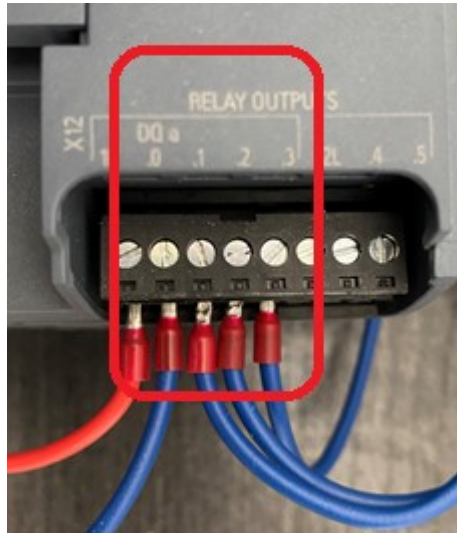
Na obrázku x. vidíme část programu, kde námi ovládané memory bity %M40.2 a %M40.3 už po splnění podmínek ovládají příslušné DO, které máme napojené na FM.

4.3.4 Bitová kombinace

%Q0.0	%Q0.1	%Q0.2	%Q0.3	Hz
1	0	0	0	7
0	0	0	1	-7

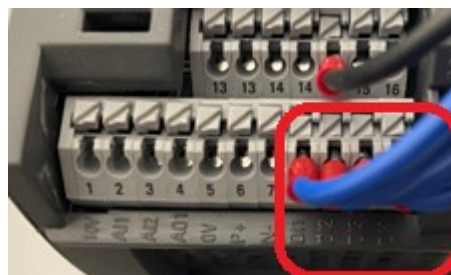
Tabulka 2 Dvou bitová tabulka

V předchozí tabulce vidíme informaci o tom, na které výstupy posíláme hodnotu 1, tak podle toho dostáváme frekvenci, která byla nastavena na měniči. Pro první úlohu použijeme kombinaci podle tabulek.



Obr. 39 Fyzické výstupy z PLC

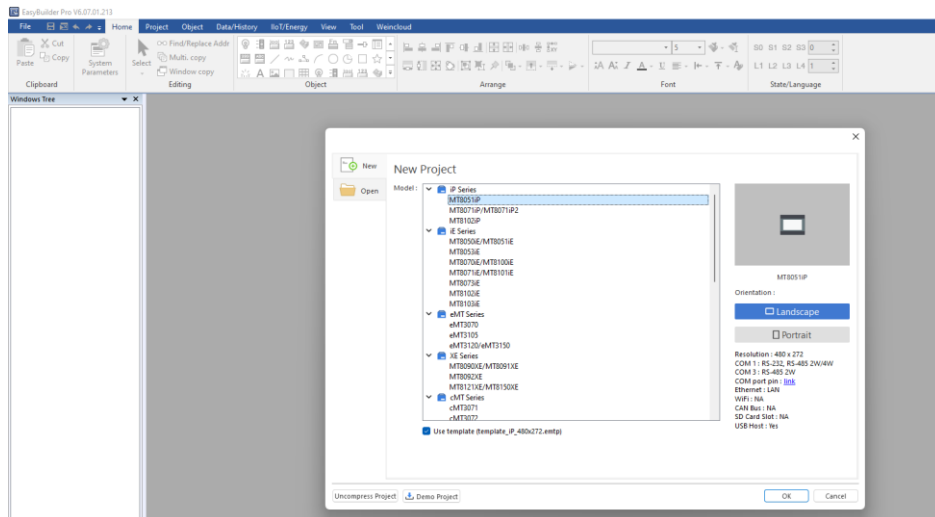
Tyto DI jsou navedeny na FM, tak aby byli propojeny na stejné místo, tedy DQ 0 na DI 1 a další 3 vodiče stejným způsobem. Nesedí to akorát číselně, jelikož PLC značí první I/O od 0 a FM od 1.



Obr. 40 Propojení výstupů z PLC

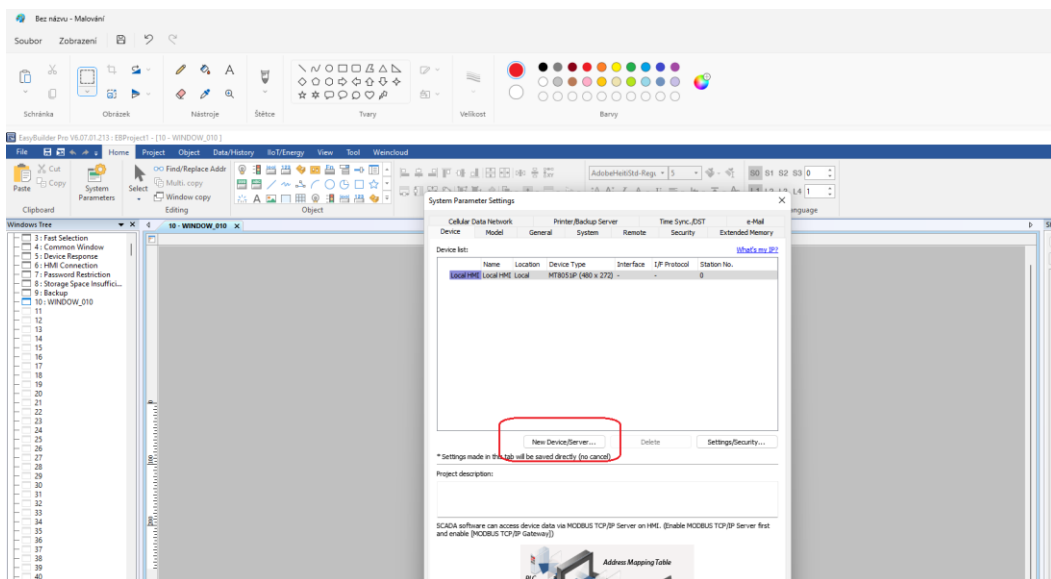
4.4 Vizualizace

Vizualizace bude probíhat v programu Easy Builder Pro od firmy Weintek, který je zdarma ke stažení po registraci. Ovládání programu je velmi intuitivní a snadné i pro nového uživatele s tímto programem.



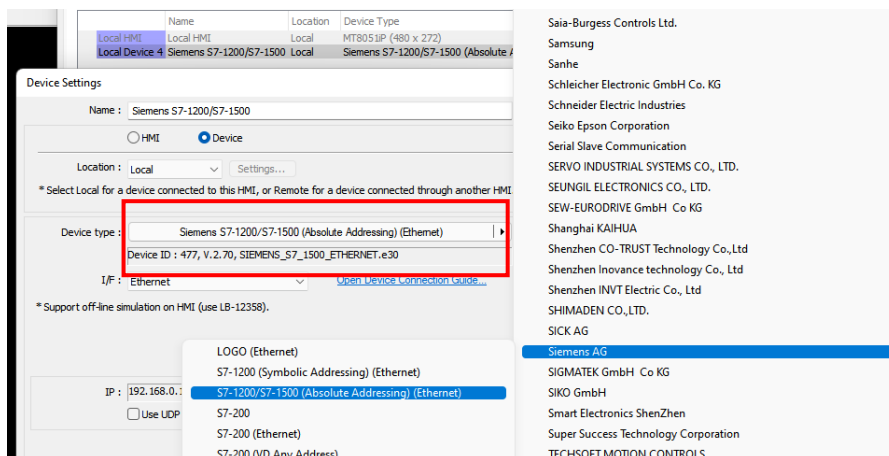
Obr. 41 Nový projekt

Po otevření programu, zvolíme volbu nový projekt a vybereme panel, který máme k dispozici. Zvolený panel se načte do programu. Teď je potřeba připojit řídicí PLC, v nastavení parametrů.



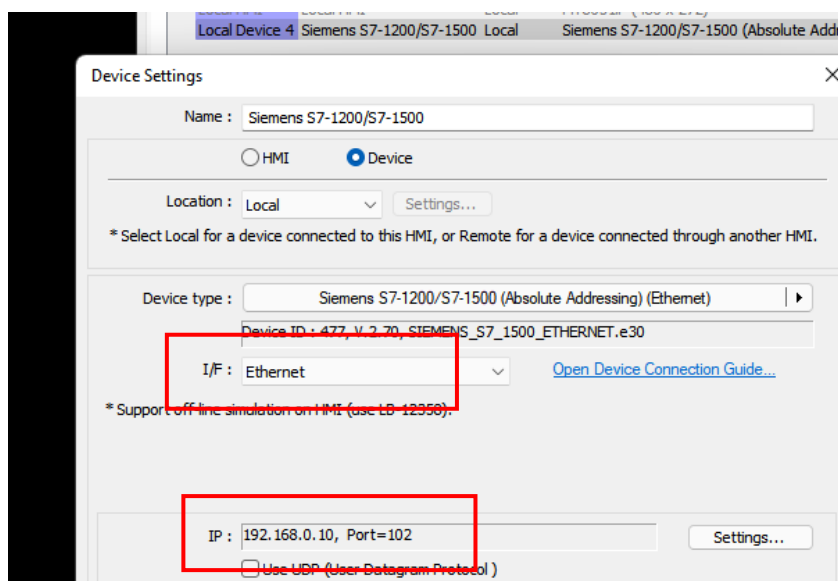
Obr. 42 Nové zařízení

Zvolíme typ PLC, které máme v sestavě, se kterým bude probíhat komunikace a který bude řídicí orgán.



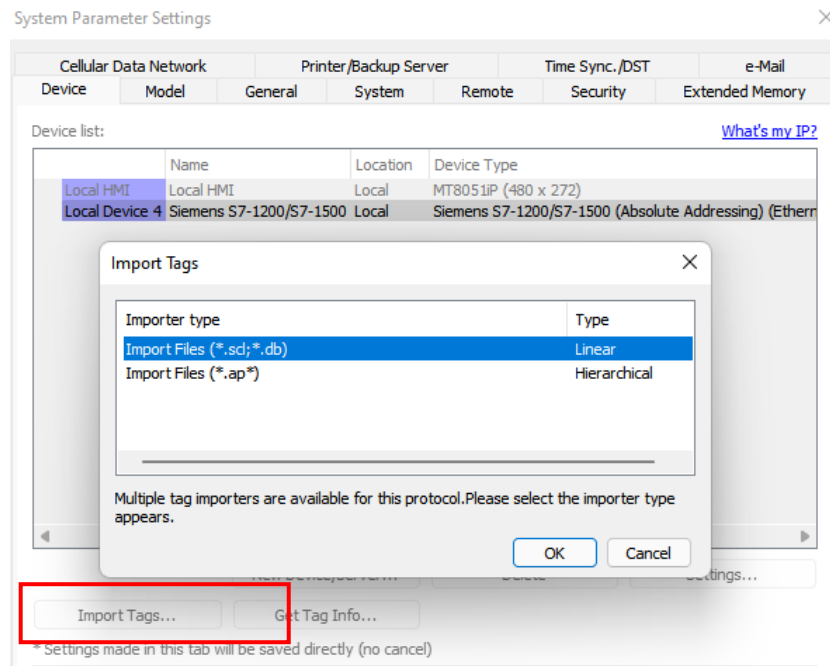
Obr. 43 Přiřazení PLC

Nastavíme druh komunikace pomocí Ethernetu a IP adresu PLC.

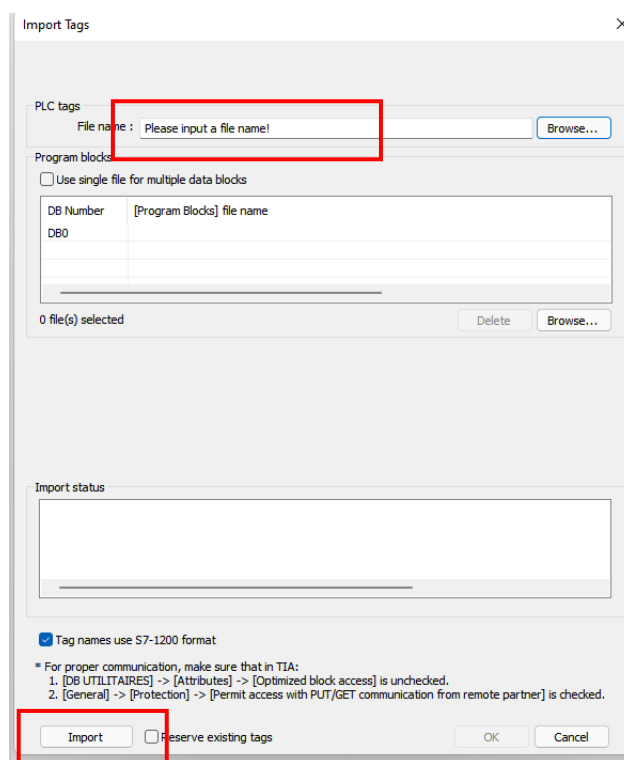


Obr. 44 Adresa zařízení

Teď když máme přiřazené PLC a máme nový projekt, musíme se dostat k tagům které jsme vytvořili v PLC, tak že je importujeme z programu PLC do panelu, který bude propojený s PLC, abychom mohli ovládat program přes panel.

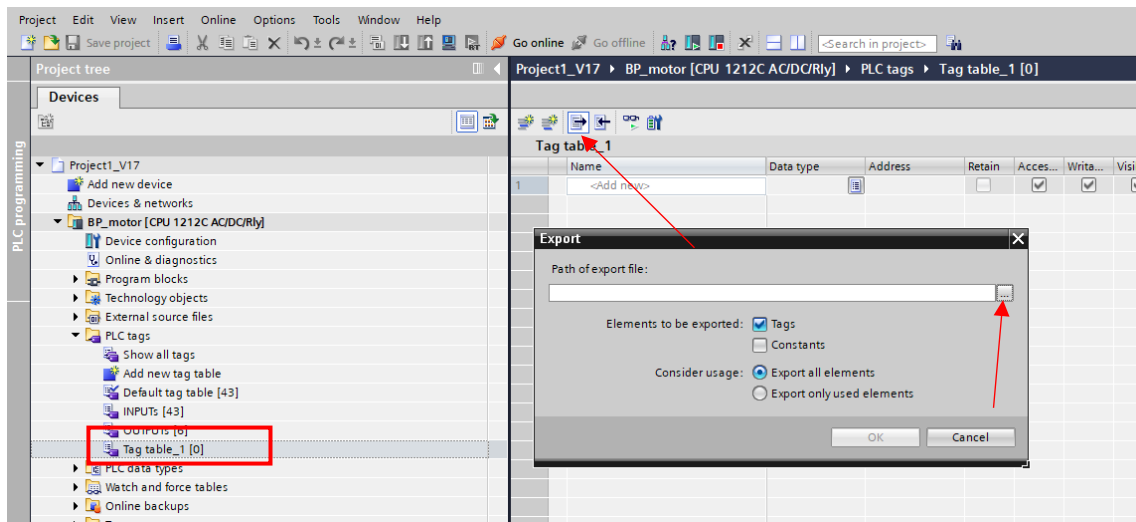


Obr. 45 Import tagů



Obr. 46 Import

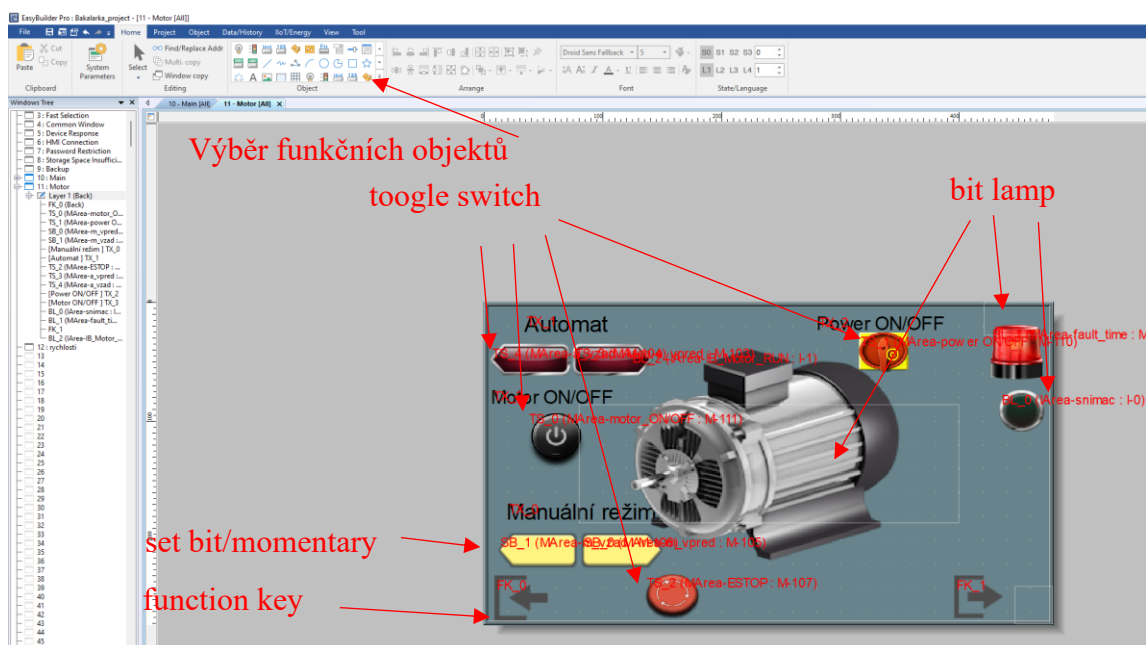
Vyhledáte soubor s tagy, který exportujete z TIA portálu ve formátu .xlsx a následně jej importujete do projektu.



Obr. 47 Postup exportu tagů z TIA portálu

Teď když máme nahrané tagy do prostředí pro tvoření vizualizace, můžeme začít s tvoření obrazovek. Vytvoříme jednoduchou vizualizaci, pomocí vložení pár tlačítek, tak jako bychom realizovaly v reálném prostředí.

ON/OFF přívod proudu, ON/OFF motor, volba směru v automatu/manuál a bezpečnostní tlačítko, navíc dvě kontrolky, které nám signalizují přítomnost kusu na snímači a alarmový stav.



Obr. 48 Screen ovládání motoru

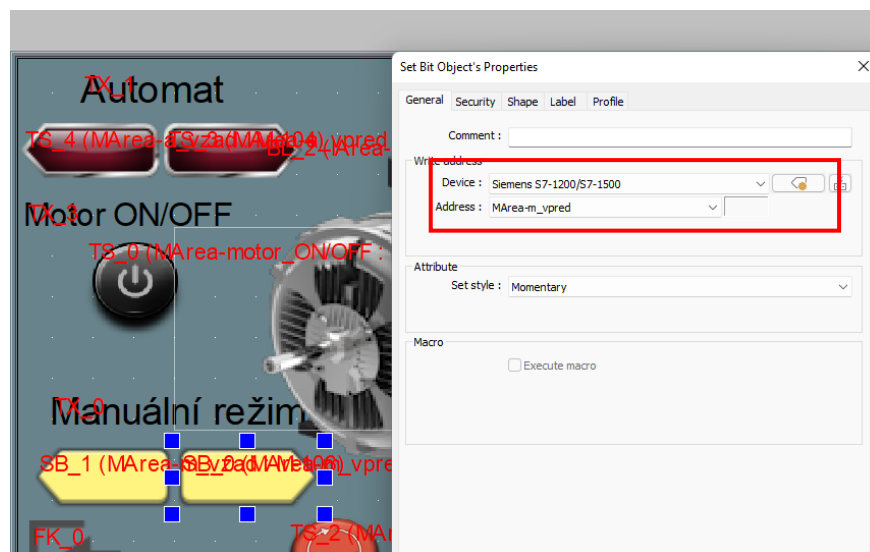
Nastavujeme jednotlivá tlačítka podle funkce programu. Využijeme přitom následující tlačítka:

Toogle switch: dvoustavové tlačítko, které nabývá hodnoty 1 při stisknutí, po opětovném stisknutí se nastaví na hodnotu 0.

Set bit: tlačítko je opatřeno atributem „momentary“, tudíž hodnoty 1 nabývá pouze při držení tlačítka, při povolení se nastaví na 0.

Function key: slouží k přesouvání mezi screeny.

Bit lamp: toto už neslouží jako tlačítko, ale k zobrazení stavu příslušného bitu.



Obr. 49 Přiřazování adres tagů

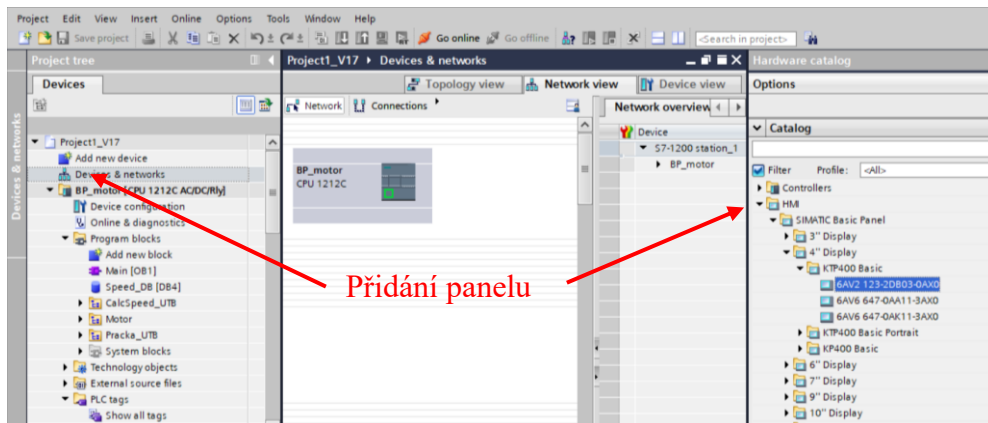


Obr. 50 Lišta projektu

Po dokončení vizualizace screenu a propojení tagů, v horní liště, pod tlačítkem „project“, provedeme kompilaci projektu a následně jeho nahrání do HMI, pokud nejsou zobrazeny žádné errorové hlášky.

4.5 Vizualizace první úlohy na panelu Siemens

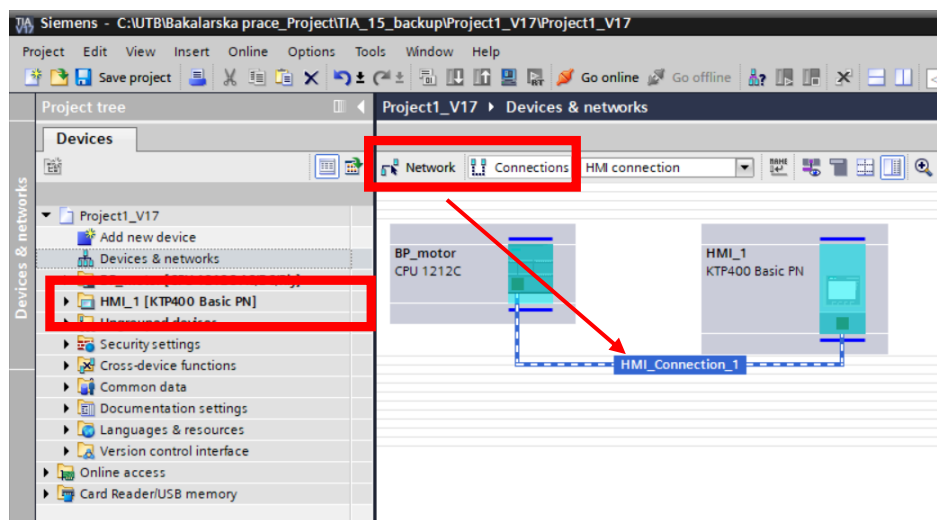
Pro názornost vytvořím, jak by vypadalo řešení vizualizace první úlohy na panelu KTP 400 Basic od značky Siemens v prostředí TIA portálu.



Obr. 51 Přidání panelu

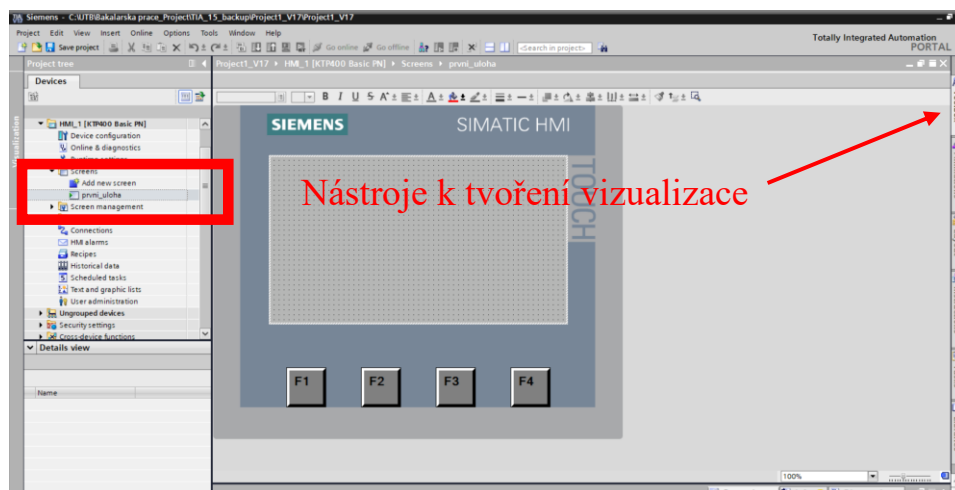
Pod záložkou HMI si vyberu panel, na kterém bude vizualizace realizována a přidám jej do hardwarové konfigurace, vybraný model jednoduše přetáhneme vedle PLC.

Jakmile přetáhneme panel do prostředí, musíme propojit HMI s panelem, kvůli propojení komunikace a automaticky se nám vytvoří složka pro panel, ve které si posléze vytvoříme vizualizaci. Navíc jakmile panel softwarově propojíme s PLC, je mu automaticky udělena IP adresa.



Obr. 52 Připojení panelu

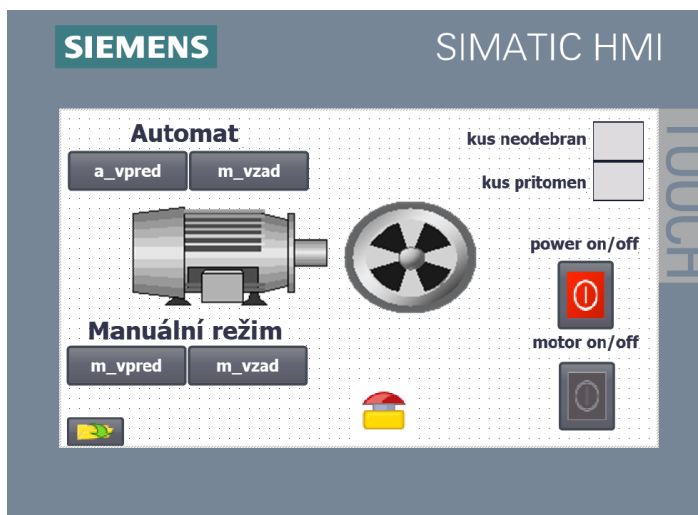
Přidám nový screen a vytvořím si vizualizaci pro první úlohu.



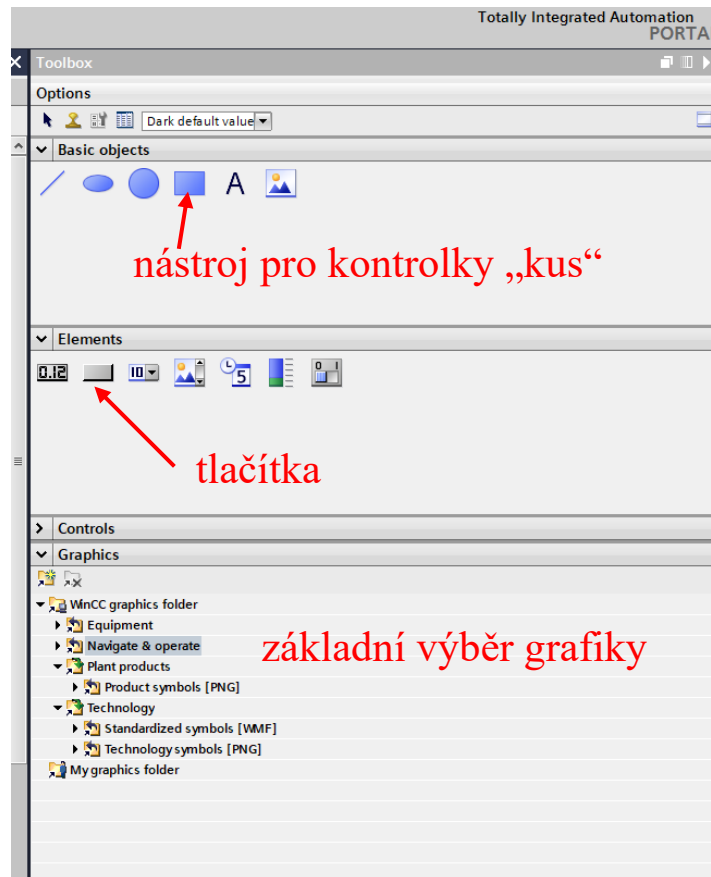
Obr. 53 Přidání screenu

Následně vytvořím objekty, které budou podobné vizualizaci na panelu Weintek. Každé prostředí pro vizualizaci má svoje pro a proti, tudíž vizualizace nebudou úplně podobné. Vizualizace v panelu Weintek nabízí vzhledově příjemnější možnosti pro tlačítka již v standardním nastavení, u Siemens, pokud chcete hezčí vzhled, musíme si obrázek stáhnout a přiřadit do vaší knihovny obrázků, které potom použijete pro vzhled tlačítka.

V této vizualizaci jsem vytvořil tlačítka ze základního výběru.



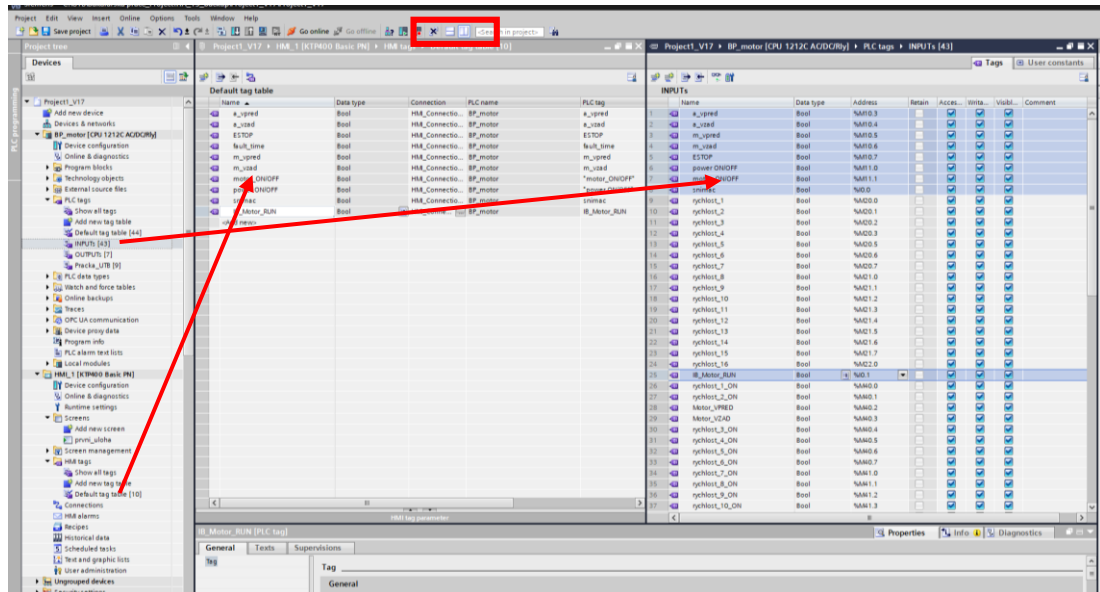
Obr. 54 Vizualizace v prostředí TIA portál



Obr. 55 Základní nástroje pro vizualizaci

Když máme vytvořenou vizualizaci, tak jak to bylo u panelu Weintek, musíme i zde jednotlivých tlačítkům a dalším objektům připojit tagy se kterými budeme pracovat. Tím že vizualizace je ve stejném prostředí jako program, je celý proces snazší na propojení tagů.

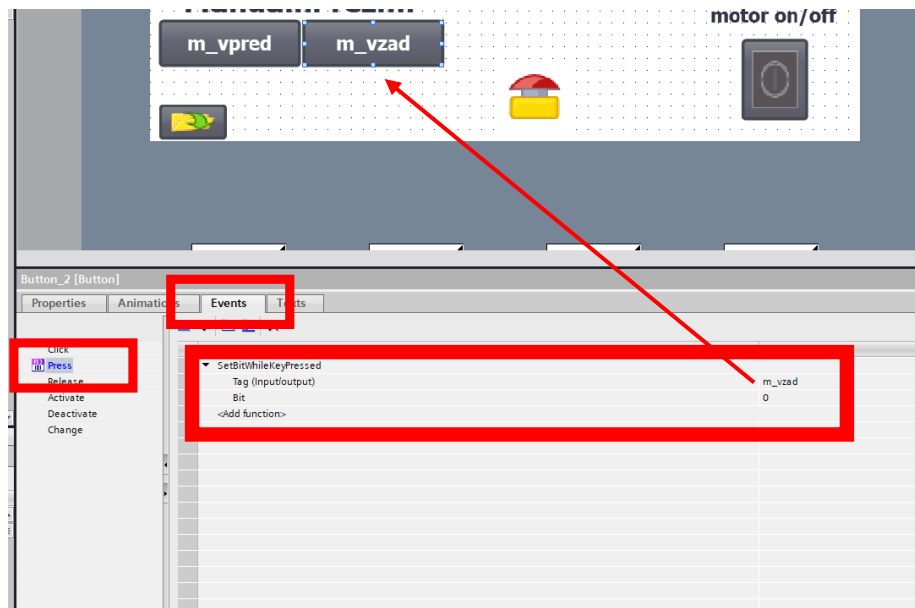
Ve složce „HMI tags“ vytvoříme nový soubor pro tagy a následně do něj připojíme tagy z PLC, jelikož HMI nepracuje s tagy, které jsou pro PLC, musí se převést na HMI tagy. V horním panelu si mohou rozdělit pracovní obrazovku prostředí na dvě, což nám ulehčí často práci, a právě tyto přesuny. Otevřeme si nový soubor pro HMI tagy a již vytvořené tagy v PLC, potom jednoduše vybereme tagy, které chce použít v panelu, označíme je a přesuneme do nového souboru.



Obr. 56 Přesun tagů

Připojení vytvořených tagů k jednotlivým tlačítkům je také velice jednoduché. Klikneme na tlačítko, které chceme upravovat a zvolíme možnost „Event“ a následně vybereme možnost, jak bude tlačítko fungovat. Pro tlačítka „m_vpřed“ a „m_vzad“ víme že jsme přiřadili funkci, že se motor bude točit pouze při stisknutém tlačítku.

Na levé straně máme na výběr z možností, jak bude tlačítko na panelu fungovat při stisknutí. Vybereme možnost „Press“ tudíž svoji funkci vykoná, jakmile je tlačítko zmáčknuto. Na pravé straně při kliknutí na šipku se nám otevře rozbalovací menu funkcí tlačítek, z tohoto výběru vybereme funkci tlačítka „SetBitWhileKeyPressed“, tudíž tlačítko změní svoji hodnotu na 1, když je tlačítko zmáčknuto, po uvolnění se vrátí hodnota na 0 a přiřadíme „tag“ se kterým pracuje.

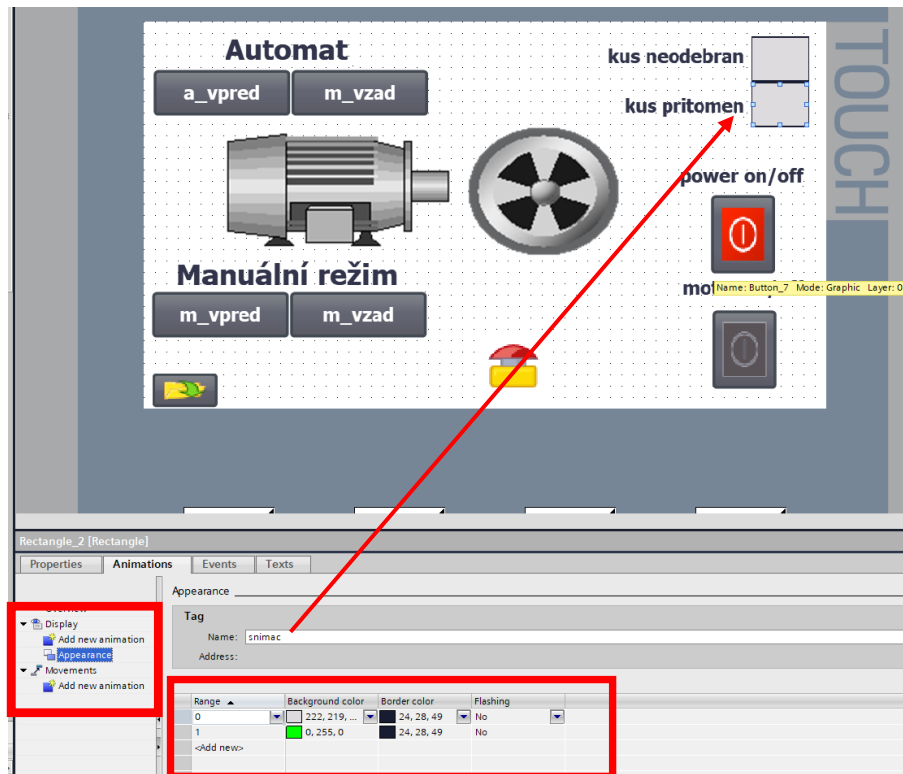


Obr. 57 Přřazení funkce tlačítka

Stejným způsobem nastavíme i ostatní tlačítka, s tou výjimkou že ostatní tlačítka mají funkci „SetBit“, tudíž při zmáčknutí tlačítka, změní svoji hodnotu bez toho abychom tlačítko museli držet, bude fungovat jako přepínač.

Pro funkci, která nám bude zobrazovat přítomnost kusu, si vytvoříme objekt, kterému následně vytvoříme funkci, díky které bude měnit barvu podle hodnoty připojeného tagu.

Tentokrát vybereme funkci „Animation“ a vytvoříme novou „Apperance“, připojíme tag a nastavíme mu barvy pro danou hodnotu.



Obr. 58 Objekt kus přítomen

Po vytvoření vizualizace a propojení tlačítek, celý projekt zkompilujeme a pokud nejsou zobrazeny žádné chyby, projekt nahrajeme do PLC a do HMI.

5 DRUHÁ ÚLOHA

5.1 Zadání

V druhé úloze rozšířte program z úlohy 1 o možnost měnit rychlost motoru.

5.2 Realizace

Využijeme stávajícího programu a doplníme program v PLC o příslušné tagy navíc a vytvoříme si nový screen na ovládání motoru a změnu rychlosti. Využijeme bitové kombinace na řízení motoru pomocí FM.

%Q0.0	%Q0.1	%Q0.2	%Q0.3	Hz	
0	0	0	0	0	VPŘED
1	0	0	0	7	
0	1	0	0	14	
1	1	0	0	21	
0	0	1	0	28	
1	0	1	0	35	
0	1	1	0	42	
1	1	1	0	50	
1	1	1	1	0	
0	0	0	1	-7	
1	0	0	1	-14	
0	1	1	1	-24	
0	1	0	1	-28	
1	1	0	1	-35	
0	0	1	1	-42	
1	0	1	1	-50	

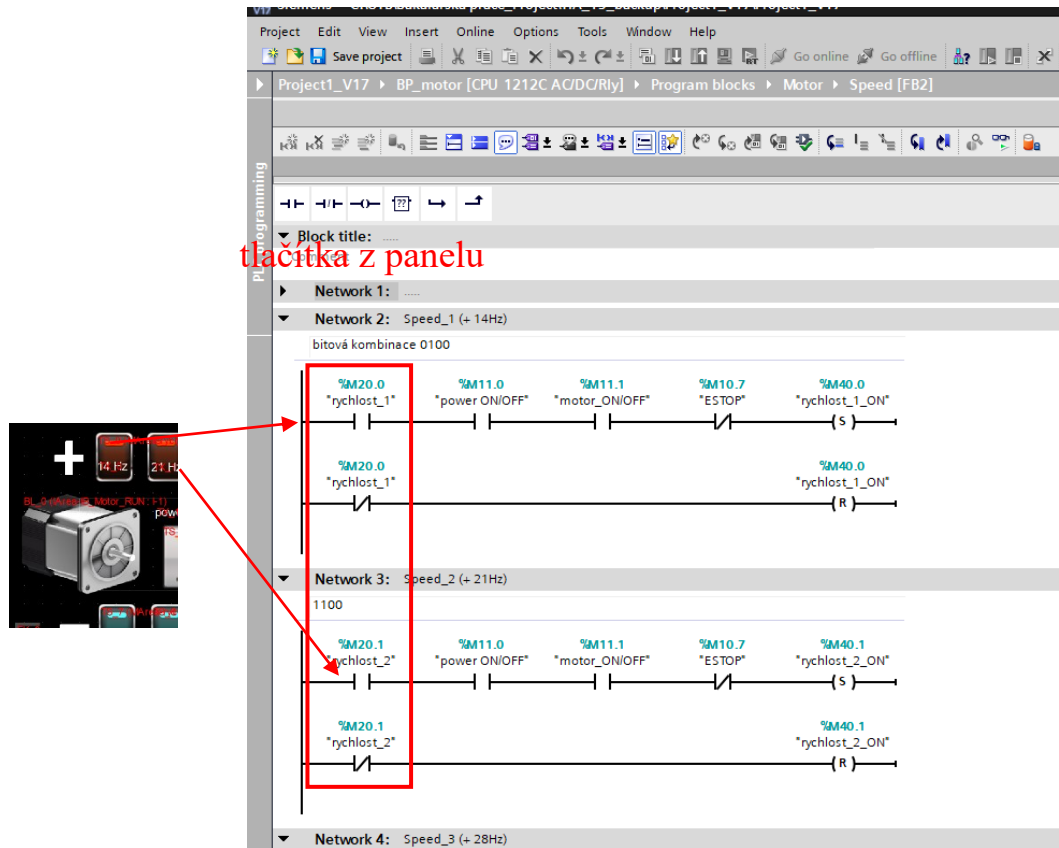
Tabulka 3 Tabulka bitová kombinace

V TIA portálu si vytvoříme nové memory tagy a doplníme program. Vytvoříme si nový funkční blok a v něm si navedeme nové memory bity, které budeme napojovat na tlačítka v panelu, tak abychom jimi ovládali příslušné výstupy.

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	a_vpřed	Bool	%M10.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	a_vzad	Bool	%M10.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	m_vpřed	Bool	%M10.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	m_vzad	Bool	%M10.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	ESTOP	Bool	%M10.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	power ON/OFF	Bool	%M11.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	motor_ON/OFF	Bool	%M11.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	snimac	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	rychlost_1	Bool	%M20.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	rychlost_2	Bool	%M20.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	rychlost_3	Bool	%M20.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	rychlost_4	Bool	%M20.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	rychlost_5	Bool	%M20.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	rychlost_6	Bool	%M20.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	rychlost_7	Bool	%M20.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	rychlost_8	Bool	%M21.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	rychlost_9	Bool	%M21.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	rychlost_10	Bool	%M21.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	rychlost_11	Bool	%M21.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	rychlost_12	Bool	%M21.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	rychlost_13	Bool	%M21.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	rychlost_14	Bool	%M21.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	rychlost_15	Bool	%M21.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
24	rychlost_16	Bool	%M22.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
25	IB_Motor_RUN	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
26	rychlost_1_ON	Bool	%M40.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
27	rychlost_2_ON	Bool	%M40.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
28	Motor_VPRED	Bool	%M40.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
29	Motor_VZAD	Bool	%M40.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
30	rychlost_3_ON	Bool	%M40.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
31	rychlost_4_ON	Bool	%M40.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
32	rychlost_5_ON	Bool	%M40.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
33	rychlost_6_ON	Bool	%M40.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
34	rychlost_7_ON	Bool	%M41.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
35	rychlost_8_ON	Bool	%M41.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
36	rychlost_9_ON	Bool	%M41.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
37	rychlost_10_ON	Bool	%M41.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
38	rychlost_11_ON	Bool	%M41.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
39	rychlost_12_ON	Bool	%M41.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
40	rychlost_13_ON	Bool	%M41.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
41	rychlost_14_ON	Bool	%M41.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
42	rychlost_15_ON	Bool	%M42.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
43	<Add new>			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Obr. 59 Rozšíření tagů

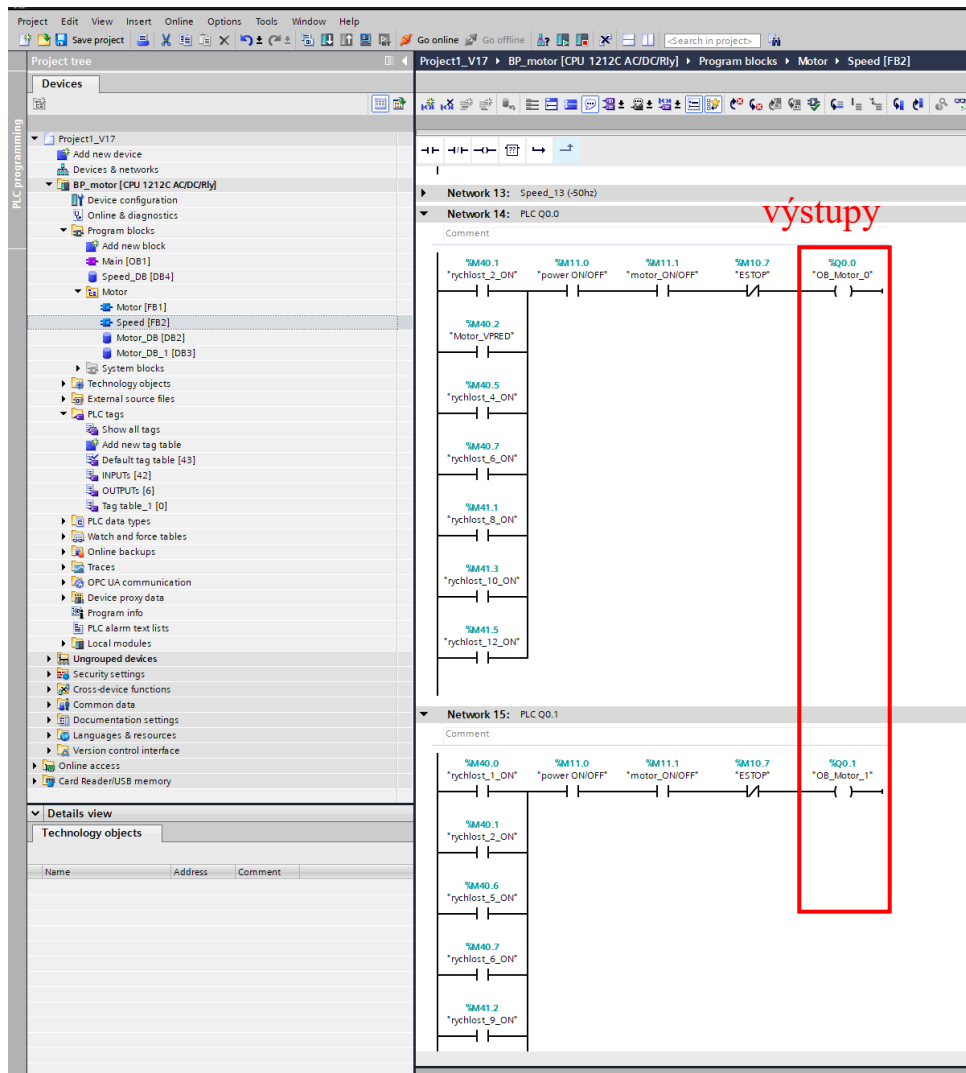
Tagy máme rozšířené a teď musíme doplnit program, použijeme funkční blok, který již máme a rozšíříme jej. Pro modelovou aplikaci a pochopení, jak systém funguje, opět nebudeme tvořit nic složitého. Tím že jsme v první úloze již použili bitovou kombinaci pro frekvenci 7 Hz a -7 Hz, tak tuto kombinaci již nepoužijeme.



tláčítka z panelu

Obr. 60 Tlačítka pro rychlost

Použitím memory bitů %M20.0 a dále, které můžeme vidět na obrázku 59, které budou ovládané z panelu, budeme opět setovat memory bit, který už ale bude použit na ovládání výstupů. Postupně rozšíříme program pro použití dalších tlačítek.



Obr. 61 Rozšíření programu

Po dokončení rozšíření v TIA portálu, rozšíříme vizualizaci. Přidáme nový screen pro tlačítka na ovládání rychlosti, tak abychom měli tlačítka pro plusovou frekvenci u sebe a zápornou také.



Obr. 62 Screen pro manipulaci rychlosti

Stejným postupem jako jsme tvořili screen pro ovládání motoru, vytvoříme i tento pro změny rychlosti.

Předtím než ale začneme, musíme nejprve opět exportovat tagy z TIA portálu, jelikož jsme vytvořili nové a importujeme je do Easy Builderu, stejným způsobem jako byl uveden dříve.

Jmile je vizualizace hotova a všechny objekty jsou napojeny na příslušné tagy, tak vizualizaci zkompilujeme a nahrajeme do HMI.

6 TŘETÍ ÚLOHA

6.1 Zadání

Pro tuto úlohu budeme mít výukové zadání z předmětu Programování PLC a to na ovládání pračky.

1. Pračku spustíme tlačítkem „start“
2. Po stisknutí tlačítka „start“ bude pračka vykonávat následující činnosti:
 - a) 3s – napouštění pro předpírku 0:00 – 0:03s
 - b) 10s – předpírka 0:03 – 0:13
 - c) 3s – vypouštění po předpírce 0:13 – 0:16
 - d) 3s – napouštění pro hlavní praní 0:16 – 0:19
 - e) 15s – hlavní praní 0:19 – 0:34
 - f) 3s – vypouštění po hlavním praní 0:34 – 0:37

Konec cyklu.

6.2 Realizace

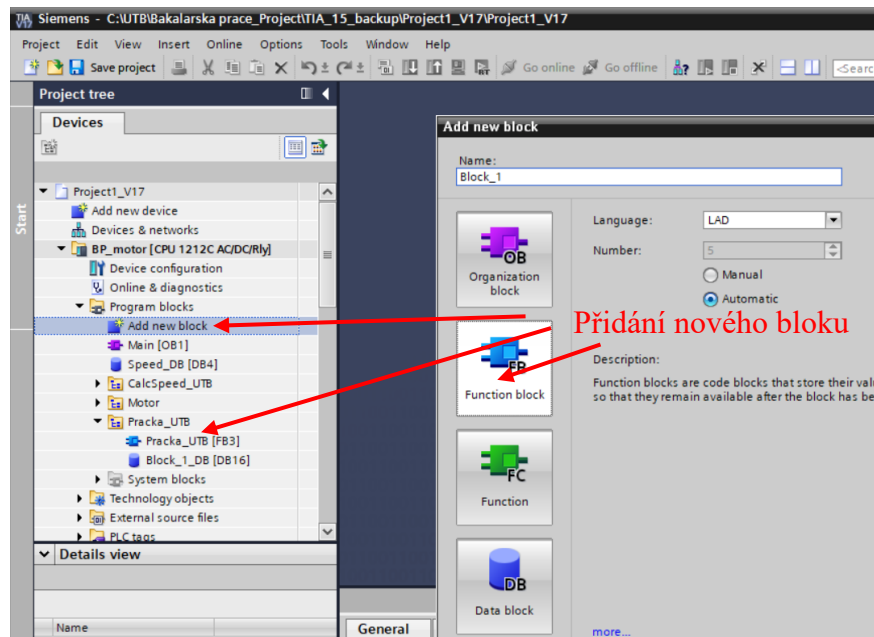
Zvolenou úlohu jde řešit mnoha způsoby, v této práci je vypracován jeden z těchto způsobů, tudíž kreativě se meze nekladou a každý si může řešení upravit podle sebe. Jelikož každý programátor má jiný styl a jiné zkušenosti. Zvolené řešení je dostatečně jednoduché i pro výukové předměty. Opět využijeme PLC Siemens CPU 1212C AC/DC/RLY a panelu MT 8051 iP.

Nejdříve je ukázáno řešení v prostředí TIA portálu a poté vizualizace.

6.2.1 Program

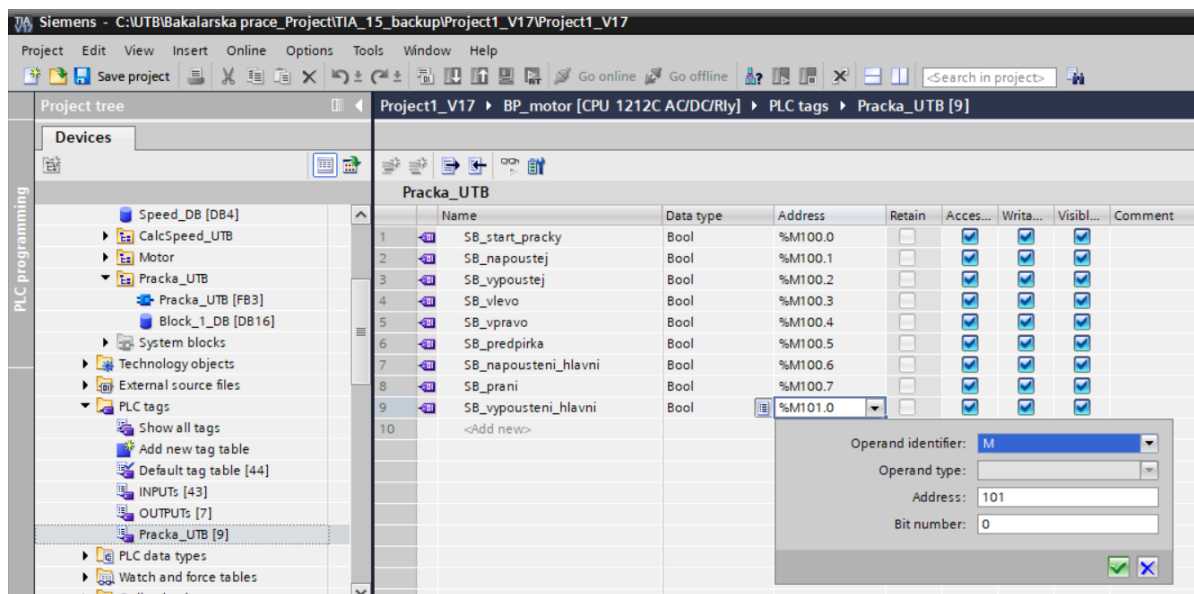
Ve stejném programu jako jsou vypracovány předchozí úlohy, tak vypracujeme i stávající úlohu na pračku.

Vytvoříme si novou složku a do ní si vložíme nový funkční blok, ve kterém vytvoříme logiku pro tento program, taktéž musíme vytvořit memory tagy, kterými budeme řídit program z panelu.



Obr. 63 Přidání nového FB

Jakmile máme vytvořený nový funkční blok a přidán do nové složky, vytvoříme si nové tagy, pro které si také vytvoříme novou složku, ať máme úlohy oddělené, kvůli lepší přehlednosti.



Obr. 64 Tagy pro program pračka

Přidáme si novou složku pro tagy a vytvoříme si zde několik tagů, tak abychom mohli ovládat program. Zvolené tagy jsou pojmenované podle jednotlivých funkcí ze zadání a počátek adresování jsem si nastavil od %M100.0.

Využijí funkce časovačů TON a funkce SET/RESET a dále postupují podle zadání, zpracovaná verze řešení je z logického řešení jednoduchá pro realizaci.

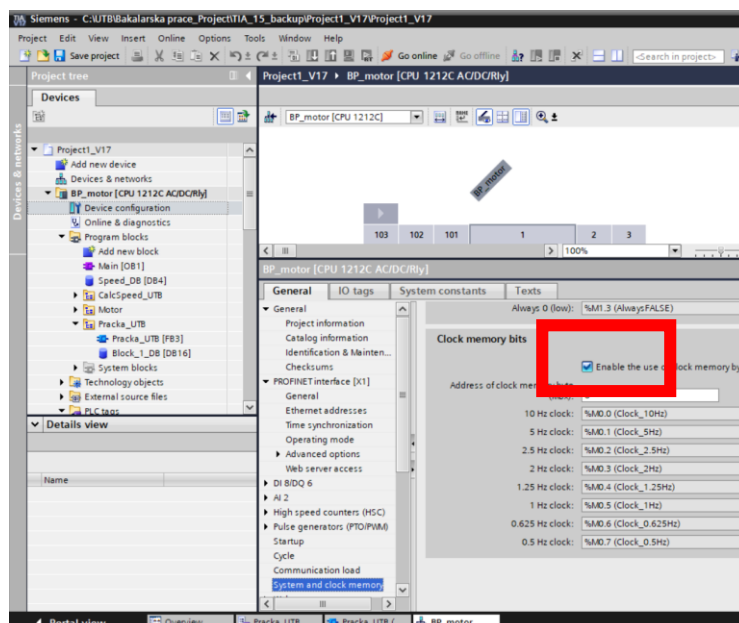
6.2.2 Postup

Tlačítkem start si funkci set nastaví bit SB_napoustej, který mi díky funkci set bude držet hodnotu 1, dokud nebude příslušný bit resetován funkcí reset. Podle zadání víme že prvotní napouštění trvá 3s. Vytvoříme si časovač TON a přidělíme mu datový blok formátu multi-instance. Jeho hodnota PT (preset time) bude 3s, podle zadání.

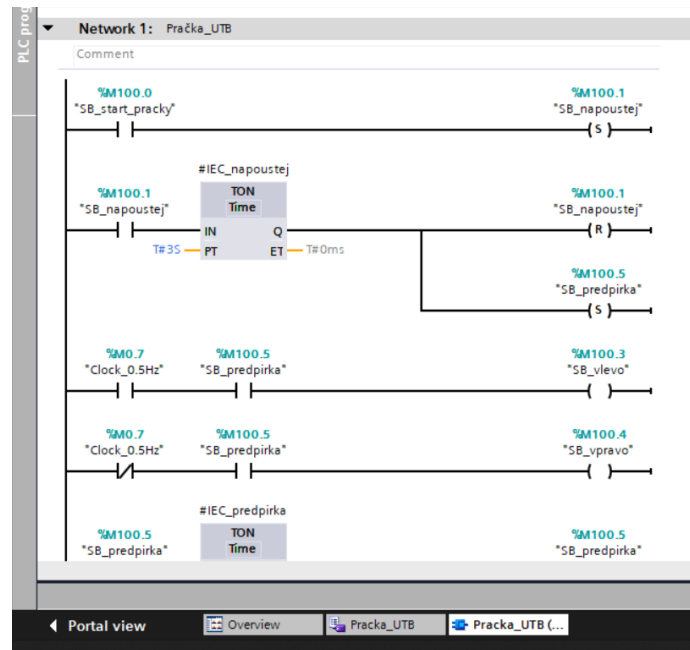
Časovač TON, nám bude držet hodnotu 1 po dobu 3s a poté posílá hodnotu 1 dále. Tím resetuje SB_napoustej a nastavuje následující funkci podle zadání. SB_predpirka. Během předpírky si chci na panelu zobrazit, že se bude buben pračky otáčet, tudíž si přidám dva systémové bity ve formě frekvencí, které jsou výhodou u Siemens, že si nemusím tyto bity vytvářet pomocí jiných funkcí, ale stačí nám zatrhnutí políčka, že budu využívat tyto funkce.

Následně využijí jejich funkce a během toho co je nasetována předpírka po dobu 10s budu pomocí těchto bitů přepínat po 0,5s na bity SB_vlevo a SB_vpravo, tím že využijí frekvenci 0,5 Hz.

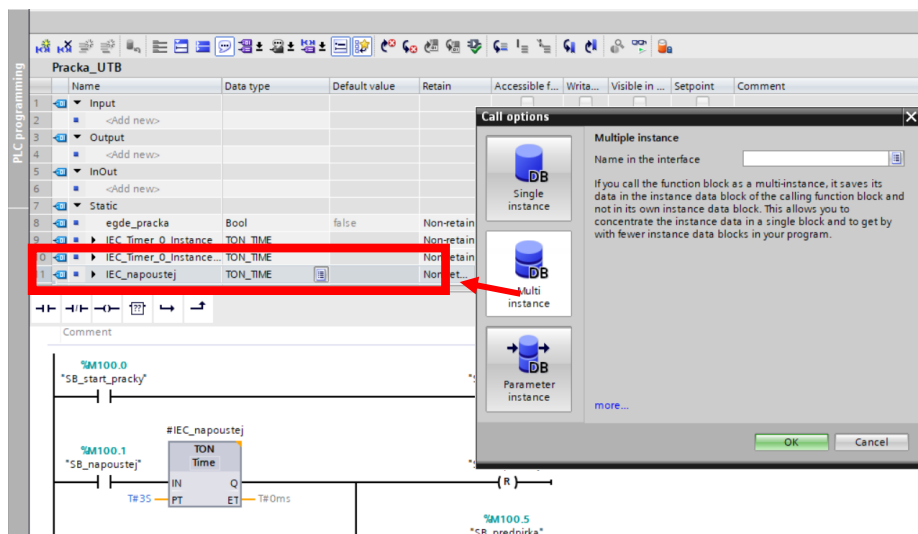
Postup kde povolit tuto možnost, Device configuration - general-System and clock memory.



Obr. 65 Clock memory

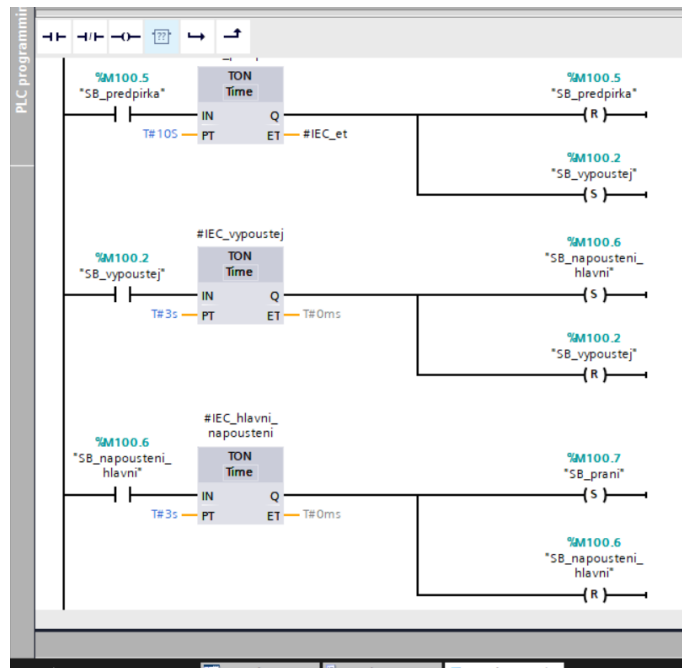


Obr. 66 I část programu



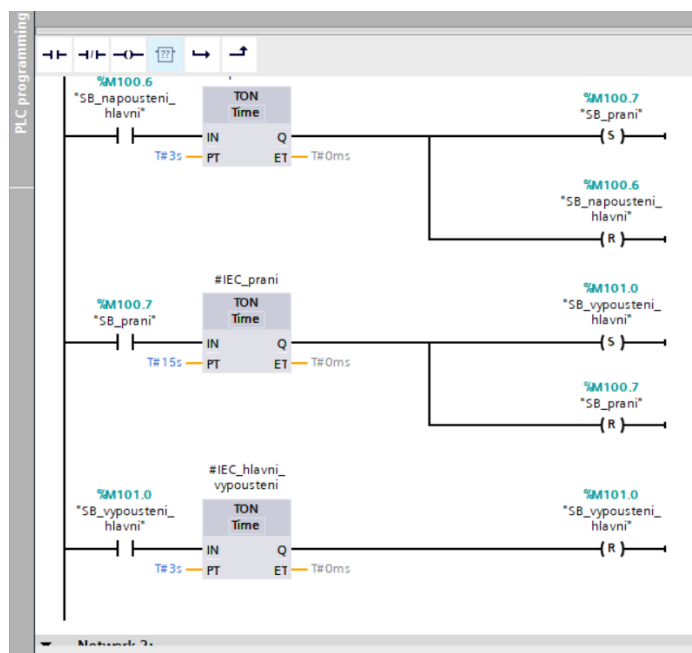
Obr. 67 Vytvoření datového bloku pro TON

Vytvořením datového bloku formátu multi-instance, znamená že nebude vytvořen datový blok nikde jinde v podsložce, ale použité hodnoty budou spojeny s tímto funkčním blokem. V rozsáhlém programu si tím zamezíme vytváření desítek datových bloků a složek, místo toho na konci budeme mít jeden datový blok pro použitý funkční blok.



Obr. 68 2 část programu

Jakmile uplyne doba 10s pro předpírku, resetují SB_predpirka a nastaví SB_vypoustej, které má trvat 3s. Poté je hlavní napouštění, také po dobu 3s a po uplynutí resetují SB_napousteni_hlavni a nastaví funkci SB_prani.



Obr. 69 3 část programu

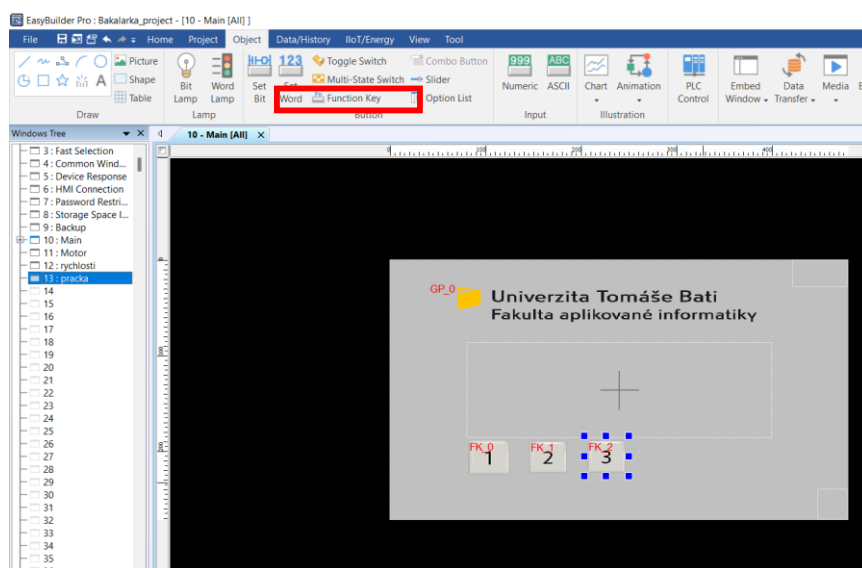
Funkce SB_prani, podle zadání má trvání 15s, opět po uplynutí doby resetují SB_prani a nastaví SB_vypousteni_hlavni, které po 3s resetují.

Zvolená varianta má trvání 37s na celý program. Možností, jak tento program vytvořit je hodně a záleží na každém, jak toto řešení pojme a zpracuje.

6.2.3 Vizualizace

U vizualizace opět jako u programu využijeme již vytvořeného programu a rozšíříme jej, přidáme nový screen pro novou aplikaci a napojíme nově vytvořené tagy, stejným způsobem jako je provedeno v předchozích úlohách.

Na hlavním screenu si vytvoříme tlačítko s funkcí „Function Key“, který otevře nově vytvořenou obrazovku, kde budeme mít vytvořenou vizualizaci pro úlohu s pračkou.



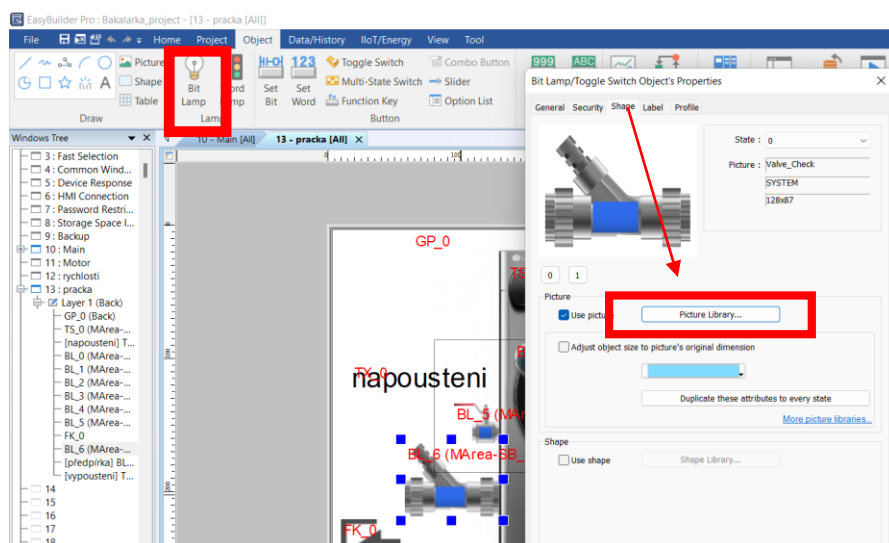
Obr. 70 Přidání screenu

Na nově vytvořeném screenu pro novou úlohu si vložíím obrázek pračky pro lepší představu o úloze a následně si přidám tlačítko pro start a následně už jen signalizační kontrolky, které nám budou signalizovat probíhající funkci.



Obr. 71 Vizualizace pro úlohu pračka

Pro každou funkci zvlášť jsem vytvořil příslušný prvek, tak aby byla vizualizace přehledná. Jednotlivé kontrolky, ventily, nebo ventilátor, který je jen jako rotující doplněk, jsou všechny vytvořené pomocí funkce Bit lamp.



Obr. 72 Funkce Bit lamp

Funkce Bit lamp nám nabízí zobrazování příslušného objektu jako proměnou nabývající dvou stavů, tudíž u kontrolky můžeme nastavit při hodnotě 0 – zhasnuto a 1 – svítí, nebo v případě ventilů, u kterých je vytvořena vizualizace pro dva stavy 0 - neteče voda, 1- teče voda.

Pod tlačítkem „Picture Library“ můžeme vybrat z mnoha možných vizualizací, které můžeme podle potřeby rozšířit stáhnutím složky s vizualizací z webových stránek Weintek.

K jednotlivým objektům připojím tagy, tak jak bylo uvedeno již u předchozích úloh. Poté již probíhá stejný postup, vizualizaci zkompilujeme a pokud je vše v pořádku, tak program nahrajeme do panelu.

7 SAIA-BURGESS

V této části práce si ověříme vzájemnou komunikaci mezi použitým panelem a PLC Saia PCD3.M 5540 a vytvoření funkčního propojení na základním příkladu, tak abychom si ověřili funkčnosti fyzických vstupů/výstupů a paměti.

7.1 Zadání

Vytvořit funkční komunikační spojení mezi PLC a HMI a toto spojení ověřit na jednoduchém funkčním příkladu.

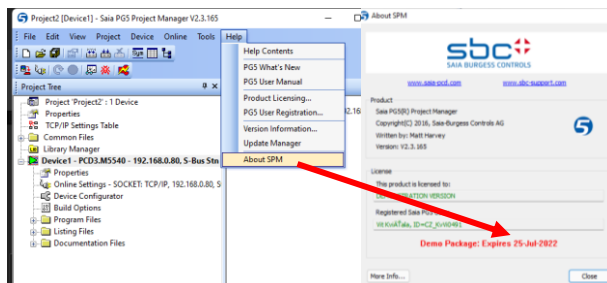
Zařízení k dispozici je Saia PCD3.M5540.



Obr. 73 Saia PCD3.M5540

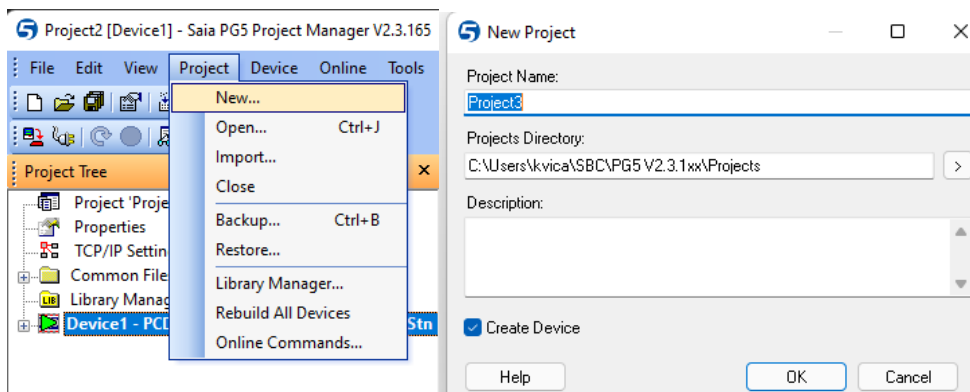
7.2 Realizace programu

V případě realizace zkušebního propojení zařízení potřebujeme vývojové prostředí pro PLC Saia, tento program je ke stažení ze stránek výrobce po registraci zdarma, ovšem tato verze je pouze demonstrační a její platnost si ověříme v prostředí.

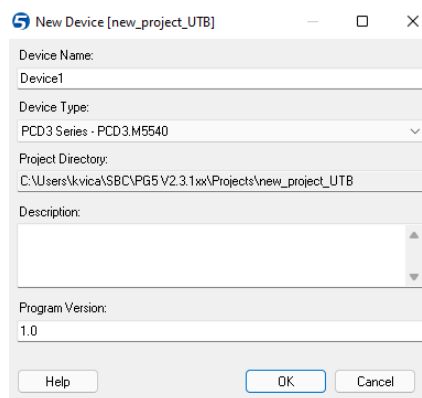


Obr. 74 Platnost programu

Jakmile si stáhneme prostředí a provedeme instalaci, začneme se zkušebním projektem. Vytvoříme si nový projekt a přidáme zařízení, které máme k dispozici. K PLC jsme připojeni pomocí S-Bus USB.

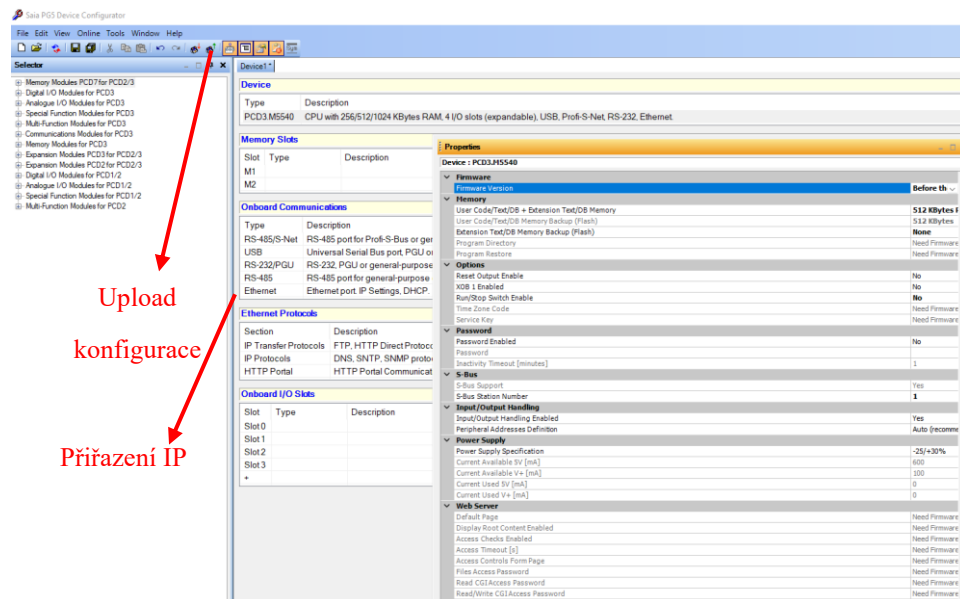


Obr. 75 Nový projekt



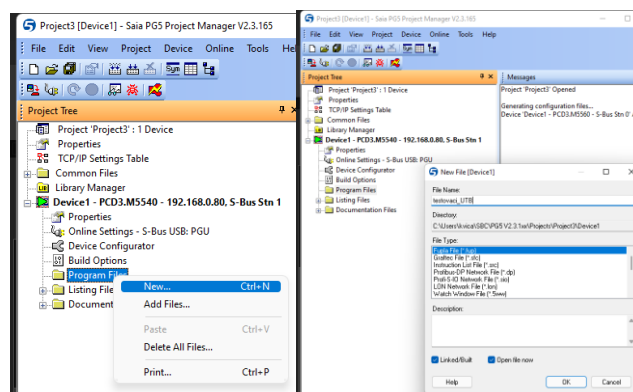
Obr. 76 Přidání nového zařízení

Když vytvoříme nový projekt, musíme naši sestavu PLC nahrát do naší konfigurace, a to uděláme tlačítkem „upload configuration“, a následně si můžeme zobrazit parametry naší sestavy. Potom přidáme IP adresu a vytvoříme si program.



Obr. 77 Parametry konfigurace

Nastaví se IP adresa, okno s konfiguračními parametry uložíme a zavřeme. Pro vytvoření nového programu je na výběr z více možností programovacích jazyků. Na zkušební aplikaci využijeme jazyk „Fupla“, který je ve své podstatě „Ladder“ od Siemens, akorát v jiném prostředí. Vytvoříme si jednoduchou aplikaci na ověření funkčnosti. Na složku „Program files“, klikneme pravým tlačítkem na myši a vybereme možnost „New“ a následně si vybereme „Fupla“. Projekt se sám otevře.



Obr. 78 Postup vytvoření nového projektu

Na levé straně máme na výběr z velkého množství prvků, které můžeme použít pro vytvoření programu. Ve zkušební aplikaci pro ověření funkčnosti použijeme fyzický vstup s adresou „I0“ a výstup „O16“, navíc si vyzkoušíme, zda bez problémů pracujeme s pamětí PLC, a to tak že si přidám paměťový bit, které se zde nazývají „Flag“. Kontakty dáme do jednoduché logiky $I0 \text{ AND } F10 = O16$, po vložení kontaktů je musíme spojit vzájemně a přidat konektory. Poté logiku uložíme a provedeme „Build“, který nám logiku zkontroluje. Pokud je vše

v pořádku není nám zobrazen žádný alarm, logika se uloží a můžeme program nahrát do PLC.

build

konektor propojení

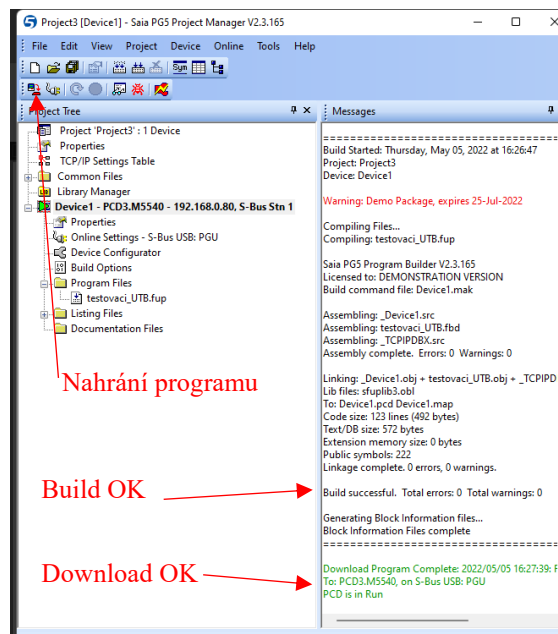
I 0 F 10 O 16

$I 0 + F 10 = O 16$

Seznam vytvořených I/O

Symbol Name	Type	Address/Value	Comment	Actual Value	Tags	Scope
COB_0	COB	0		0		Local
I 0	I	0		0		Local
F 10	F	10		10		Local
O 16	O	16		16		Local

Obr. 79 Zkušební logika programu

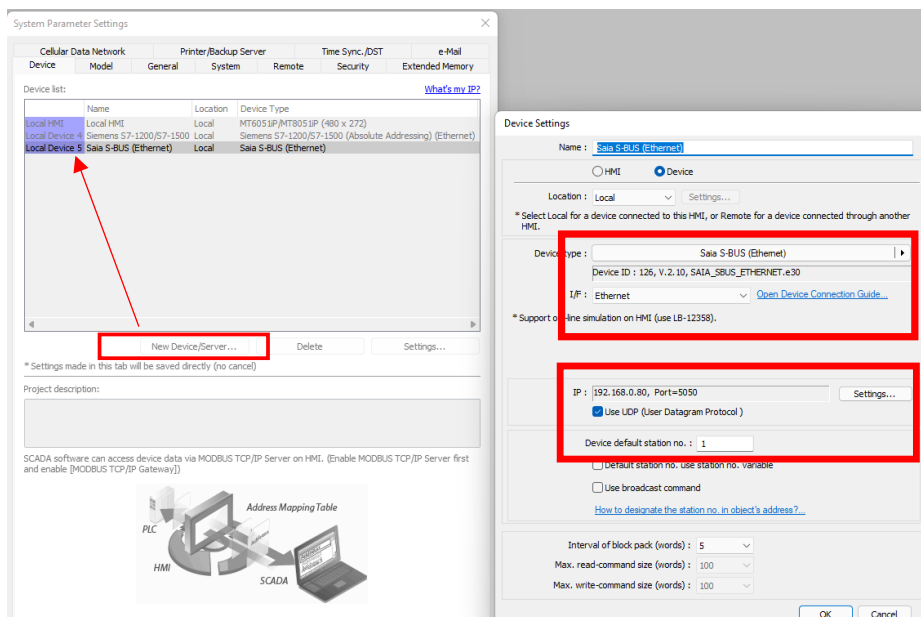


Obr. 80 Nahrání programu

7.3 Vizualizace

Ted' když máme program hotov a vše proběhlo v pořádku, vytvoříme si zobrazení na HMI. V stejném projektu jako máme vytvořené úlohy pro Siemens, si přidáme nový screen a vytvoříme si jednoduché zobrazení. Nejdříve si ale vytvoříme spojení HMI s PLC.

Přidáme nové zařízení, které vybereme ze seznamu nabízených možností, vybereme komunikaci po Ethernetu, s využitím protokolu TCP/IP, pod ikonou „Settings“ přiřadíme IP adresu PLC se kterým bude HMI komunikovat, nesmíme zapomenout, že všechny adresy zařízení, které jsme použili, musí být ve stejném rozsahu. Zatrhneme možnost „Use UDP“ a zkontrolujeme si číslo zařízení, zda voláme stejné jako máme nastavené v programu PLC.

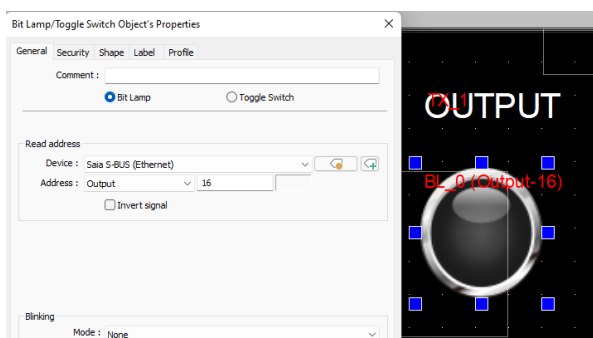


Obr. 81 Přidání nového PLC



Obr. 82 Vizualizace pro PLC Saia

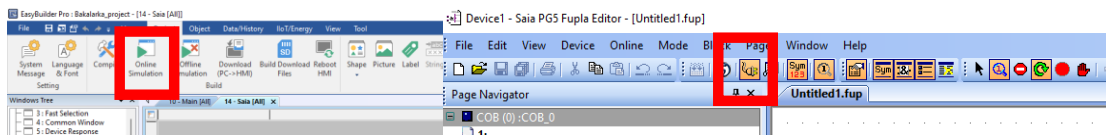
Pro vytvoření vizualizace použijeme funkce „bit lamp“, které nám budou sloužit v případě „I0“ k tomu, že i vizuálně si potvrdíme to, že bit „I0“ má hodnotu 1. To stejné platí pro „O16“. Pro „F10“ použijeme funkci „Toggle switch“ a vytvoříme si přepínač, který budeme moct ovládat, oproti „I0“ a „O16“, které pouze zobrazují svůj stav.



Obr. 83 Adresování pro O16

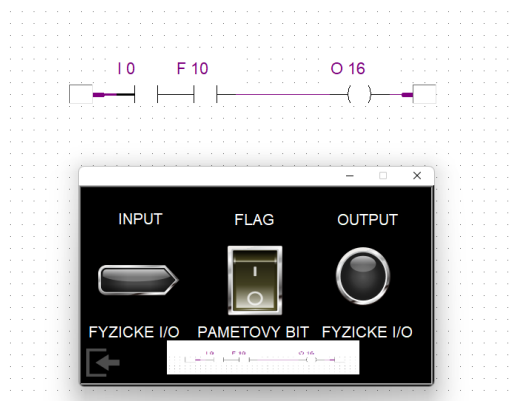
Vizualizace je velmi jednoduchá a splňuje potřeby pro vyzkoušení komunikace. Provedeme kompilaci projektu a jeho nahrání do HMI. Následně si ověříme funkčnost programu ve vizualizaci.

V prostředí HMI si pustíme online simulaci a v PG5 si otevřeme vytvořený projekt a pustíme zařízení do stavu „online“.



Obr. 84 Spuštění do „Online“ režimu

Otevřeme si vzájemně online režimy obou zařízení. Ve výchozích stavech nemá žádný I/O hodnotu 1.



Obr. 85 Hodnota logického součtu je rovna 0

Po přepnutí fyzického vstupu „I0“ na PLC, zobrazí se jeho hodnota rovna 1, tím že proměnná „bit lamp“ změní barvu a zároveň tuto změnu jde sledovat v online režimu v PG5.



Obr. 86 „I0“ = hodnota 1

Můžeme vidět že změnou vstupu „I0“ na hodnotu 1, pořád nesplňujeme podmínku pro „O16“ = 1, toho dosáhneme až po přepnutí „F10“ na hodnotu 1.



Obr. 87 *Splnění podmínky*

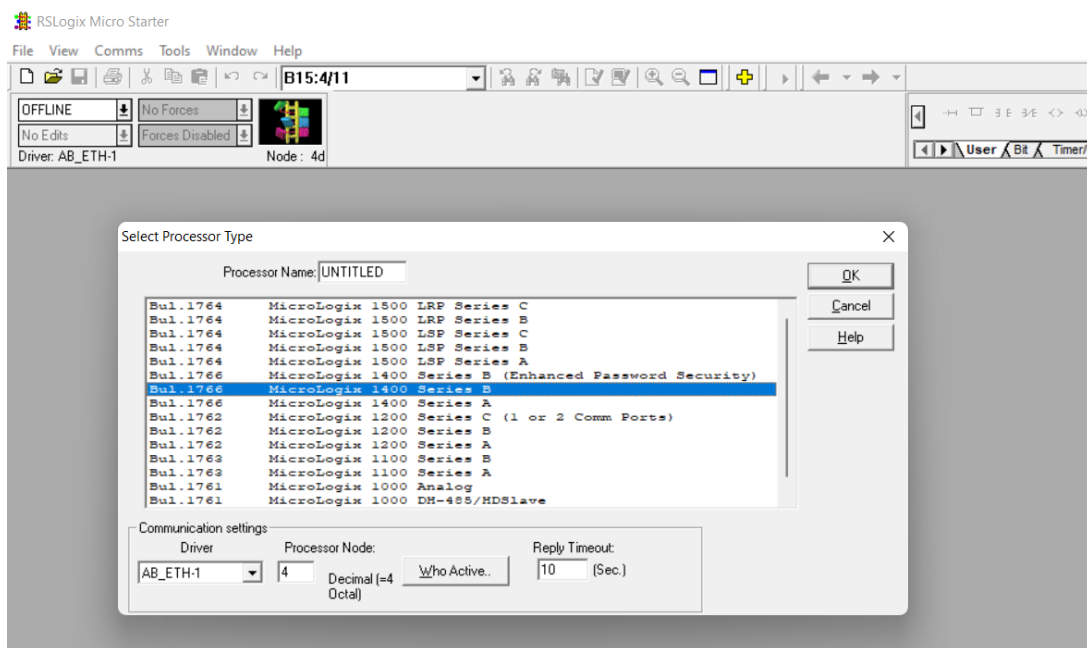
Jakmile splníme podmínku. Můžeme vidět že „O16“ má hodnotu 1 a tím jsme dosáhli ověření funkčnosti programu a celkové komunikace obou zařízení.

8 ALLAN BRADLEY

V této kapitole je pouze pro názornost představena realizace prvních dvou úloh v programovacím prostředí pro PLC systémy od firmy Allen Bradley.

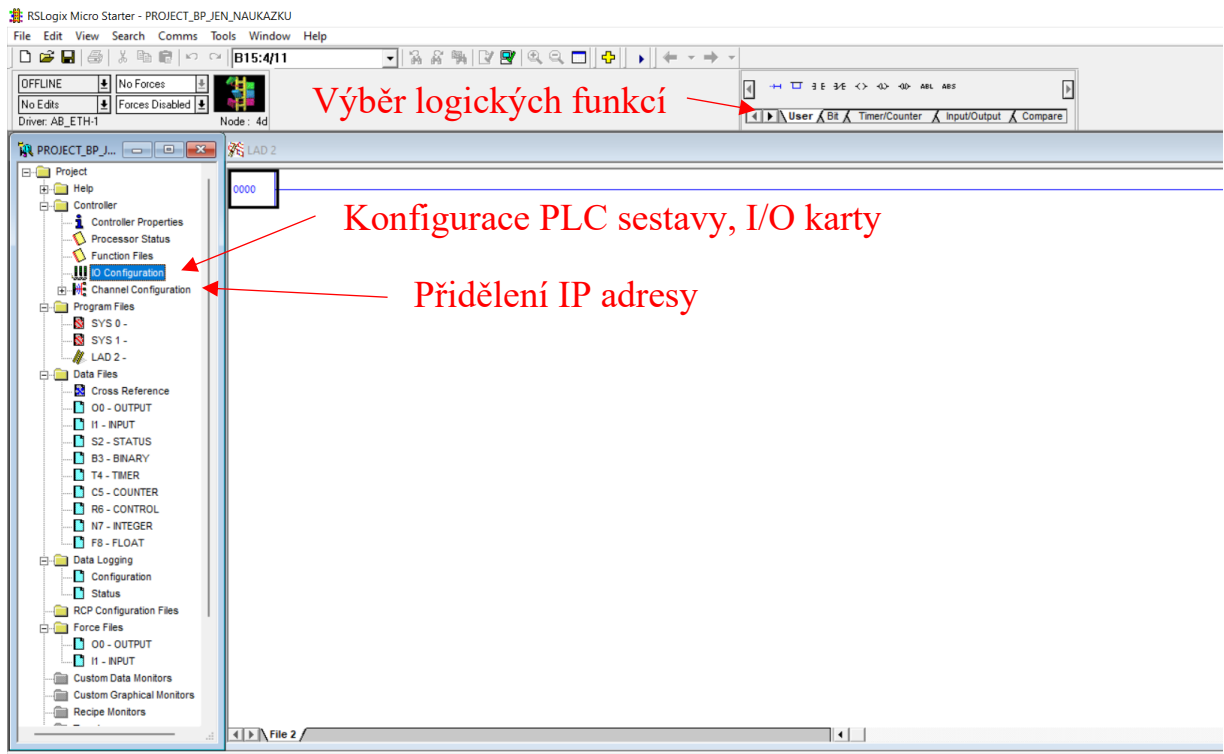
8.1 Realizace

V programu pro PLC Bradley vyberu typ PLC, se kterým bude práce probíhat. Po zvolení se otevře prostředí.

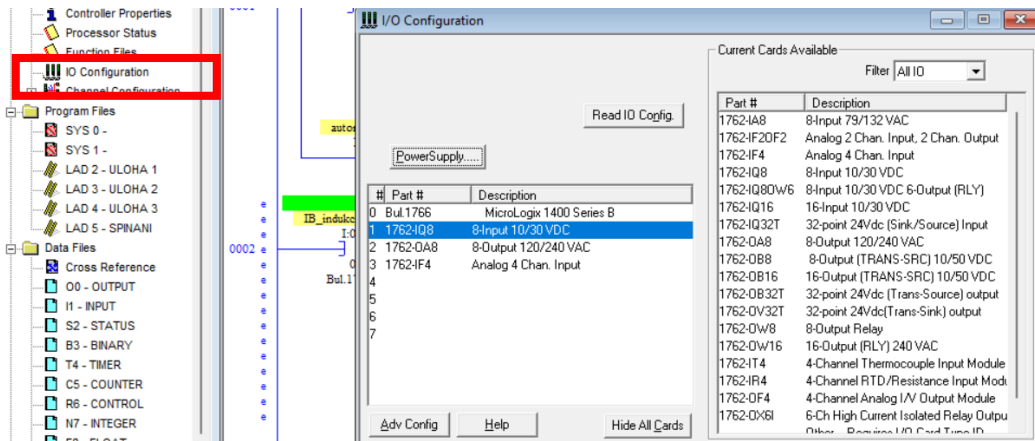


Obr. 88 Výběr PLC v programu RSLogix Micro

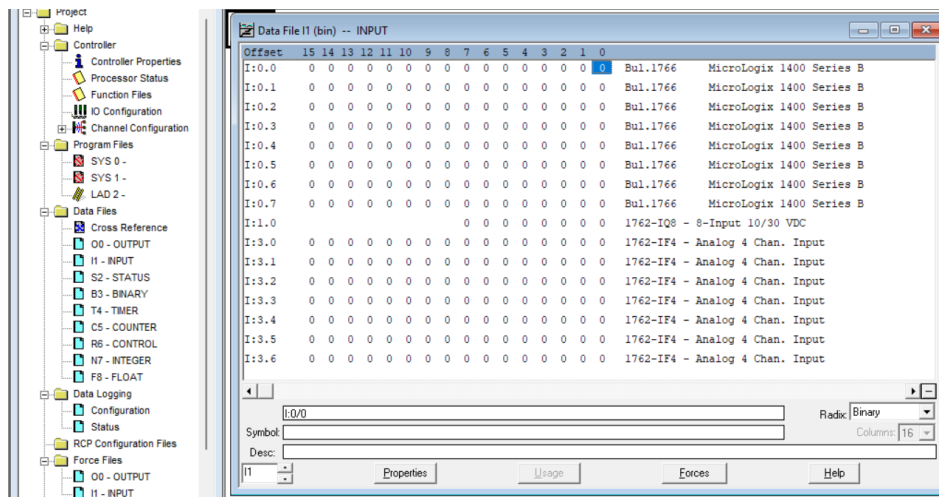
K provedení aplikací si připojím PLC a I/O karty, které budou v konfiguraci. Přidělím PLC IP adresu.



Obr. 89 Konfigurace sestavy



Obr. 90 I/O konfigurace

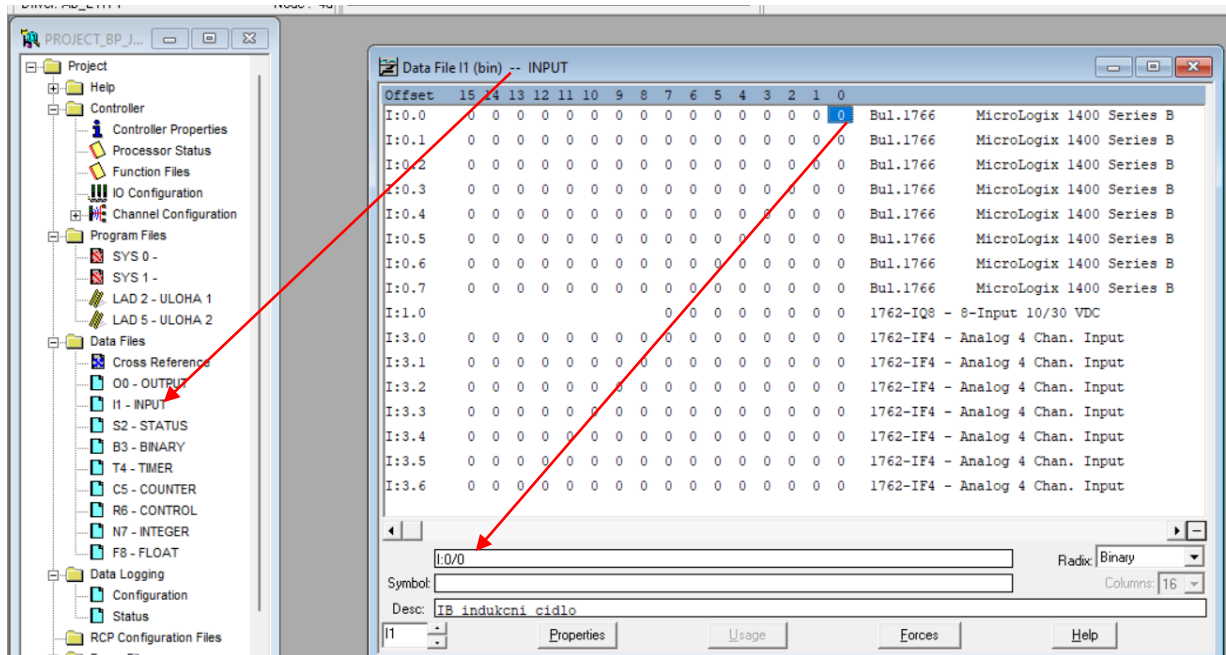


Obr. 91 INPUT file

Oproti systému Siemens, zde netvoříme tagy, které pak odkazujeme na příslušnou bitovou adresu, kterou máme k dispozici na PLC, nebo na I/O kartě, popřípadě memory bitu, ale připsujeme proměnné na bity, tak jak je zobrazeno na obrázku 91.

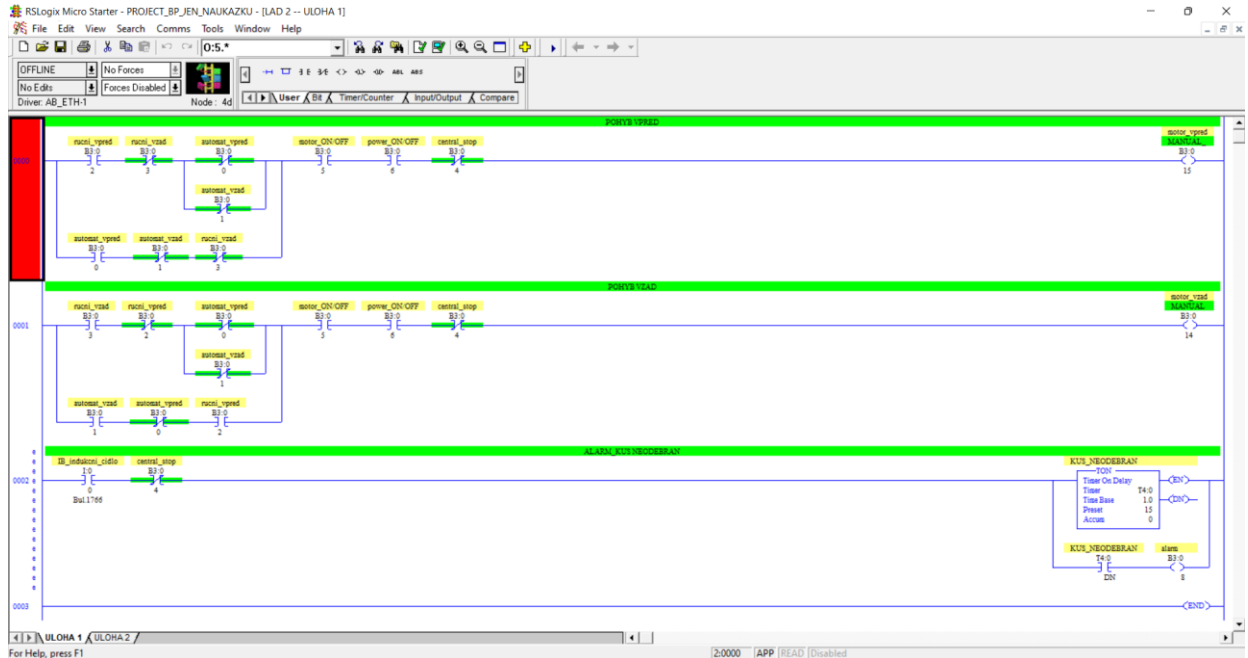
Vytvoříme si bity, se kterými budeme pracovat, tudíž do 00 - OUTPUT si vypíšeme výstupové bity, kterými se bude ovládat motor, do I1-INPUT přijdou vstupní signály.

Pro použití tlačítek z panelu se využije B3-BINARY, obdoba vytvoření memory bitů u Siemens, kdy se vytvoří pomocné bity, se kterými se pracuje.



Obr. 92 Zobrazení INPUT file

Stejným postupem jako na předchozím obrázku, se popíšu všechny potřebné bity pro úlohy. Jakmile jsou bity popsány, tak se může začít s psaním logiky programu. Stejně jako u Siemens, i tady je řešení v jazyce LAD, tudíž zpracování úloh je velmi podobné.



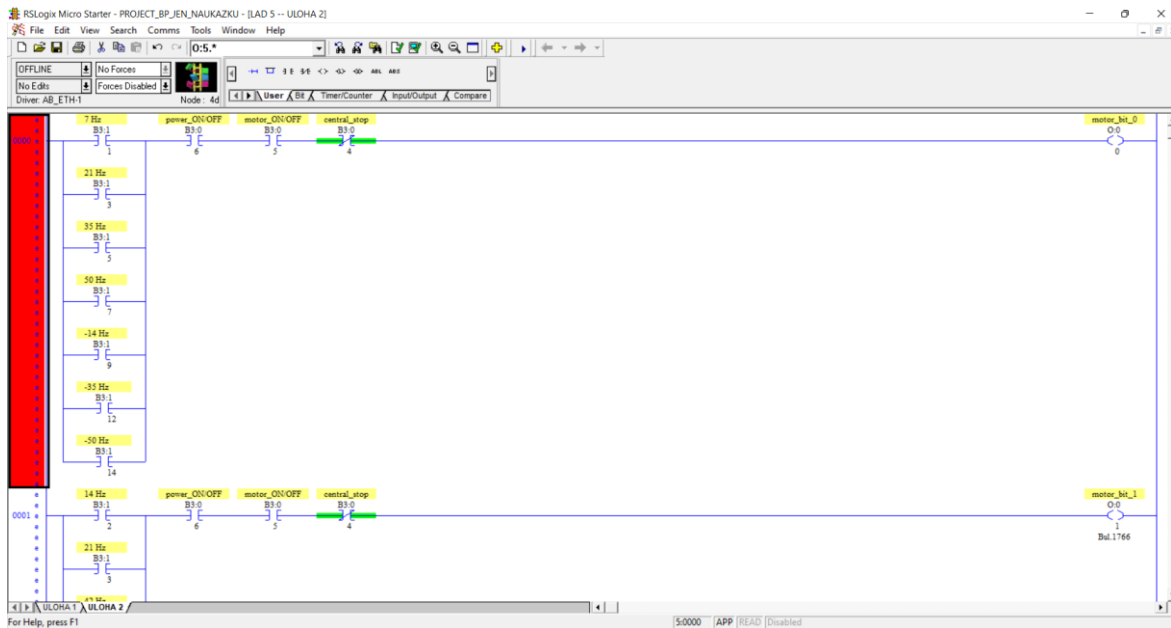
Obr. 93 Řešení úlohy 1

Nyní budeme řešit úlohu 2. Z první úlohy už máme vypracovanou tabulku bitové kombinace. Ta bude použita i u tohoto řešení, stejným způsobem, kdy budeme posílat kombinaci bitů na příslušný výstup.

V „Data files B3-BINARY“ rezervujeme celý word pro rychlosti. Následně jednotlivé bity jsou popsány pro rychlostní varianty, dle tabulky 3.

Data File B3 (bin) -- BINARY																		
Offset	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	(Symbol)	Des
B3:0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
B3:1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	rychlosti	
B3:2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
B3:3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
B3:4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Obr. 94 Bity pro rychlosti

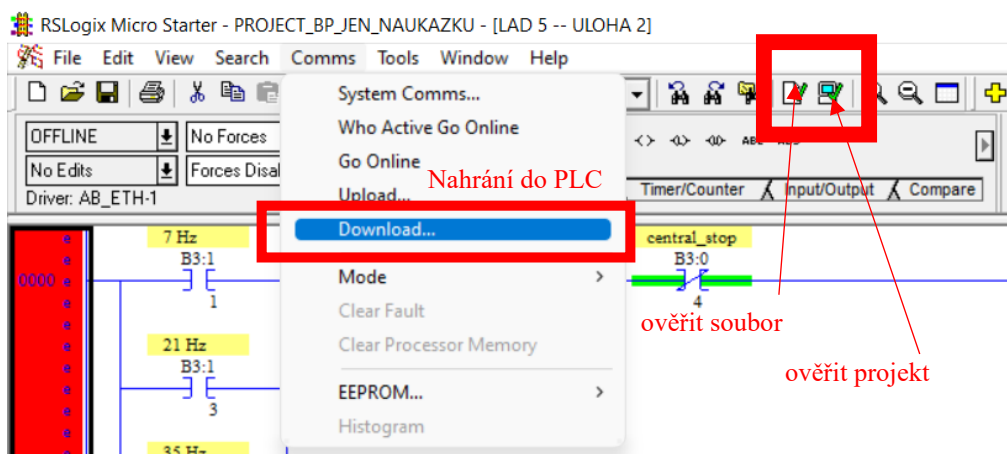


Obr. 95 Úloha 2

Pro řešení druhé úlohy bude varianta obdobná jako u Siemens, s tím že nebude využito memory bitů, ale použité bity pro rychlosti budou svojí kombinací ovládat příslušné digitální výstupy, tudíž použitý motor.

Tak jak je vidět na předchozím obrázku, tak je program situován stejně i pro ostatní výstupy, které nejsou na obrázku vidět.

Jakmile máme vytvořenou logiku, tak program musíme zkontrolovat, zkompilovat a poté můžeme nahrát do PLC.

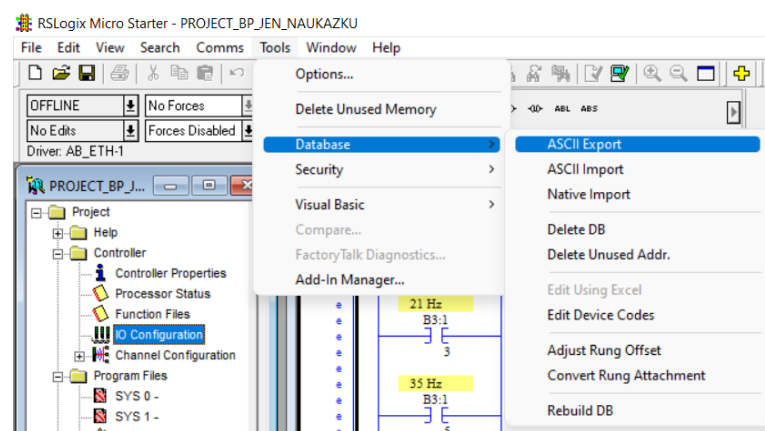


Obr. 96 Ověření projektu

Ted' když je logika vytvořená, tak se dostáváme k vizualizaci a propojení s panelem. Vizualizace bude použita stejná, není potřeba tvořit novou vizualizaci, když tvoříme stejnou aplikaci, jen v odlišném prostředí.

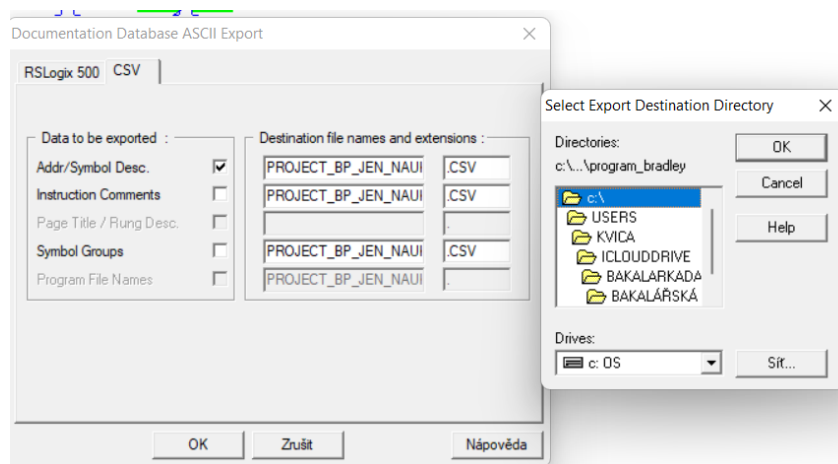
Nejprve si na webových stránkách od výrobce panelu www.weintek.com najdeme příslušný soubor, kde mají popsán postup, jakým komunikují s PLC ML 1400 od Rockwell, jakým způsobem se odkazují na příslušné bity, nesoucí informaci a poté postupují dále.

Následný postup slouží k připojení použitých dat, k panelu, tak abychom opět ovládali aplikaci přes panel.



Obr. 97 Exportování dat

Jak to bylo u Siemens, tak i zde musíme exportovat data ze systému PLC ve formátu .xlsx, do naší zvolené složky, ze které poté importujeme data pro panel.



Obr. 98 Exportování dat

Jakmile exportujeme data z PLC do souboru, musíme tyto data nahrát do panelu, tak abychom s nimi mohli pracovat.

Tak jak jsme importovali data ze souboru excel, pro provedení aplikace pro Siemens, je tento postup stejný, jen na začátku musíme zvolit příslušný typ PLC se kterým bude panel propojen.

9 ZÁVĚR

První polovina teoretické části je věnována popisu systému PLC a HMI tak, aby čtenář v souvislosti s těmito informacemi, byl schopen dále v textu porozumět vypracovaným úlohám, ale aby měl i tyto informace kdykoliv k použití. V druhé polovině jsou popsány nejčastější komunikační protokoly využívané v komunikaci u systému PLC a PLC s HMI.

Druhá část bakalářské práce obsahuje hlavní náplň. Tato část je věnována vytvořením funkčních úloh na rozdílných typech PLC s jedním panelem, na kterých je demonstrováno stabilní propojení zařízení a jejich vzájemné komunikace. Jedním z cílů této práce bylo i vytvoření manuálů k těmto úlohám. Tento bod je zde splněn a u vypracovaných úloh je vždy podrobný návod, jak tyto úlohy vytvořit. Přes postupnou konfiguraci PLC a napsání programu až k vytvoření vizualizace a nastavení vzájemné konfigurace.

Během celkové konfigurace propojených zařízení, byla ověřena vzájemná kompatibilita HMI od firmy Weintek a PLC od Siemens a Saia, navíc jsou některé vybrané aplikace vytvořeny i pro PLC Allen Bradley, které jsou pouze pro názornost.

Ve fyzickém zapojení byla ověřena funkčnost dvou souběžně uvedených zařízení do stavu „run“ na jeden panel HMI a jejich možné ovládání současně při tomto stavu.

Navíc je v práci možnost porovnání vývojových prostředí tří rozdílných programovatelných automatů.

SEZNAM POUŽITÉ LITERATURY

- [1] *CO JE PLC NEBOLI PROGRAMOVATELNÝ LOGICKÝ AUTOMAT* [online]. [cit. 2022-04-17]. Dostupné z: <https://dreamland-plc.cz/plc-programovatelny-logicky-automat/>
- [2] BOLTON, William. *Programmable Logic Controllers*. 6th Edition. 2015. ISBN 9780128029299.
- [3] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. *PLC a automatizace*. Praha: BEN - technická literatura, 1999. ISBN 80-86056-58-9.
- [4] ŠMEJKAL, Ladislav. *PLC a automatizace*. Praha: BEN - technická literatura, 2005. ISBN 80-7300-087-3.
- [5] *Human-Machine Interfaces (HMIs): Chapter 29 - Basics of Programmable Logic Controllers (PLCs)* [online]. 2019 [cit. 2022-04-17]. Dostupné z: <https://control.com/textbook/programmable-logic-controllers/human-machine-interfaces/>
- [6] MUELANER, Jody. *Application Layer Protocol Options for M2M and IoT Functionality* [online]. 2021-04-27 [cit. 2022-04-17]. Dostupné z: <https://www.digikey.com/en/articles/application-layer-protocol-options-for-m2m-and-iot-functionality>
- [7] *PROFINET - the leading Industrial Ethernet Standard* [online]. [cit. 2022-04-17]. Dostupné z: <https://www.profibus.com/technology/profinet#tab2-208895>
- [8] *PLC BASICS* [online]. [cit. 2022-04-17]. Dostupné z: <https://real-pars.com/plc-basics/>
- [9] *AC 800M: Communication Protocols*. ABB, 2013.
- [10] *IO-LINK* [online]. [cit. 2022-04-17]. Dostupné z: <https://www.profibus.com/technology/io-link>

- [11] *EtherCAT – the Ethernet Fieldbus* [online]. [cit. 2022-04-17]. Dostupné z: <https://www.beckhoff.com/en-us/products/i-o/ethercat/>
- [12] *EtherNet/IP Network Devices* [online]. [cit. 2022-05-09]. Dostupné z: <https://www.rockwellautomation.com/en-us/support/documentation/literature-library.html>
- [13] *Referenční model ISO/OSI* [online]. [cit. 2022-04-17]. Dostupné z: <http://ijs.8u.cz/index.php/standardizace-v-pocitacovych-sitich/referenci-model-iso-osi>
- [14] EDEMA, Victory. *An Introduction to the Modbus Communication Protocol* [online]. [cit. 2022-04-17]. Dostupné z: <https://www.solisplc.com/tutorials/modbus?msclkid=7d180ff1be6f11ec90668ea4baf8c05e>
- [15] *KOMUNIKAČNÍ PERIFERIE* [online]. [cit. 2022-04-17]. Dostupné z: <http://plc-automatizace.cz/knihovna/periferie/komunikacni-periferie.htm>
- [16] *Master-Slave Communication via a CM PtP using the Modbus RTU Protocol* [online]. 2019 [cit. 2022-04-17]. Dostupné z: <https://support.industry.siemens.com>
- [17] PARDEEP, Kumar. *Human Machine Interface (HMI): Software Projects*. LAP Lambert Academic Publishing, 2019. ISBN 978-620-0-47941-9.
- [18] YEMAO, Man. *Human-Machine Interface Considerations for Design and Testing in Distributed Sociotechnical Systems*. Gothenburg, Sweden, 2015. CHALMERS UNIVERSITY OF TECHNOLOGY.
- [19] *Jak vyrobit/opravit síťový UTP kabel - Internetový RJ45* [online]. 2020 [cit. 2022-04-18]. Dostupné z: <http://www.servispckupka.cz/jak-vyrobit-opravit-sitovy-utp-kabel.php>
- [20] *MT851iP* [online]. Dostupné také z: https://dl.weintek.com/public/MT8000iP/Datasheet/eng/MT8051iP1_Datasheet_ENG.pdf

[21] *Specification* [online]. [cit. 2022-04-25]. Dostupné z: https://www.wein-tek.com/globalw/Product/Product_spec.aspx?Series=iP

[22] FOROUZAN, Behrouz A. *TCP/IP protocol suite*. 4th ed. Boston: McGraw-Hill Higher Education, c2010. ISBN 978-0-07-337604-2.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PLC	Programmable Logic Controller.
HMI	Human Machine Interface.
DI	Digital Input.
DO	Digital Output
AI	Analog Input.
FBD	Function Block Diagram
I/O	Input/Output
LAD	Ladder Diagram
PC	Personal Computer
RAM	Random Access Memory
ROM	Read Only Memory
ST	Structure Text
TIA	Totally Integrated Automation

SEZNAM OBRÁZKŮ

Obr. 1 Schématická struktura PLC [2]	10
Obr. 2 Ukázkové schéma propojení PLC s analogovými vstupy a výstupy [8].....	12
Obr. 3 Otočka cyklu [2].....	13
Obr. 4 Ukázka IL.....	13
Obr. 5 Ladder diagram	14
Obr. 6 Jazyk FBD.....	15
Obr. 7 Jazyk strukturovaný text	15
Obr. 8 Použití HMI jako ovládací člen [7]	16
Obr. 9 Cyklický model interakce uživatel a zařízení [18]	16
Obr. 10 Obecná struktura HMI obrazovek [15]	17
Obr. 11 Ukázka tagů v PLC	18
Obr. 12 Rozměry panelu MT8051iP [20]	19
Obr. 13 Konektory panelu MT8051iP [20].....	19
Obr. 14 Parametry k panelu MT8051 iP [20]	20
Obr. 15 Modelová řada iP [21].....	21
Obr. 16 Nastavení zařízení.....	22
Obr. 17 Náhled na návod pro připojení komunikace.....	22
Obr. 18 Referenční model ISO/OSI [12]	23
Obr. 19 Rozdíl mezi OSI modelem a TCP/IP [22]	25
Obr. 20 Adresování u TCP/IP [22]	26
Obr. 21 Křížové propojení u RJ45 [19]	27
Obr. 22 Možné zasíťování zařízení pomocí Profinetu [9].....	28
Obr. 23 Náhled na schéma komunikace [7].....	29
Obr. 24 EtherCat frame [11].....	30
Obr. 25 PLC S7-1200 CPU 1212C AC/DC/RLY.....	34
Obr. 26 Operátorský panel MT8051iP	35
Obr. 27 Frekvenční měnič Sinamics V20	35
Obr. 28 tři fázový motor.....	35
Obr. 29 Indukční snímač	36
Obr. 30 Značení PLC	36
Obr. 31 Naznačené části postupu výběru.....	37
Obr. 32 Přiřazení jména PLC	37

Obr. 33	Přiřazení IP adresy.....	38
Obr. 34	Vložení nového souboru pro tagy	39
Obr. 35	Seznam požitých vstupů	39
Obr. 36	Seznam použitých výstupů	39
Obr. 37	Schéma programu.....	41
Obr. 38	Propis na výstupy.....	42
Obr. 39	Fyzické výstupy z PLC	43
Obr. 40	Propojení výstupů z PLC	43
Obr. 41	Nový projekt.....	44
Obr. 42	Nové zařízení	44
Obr. 43	Přiřazení PLC.....	45
Obr. 44	Adresa zařízení	45
Obr. 45	Import tagů	46
Obr. 46	Import	46
Obr. 47	Postup exportu tagů z TIA portálu	47
Obr. 48	Screen ovládání motoru.....	47
Obr. 49	Přiřazování adres tagů	48
Obr. 50	Lišta projektu	48
Obr. 51	Přidání panelu	49
Obr. 52	Připojení panelu	49
Obr. 53	Přidání screenu.....	50
Obr. 54	Vizualizace v prostředí TIA portál.....	50
Obr. 55	Základní nástroje pro vizualizaci	51
Obr. 56	Přesun tagů.....	52
Obr. 57	Přiřazení funkce tlačítka.....	53
Obr. 58	Objekt kus přítomen.....	54
Obr. 59	Rozšíření tagů	56
Obr. 60	Tlačítka pro rychlost.....	57
Obr. 61	Rozšíření programu	58
Obr. 62	Screen pro manipulaci rychlosti.....	59
Obr. 63	Přidání nového FB.....	61
Obr. 64	Tagy pro program pračka.....	61
Obr. 65	Clock memory	62
Obr. 66	1 část programu.....	63

Obr. 67 Vytvoření datového bloku pro TON.....	63
Obr. 68 2 část programu.....	64
Obr. 69 3část programu.....	64
Obr. 70 Přidání screenu.....	65
Obr. 71 Vizualizace pro úlohu pračka.....	66
Obr. 72 Funkce Bit lamp.....	66
Obr. 73 Saia PCD3.M5540.....	68
Obr. 74 Platnost programu.....	68
Obr. 75 Nový projekt.....	69
Obr. 76 Přidání nového zařízení.....	69
Obr. 77 Parametry konfigurace.....	70
Obr. 78 Postup vytvoření nového projektu.....	70
Obr. 79 Zkušební logika programu.....	71
Obr. 80 Nahrání programu.....	72
Obr. 81 Přidání nového PLC.....	73
Obr. 82 Vizualizace pro PLC Saia.....	73
Obr. 83 Adresování pro O16.....	73
Obr. 84 Spuštění do „Online“ režimu.....	74
Obr. 85 Hodnota logického součtu je rovna 0.....	74
Obr. 86 „I0“ = hodnota 1.....	74
Obr. 87 Splnění podmínky.....	75
Obr. 88 Výběr PLC v programu RSLogix Micro.....	76
Obr. 89 Konfigurace sestavy.....	77
Obr. 90 I/O konfigurace.....	77
Obr. 91 INPUT file.....	78
Obr. 92 Zobrazení INPUT file.....	78
Obr. 93 Řešení úlohy 1.....	79
Obr. 94 Bity pro rychlosti.....	79
Obr. 95 Úloha 2.....	80
Obr. 96 Ověření projektu.....	80
Obr. 97 Exportování dat.....	81
Obr. 98 Exportování dat.....	81

SEZNAM TABULEK

Tabulka 1 <i>Profinet x Profibus</i>	29
Tabulka 2 <i>Dvou bitová tabulka</i>	42
Tabulka 3 <i>Tabulka bitová kombinace</i>	55